

UNIVERSITÄT LEIPZIG

**Institut für Produktionswirtschaft
und Industrielle Informationswirtschaft**

Marschnerstraße 31, 04109 Leipzig

Tel.: 0341/4941-182, Fax: -125

Arbeitsbericht Nr. 10

**Petrinetzbasierte Modellierung
komplexer Produktionssysteme**

**Band 5: Einführung in Synthetische Netze
Teilband 5.2: Auswertungsmöglichkeiten**

von

Univ.-Prof. Dr. Stephan Zelewski

<zelewski@hpswifa.wifa.uni-leipzig.de>

Leipzig 1995

Alle Rechte vorbehalten.

Inhaltsverzeichnis zu Band 5.2

	Seite
5.2	Auswertungsmöglichkeiten für Synthetische Netze 1
5.2.1	Einführung 1
5.2.2	Definition von Netzeigenschaften 7
5.2.2.1	Statische Eigenschaften 7
5.2.2.2	Dynamische Eigenschaften 9
5.2.2.2.1	Überblick 9
5.2.2.2.2	Beschränktheit 11
5.2.2.2.3	Deadlocks, Lebendigkeit und Livelocks 15
5.2.2.2.4	Finalität, Optimalität und Terminieren 57
5.2.2.2.5	Robustheit, Promptheit und Reversibilität 67
5.2.3	Untersuchung von Netzeigenschaften 85
5.2.3.1	Überblick über Auswertungstechniken 85
5.2.3.2	Die Deduktionsanalyse 87
5.2.3.3	Die Invariantenanalyse 91
5.2.3.4	Die Reduktionsanalyse 114
5.2.3.5	Die Erreichbarkeitsanalyse 117
5.2.3.5.1	Grundsätzliches 117
5.2.3.5.2	Konstruktion von Erreichbarkeitsgraphen 144
5.2.3.5.3	Auswertung von Erreichbarkeitsgraphen 206
5.2.3.5.4	Optimierung von Netzmodellen 239
5.2.3.5.5	Robustheitsuntersuchungen 279
5.2.3.6	Die Simulationsanalyse 296
5.2.4	Beiträge der Netzanalyse zur Prädikatenlogik 320
	Literaturverzeichnis zu Teilband 5.2 331

5.2 Auswertungsmöglichkeiten für Synthetische Netze

5.2.1 Einführung

Das analytische¹⁾ Potential des Petrinetz-Konzepts umfaßt die Gesamtheit aller Eigenschaften von Netzmodellen²⁾, die sich mit der Hilfe von Auswertungstechniken untersuchen lassen³⁾. Dieses Analysepotential wird hier nicht vollständig entfaltet. Statt dessen werden nur jene Netzeigenschaften vorgestellt, die für Prozeßkoordinierungen in komplexen Produktionssystemen von Interesse sein können⁴⁾. Ebenso werden die Auswertungstechniken des Petrinetz-Konzepts nicht erschöpfend behandelt. Vielmehr wird nur auf diejenigen Auswertungstechniken eingegangen, die bei der Analyse von Netzmodellen vorherrschen und zugleich für die Untersuchung von produktionswirtschaftlich interessanten Netzeigenschaften geeignet sind. Darüber hinaus wird später gezeigt, daß eine dieser Auswertungstechniken für die Analyse aller produktionswirtschaftlich interessanten Netzeigenschaften ausreicht.

Die Eigenschaften eines Netzmodells lassen sich in zwei Dimensionen gruppieren. Die erste Dimension bezieht sich auf die Struktur von Netzmodellen. Aus dieser Perspektive wird zwischen statischen und dynamischen Netzeigenschaften unterschieden. Die zweite Dimension beleuchtet den Bezugspunkt, an dem die Untersuchung eines Netzmodells vornehmlich anknüpft. Aus diesem Blickwinkel wird zwischen objekt- und subjektbezogenen Netzeigenschaften differenziert.

Statische Netzeigenschaften hängen nur von der Topologie eines Netzmodells ab, nicht jedoch von der Netzmarkierung. Sie drücken daher Charakteristika der statischen Modellstruktur aus. Dynamische Netzeigenschaften betreffen dagegen die Menge erreichbarer Netzmarkierungen. Damit charakterisieren sie das gesamte Verhaltenspotential eines Netzmodells, d.h. seine dynamische Netzstruktur.

Objektbezogene Netzeigenschaften gelten für das jeweils modellierte Objekt unabhängig von der Art und Weise, in der sein Netzmodell vom Modellierungsträger angewendet wird⁵⁾. Es handelt sich um alle Eigenschaften, die sich an einem Netzmodell feststellen lassen, ohne daß auf Präferenzen des Modellierungsträgers Bezug genommen wird. Objektbezogene Netzeigenschaften können sowohl statischer als auch dynamischer Natur sein.

Subjektbezogene Netzeigenschaften rücken dagegen das Zielsystem in den Vordergrund, das ein Modellierungsträger mit der *Anwendung* eines *vorliegenden* Netzmodells⁶⁾ zu erfüllen trachtet. Dabei werden Modellanwendungen grundsätzlich mit Prozessen identifiziert, die in dem jeweils betrachteten Netzmodell durch Ausführungen von Schaltfolgen realisiert werden⁷⁾. Daher kommen als subjektbezogene lediglich die dynamischen Netzeigenschaften in Betracht. Aus dieser Perspektive gilt es zu untersuchen, in welchem Ausmaß die Schaltprozesse in einem Netzmodell den Zielsetzungen des Modellierungsträgers gerecht werden. Dadurch wird die pragmatische Modellierungsdimension erschlossen, die eingangs als letzte Komponente des semiotischen Bezugsrahmens für formale Modellierungskonzepte vorgestellt wurde. Zugleich lassen sich drei Analysekatégorien unterschieden, die mit der Untersuchung verschiedenartiger Zielkategorien in Entscheidungsmodellen⁸⁾ korrespondieren⁹⁾:

- Projektive Analysen untersuchen, wie sich *ein* zulässiger Schaltprozeß im Netzmodell auf die Zielsetzungen des Modellierungsträgers auswirkt. Hierdurch läßt sich vor allem entscheiden, ob ein Netzmodell Satisfizierungszielen gerecht wird. Darüber hinaus kann im Rahmen der projektiven Modellanalyse untersucht werden, ob ein denkmöglicher Schaltprozeß im Netzmodell überhaupt zulässig ist¹⁰⁾. Dies gestattet die Überprüfung, ob das allgemeine Formalziel der Konsistenz erfüllt wird.

- Komparative Analysen vergleichen die Zielerreichungsprofile *mehrerer*, aber nur endlich vieler zulässiger Schaltprozesse miteinander, die sich gegenseitig ausschließen¹¹⁾. Falls dabei das Zielerreichungsprofil eines Schaltprozesses fest vorgegeben ist, ist es auf diese Weise möglich zu untersuchen, ob alternative Schaltprozesse Meliorisierungsziele erfüllen.
- Optimierende Analysen bestimmen in der Menge *aller* Schaltprozesse¹²⁾, die in einem Netzmodell überhaupt zulässig sind, mindestens einen oder die Teilmenge aller optimalen Schaltprozesse. Im ersten Fall wird von einer schwachen, im zweiten von einer starken Netzoptimierung gesprochen. Solche Netzoptimierungen erlauben es, das Optimierungsinteresse eines Modellierungsträgers zu befriedigen.

Projektive und komparative Analysen umschließen eine Analyseform, die sich in der betrieblichen Praxis großer Beliebtheit erfreut. Es handelt sich um die Beantwortung von "What if"-Fragen¹³⁾. Zu diesem Zweck werden die Konsequenzen von Entscheidungsalternativen auf das Zielsystem eines Entscheidungsträgers "projiziert" und - gegebenenfalls - miteinander verglichen¹⁴⁾.

Die Projektionen und Komparationen können einerseits erfolgen, um spätere Entscheidungen zugunsten einer der betrachteten Koordinierungsalternativen zu begründen. Dann werden die Modelle der untersuchten Produktionssysteme als *Entscheidungsmodelle* behandelt, die sich durch einen generellen Optimierungsverzicht auszeichnen. Andererseits lassen sich projektive und komparative Modellanalysen ebenso benutzen, um das koordinierungsrelevante Verhalten eines komplexen Produktionssystems zu erforschen. In diesem Fall dient die Untersuchung von "What if"-Fragen noch nicht der unmittelbaren Vorbereitung von Koordinierungsentscheidungen. Vielmehr erfüllt das Modell eines Produktionssystems dann die Funktion eines *Orientierungsmodells*. Seine Exploration durch "What if"-Fragen zielt darauf ab, zunächst nur ein verbessertes Verständnis für die Eigenarten des modellierten Produktionssystems zu erlangen¹⁵⁾. Dieses Orientierungswissen kann aber später genutzt werden, um anhand eines Entscheidungsmodells konkrete Koordinierungsentscheidungen zu treffen.

Bei der projektiven "What if"-Analyse eines Netzmodells wird in der Regel¹⁶⁾ eine fest vorgegebene Ausgangsmarkierung betrachtet. Die Ausgangsmarkierung repräsentiert die aktuelle Entscheidungssituation. Entscheidungen, die dem Modellierungsträger in dieser Situation offenstehen, werden durch die Schaltschritte angezeigt, die unter der Ausgangsmarkierung des Netzmodells aktiviert sind. Der Modellierungsträger kann einen aktivierten Schaltschritt auswählen und im Netzmodell tatsächlich ausführen. Es wird festgestellt, wie sich die Schaltschrittausführung und die unmittelbar eintretende Folgemarkierung auf das Zielsystem des Modellierungsträgers auswirken¹⁷⁾. Diese Projektion der Zielwirksamkeit einer Schaltschrittausführung läßt sich auf die Schaltschritte ausdehnen, die unter der Folgemarkierung aktiviert sind (usw.). Auf diese Weise ist es möglich, sowohl die unmittelbare Wirkung einer Entscheidung als auch ihre möglichen Folgewirkungen¹⁸⁾ zu ermitteln.

Die projektive Analyse der unmittelbaren oder auch mittelbaren Zielwirksamkeit einer Entscheidung kann auf die komparative Analyse mehrerer Entscheidungen ausgedehnt werden. Dies ist immer dann möglich, wenn unter der Ausgangsmarkierung des Netzmodells mehrere Schaltschritte aktiviert sind. In diesem Fall wird für Auswahlentscheidungen zugunsten unterschiedlicher Schaltschritte jeweils auf projektive Weise bestimmt, wie sie sich auf das Zielsystem des Modellierungsträgers auswirken. Die Zielwirksamkeiten der alternativen Auswahlentscheidungen, die unter derselben Ausgangsmarkierung offenstehen, lassen sich dann miteinander vergleichen¹⁹⁾.

Komparative und optimierende Analysen setzen voraus, daß Auswahlkonflikte zwischen mehreren alternativen Schaltprozessen vorliegen. Solche Prozeßoptionen erstrecken sich im Rahmen des Petrinetz-Konzepts grundsätzlich auf die Indeterminiertheit des Netzverhaltens. Die unterschiedlichen Quellen dieser Indeterminiertheit wurden bereits früher als Auswahlkonflikte identifiziert und systematisiert, die zwischen alternativen Schaltoptionen für die Transitionen eines Petrinetzes bestehen²⁰⁾. Fortan wird für alle komparativen und optimierenden Netzanalysen

unterstellt, daß im jeweils vorgegebenen Netzmodell mindestens ein solcher Schaltkonflikt existiert²¹⁾. Dann steht es in der Dispositionsfreiheit des Modellierungsträgers, wie er die Schaltkonflikte des Netzmodells auflöst. Die Weise, in der dies erfolgt, wurde bereits als Konfliktstrategie eingeführt. Daher läßt sich der Subjektbezug der Netzanalyse bei Meliorierungszielen dadurch konkretisieren, daß nach Konfliktstrategien gesucht wird, die der Interessenlage eines Modellierungsträgers näher kommen als andere Strategien. In analoger Weise wird bei Optimierungszielen nach bestmöglichen Konfliktstrategien geforscht.

Anmerkungen zum Kapitel:

- 1) Die Begriffe "Analyse", "Auswertung" und "Untersuchung" werden in dieser Arbeit synonym verwendet.
- 2) Eigenschaften von Netzmodellen werden fortan auch kurz als Modell- oder Netzeigenschaften bezeichnet.
- 3) Zum analytischen Potential werden also nur solche Netzeigenschaften gerechnet, die sich durch entsprechende Auswertungstechniken tatsächlich untersuchen lassen. Eine Netzeigenschaft, die zwar definiert, aber nicht untersucht werden kann, gehört dagegen nicht zum analytischen Potential des Petrinetz-Konzepts. Die Erweiterung des Petrinetz-Konzepts um neuartige Auswertungstechniken kann durchaus dazu führen, daß sich Netzeigenschaften erstmals untersuchen lassen. Auf diese Weise wird das analytische Potential erweitert.
- 4) Die vorgestellten Netzeigenschaften werden daher auch als produktionswirtschaftlich interessante - oder kurz: interessante - Eigenschaften von Netzmodellen thematisiert. Bei der Erörterung der einzelnen Eigenschaften wird jeweils angedeutet werden, in welcher Hinsicht sie für die Modellierung von Produktionssystemen und -prozessen bedeutsam sein können.
- 5) Dabei wird vorausgesetzt, daß das Netzmodell bereits vorliegt. Daher werden die vielfältigen Einflüsse, die vom Modellierungsträger auf die Modellkonstruktion ausgehen können, *hier* nicht mehr berücksichtigt. Sie wurden bereits an früherer Stelle grundsätzlich gewürdigt. In Anerkennung der Teilhabe des Modellierungsträgers an der Modellkonstruktion beanspruchen die objektbezogenen Eigenschaften eines Netzmodells aber keineswegs, "objektive" - d.h. subjektunabhängige - Netzeigenschaften darzustellen.
- 6) Es geht hier nicht um Entscheidungen eines Modellierungsträgers, aus alternativen Netzmodellen eines auszuwählen oder zwischen einem Netzmodell und anderen Objektmodellen, die nicht auf der Basis des Petrinetz-Konzepts konstruiert wurden, zu wählen. Vielmehr interessieren nur Einflußmöglichkeiten auf den Gebrauch von genau einem Netzmodell, das bereits als Abbildung des zu modellierenden Realproblems fixiert wurde.
- 7) Jede Prozeßausführung wird durch eine zugrundeliegende Schaltfolge und die - implizit vorausgesetzte - Ausgangsmarkierung eindeutig determiniert. Daher ist es zulässig, fortan Modellanwendungen in synonyme Weise sowohl mit Prozessen als auch mit Schaltfolgen zu identifizieren. Ebenso kann von Schaltprozessen geredet werden. Vgl. zur Definition von Prozessen und Schaltfolgen für Petrinetze die früheren Festlegungen im Kontext von Stelle/Transition-Netzen.
- 8) Der Einfachheit halber wird nachfolgend zwischen Modellierungs- und Entscheidungsträgern nicht unterschieden. Damit wird aber nicht ausgeschlossen, daß bei Modellanwendungen beide Personengruppen auseinanderfallen können.
- 9) Bei der nachfolgenden Differenzierung wird vorausgesetzt, daß im untersuchten Netzmodell jeweils mindestens ein Schaltprozeß zulässig ist. Andernfalls läßt sich keine der drei Analysekatgorien anwenden. Ein solcher Ausnahmefall liegt immer dann vor, wenn ein Netzmodell unter seiner Ausgangsmarkierung tot ist. Diese Netzeigenschaft wird in Kürze präzisiert. Darüber hinaus werden hier die Modellanalysen vornehmlich auf zulässige Schaltprozesse bezogen. Dies ist insofern berechtigt, als im Zusammenhang mit den ebenso angesprochenen Formalzielkategorien nur solche Schaltprozesse interessieren, die in den netzartigen Entscheidungsmodellen zulässige Entscheidungsalternativen darstellen. Es läßt sich jedoch durchaus vorstellen, im Rahmen der Modellanalysen ebenso denkmögliche Schaltprozesse hinsichtlich ihrer Zulässigkeit zu untersuchen. Dies wird anläßlich der projektiven Modellanalyse im Hinblick auf Konsistenzziele ausdrücklich hervorgehoben.
- 10) Damit wird vor dem Hintergrund aller horizontalen und vertikalen Restriktionen des modellierten Entscheidungsfelds geprüft, ob der Schaltprozeß eine zulässige Entscheidungsalternative darstellt.
- 11) In Netzmodellen, in denen nur *endlich* viele verschiedene Schaltprozesse zulässig sind, kann die komparative Modellanalyse mit der nachfolgenden optimierenden Modellanalyse zusammenfallen. Dies ist genau dann der Fall, wenn sich die komparative Analyse auf die Menge aller zulässigen Schaltprozesse erstreckt. Im Regelfall decken komparative Analysen aber nur eine echte Teilmenge derjenigen Schaltprozesse ab, die in einem Netzmodell zulässig sind. Das gilt notwendig immer dann, wenn im untersuchten Netzmodell unendlich viele zulässige Schaltprozesse definiert sind. Denn dann kann die komparative Modellanalyse, die sich per definitionem auf den Vergleich von *endlich* vielen Schaltprozessen erstreckt, unmöglich die Menge aller zulässigen Schaltprozesse erfassen. Komparative Modellanalysen verhalten sich auch nicht disjunkt gegenüber der projektiven Analysevariante. Statt dessen kann jede komparative Modellanalyse als der Vergleich von mindestens zwei und höchstens endlich vielen projektiven Modellanalysen aufgefaßt werden. Da projektive Modellanalysen auch die Untersuchung denkmöglicher Schaltprozesse hinsichtlich ihrer Zulässigkeit umfassen, schließen komparative Analysen diese Untersuchungsmöglichkeit ebenso ein.
- 12) Es erfolgt keine Einschränkung dieser Prozeßmenge. Sie darf daher auch unendlich viele zulässige Schaltprozesse enthalten.

13) Vgl. zur Beliebtheit von "What if"-Fragen MÜLLER-MERBACH (1978), S. 414 u. 428; GODIN (1978), S. 334; GEOFFRION (1980), S. 26ff.; MEYER ZU SELHAUSEN (1980b), S. 135 (dort als "Wenn - Dann - Modell" angesprochen); PRESSMAR (1982), S. 340 u. 344; WYNNE (1982), S. 34; SCHEER (1983b), S. 153; DIRUF (1984), S. 123; ROSENFELD (1985), S. 90ff., insbesondere S. 91; FREDMAN (1988b), S. 330; ERICKSON (1989), S. 291; GARRATT (1990), S. 26ff.; STAUDINGER (1990), S. 173; HAUN (1990), S. 14; GLOVER (1990), S. 12 u. 15; WINTER,RO. (1991), S. 193 u. 237; BACK-HOCK (1991a), S. 98. Vgl. auch - mit speziellem Bezug auf Petrinetze - WINAND (1980), S. 1252, und ROSENSTENGEL (1991), S. 46 u. 100. Zum Typ der "What if"-Fragen gehört auch "EDV-unterstütztes Probieren", auf das sich z.B. im Kontext von PPS-Systemen HEB-KINZER (1979), Sp. 1991, beruft.

Noch weiter als "What if"-Fragen reichen die eng verwandten "What to do to achieve"- oder "How to achieve"-Fragen. Dort wird ein erwünschter Endzustand vorgegeben und danach gefragt, wie dieser Endzustand erreicht werden kann. Diese Fragevariante wird in den vorgenannten Quellen des öfteren gleichrangig mit den "What if"-Fragen angeschnitten; vgl. z.B. SCHEER (1983b), S. 153; WINTER,RO. (1991), S. 193. Dennoch werden "How to achieve"-Fragen im folgenden nicht ausdrücklich gewürdigt. Dieses Außerachtlassen beruht im wesentlichen auf zwei Gründen. Entweder handelt es sich bei "How to achieve"-Fragen um eine verkappte Iteration von "What if"-Fragen: Die letztgenannten Fragen werden so lange variiert, bis ihre Antwort ein Ergebnis aufzeigt, das von der zugrundeliegenden "How to achieve"-Frage intendiert wurde. Die Variation der Anfragen erfolgt dabei systematisch als der Intuition des Fragestellers folgend. In diesem ersten Fall liegt - abgesehen vom hinzukommenden repetitiven Moment - keine wesentliche Abweichung von konventionellen "What if"-Fragen vor. Oder aber der Fragesteller beschränkt sich auf genau eine "How to achieve"-Anfrage. Dies entspricht der Vorgabe eines Sachziels, das den erwünschten Endzustand spezifiziert. Eventuell können noch ergänzende Satisfizierungs-, Meliorisierungs- oder Optimierungsziele hinzukommen. Dieser zweite Fall wird durch jene Auswertungskonzepte abgedeckt, die hier für die sach- und formalzielbezogene Analyse von Netzmodellen entfaltet werden. Daher besteht kein Bedarf, "How to achieve"-Fragen gesondert zu behandeln.

14) Daher lassen sich zum Problemtyp der "What if"-Fragen auch alle betriebswirtschaftlichen Ermittlungsprobleme rechnen, sofern sie auf das Zielsystem eines Entscheidungsträgers bezogen werden. Dazu gehören z.B. die Kosten- und Leistungsprognosen von DIRUF (1984), S. 119 u. 121.

15) Diesem Gedanken entspricht der Ansatz, Modelle als "flexible Lernvehikel" einzusetzen. Vgl. dazu DIRUF (1983), S. 237ff.; DIRUF (1984), S. 125f.

16) Eine abweichende Variante findet sich bei WINAND (1980), S. 1252, und ROSENSTENGEL (1991), S. 45f., 81f., 90f. u. 100. Dort werden hypothetische Entscheidungen eines Modellierungsträgers dadurch erfaßt, daß für das vorliegende Netzmodell alternative Ausgangsmarkierungen zugelassen werden. Dann wird untersucht, wie im Netzmodell zulässige Schaltprozesse auf hypothetische Variationen der Ausgangsmarkierung reagieren. Insbesondere läßt sich dabei feststellen, wie sich die entscheidungsbedingten Variationen der Ausgangsmarkierung in entsprechend veränderten Zielerreichungsprofilen der Schaltprozesse niederschlagen. Auf diese Weise erfolgt von vornherein eine komparative Analyse. Denn es werden die Zielwirksamkeiten von Entscheidungen zwischen unterschiedlichen Ausgangsmarkierungen miteinander verglichen.

Allerdings ist zu beachten, daß schon eine kleine Veränderung der Ausgangsmarkierung zu erheblichen Modifizierungen der Menge zulässiger Schaltprozesse führen kann. Netzmodelle erfüllen in dieser Hinsicht nur das Prinzip schwacher Kausalität. Folglich kann der Fall eintreten, daß ein Schaltprozeß, der unter einer Ausgangsmarkierung noch zulässig war, unter einer veränderten Ausgangsmarkierung unzulässig wird. Dann muß die Zielwirksamkeit der variierten Ausgangsmarkierung anhand eines neuen zulässigen Schaltprozesses bewertet werden. In der Regel kommt dafür aber eine große Vielfalt von zulässigen Schaltprozessen in Betracht. Daher erweist sich die Bewertungsbasis bei veränderten Ausgangsmarkierungen unter Umständen als hochgradig unterbestimmt. Zwar könnte daran gedacht werden, einen "möglichst ähnlichen" Schaltprozeß auszuwählen. Aber es bleibt - zumindest in den o.a. Quellen - offen, anhand welches Maßstabs die Prozeßähnlichkeit gemessen werden sollte. Darüber hinaus ist nicht geklärt, ob die Entscheidungsrelevanz eines solchen Maßstabs gerechtfertigt werden kann. Aufgrund der voranstehend skizzierten Schwierigkeiten verfolgt der Verf. den Vorschlag von ROSENSTENGEL und WINAND, wie "What if"-Analysen in Netzmodellen durchgeführt werden sollten, nicht weiter.

17) Die Schaltschrittausführung kann sich durch Schaltwerte, die den zugehörigen Transitionen zugeordnet sind, auf das Formalzielsystem auswirken. Die Folgemarkierung vermag zur Erreichung von Sachzielen beizutragen, indem sie einer erwünschter Endmarkierung näher kommt oder mit ihr sogar zusammenfällt.

18) Die Folgewirkungen können der initialen Auswahlentscheidung, die unter der Ausgangsmarkierung zugunsten eines aktivierten Schaltschritts getroffen wurde, im allgemeinen nicht mehr eindeutig zugerechnet werden. Denn sie beruhen ebenso auf den Folgeentscheidungen, aus denjenigen Schaltschritten, die unter den Folgemarkierungen aktiviert sind, jeweils genau einen auszuwählen. Vgl. zu solchen Folgeentscheidungen das instruktive Beispiel von KLEINER,F. (1991), S. 101 u. 105. Er zeigt auf, welche Folgeentscheidungen anstehen können, wenn auf die Herstellung einer Produktart verzichtet wird. Die Folgeentscheidungen erstrecken sich z.B. darauf, welche Arbeitsgänge jenen Bearbeitungsstationen zugeordnet werden können, die zuvor von jenen Arbeitsgängen belegt wurden, die zur

Herstellung der nunmehr eliminierten Produktart erforderlich waren. Vgl. zu ähnlichen Folgeentscheidungen KLEINER, F. (1991), S. 153 i.V.m. S. 139ff. u. 151f.

Aufgrund solcher Folgeentscheidungen besteht ein Freiheitsgrad, wie die Folgewirkungen einer ursprünglichen Entscheidung bestimmt werden sollen. Für diesen Zweck bieten sich drei Alternativen an. Erstens ist es möglich, "typische" Folgeentscheidungen zu unterstellen. Diese Variante wird bei praktischen "What if"-Analysen oftmals angewendet. Allerdings bleibt dabei die inhaltliche Füllung dessen, was als "typisch" betrachtet wird, im Dunkeln. Zweitens kann von allen zulässigen Folgeentscheidungen jeweils eine Gesamtheit von Folgeentscheidungen ausgewählt werden, die angesichts eines vorausgesetzten Zielsystems optimal sind. Diese Vorgehensweise liegt der adaptiven "What if"-Analyse zugrunde. Drittens läßt sich vorstellen, daß die Folgewirkungen einer Entscheidung die Zielwirksamkeiten von *allen* Schaltprozessen umfassen, die nach dem Treffen der initialen Auswahlentscheidung im Netzmodell zukünftig noch zulässig sind. Es wird dann ein informationsreiches, aber mehrdeutiges Bündel von verursachungsgerecht ermittelten Folgewirkungen berücksichtigt.

19) Die Auswahl von Entscheidungsalternativen auf der Basis solcher Vergleichsergebnisse entspricht den Ansätzen der inkrementellen Planungskonzepte und der komparativen Verbesserung. Sie wurden bereits anläßlich der entscheidungstheoretischen Rahmenlegung erwähnt.

20) Solche Auswahlkonflikte bestehen immer dann, wenn unter einer Netzmarkierung mindestens ein Schaltschritt aktiviert ist. Jeder dieser Schaltschritte kann zunächst vom Modellierungsträger ausgeführt oder aber unterlassen werden. Dies folgt aus der Permissivität der Schaltregel von Petrinetzen. Falls unter derselben Markierung mehrere Schaltschritte aktiviert sind, kommen noch die Auswahlkonflikte zwischen diesen alternativen Schaltschritten hinzu. Das umfangreiche Spektrum solcher Auswahlkonflikte wurde bereits an früherer Stelle dargelegt und durch die Einführung verschiedener Schaltfarben erweitert.

21) Andernfalls erweist sich die subjektbezogene Netzanalyse als gegenstandslos.

5.2.2 Definition von Netzeigenschaften

5.2.2.1 Statische Eigenschaften

Das Petrinetz-Konzept bietet eine breite Palette statischer Netzeigenschaften an, um die Topologie von Netzmodellen zu charakterisieren¹⁾. Doch diese Netzeigenschaften sind in erster Linie nur von theoretischem Interesse. Im Vordergrund stehen dabei zwei Anwendungsmöglichkeiten²⁾. Entweder werden die Charakterisierungen von Netztopologien genutzt, um Netzklassen mit jeweils ähnlichen Topologien zu bilden. Oder es wird versucht, aus der Kenntnis von statischen Netzeigenschaften auf das Vorliegen oder Fehlen von dynamischen Netzeigenschaften zu schließen. Falls letztes möglich ist, kann die aufwendigere Untersuchung des Verhaltenspotentials eines Netzes durch die einfachere Analyse seiner Topologie ersetzt werden³⁾. Bislang konnten Erkenntnisse über Zusammenhänge zwischen statischen und interessanten⁴⁾ dynamischen Netzeigenschaften jedoch nur für Netzklassen mit sehr einfachen Topologien gewonnen werden. Daher hat die Rückführung dynamischer auf statische Netzeigenschaften noch keine nennenswerte praktische Bedeutung erlangt.

Die Untersuchung statischer Netzeigenschaften spielt in dieser Arbeit keine größere Rolle. An der reinen Klassifizierung von Netzmodellen besteht - abgesehen von zwei Ausnahmen⁶⁾ - kein Interesse. Die Ableitung dynamischer aus statischen Netzeigenschaften wäre zwar willkommen, um den Aufwand für Modellanalysen möglichst gering zu halten. Doch steht ihr die Prämisse entgegen, ein Modellierungskonzept mit möglichst großer Ausdrucksmächtigkeit anzustreben. Denn unter dieser Voraussetzung wäre es verfehlt, sich von vornherein auf Netzklassen mit stark beschränktem topologischen Strukturierungsmöglichkeiten festzulegen. Leider stehen für Netze, die das topologische Ausdrucksvermögen von Allgemeinen Netzen vollständig ausschöpfen dürfen, entsprechende Erkenntnisse über den Reduktionszusammenhang zwischen dynamischen und statischen Netzeigenschaften noch nicht zur Verfügung.

Für die hier vorgelegten Untersuchungen sind nur zwei statische Netzeigenschaften von Bedeutung: die Freiheit von 1-Zyklen und das Fehlen von Stellen mit mehreren Ein- oder Ausgangskanten. Alle Netze, deren Topologien keine 1-Zyklen enthalten, bilden die Klasse der reinen Netze. Andernfalls wird von unreinen Netzen gesprochen. Auf die Bedeutung von 1-Zyklen für das Schaltverhalten von Netzen wurde schon früher eingegangen. Auf sie wird noch einmal im Zusammenhang mit der Invariantenanalyse zurückgekommen.

Neu eingeführt werden dagegen Netze, deren Stellen jeweils höchstens eine Ein- und höchstens eine Ausgangstransition besitzen⁷⁾. Solche Netze werden zur Klasse der Synchronisationsnetze⁸⁾ zusammengefaßt. In dieser Arbeit wird die Klasse der Synchronisationsnetze noch enger gefaßt: Sie betrifft nur Netze mit unverzweigten Stellen, die zur Klasse der Stelle/Transition-Netze gehören⁹⁾. Synchronisationsnetze erfreuen sich in der Netzliteratur einer großen Beliebtheit¹⁰⁾. Denn ihre besonders einfache topologische Struktur gestattet es, zahlreiche dynamische Eigenschaften mit relativ¹¹⁾ geringem Aufwand zu untersuchen¹²⁾. Insbesondere sind für Synchronisationsnetze mehrere Theoreme bekannt, die es gestatten, aus ihrer statischen Struktur und ihrer Ausgangsmarkierung abzuleiten, ob interessante dynamische Netzeigenschaften erfüllt werden. Dies gilt insbesondere für die Untersuchung ihrer Deadlockfreiheit oder Lebendigkeit¹³⁾. Dennoch wird in dieser Arbeit die Einschränkung auf Synchronisationsnetze bewußt ausgeschlossen. Denn Synchronisationsnetze zeichnen sich dadurch aus, daß ihre Topologie die Modellierung von Ressourcenkonflikten grundsätzlich nicht zuläßt¹⁴⁾. Gerade diese Ressourcenkonflikte stellen aber jene Konfliktkategorie¹⁵⁾ dar, die für betriebswirtschaftliche Modellierungsaufgaben besonders interessant ist. Daher ist es wesentlich, eine Verengung auf Synchronisationsnetze grundsätzlich nicht zuzulassen. Diese Festlegung wird später bei der Erweiterung von Synthetischen Netzen um stochastische Einflußgrößen eine herausragende Rolle spielen.

Anmerkungen zum Kapitel:

- 1) Vgl. zur Vielfalt statischer Netzeigenschaften PAGNONI (1990), S. 126ff.
- 2) Beide Anwendungsmöglichkeiten schließen sich keineswegs aus. Vielmehr kann die zweite auf der ersten aufbauen. Vgl. dazu die nachfolgende Anmerkung.
- 3) Auf diese Weise lassen sich die beiden Anwendungsmöglichkeiten der Analyse statischer Netzeigenschaften miteinander kombinieren: Zunächst wird durch Bestimmung der Netztopologie die vorliegende Netzklasse ermittelt. Dann ist es im günstigen Fall möglich, durch genauere Analyse der Netztopologie und durch klassenspezifische Theoreme über den Zusammenhang zwischen statischen und dynamischen Netzeigenschaften zu erschließen, ob dynamische Netzeigenschaften vorliegen oder nicht.
- 4) In dieser Arbeit gelten diejenigen dynamischen Netzeigenschaften als interessant, die im Abschnitt 5.2.2.2 explizit definiert werden.
- 6) Sie werden im anschließenden Abschnitt behandelt.
- 7) Stellen, die diese topologische Anforderung erfüllen, werden auch als unverzweigte Stellen bezeichnet. Äquivalent kann auch davon gesprochen werden, daß diese Stellen jeweils höchstens eine Ein- und höchstens eine Ausgangskante besitzen.
- 8) In der Netzliteratur wird zumeist von Synchronisationsgraphen gesprochen. Vgl. z.B. ROSENSTENGEL (1991), S. 85 u. 115. Dies könnte zu dem Mißverständnis verleiten, mit den Synchronisations"graphen" wären mathematische Graphen gemeint. Da in dieser Arbeit deutlich zwischen Netzen und ihren zugehörigen mathematischen Graphen differenziert wurde, wird der Klarheit halber von vornherein von Synchronisationsnetzen geredet.
- 9) Diese zusätzliche Anforderung wird in der Netzliteratur zumeist implizit geteilt, weil dort Synchronisationsnetze in der Regel nur im Kontext von Stelle/Transition-Netzen thematisiert werden. Die Einschränkung auf Stelle/Transition-Netze läßt sich dadurch erklären, daß der tiefere Grund der Beliebtheit von Synchronisationsnetzen in ihrem einfachen Schaltverhalten liegt. Das Schaltverhalten eines Netzes erweist sich als besonders übersichtlich, wenn alle Komplizierungen, die aus der konfliktionären Aktivierung von Schaltschritten resultieren können, ausgeschlossen werden. Dafür reicht im Fall der Stelle/Transition-Netze die o.a. Eigenschaft aus, daß sie keine verzweigten Stellen besitzen. Denn in dieser Netzklasse können konfliktionäre Schaltschrittaktivierungen nur durch Ressourcenkonflikte zustande kommen. Solche Ressourcenkonflikte sind jedoch nur dann möglich, wenn mehrere Transitionen über gemeinsame Ein- oder Ausgangsstellen verfügen. Da dies notwendig zu verzweigten Stellen führt, wird die Möglichkeit konfliktionärer Schaltschrittaktivierungen in Synchronisationsnetzen grundsätzlich ausgeschlossen. In Höheren Netzen - z.B. in Synthetischen Netzen - können jedoch zwei Schaltschritte auch dadurch konfliktionär aktiviert sein, daß sie dieselbe Transition enthalten, die bezüglich alternativer Schaltfarben aktiviert ist. Dieser Fall ist auch dann möglich, wenn die betroffene Transition weder mehrere Ein- noch mehrere Ausgangsstellen besitzt. Er kann also auch in Höheren Netzen ohne verzweigte Stellen eintreten. Als Abundanzkonflikt wurde er schon früher thematisiert. Um solche konfliktionären Schaltschrittaktivierungen aus Synchronisationsnetzen auszuschließen, werden sie hier von vornherein auf die Klasse der Stelle/Transition-Netze eingeschränkt.
- 10) Vgl. zum Studium von Synchronisationsnetzen z.B. ROSENSTENGEL (1982), S. 78ff. u. 102ff.; ABEL,D. (1990), S. 30f.; ROSENSTENGEL (1991), S. 76, 79ff., 85ff., 100 u. 115ff.
- 11) Bezugspunkt sind alle Netze, deren Stellen mehrere Ein- oder Ausgangskanten besitzen können.
- 12) Vgl. dazu die Auswertung von Synchronisationsnetzen, die einfache Netzpläne abbilden.
- 13) Die dynamischen Netzeigenschaften der Deadlockfreiheit und Lebendigkeit werden alsbald näher erläutert. Vgl. zu Theoremen, welche die Deadlockfreiheit oder Lebendigkeit von Synchronisationsnetzen auf Eigenschaften ihrer Netztopologien und ihre Ausgangsmarkierungen zurückführen, ABEL,D. (1990), S. 31.
- 14) Vgl. ABEL,D. (1990), S. 31.
Im Kontext von Stelle/Transition-Netzen wurde aufgezeigt, daß eine notwendige Bedingung für die konfliktionäre Aktivierung von Transitionen, die um knappe Ressourcen konkurrieren, darin besteht, daß die Transitionen mindestens eine gemeinsame Ein- oder Ausgangsstelle besitzen. Diese gemeinsame Stelle weist dann mehrere Aus- bzw. Eingangskanten auf, welche die Stelle mit den konfliktionär aktivierbaren Transitionen verknüpft. Stellen mit mehreren Ein- oder Ausgangskanten werden jedoch in Synchronisationsnetzen nicht zugelassen. Folglich kann die notwendige Bedingung für die konfliktionäre Aktivierung mehrerer Transitionen auch niemals erfüllt sein. Daher ist es unmöglich, in Synchronisationsnetzen jemals einen Ressourcenkonflikt zu modellieren; q.e.d.
- 15) An früherer Stelle wurden die Konfliktkategorien vorgestellt, die für Petrinetze grundsätzlich in Betracht kommen.

5.2.2.2 Dynamische Eigenschaften

5.2.2.2.1 Überblick

Die dynamischen Netzeigenschaften, die das Verhaltenspotential eines Netzmodells charakterisieren, besitzen für die Netzanalyse eine weitaus größere Bedeutung als die zuvor behandelten statischen Eigenschaften. Insbesondere knüpfen alle subjektbezogenen Netzanalysen an dynamische Netzeigenschaften an. Erst diese Analysen eröffnen die pragmatische Modelldimension, in der auf Zielsetzungen der Modellierungsträger Bezug genommen werden kann. Daher liegt das Schwergewicht dieser Arbeit auf der Untersuchung von dynamischen Netzeigenschaften¹⁾. Von Interesse sind dabei die Eigenschaften der Beschränktheit, der Deadlockfreiheit und Lebendigkeit, der Livelockfreiheit, der Finalität, der Optimalität, des Terminierens, der Robustheit, der Promptheit sowie der Reversibilität. Sie werden auch gemeinsam als interessante dynamische Netzeigenschaften angesprochen. Eine Vielzahl anderer dynamischer Netzeigenschaften, die im Rahmen des Petrinetz-Konzepts definiert werden können, bleibt hier unberücksichtigt. Sie lassen keine Relevanz für die hier interessierende Prozeßkoordinierung in komplexen Produktionssystemen erkennen²⁾.

Anmerkungen zum Kapitel:

1) Diese Fokussierung äußert sich sowohl im Umfang der betrachteten dynamischen Eigenschaften als auch in dem Sachverhalt, daß im Abschnitt 5.2.3 ausschließlich Analysetechniken für dynamische Netzeigenschaften behandelt werden.

2) Dies wird - als pars pro toto - für die Eigenschaft der Persistenz, Konflikt- oder Entscheidungsfreiheit verdeutlicht. Ein Netz heißt persistent (konflikt- oder entscheidungsfrei), wenn alle seine Transitionen persistent sind. Eine einzelne Transition ist persistent, falls ihre Aktivierung nur durch ihr eigenes Schalten aufgehoben werden kann. Die Persistenz eines Netzes ist äquivalent damit, daß niemals mehrere Transitionen zueinander konfliktionär aktiviert sein können. Wäre dies nämlich der Fall, dann würde das Schalten einer solchen Transition die Aktivierung aller anderen zu ihr konfliktionär aktivierten Transitionen aufheben. Dies ist aber per definitionem unmöglich.

Persistenz, Konflikt- oder Entscheidungsfreiheit von einzelnen Transitionen und ganzen Netzen genießen in der Netzliteratur große Aufmerksamkeit; vgl. etwa PATIL (1971), S. 15; PATIL (1972a), S. 2 u. 9; KELLER, R. (1972a), S. 41; PATIL (1973), S. 59; HACK, M. (1975a), S. 25 u. 37ff.; CRESPI-REGHIZZI (1975), S. 78; ULLRICH (1976), S. 2/14; JONES, N. (1977), S. 279; LANDWEBER (1978), S. 352ff.; RAMMIG (1978), S. 38f.; HEIMERDINGER (1978), S. 162; AGERWALA (1978a), S. 152; MOALLA (1978a), S. 107; PAKAS-SKEWES (1979), S. 22, 24ff. u. 70ff.; AGERWALA (1979), S. 91; KWONG (1979), S. 186; PRIESE (1979), S. 27; STARKE (1979b), S. 424f.; PETRI, C. (1979d), S. 154; GRABOWSKI, J. (1980b), S. 20ff.; GRABOWSKI, J. (1980c), S. 19f. u. 25ff.; JANTZEN (1980a), S. 190 u. 193f.; RAMAMOORTHY (1980), S. 446f.; GENRICH (1980b), S. 529; STARKE (1980), S. 121f. u. 160; MÜLLER, H. (1980b), S. 429; MAYR, E. (1981c), S. 309f. u. 314ff.; YAMASAKI (1981), S. 94f.; GRABOWSKI, J. (1981b), S. 429; BURKHARD (1982a), S. 86; ROUCAIROL (1982b), S. 5; PRIESE (1983), S. 225.

KARP (1969), S. 182; YOELI (1973), S. 4 u. 6; PETRI, C. (1976b), S. 8; JONES, N. (1977), S. 279; LANDWEBER (1978), S. 352ff.; RAMAMOORTHY (1980), S. 443; und BATHELT (1982), S. 27, vertreten enger gefaßte Definitionen von konflikt- oder entscheidungsfreien Netzen, so daß diese dort eine echte Teilklasse der persistenten Netze darstellen; vgl. vor allem JONES, N. (1977), S. 279; LANDWEBER (1978), S. 352, 354 u. 359; RAMAMOORTHY (1980), S. 446; mittelbar auch PETRI, C. (1976b), S. 9.

Trotz des beachtlichen Interesses, das der Eigenschaft der Persistenz oder Konfliktfreiheit in der Netzliteratur zuteil wird, wird sie in dieser Arbeit aus dem gleichen Grund nicht untersucht, der schon anläßlich der Ausgrenzung von Synchronisationsnetzen angeführt wurde: Persistente Netzmodelle sind uninteressant, weil sie aufgrund ihrer Konfliktfreiheit nicht in der Lage sind, Ressourcenkonflikte abzubilden. Zugleich läßt sich anhand dieser Netzeigenschaft die o.a. Schwierigkeit verdeutlichen, die bei der Rückführung von dynamischen auf statische Netzeigenschaften besteht: Die dynamische Netzeigenschaft "Persistenz" wird mit Sicherheit von allen Netzen erfüllt, welche die statische Netzeigenschaft besitzen, zur Klasse der Synchronisationsnetze zu gehören. Vgl. dazu auch die Ausführungen bei PAKAS-SKEWES (1979), S. 24 i.V.m. S. 22, und ROUCAIROL (1982b), S. 5, zur Persistenz von Netzen, deren Transitionen keine gemeinsamen (Aus- oder) Eingangsstellen besitzen. Für Synchronisationsnetze läßt sich daher die Frage nach ihrer Persistenz ohne jede weitere Analyse positiv beantworten. Doch wird dieser Vorzug der Analyseeinfachheit damit erkauft, von einer Netzklasse ausgehen zu müssen, deren Topologie die Modellierung von Ressourcenkonflikten ausschließt. Dies ist für die Modellierung betriebswirtschaftlicher Problemstellungen im allgemeinen inakzeptabel.

5.2.2.2.2 Beschränktheit

Die Beschränktheit¹⁾ eines Netzes besitzt fundamentale Bedeutung für seine Erreichbarkeitsanalyse. Denn diese Analyse kann sich immer dann auf einen *endlichen* Erreichbarkeitsgraphen²⁾ stützen, wenn das zugrundeliegende Netz beschränkt ist³⁾.

Eine Stelle s_m heißt genau dann K_m -beschränkt, wenn sichergestellt ist, daß sich auf ihr unter keiner erreichbaren Markierung mehr als K_m Marken⁴⁾ befinden. Die Anzahl K_m der maximal vorhandenen Marken wird auch als obere Markenschranke der Stelle s_m bezeichnet. Ein Netz⁵⁾ wird als K -beschränkt bezeichnet, wenn alle Stellen aus einer Stellenmenge S jeweils K -beschränkt sind⁶⁾. Es wird auch einfach als beschränkt angesprochen, falls es eine obere Markenschranke besitzt, der konkrete Wert dieser Schranke aber nicht näher interessiert. Die K_m -Beschränktheit einer Stelle s_m , die K -Beschränktheit eines Netzes SN und die Beschränktheit eines Netzes SN lassen sich formal präzisieren durch die Prädikate $BES(s_m, K_m)$, $BES(SN, K)$ bzw. $BES(SN)$:

$$\begin{aligned}
 & BES(s_m, K_m) \\
 :\Leftrightarrow & \quad \forall (M_r \in RM(M_0)): M_r(s_m) \leq K_m \\
 & BES(SN, K) \\
 :\Leftrightarrow & \quad \forall (s_m \in S): BES(s_m, K) \\
 \Leftrightarrow & \quad \forall (s_m \in S) \forall (M_r \in RM(M_0)): M_r(s_m) \leq K \\
 & BES(SN) \\
 :\Leftrightarrow & \quad \exists (K \in \mathcal{N}_+): BES(SN, K) \\
 \Leftrightarrow & \quad \exists (K \in \mathcal{N}_+) \forall (s_m \in S) \forall (M_r \in RM(M_0)): M_r(s_m) \leq K
 \end{aligned}$$

Eine hinreichende Bedingung für die Beschränktheit eines Netzes ist es, daß alle seine Stellen endliche Markenkapazitäten besitzen⁷⁾. Dann sorgt die Schaltregel dafür, daß unter allen erreichbaren Markierungen auf keiner Stelle mehr Marken liegen, als durch ihre Markenkapazität erlaubt ist. Das Maximum aller stellenspezifischen Markenkapazitäten ist auf jeden Fall eine obere Markenschranke des Netzes⁸⁾.

Mit Hilfe der K -Beschränktheit kann z.B. für Einrichtungen mit Lagerfunktion ermittelt werden, welche Kapazitäten für sie ausreichen, damit sie ihre Funktion immer erfüllen können⁹⁾. Zu diesem Zweck wird der jeweils interessierende Realitätsausschnitt mit der Hilfe eines Netzmodells abgebildet, in dem die Lagereinrichtungen jeweils durch eine Stelle mit unbeschränkter¹⁰⁾ Markenkapazität dargestellt werden. Falls sich alle Stellen, die Lagereinrichtungen modellieren, jeweils als K_m -beschränkt erweisen, reicht es aus, jeder Stelle die Markenkapazität $KAP_m = K_m$ zuzuordnen. Dann kann kein Netzverhalten, das im ursprünglichen Netzmodell mit unbeschränkten Markenkapazitäten als Schaltprozeß zulässig war, im modifizierten Netzmodell mit beschränkten Markenkapazitäten dadurch unzulässig werden, daß es im modifizierten Netzmodell zum Überschreiten der Markenkapazität von mindestens einer Stelle führen würde. Daher ist die Funktionserfüllung der Lagereinrichtungen aus der Perspektive beschränkter Kapazitäten gewährleistet, wenn ihre realen Kapazitäten KAP_m^* ¹¹⁾ mindestens so groß ausgelegt werden, wie es die Markenkapazitäten KAP_m im Netzmodell angeben¹²⁾. Infolgedessen werden die Markenkapazitäten KAP_m auch als friktionsfreie Mindestkapazitäten bezeichnet¹³⁾.

Es kann sich aber auch herausstellen, daß die reale Kapazität KAP_m^* einer Lagereinrichtung kleiner ist, als es im Netzmodell für die Markenkapazität KAP_m der lagerrepräsentierenden Stelle s_m durch $KAP_m = K_m$ mindestens gefordert wird. In diesem Fall erweist sich die reale Kapazität

der Lagereinrichtung als ein Engpaß. Sowohl das Ausmaß als auch die Auswirkungen dieses Engpasses lassen sich anhand des Netzmodells exakt bestimmen. Die Engpaßgröße ergibt sich als Differenz zwischen der friktionsfreien Mindestkapazität KAP_m und der realen Kapazität KAP_m^* . Die Konsequenzen dieses Engpasses sind alle Schaltprozesse, die im Netzmodell mit der friktionsfreien Mindestkapazität KAP_m zulässig sind, aber im Netzmodell mit der realen Markkapazität KAP_m^* unzulässig werden. Darüber hinaus läßt sich die wirtschaftliche Vorteilhaftigkeit der Engpaßbeseitigung¹⁴⁾ untersuchen¹⁵⁾. Zu diesem Zweck können einerseits die Kosten ermittelt werden, die von der Erhöhung der realen Lagerkapazität um den Engpaßbetrag $KAP_m - KAP_m^*$ verursacht würden. Ihnen ist andererseits der Nutzenentgang gegenüberzustellen, der durch die engpaßbedingt unzulässigen Schaltprozesse entsteht¹⁶⁾.

Anmerkungen zum Kapitel:

1) Vgl. zur Beschränktheit von Stellen und Netzen MARIN (1975), S. 4/4f. u. 4/55; AGERWALA (1978a), S. 152; ABEL,D. (1990), S. 15f.

Oftmals wird ein beschränktes Netz in den vorgenannten Quellen als sicheres Netz bezeichnet. Mitunter wird der Begriff der Netzsicherheit aber auch für 1-beschränkte Netze reserviert; vgl. z.B. ABEL,D. (1990), S. 16. Von diesen Nuancierungen wird hier abgesehen, indem nur die Beschränktheit von Netzen thematisiert wird.

2) Die Vorteilhaftigkeit endlicher Erreichbarkeitsgraphen wird an späterer Stelle näher ausgeführt.

3) Vgl. ABEL,D. (1990), S. 15. Allerdings ist zu betonen, daß die Endlichkeit des Erreichbarkeitsgraphen eines Synthetischen Netzes nur dann eine logische Konsequenz seiner Beschränktheit ist, wenn die Definitionsbereiche aller Attribute endliche Mengen darstellen. Sonst könnte der Fall eintreten, daß eine zyklische Schaltfolge durchlaufen wird, bei der immer wieder neue Kopien von Attributmarken erzeugt werden, die noch unter keiner bisher erreichten Markierung vorhanden waren. Eine unendliche Fortsetzung dieser Schaltfolge würde zu einem unendlichen Erreichbarkeitsgraphen führen. Dieser Sonderfall wird jedoch dadurch ausgegrenzt, daß schon früher die Definitionsbereiche aller Attribute als endliche Objektmengen vorausgesetzt wurden.

4) Strenggenommen handelt es sich um Markenkopien. Aufgrund der früher eingeführten terminologischen Vereinfachung wird hier jedoch nur von Marken gesprochen.

5) Es werden hier nur Synthetische Netze betrachtet. Die nachfolgenden Definitionen gelten aber für alle Netzklassen, für die Netzmarkierungen definiert sind, in derselben Weise.

6) Dies muß keineswegs der Fall sein. Statt dessen können schon recht einfache Netze ein Verhaltenspotential besitzen, das die Markenanzahl mindestens einer seiner Stellen über jede endliche Schranke hinaus anwachsen läßt. Vgl. dazu das Beispielnetz aus Abb. 65; vgl. ebenso MARIN (1975), S. 4/6, Fig. IV-3, u. S. 4/55.

7) Diese Bedingung ist aber keineswegs notwendig. Denn ein Netz kann auch dann beschränkt sein, wenn nicht alle seine Stellen beschränkte Markenzapazitäten aufweisen. Denn die dynamische Struktur eines Netzes kann so ausgelegt sein, daß durch das Zusammenwirken von Netztopologie, Ausgangsmarkierung und Schaltregel nur solche Schaltprozesse zugelassen werden, die auf keiner Stelle die Markenanzahl über eine obere Schranke K mit $K \in \mathcal{N}_+$ hinaus anwachsen lassen.

8) Es muß sich nicht um die kleinstmögliche obere Markenschranke handeln. Denn es ist durchaus möglich, daß die maximale Markenzapazität unter keiner erreichbaren Markierung von irgendeiner Stellenmarkierung vollständig ausgeschöpft wird.

9) Ein ähnlicher, aber nicht weiter ausgeführter Gedanke findet sich bei ABEL,D. (1990), S. 43.

10) Statt dessen kann auch die größtmögliche beschränkte Markenzapazität gewählt werden, falls in einer Modellierungsumgebung nur endliche Markenzapazitäten definiert werden können.

11) Unter der realen Kapazität KAP_m^* einer Lagereinrichtung wird hier diejenige Markenzapazität verstanden, die eine lagerrepräsentierende Stelle s_m im Netzmodell besitzen müßte, um die tatsächliche Lagerkapazität korrekt wiederzugeben.

12) Darüber hinaus lassen sich auch abundant vorgehaltene Kapazitäten aufdecken. Dies ist immer dann der Fall, wenn die real vorhandene Kapazität KAP_m^* einer Lagereinrichtung deren friktionsfreie Mindestkapazität KAP_m übersteigt. Es steht zur Disposition eines Entscheidungsträgers, ob der Abundanzbetrag $KAP_m^* - KAP_m$ entweder abgebaut oder aber bewußt aufrechterhalten werden soll. Letztes kommt z.B. als Strategie im Rahmen eines passiven Störungsmanagements in Betracht.

13) Die Ermittlung friktionsfreier Lagerkapazitäten stimmt prima facie mit konventionellen Simulationsanalysen überein. In ihnen wird das Modell eines Produktionssystems mit alternativen Lagerkapazitäten simuliert. Durch den Vergleich der Simulationsergebnisse läßt sich feststellen, welche Lagerkapazität am vorteilhaftesten ist. Vgl. zu dieser Vorgehensweise z.B. KNOOP (1986), S. 166, 196ff. u. 215; MÜLLER,A. (1987), S. 353ff. (für die Ermittlung der Kapazitäten von Pufferlagern).

Allerdings lassen sich solche Simulationsstudien nur mit alternativen, jeweils fest vorgegebenen Lagerkapazitäten durchführen. Daher kann die friktionsfreie Kapazität eines Lagers im allgemeinen nicht auf Anhieb ermittelt werden. Statt dessen muß mit schrittweise variierten Lagerkapazitäten so lange experimentiert werden, bis die gesuchte friktionsfreie Lagerkapazität aufgefunden ist. In dieser Hinsicht zeigt sich die Auswertungsmöglichkeit von Netzmodellen überlegen. Denn die Überprüfung ihrer Beschränktheit gestattet es, in genau einem Durchlauf die friktionsfreie Lagerkapazität zu ermitteln.

14) Der Übersichtlichkeit halber wird hier nur die vollständige Engpaßbeseitigung thematisiert. In analoger Weise kann aber auch analysiert werden, wie sich jede unvollständige, aber ganzzahlige Verringerung des Engpaßausmaßes auf Kosten- und Nutzengrößen auswirken würde.

15) Diese Analysemöglichkeit klingt schon bei HERZOG,O. (1973), S. 3, an.

16) Der Nutzenentgang läßt sich ermitteln, indem die Transitionen mit Schaltwerten assoziiert werden. Das wird an anderer Stelle ausführlicher behandelt. Die Summe der Schaltwerte aller Transitionsschaltakte, die in einem der unzulässig gewordenen Schaltprozesse stattfinden würden, ergeben dann den Nutzenentgang des betrachteten Schaltprozesses.

5.2.2.2.3 Deadlocks, Lebendigkeit und Livelocks

Deadlockfreiheit und Lebendigkeit spielen für die Gestaltung oder Beurteilung der Prozeßkoordination in dynamischen Systemen eine herausragende Rolle¹⁾. Livelocks werden zwar weitaus seltener behandelt. Dennoch werden sie später ebenso berücksichtigt, weil sie mit den beiden vorgenannten dynamischen Netzeigenschaften konzeptionell eng verwandt sind.

Deadlocks, Lebendigkeit und Livelocks erweisen sich auch für die hier interessierende Koordination von Produktionsprozessen als bedeutsam. Darauf wird später in einem veranschaulichenden Beispiel näher eingegangen. In konventionellen produktionswirtschaftlichen Modellierungen finden die vorgenannten Systemeigenschaften allerdings keine Beachtung. Daher werden sie zunächst aus der Perspektive von Netzmodellen²⁾ ausführlicher erläutert.

Erschwerend wirkt sich aus, daß in der Netzliteratur eine Vielfalt unterschiedlicher Definitionsansätze für Dead- und Livelocks sowie für die Netzlebendigkeit existiert³⁾. Sie harmonieren nicht immer miteinander⁴⁾. Ein wesentlicher Grund dafür ist, daß die dynamischen Netzeigenschaften, die hier aus den Perspektiven von blockierenden Markierungen und lebendigen Transitionen entfaltet werden, dort auch unter anderen Blickwinkeln Berücksichtigung finden⁵⁾. Schließlich konkurrieren verschiedenartige terminologische "Schulen" miteinander⁶⁾. Leider ist es oftmals nicht möglich, das Verhältnis dieser Terminologien in übersichtlicher Weise auf die synonyme Verwendung unterschiedlicher Bezeichnungen zu reduzieren⁷⁾. Angesichts der vorgenannten Schwierigkeiten erscheint es unmöglich, in allgemein akzeptabler Weise die Dead- und Livelockfreiheit sowie die Lebendigkeit von Netzen zu definieren. Statt dessen nutzt der Verf. die Freiheit des nominalistischen Begriffskonzepts, jene Definitionsvarianten festzulegen, die für die Erkenntniszwecke dieser Arbeit ausreichen.

Ein Deadlock⁸⁾ ist eine erreichbare Markierung, unter der kein Schaltschritt aktiviert ist⁹⁾. A fortiori kann in einem Deadlock auch kein Schaltschritt ausgeführt werden¹⁰⁾. Da alle Netzverhaltensweisen Prozesse darstellen, in denen Netzmarkierungen und Schaltschrittausführungen alternieren, muß jedes Netzverhalten enden, sobald es eine Deadlockmarkierung erreicht hat¹¹⁾. Es wird plastisch vom "Verklemmen" oder "Blockieren" aller Prozesse gesprochen¹²⁾. Entsprechend heißt eine Deadlockmarkierung auch "tot"¹³⁾.

Die formale Definition eines Deadlocks bereitet insofern Schwierigkeiten, als die Gesamtheit aller denkmöglichen Schaltschritte betroffen ist. Diese Schaltschrittmenge stellt für Synthetische Netze ein kompliziertes Konstrukt dar¹⁴⁾. Es läßt sich jedoch der Sachverhalt nutzen, daß unter einer Markierung M_r *genau dann* überhaupt kein Schaltschritt aktiviert ist, wenn unter ihr keine Transition bezüglich einer beliebigen Schaltfarbe aktiviert ist¹⁵⁾. Für die Aktivierung einer Transition t_n durch mindestens eine - nicht näher bestimmte - Schaltfarbe wurde bereits das Aktivierungsprädikat $AKT(t_n, M_r)$ eingeführt. Daher läßt sich das Prädikat $DEAD(M_r)$, das die Deadlockeigenschaft der Markierung M_r ausdrückt, formal definieren durch:

$$DEAD(M_r) \\ :\Leftrightarrow \quad \forall(t_n \in T): \neg AKT(t_n, M_r)$$

Ein totes Netz¹⁶⁾ liegt vor, falls seine Ausgangsmarkierung M_0 einen Deadlock darstellt. Dann läßt sich nur der entartete Schaltprozeß ausführen, der vermittelt der Nullschaltfolge die Ausgangsmarkierung in sich selbst überführt¹⁷⁾. In einem solchen Netz ist niemals irgendein Schaltschritt aktiviert. Für das Prädikat $TOT(SN)$, das die Totheit eines Netzes SN ausdrückt, gilt:

$$\begin{aligned} & \text{TOT}(\text{SN}) \\ \Leftrightarrow & \text{DEAD}(\text{M}_0) \end{aligned}$$

Die Erreichbarkeitsmenge eines toten Netzes schrumpft auf seine Ausgangsmarkierung M_0 zusammen: $\text{RM}(M_0) = M_0$. Die Erreichbarkeitsrelation degeneriert zur leeren Menge¹⁸: $\text{RR}(M_0) = \emptyset$. Der Erreichbarkeitsgraph nimmt die extreme Gestalt eines isolierten Knotens für die Ausgangsmarkierung M_0 an. Schaltkanten existieren in diesem Graphen nicht. Entsprechend ist die Beschriftungsfunktion "bk" des Erreichbarkeitsgraphen nicht definiert¹⁹.

Falls eine erreichbare Markierung M_r keinen Deadlock darstellt, wird sie als lebendig bezeichnet. Dies wird durch das Prädikat $\text{LEB}(M_r)$ notiert²⁰. Aus der Negation des Deadlockprädikats folgt unmittelbar, daß eine Netzmarkierung genau dann lebendig ist, wenn unter ihr mindestens eine Transition aktiviert ist:

$$\begin{aligned} & \text{LEB}(M_r) \\ \Leftrightarrow & \neg \text{DEAD}(M_r) \\ \Leftrightarrow & \neg (\forall (t_n \in T): \neg \text{AKT}(t_n, M_r)) \\ \Leftrightarrow & \exists (t_n \in T): \text{AKT}(t_n, M_r) \end{aligned}$$

Ein Netz heißt deadlockfrei, wenn seine Erreichbarkeitsmenge keine Deadlockmarkierung enthält. Für das Prädikat $\text{DFR}(\text{SN})$, das die Deadlockfreiheit des Netzes SN feststellt, gilt:

$$\begin{aligned} & \text{DFR}(\text{SN}) \\ \Leftrightarrow & \forall (M_r \in \text{RM}(M_0)): \neg \text{DEAD}(M_r) \end{aligned}$$

Aus der Deadlockfreiheit eines Netzes folgt, daß unter jeder seiner erreichbaren Markierungen mindestens eine Transition aktiviert ist und deshalb auch geschaltet werden kann²¹). Dies ergibt sich unmittelbar durch Einsetzen der Definition des Lebendigkeitsprädikats für das negierte Deadlockprädikat:

$$\begin{aligned} & \text{DFR}(\text{SN}) \\ \Leftrightarrow & \forall (M_r \in \text{RM}(M_0)): \text{LEB}(M_r) \\ \Leftrightarrow & \forall (M_r \in \text{RM}(M_0)) \exists (t_n \in T): \text{AKT}(t_n, M_r) \end{aligned}$$

Deadlockfreiheit bedeutet ebenso, daß in einem Netz jeder endliche Schaltprozeß mindestens eine - beliebig weit reichende - Prozeßfortsetzung besitzt²²). Denn am Ende des endlichen Prozesses liegt eine Markierung vor, die in einem deadlockfreien Netz keine Deadlockmarkierung sein kann. Unter jener Markierung ist mindestens eine Transition aktiviert, durch deren Schalten sich der Prozeß fortsetzen läßt. Daraus resultiert eine neue Markierung, die wiederum keine Deadlockmarkierung darstellt usw. ad infinitum.

Die Netzeigenschaft der Deadlockfreiheit ist im Konzept der Netzmarkierungen verwurzelt, die jeweils einen Deadlock darstellen können oder nicht. Ein anderer konzeptioneller Ausgangspunkt liegt der Netzlebendigkeit zugrunde. Sie geht grundsätzlich von den Schaltmöglichkeiten einzelner Transitionen aus²³). Es wird jedoch später deutlich, daß sich die Deadlockfreiheit als ein zentraler Aspekt der Lebendigkeit von Netzen wiederfindet.

Eine Transition t_n heißt lebendig bezüglich einer erreichbaren Markierung M_r oder kurz M_r -lebendig²⁴), wenn von der Referenzmarkierung M_r aus durch Ausführen einer endlichen Schaltfolge mindestens eine Folgemarkierung M_f erreicht werden kann, unter der die Transition akti-

viert ist²⁵). Einen Spezialfall der M_r -Lebendigkeit liegt vor, wenn es sich bei der Referenz- um die Ausgangsmarkierung handelt: Für jede M_0 -lebendige²⁶ Transition existiert stets eine erreichbare Markierung $M_r \in \text{RM}(M_0)$, unter der die Transition aktiviert ist. M_r - und M_0 -lebendige Transitionen werden gemeinsam als lebendige Transitionen angesprochen. Für sie gilt in analoger Weise:

$$\begin{aligned} & \text{LEB}(t_n, M_r) \\ \Leftrightarrow & \exists(M_r \in \text{RM}(M_0)): \text{AKT}(t_n, M_r) \\ & \text{LEB}(t_n, M_0) \\ \Leftrightarrow & \exists(M_r \in \text{RM}(M_0)): \text{AKT}(t_n, M_r) \end{aligned}$$

Neben dem Spezialfall der M_0 -Lebendigkeit ist auch eine Verallgemeinerung der M_r -Lebendigkeit möglich. Die Generalisierung erfolgt durch das Konzept der Unsterblichkeit²⁷). Eine Transition wird als unsterblich bezeichnet, wenn sie bezüglich jeder erreichbaren Markierung M_r mit $M_r \in \text{RM}(M_0)$ M_r -lebendig ist. Für die Prädikate $\text{LEB}(t_n, M_r)$ und $\text{UNS}(t_n)$, welche die M_r -Lebendigkeit bzw. die Unsterblichkeit der Transition t_n ausdrücken, gilt:

$$\begin{aligned} & \text{UNS}(t_n) \\ \Leftrightarrow & \forall(M_r \in \text{RM}(M_0)): \text{LEB}(t_n, M_r) \\ \Leftrightarrow & \forall(M_r \in \text{RM}(M_0)) \exists(M_r \in \text{RM}(M_r)): \text{AKT}(t_n, M_r) \end{aligned}$$

Lebendigkeit und Unsterblichkeit einer Transition drücken aus, daß sich von mindestens einer bzw. von allen erreichbaren Netzmarkierungen aus jeweils mindestens eine endliche Schaltfolge so ausführen läßt, daß die betrachtete Transition am Ende der Schaltfolge aktiviert ist. Es liegt also eine mindestens einmal bzw. immer wieder aktivierbare Transition vor. Da jede aktivierbare Transition geschaltet werden kann, wenn das Ende ihrer aktivierenden Schaltfolge erreicht ist, gilt ebenso: Wenn eine Transition lebendig ist, dann kann sie mit Sicherheit irgendwann einmal geschaltet werden. Die Unsterblichkeit einer Transition bedeutet dagegen, daß sich die Transition immer wieder schalten läßt. Die Lebendigkeit oder Unsterblichkeit einer Transition sagt jedoch nichts darüber aus, ob die Transition dann, wenn sie unter einer erreichbaren Markierung aktiviert ist, auch tatsächlich geschaltet wird. Daher kann es durchaus dazu kommen, daß lebendige oder sogar unsterbliche Transitionen trotz ihrer erreichbaren Aktivierungen niemals geschaltet werden²⁸). Dies wird später anhand des Phänomens des "Verhungerns" verdeutlicht.

Eine Transition, die unter der Markierung M_r nicht lebendig ist, heißt bezüglich dieser Markierung blockiert oder kurz M_r -blockiert²⁹). Sie ist unter allen Folgemarkierungen, die sich von der Referenzmarkierung M_r aus durch beliebige endliche Schaltfolgen erreichen lassen, niemals aktiviert. Eine tote Transition³⁰) stellt den Sonderfall einer M_0 -blockierten Transition dar. Sie ist unter keiner erreichbaren Markierung M_r aktiviert. Deshalb ist eine tote Transition auch unter jeder erreichbaren Markierung M_r -blockiert³¹). Wenn das Prädikat $\text{BLO}(t_n, M_r)$ die M_r -Blockierung einer Transition t_n und das Prädikat $\text{TOT}(t_n)$ das Totsein dieser Transition ausdrücken, dann gilt:

$$\begin{aligned} & \text{BLO}(t_n, M_r) \\ \Leftrightarrow & \neg \text{LEB}(t_n, M_r) \\ \Leftrightarrow & \neg (\exists(M_r \in \text{RM}(M_r)): \text{AKT}(t_n, M_r)) \\ \Leftrightarrow & \forall(M_r \in \text{RM}(M_r)): \neg \text{AKT}(t_n, M_r) \end{aligned}$$

$$\begin{aligned}
& \text{BLO}(t_n, M_0) \\
:\Leftrightarrow & \quad \neg \text{LEB}(t_n, M_0) \\
\Leftrightarrow & \quad \neg (\exists (M_r \in \text{RM}(M_0)): \text{AKT}(t_n, M_r)) \\
\Leftrightarrow & \quad \forall (M_r \in \text{RM}(M_0)): \neg \text{AKT}(t_n, M_r) \\
& \text{TOT}(t_n) \\
:\Leftrightarrow & \quad \text{BLO}(t_n, M_0) \\
\Leftrightarrow & \quad \forall (M_r \in \text{RM}(M_0)): \neg \text{AKT}(t_n, M_r) \\
\Leftrightarrow & \quad \forall (M_r \in \text{RM}(M_0)): \text{BLO}(t_n, M_r)
\end{aligned}$$

M_r -Lebendigkeit und M_r -Blockiertheit einer Transition verhalten sich kontradiktorisch: Eine Transition ist genau dann M_r -lebendig, wenn sie nicht M_r -blockiert ist. Unsterblichkeit und Totheit einer Transition t_n stellen hingegen keine kontradiktorischen, sondern konträre Gegenpole dar. Denn jede Transition, die unter mindestens einer erreichbaren Markierung M_r aktiviert, aber unter mindestens einer der nachfolgenden Markierungen M_f mit $M_f \in \text{RM}(M_r)$ M_r -blockiert ist, erweist sich weder als tot noch als unsterblich³²).

Deadlocks lassen sich auf die M_r -Blockiertheit von Transitionen zurückführen³³. Aus dieser Perspektive liegt ein Deadlock M_r genau dann vor, wenn unter der erreichbaren Markierung M_r alle Transitionen eines Netzes M_r -blockiert sind³⁴:

$$\begin{aligned}
& \text{DEAD}(M_r) \\
:\Leftrightarrow & \quad \forall (t_n \in T): \text{BLO}(t_n, M_r)
\end{aligned}$$

Auf den ersten Blick weicht diese Deadlockdefinition von der oben präsentierten ab. Denn es wird nicht die fehlende Aktivierung, sondern die Blockiertheit aller Transitionen unter der Markierung M_r gefordert. In bezug auf eine *einzelne* Transition³⁵ fallen deren Nichtaktivierung und ihre M_r -Blockierung keineswegs zusammen. Denn eine Transition, die unter einer Markierung M_r nicht aktiviert ist, kann durchaus unter einer Folgemarkierung M_f wieder aktiviert sein. Dies ist immer dann der Fall, wenn unter der Referenzmarkierung M_r mindestens eine Schaltfolge ausgeführt werden kann, welche die Folgemarkierung M_f mit $M_f \in \text{RM}(M_r)$ und $\text{AKT}(t_n, M_f)$ hervorbringt. Diese spätere Aktivierungsmöglichkeit ist dagegen für jede M_r -blockierte Transition per definitionem ausgeschlossen. Trotzdem erweisen sich die beiden Deadlockdefinitionen, die sich einerseits auf Aktivierungen und andererseits auf Blockierungen von Transitionen erstrecken, als äquivalent. Dies liegt an der gemeinsamen Betrachtung *aller* Transitionen eines Netzes. Sie führt dazu, daß die universelle M_r -Blockierung eines Netzes äquivalent ist mit der universellen Nichtaktivierung aller Transitionen unter derselben Markierung M_r ³⁶.

$$\begin{aligned}
& \forall (t_n \in T): \text{BLO}(t_n, M_r) \\
\Leftrightarrow & \quad \forall (t_n \in T): \neg \text{AKT}(t_n, M_r)
\end{aligned}$$

Deswegen sind die beiden oben vorgelegten Varianten der Deadlockdefinition äquivalent. Es spielt keine Rolle, ob sie vom markierungsbezogenen Ansatz aus auf die fehlende Aktivierung aller Transitionen bezogen wird - oder ob sie aus der transitionenfundierte Perspektive aus der Blockierung aller Transitionen abgeleitet wird³⁷. Auch die oben eingeführte Totheit eines Netzes kann mit Hilfe der M_r -Blockiertheit in äquivalenter Weise³⁸ reformuliert werden. Ein Netz ist genau dann tot, wenn alle seine Transitionen unter der Ausgangsmarkierung M_0 blockiert - also tot - sind:

$$\text{TOT}(\text{SN})$$

$$:\Leftrightarrow \forall (t_n \in T): \text{TOT}(t_n)$$

Oftmals bilden Transitionen in einem Netzmodell aktive Komponenten ab, deren Aktivitäten im jeweils modellierten realen Objekt komponentenspezifische Funktionen erfüllen sollen. Falls solche Transitionen vorliegen, können die Transitionseigenschaften der Lebendigkeit und Blockierung in ihren verschiedenen Varianten wertvolle Erkenntnisse über die Funktionsweise der modellierten Objektkomponenten vermitteln³⁹⁾.

So vermag die M_r -Blockierung⁴⁰⁾ einer Transition darauf hinzuweisen, daß eine Komponente des abgebildeten Objekts nicht mehr in der gewünschten Weise funktioniert. Denn sie läßt sich spätestens ab der Netzmarkierung M_r niemals mehr schalten. Daher ist es unmöglich, ab Erreichen des Objektzustands "r" die Funktion der modellierten Objektkomponente noch zu erfüllen. Falls die Transition sogar M_0 -blockiert - also tot - ist, vermag die abgebildete Komponente ihre erwünschte Funktion sogar niemals zu erbringen. Daher bieten sich M_r -blockierte, insbesondere tote Transitionen dazu an, um in Netzmodellen Fehlfunktionen der modellierten Objekte aufzudecken⁴¹⁾. Beispielsweise kann eine tote Transition eine erwünschte Objektaktivität repräsentieren, die sich niemals verwirklichen läßt, weil die Koordinierung der Objektaktivitäten nicht in der gewünschten Weise erfolgt⁴²⁾. Oder die tote Transition stellt ein Ereignis dar, das genau dann stattfindet, wenn ein vorgegebenes Sachziel - wie etwa die Abwicklung eines Produktionsauftrags - verwirklicht ist. Da die tote Transition niemals schaltet, läßt sich auch das betrachtete Sachziel niemals erreichen. Auch darin manifestiert sich eine Fehlfunktion des modellierten Objekts⁴³⁾. Ebenso ist es möglich, daß die tote Transition eine Objektkomponente abbildet, deren Aktivitäten zur Erfüllung der intendierten Objektfunktionen überhaupt nicht benötigt werden. Die modellierte Objektkomponente liegt dann redundant oder sogar abundant⁴⁴⁾ vor⁴⁵⁾. Darüber hinaus läßt sich sogar vorstellen, daß die Totheit der Transition für eine korrekte Funktionsweise des modellierten Objekts notwendig ist⁴⁶⁾. Beispielsweise stellen netztheoretische "Fakten" spezielle Transitionen dar, die in einem korrekten Netzmodell niemals aktiviert sein dürfen und infolgedessen tot sein müssen. Mit Ihrer Hilfe lassen sich z.B. Integritätsbedingungen für Netzmodelle formulieren, deren Integritätsformeln stets ungültig sein sollen.

Die voranstehenden Beispiele belegen nicht nur, daß aus der Blockierung oder Totheit⁴⁷⁾ von Transitionen eine Fülle interessanter Einsichten in die Funktionsweise eines modellierten Objekts gewonnen werden können. Darüber hinaus verdeutlichen sie ebenso, daß die gleichen Transitionseigenschaften in unterschiedlichen Modellierungskontexten vollkommen verschiedene Bedeutungen für den Modellierungsträger besitzen können. Es ist daher auf der einen Seite unmöglich, allgemeine Interpretationsregeln für die analysierten Transitionseigenschaften aufzustellen. Statt dessen erfordert jeder Einzelfall von neuem, die Bedeutung der erkannten Netzeigenschaften im jeweils aktuellen Modellierungskontext zu ermitteln. Andererseits zeigen die Beispiele auch, wie flexibel sich Netzmodelle an variierende Modellierungsaufgaben anpassen lassen.

An dieser Stelle bietet es sich an, in einem kurzen Exkurs auf die Bedeutung der strengen semiotischen Strukturierung des hier verfolgten Modellierungskonzepts hinzuweisen. Zunächst wurden Netze als Stelle/Transition-Netze und als Synthetische Netze in rein formaler Weise definiert. Ohne die formale Konstitution von Netzen zu verändern, können sie herangezogen werden, um die verschiedensten, jeweils prädikatenlogisch beschriebenen Objekte zu modellieren. Dadurch lassen sich syntaktisch gleichartige Netze mit unterschiedlichsten denotationalen Semantiken kombinieren. Die derart materiell interpretierten Netzmodelle können dann mit den gleichen formalen Analysetechniken⁴⁸⁾ hinsichtlich ihrer charakteristischen Eigenschaften untersucht werden. Dies hat den Vorteil, bei der Netzanalyse zunächst von Komplizierungen absehen zu können, die aus der Berücksichtigung von materiellen Netzbedeutungen resultieren würden⁴⁹⁾. Statt dessen reicht es aus, sich auf wenige, rein syntaktisch definierte Analysetechniken zu kon-

zentrieren⁵⁰). Dennoch bietet sich die Möglichkeit, die Bedeutung der Analyseergebnisse vor dem einzelfallspezifischen Hintergrund des jeweils modellierten Objekts zu ermitteln. Dies wurde oben exemplarisch skizziert. Dadurch findet die denotationale Semantik der Netzmodelle ihre wünschenswerte Beachtung, ohne die formalen Analysetechniken zu belasten. Darüber hinaus lassen sich neben der denotational-semantischen Modelldimension auch pragmatische Aspekte beachten. Sie fließen in Netzmodelle ein, indem das Zielsystem des Modellierungsträgers explizit abgebildet wird. Das wurde schon kurz angedeutet und findet später noch ausführlichere Würdigung.

Die weitreichenden Interpretationsmöglichkeiten, die sich an die Blockierung und Lebendigkeit von Transitionen oder Markierungen anschließen, legen es nahe, diese Transitions- und Markierungseigenschaften zu Netzeigenschaften zu verallgemeinern. Bei dieser Generalisierung wird jeweils die Gesamtheit aller Transitionen eines Netzes oder die Menge aller seiner erreichbaren Markierungen betrachtet. Aus dieser generalisierenden Perspektive heraus werden im folgenden die Prädikate der transitionsschwachen, der markierungsschwachen und der starken Netzlebendigkeit entfaltet.

Ein Netz wird genau dann als transitionsschwach lebendig⁵¹) eingestuft⁵²), wenn alle seine erreichbaren Markierungen lebendig sind. Dies bedeutet, daß unter jeder erreichbaren Markierung des Netzes mindestens eine Transition aktiviert ist. Daher kann von seiner Ausgangsmarkierung aus niemals eine Deadlockmarkierung erreicht werden. Folglich ist jedes transitionsschwach lebendige Netz notwendig deadlockfrei⁵³). Die Umkehrung dieser Folgerung trifft ebenso zu⁵⁴). Wegen dieser Äquivalenz bezeichnen Deadlockfreiheit und transitionsschwache Lebendigkeit eines Netzes dieselbe dynamische Netzeigenschaft. Deshalb können beide Begriffe synonym verwendet werden⁵⁵). Für das Prädikat $LEB_{ts}(SN)$, das die Eigenschaft der transitionsschwachen Netzlebendigkeit ausdrückt, gilt:

$$\begin{aligned}
 &LEB_{ts}(SN) \\
 :\Leftrightarrow &\forall(M_r \in RM(M_0)): LEB(M_r) \\
 \Leftrightarrow &\forall(M_r \in RM(M_0)) \exists(t_n \in T): AKT(t_n, M_r) \\
 \Leftrightarrow &\forall(M_r \in RM(M_0)): \neg DEAD(M_r) \\
 \Leftrightarrow &DFR(SN)
 \end{aligned}$$

Aus der Deadlockfreiheit eines transitionsschwach lebendigen Netzes folgt, daß unter jeder erreichbaren Markierung mindestens ein Schaltschritt aktiviert ist⁵⁶). Das Verhalten eines lebendigen Netzes kann daher niemals durch Erreichen einer toten Markierung "absterben"⁵⁷). Statt dessen läßt es sich potentiell unendlich fortsetzen⁵⁸).

Die transitionsschwache Netzlebendigkeit schließt allerdings nicht aus, daß ein Netz einzelne tote Transitionen enthält⁵⁹). Daher rührt das präzisierende Attribut dieser zuerst eingeführten Variante der Netzlebendigkeit⁶⁰). Tote Transitionen können redundante oder abundante Komponenten eines modellierten Objekts oder mangelhafte Koordinierungen der Objektaktivitäten anzeigen. Daher besteht des öfteren ein Interesse daran, ebenso feststellen zu können, ob ein Netzmodell tote Transitionen enthält⁶¹). Diese Netzeigenschaft läßt sich anhand der markierungsschwachen Lebendigkeit überprüfen.

Ein Netz erweist sich genau dann als markierungsschwach lebendig⁶²), wenn alle seine Transitionen unter seiner Ausgangsmarkierung M_0 lebendig sind. In einem markierungsschwach lebendigen Netz gibt es daher keine M_0 -blockierten oder toten Transitionen⁶³). Folglich läßt sich für jede Transition mindestens eine Markierung erreichen, unter der die Transition aktiviert ist. Daher kann jede Transition irgendwann mindestens einmal geschaltet werden⁶⁴). Das Prädikat $LEB_{ms}(SN)$, das die markierungsschwache Lebendigkeit eines Netzes ausdrückt, ist definiert durch:

$$\begin{aligned}
& \text{LEB}_{\text{ms}}(\text{SN}) \\
:\Leftrightarrow & \quad \forall(t_n \in T): \text{LEB}(t_n, M_0) \\
\Leftrightarrow & \quad \forall(t_n \in T): \neg \text{BLO}(t_n, M_0) \\
\Leftrightarrow & \quad \forall(t_n \in T): \neg \text{TOT}(t_n) \\
\Leftrightarrow & \quad \forall(t_n \in T) \exists(M_f \in \text{RM}(M_0)): \text{AKT}(t_n, M_f)
\end{aligned}$$

Das präzisierende Attribut der markierungsschwachen Lebendigkeit verweist darauf, daß diese Lebendigkeitsvariante die Deadlockfreiheit eines Netzes nicht zu garantieren vermag. Denn es mag durchaus der Fall sein, daß zunächst jede Transition mindestens einmal geschaltet wird, dann aber eine Markierung erreicht wird, unter der überhaupt keine Transition mehr aktiviert ist. Es liegt dann eine Deadlockmarkierung vor, die der Definition der markierungsschwachen Netzlebendigkeit in keiner Weise widerspricht.

Die markierungsschwache und transitionsschwache Lebendigkeit verhalten sich zueinander konträr: Die markierungsschwache verhindert zwar tote Transitionen, versagt aber beim gesicherten Ausschluß von Deadlockmarkierungen. Die transitionsschwache Lebendigkeit garantiert zwar die Deadlockfreiheit, kann aber keine toten Transitionen vermeiden. Der konträre Charakter der beiden schwachen Lebendigkeitsvarianten wird besonders deutlich, wenn sie durch jene definitorischen Äquivalenzen ausgedrückt werden, die auf Transitionsaktivierungen Bezug nehmen:

$$\begin{aligned}
& \text{LEB}_{\text{ms}}(\text{SN}) \\
\Leftrightarrow & \quad \forall(t_n \in T) \exists(M_f \in \text{RM}(M_0)): \text{AKT}(t_n, M_f) \\
& \text{LEB}_{\text{ts}}(\text{SN}) \\
\Leftrightarrow & \quad \forall(M_f \in \text{RM}(M_0)) \exists(t_n \in T): \text{AKT}(t_n, M_f)
\end{aligned}$$

Das Defizit der Deadlockvermeidung bei markierungsschwacher Lebendigkeit *und* der mangelhafte Ausschluß toter Transitionen bei transitionsschwacher Lebendigkeit werden geheilt, wenn zur starken Netzlebendigkeit übergegangen wird.

Die Eigenschaft der starken Netzlebendigkeit⁶⁵⁾ liegt genau dann vor, wenn alle Transitionen eines Netzes unsterblich sind. Die Transitionen verhalten sich dann unter allen erreichbaren Markierungen M_f mit $M_f \in \text{RM}(M_0)$ lebendig. Daher ist in einem stark lebendigen Netz keine Transition unter keiner erreichbaren Markierung M_f jemals M_f -blockiert⁶⁶⁾. Deshalb gilt für jede erreichbare Netzmarkierung: Von dieser Markierung M_f aus kann für jede Transition eine endliche Schaltfolge ausgeführt werden, an deren Ende die Transition aktiviert ist. Für das Prädikat $\text{LEB}_s(\text{SN})$ der starken Lebendigkeit eines Netzes gilt also:

$$\begin{aligned}
& \text{LEB}_s(\text{SN}) \\
:\Leftrightarrow & \quad \forall(t_n \in T): \text{UNS}(t_n) \\
\Leftrightarrow & \quad \forall(t_n \in T) \forall(M_f \in \text{RM}(M_0)): \text{LEB}(t_n, M_f) \\
\Leftrightarrow & \quad \forall(t_n \in T) \forall(M_f \in \text{RM}(M_0)): \neg \text{BLO}(t_n, M_f) \\
\Leftrightarrow & \quad \forall(t_n \in T) \forall(M_f \in \text{RM}(M_0)) \exists(M_f \in \text{RM}(M_f)): \text{AKT}(t_n, M_f)
\end{aligned}$$

Die starke Lebendigkeit eines Netzes sichert dessen Deadlockfreiheit, während ein deadlockfreies Netz keineswegs stark lebendig sein muß⁶⁷⁾. Ebenso garantiert die starke Lebendigkeit eines Netzes die Nichtexistenz toter Transitionen. Ein Netz, das keine tote Transitionen enthält, braucht jedoch nicht stark lebendig zu sein. Die garantierte Deadlockfreiheit und die Abwesen-

heit von toten Transitionen folgen unmittelbar aus der o.a. Definition der starken Netzlebendigkeit. Gleiches gilt für die äquivalenten Netzeigenschaften der transitions- bzw. markierungsschwachen Lebendigkeit:

$$\begin{aligned}
& \text{LEB}_s(\text{SN}) \\
\Leftrightarrow & \quad \forall (t_n \in T) \forall (M_r \in \text{RM}(M_0)): \text{LEB}(t_n, M_r) \\
\Rightarrow & \quad \forall (t_n \in T) \forall (M_r \in \text{RM}(M_0)): \text{AKT}(t_n, M_r) \\
\Rightarrow & \quad \forall (M_r \in \text{RM}(M_0)) \forall (t_n \in T): \text{AKT}(t_n, M_r) \\
\Rightarrow & \quad \forall (M_r \in \text{RM}(M_0)) \exists (t_n \in T): \text{AKT}(t_n, M_r) \\
\Rightarrow & \quad \text{LEB}_{ts}(\text{SN}) \\
\Rightarrow & \quad \text{DFR}(\text{SN})
\end{aligned}$$

$$\begin{aligned}
& \text{LEB}_s(\text{SN}) \\
\Leftrightarrow & \quad \forall (t_n \in T) \forall (M_r \in \text{RM}(M_0)): \text{LEB}(t_n, M_r) \\
\Rightarrow & \quad \forall (t_n \in T): \text{LEB}(t_n, M_0) \\
\Rightarrow & \quad \forall (t_n \in T): \neg \text{TOT}(t_n) \\
\Rightarrow & \quad \text{LEB}_{ms}(\text{SN})
\end{aligned}$$

Darüber hinaus verschärft die starke Lebendigkeit einen weiteren Aspekt der markierungsschwachen Lebendigkeit. Denn in einem markierungsschwach lebendigen Netz wird nur garantiert, daß keine M_0 -blockierten (toten) Transitionen existieren. Dadurch wird lediglich sichergestellt, daß für jede Transition mindestens eine erreichbare Markierung existiert, unter der die Transition aktiviert ist. Die markierungsschwache Netzlebendigkeit sagt jedoch nichts darüber aus, wie sich diese Transition verhält, nachdem jene aktivierende Markierung erreicht worden ist. Es kann durchaus der Fall eintreten, daß sich die Transition unter Folgemarkierungen überhaupt nicht mehr aktivieren läßt. Die markierungsschwache Netzlebendigkeit schließt also nicht aus, daß ein Netz unter mindestens einer erreichbaren Markierung M_r mit $M_r \neq M_0$ ⁶⁸⁾ mindestens eine M_r -blockierte Transition besitzt⁶⁹⁾. Die starke Netzlebendigkeit garantiert hingegen, daß unter allen erreichbaren Markierungen keine M_r -blockierte Transition existiert. Mit anderen Worten läßt sich diese Verschärfung auch so ausdrücken: Ein markierungsschwach lebendiges Netz ist nur in der Lage, die M_0 -Lebendigkeit aller Transitionen sicherzustellen. Ein stark lebendiges Netz verheißt hingegen die M_r -Lebendigkeit aller Transitionen bezüglich aller erreichbaren Markierungen M_r .

Abb. 57 auf der nächsten Seite verdeutlicht den logischen Zusammenhang derjenigen Prädikate, die voranstehend eingeführt wurden, um die Deadlockfreiheit und Lebendigkeit von Netzen zu präzisieren⁷⁰⁾. Die Darstellungsform eines Kanal/Instanz-Netzes mit kombinierter natürlich- und formalsprachlicher Knotenbeschriftung läßt erkennen, wie sehr sich Netze aufgrund ihrer graphischen Visualisierung eignen, selbst komplexere logische Beziehungsmuster zugleich kompakt und übersichtlich wiederzugeben⁷¹⁾. In der Netzliteratur existieren zwar noch weitere Definitionsvarianten für Netzeigenschaften, die zum semantischen Feld der Netzlebendigkeit gehören⁷²⁾. Sie werden hier jedoch nicht berücksichtigt. Denn sie lassen keine Einblicke in die dynamische Struktur von Netzen erwarten, die für Modellierungszwecke dieser Arbeit von Interesse sein könnten. In der anschließenden Abb. 58 auf der übernächsten Seite werden die charakteristischen Eigenschaften der drei Lebendigkeitsprädikate, die in dieser Arbeit Berücksichtigung finden, gesondert hervorgehoben.

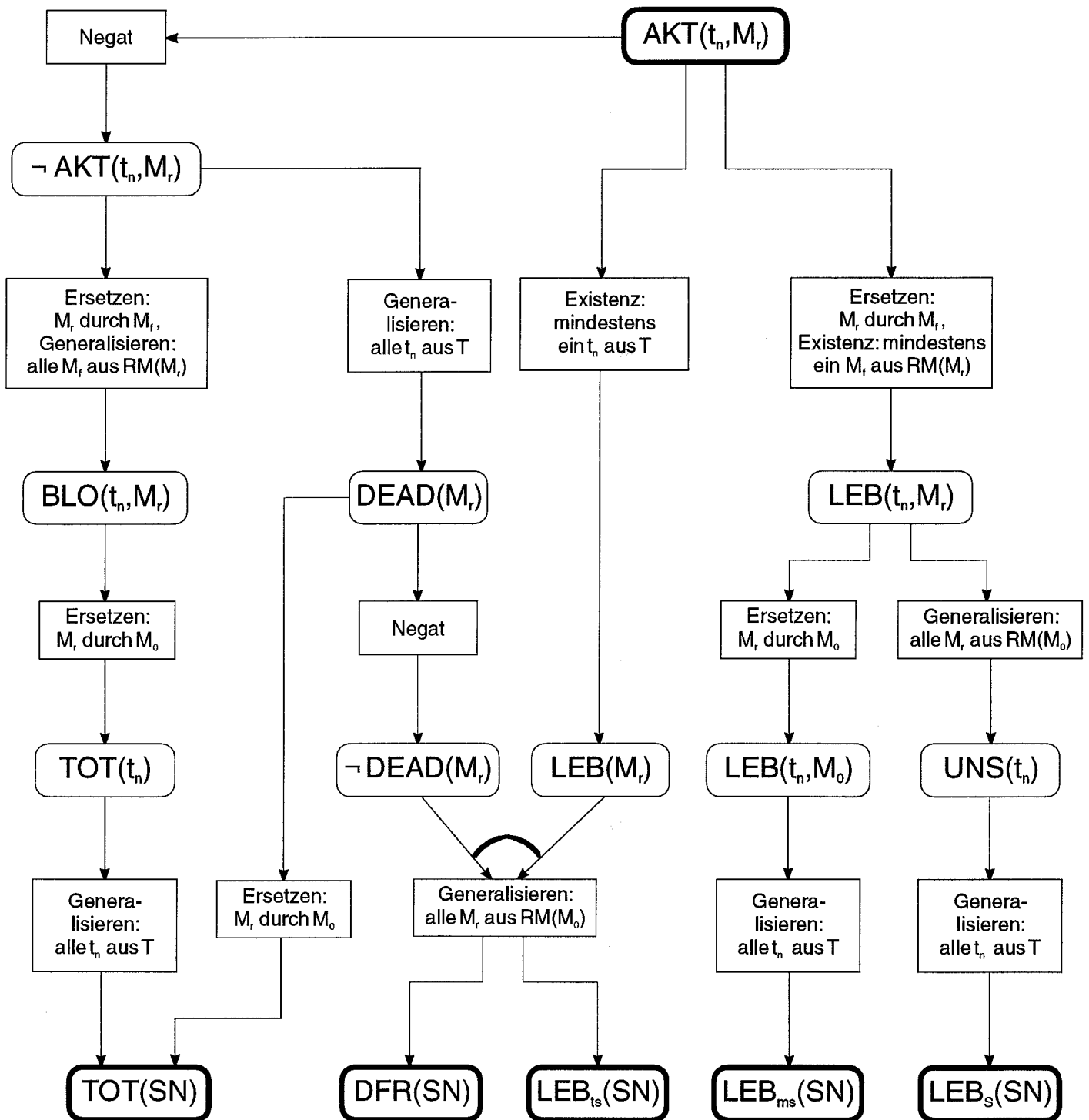


Abb. 57: Logisches Beziehungsmuster für Prädikate aus dem Umfeld von Deadlocks und Lebendigkeit

transitionsschwache Lebendigkeit	markierungsschwache Lebendigkeit	starke Lebendigkeit
<p>_____</p> <p>_____</p> <p>⇔ jede erreichbare Markierung ist lebendig</p>	<p>_____</p> <p>⇔ jede Transition ist unter der Ausgangsmarkierung lebendig</p> <p>_____</p>	<p>⇔ jede Transition ist unsterblich</p> <p>⇒ jede Transition ist unter der Ausgangsmarkierung lebendig</p> <p>⇒ jede erreichbare Markierung ist lebendig</p>
<p>⇔ Deadlockfreiheit</p> <p>tote Transitionen können existieren</p>	<p>Deadlocks können existieren</p> <p>⇔ es existieren keine toten Transitionen</p>	<p>⇒ Deadlockfreiheit</p> <p>⇒ es existieren keine toten Transitionen</p>
<p>⇔ unter jeder erreichbaren Markierung existiert mindestens eine aktivierte Transition</p> <p>_____</p> <p>_____</p> <p>_____</p>	<p>_____</p> <p>⇔ jede Transition ist unter mindestens einer erreichbaren Markierung aktiviert</p> <p>⇔ jede Transition ist M_0-lebendig</p> <p>⇔ keine Transition ist M_0-blockiert</p>	<p>⇔ unter jeder erreichbaren Markierung sind alle Transitionen lebendig</p> <p>⇔ jede Transition kann von jeder erreichbaren Markierung aus aktiviert werden</p> <p>⇔ jede Transition ist M_r-lebendig für jede erreichbare Markierung M_r</p> <p>⇔ keine Transition ist M_r-blockiert unter irgendeiner erreichbaren Markierung M_r</p>
<p>⇔ unter jeder erreichbaren Markierung kann irgendeine Transition geschaltet werden</p>	<p>⇔ jede Transition kann irgendwann einmal geschaltet werden</p>	<p>⇔ jede Transition kann immer wieder geschaltet werden</p>

Abb. 58: Charakteristische Eigenschaften der drei Lebendigkeitsprädikate

Unabhängig von allen definitorischen Variationen besteht in der Netzliteratur weitreichende Übereinstimmung, Deadlockfreiheit und Lebendigkeit als die herausragenden dynamischen Eigenschaften von Netzen zu betrachten⁷³). Dahinter steht zumeist die Vorstellung, in Netzen Prozesse ablaufen zu lassen, die nach Möglichkeit niemals abbrechen⁷⁴). Entsprechend werden Deadlocks und fehlende (transitionsschwache) Netzlebendigkeit als schwerwiegende Netz-mängel eingestuft⁷⁵). Darüber hinaus sollen die Netze keine "aktiven" Komponenten enthalten, die überflüssig sind, weil sie infolge Fehlerhaftigkeit niemals aktiv werden können. Denn solche Komponenten würden für ihre eigene Darstellung Ressourcen binden, ohne jemals in der Lage zu sein, das Netzverhalten auf irgendeine Weise zu beeinflussen⁷⁶). Daher gelten tote Transitionen und fehlende markierungsschwache Netzlebendigkeit ebenso als ein Netzdefizit⁷⁷). Schließlich wird zumeist noch die Netzbeschränktheit gefordert⁷⁸). Dies reicht im Regelfall aus, um sicherzustellen, daß die Erreichbarkeitsgraphen der zugrundeliegenden Netze endlich bleiben⁷⁹). Trotzdem können in solchen Netzen weiterhin zyklische, nicht endende Prozesse dargestellt werden⁸⁰). Derart deadlockfreie, lebendige⁸¹) und beschränkte Netze werden als "wohlgeformte"⁸²), "adäquate"⁸³) oder "konforme"⁸⁴) Netze besonders ausgezeichnet⁸⁵).

Allerdings wäre die uneingeschränkte Präferenz deadlockfreier und lebendiger Netze zu einseitig. Denn es kann sehr wohl von Interesse sein, deadlockbehaftete oder nicht-lebendige Netze zuzulassen. Beispielsweise existieren zahlreiche produktionswirtschaftlich interessante Fälle, in denen die starke Lebendigkeit eines Netzmodells überhaupt nicht erwünscht ist⁸⁶). So führen etwa Modellierungen von Produktionssystemen, in denen separate Prozesse für Anlauf- oder Notbetriebsphasen vorgesehen sind, stets zu Netzmodellen, die sich nicht stark lebendig verhalten. Denn einerseits sollen die Transitionen aus den Anlaufprozessen im allgemeinen nicht mehr aktiviert sein, wenn die Anlaufphase beendet ist. Diese Transitionen verhalten sich daher nach dem Ende der Anlaufphase M_r -blockiert. Folglich wird die Anforderung der starken Netzlebendigkeit verletzt. Andererseits werden Notprozesse in der Regel nicht alle Transitionen aus der Modellierung von Routineprozessen umfassen. Daher erweisen sich alle Transitionen der Routineprozesse, die in den Notprozessen nicht vorkommen, als M_r -blockiert, sobald eine Notbetriebsphase erreicht wird. Abermals trifft die starke Lebendigkeit des gesamten Netzmodells nicht zu.

Des weiteren spielen in Netzmodellen für Produktionssysteme oftmals solche Prozesse eine Rolle, die in wohldefinierten Markierungen enden⁸⁷). Diese Endmarkierungen stellen immer dann erwünschte Deadlocks⁸⁸) dar, wenn zwei Bedingungen erfüllt sind. Erstens müssen die Prozesse, deren Ausführung zum Stillstand gekommen ist, selbst erwünscht sein. Zweitens darf unter den erreichten Endmarkierungen kein Schaltschritt mehr aktiviert sein. Diese beiden Voraussetzungen können z.B. von Prozessen erfüllt werden, die zur Abwicklung eines Produktionsauftrags dienen. Die Markierungen, welche die Fertigstellung des Auftrags repräsentieren, stellen erwünschte Netzmarkierungen dar. Falls zusätzlich die Auftragsabwicklung so modelliert wird, daß nach der Auftragsfertigstellung keine weitere (auftragsbezogene) Aktivität mehr möglich sein soll, dann ist im Netzmodell unter den erwünschten Markierungen der Auftragsfertigstellung auch kein Schaltschritt mehr aktiviert. Den beiden Bedingungen für erwünschte Deadlocks werden häufig auch konventionelle⁸⁹) informationsverarbeitende Prozesse gerecht⁹⁰). Sie sollen nach endlicher Zeit mit dem intendierten Verarbeitungsergebnis abschließen. Auf solche terminierenden Prozesse, die dabei intendierten Endmarkierungen und die zugehörigen erwünschten Deadlocks wird im nächsten Kapitel zurückgekommen. Dabei werden die Aspekte der Finalität und des Terminierens von Netzen beleuchtet.

Dagegen werden Deadlocks, die keine Endmarkierungen von intendierten Prozessen darstellen, als unerwünschte Deadlocks bezeichnet⁹¹). Sie spielen im Rahmen des Petrinetz-Konzepts eine herausragende Rolle. Daher werden sie im folgenden näher betrachtet. Die unerwünschten Deadlocks vereiteln nicht nur die zunächst angesprochenen unendlichen Prozesse. Vielmehr führen sie auch dazu, den erfolgreichen Abschluß aller endlichen Prozesse⁹²) zu verhindern, die bis zum Eintreten der Deadlocks noch nicht zu Ende ausgeführt wurden⁹³).

Die Analyse unerwünschter Deadlocks beruht auf dem Konzept der Teilprozesse⁹⁴). Solche Teilprozesse werden in einem Netz durch das Schalten von prozeßspezifischen Transitionen ausgeführt. Die Teilprozesse benötigen zu ihrer Ausführung Ressourcen. Diese Ressourcen werden in Netzen als Marken oder freie Markkapazitäten dargestellt. Sie sind erforderlich, um die Aktivierungsbedingungen der prozeßzugehörigen Transitionen zu erfüllen. Nach Maßgabe der zugrundeliegenden Schaltregel können die betroffenen Ressourcen durch die Schaltakte der Transitionen sowohl verbraucht⁹⁵) als auch wieder freigegeben⁹⁶) werden⁹⁷).

Die Ausführung eines Teilprozesses wird unter einer Netzmarkierung unterbrochen⁹⁸), wenn drei Bedingungen erfüllt sind. Erstens muß die Ausführung des Teilprozesses bereits begonnen haben. Zweitens darf sie noch nicht beendet worden sein⁹⁹). Drittens dürfen unter der Netzmarkierung die verfügbaren Ressourcen nicht ausreichen, um die Aktivierungsbedingungen derjenigen Transitionen¹⁰⁰) zu erfüllen, die zur Fortsetzung der Teilprozeßausführung als nächste geschaltet werden müßten¹⁰¹). Der Teilprozeß ist blockiert¹⁰²), wenn er unterbrochen wurde und falls es unter allen erreichbaren Folgemarkierungen unmöglich ist, den Teilprozeß wieder fortzusetzen. Bei einem unerwünschten Deadlock handelt es sich um eine erreichbare Markierung, unter der alle Teilprozesse blockiert sind.

Im Regelfall¹⁰³) tritt eine solche universelle Blockade aller Teilprozesse dadurch ein, daß mehrere Teilprozesse für ihre Ausführung um mindestens eine gemeinsam benötigte, aber insgesamt knappe Ressource konkurrieren. Unter ungünstigen Umständen kann es dazu kommen, daß die Teilprozesse in einer Weise auf die verfügbaren Einheiten der knappen Ressource zugreifen, die verhindert, daß sich überhaupt noch eine Prozeßausführung fortsetzen läßt¹⁰⁴). Das Netz befindet sich dann in einer Deadlockmarkierung. Es wird von einem ressourcenbezogenen Deadlock gesprochen. Er beruht darauf, daß sich ein Zugriffskonflikt auf die knappe Ressource nicht mehr auflösen läßt. Dies ist die typische¹⁰⁵) Konfiguration eines unerwünschten Deadlocks¹⁰⁶).

Ressourcenbezogene Deadlocks finden nicht nur in der Netzliteratur breite Beachtung¹⁰⁷). Sie können auch für die Prozeßkoordination in komplexen Produktionssystemen - wie etwa Flexiblen Fertigungssystemen - eine erhebliche Rolle spielen¹⁰⁸). Dieses Relevanzurteil mag auf den ersten Blick befremden, weil konventionelle Modellierungen von Prozeßplanungen und -steuerungen in Flexiblen Fertigungssystemen entsprechende Deadlockuntersuchungen zumeist vermissen lassen¹⁰⁹). Dies gilt zumindest so lange, wie sich die Modellierungen im Rahmen von OR-Programmen bewegen¹¹⁰). Die mangelnde produktionswirtschaftliche Deadlockbeachtung beruht jedoch nach Einschätzung des Verf. nicht auf der Irrelevanz von Deadlocksituationen für reale Koordinierungsprobleme. Vielmehr drängt sich die Vermutung auf, die Vernachlässigung von Deadlocks folge aus der Unmöglichkeit, sie mit den Ausdrucksmitteln von OR-Programmen adäquat zu erfassen¹¹¹).

Die tatsächliche Bedeutung, die ressourcenbezogene Deadlocks für die Koordination von Teilprozessen in Flexiblen Fertigungssystemen erlangen können, wird in einem Exkurs veranschaulicht¹¹²). Dabei wird ein bewußt einfach strukturiertes Produktionssystem betrachtet¹¹³), um die hier interessierende Deadlockproblematik anhand eines übersichtlichen Netzmodells hervortreten zu lassen. Ausgangspunkt sind zwei parallele Fertigungslinien, auf denen die gleichen Vorprodukte zu Endprodukten verarbeitet werden können. Den Transport zwischen den beiden Fertigungslinien einerseits sowie ihren gemeinsamen Vorprodukt- und Endprodukt-Lagern leistet ein Transportroboter. Es handelt sich um eine knappe Transportressource, von der nur eine Einheit zur Verfügung steht. Darüber hinaus handelt es sich bei den beiden Fertigungslinien um knappe Bearbeitungsressourcen, auf denen sich in jedem Zeitpunkt nur höchstens eine Produkteinheit befinden kann¹¹⁴).

Die nachfolgende Abb. 59 zeigt ein Netzmodell für das voranstehend skizzierte rudimentäre Flexible Fertigungssystem. Um das Modell an realistische Produktionsbedingungen anzunähern, wurden Transport- und Bearbeitungsdauern als Stellenanschriften¹¹⁵) eingeführt¹¹⁶). Zusammen mit der Knappheit von Transport- und Bearbeitungsressourcen führen sie dazu, daß das Ferti-

gungssystem in einen Deadlock geraten kann. Der verursachende Schaltprozeß läßt sich unmittelbar aus dem Erreichbarkeitsgraphen erkennen, der in Abb. 60 für das Netzmodell aus Abb. 59 in einem kleinen Ausschnitt dargestellt ist¹¹⁷). Die resultierende Deadlockmarkierung des Netzmodells ab dem Zeitpunkt $T=6$ gibt Abb. 61 wieder. Sie zeigt, daß eine bearbeitete Produkteinheit die erste Fertigungslinie im Zeitpunkt $T=6$ nicht verlassen kann, weil der eine verfügbare Transportroboter noch mit der Zuführung einer weiteren Vorprodukteinheit zur selben Fertigungslinie beschäftigt ist. Der Transportroboter erreicht zwar die Fertigungslinie im Zeitpunkt $T=8$. Dennoch kann er dort seine Vorprodukteinheit nicht übergeben, weil die Fertigungslinie durch die bearbeitete Produkteinheit weiterhin blockiert wird. Daher bleibt auch der Transportroboter belegt. Es ist unmöglich, ihn dem Abtransport der bearbeiteten Produkteinheit zuzuordnen. Auf diese Weise blockieren sich die beiden Transport- und Bearbeitungsprozesse wechselseitig in ihren Zugriffsversuchen auf die jeweils knappe Bearbeitungs- bzw. Transportkapazität.

Es wird nicht in Abrede gestellt, daß ein anderes Netzmodell für das einfache Flexible Fertigungssystem hätte entworfen werden können, in dem der vorgestellte Deadlock nicht aufgetreten wäre. Die Pointe des skizzierten Beispiels liegt vielmehr darin, daß bei der Koordinierung von Teilprozessen in komplexen Produktionssystemen und ihrer Modellierung stets mit der Gefahr von unerwünschten Deadlocks gerechnet werden muß. Das Petrinetz-Konzept stellt die Ausdrucks- und die Analysemittel zur Verfügung, um solche Deadlocks identifizieren zu können. Auf konventionelle Modellierungstechniken trifft dies nicht zu, zumindest wenn der Rahmen von OR-Programmen nicht verlassen wird.

Darüber hinaus bietet das Petrinetz-Konzept sogar Mittel an, um die Koordinierungsschwierigkeiten von Deadlocks zu vermeiden. Dafür kommen grundsätzlich zwei alternative Strategien in Betracht. Die erste Strategie zielt darauf ab, deadlockfreie Netze zu konstruieren. Diese Strategie der Deadlockfreiheit läßt sich nochmals in zwei Substrategien aufspalten:

- Bei der Kompositionsstrategie¹¹⁸) werden einfache Netzmodule verwendet, deren Deadlockfreiheit bereits nachgewiesen ist. Hinzu kommen Kompositionsregeln für den systematischen Aufbau von modularen Netzmodellen. Die Regeln stellen sicher, daß die Deadlockfreiheit der Gesamtmodelle bei der Komposition der Netzmodule nicht verloren geht¹¹⁹).
- Bei der Modifizierungsstrategie¹²⁰) wird von beliebig konstruierten Netzmodellen ausgegangen. Sie werden - etwa mit Hilfe der Erreichbarkeitsanalyse - hinsichtlich ihrer Deadlockfreiheit untersucht. Werden dabei unerwünschte Deadlocks erkannt, erfolgt eine nachträgliche Netzmodifizierung. Dabei wird angestrebt, daß das modifizierte Netz schließlich deadlockfrei ist. Allerdings kann dies nicht garantiert werden¹²¹). Statt dessen bleibt es weitgehend der Intuition des Modellierungsträgers überlassen, eine deadlockfreie Netzmodifizierung aufzuspüren¹²²).

Die zweite Strategie verzichtet darauf, alle unerwünschten Deadlocks zu entfernen, die in der dynamischen Struktur eines Netzes enthalten sind. Statt dessen wird versucht, das Schaltverhalten des Netzes so zu koordinieren, daß sich die unerwünschten Deadlocks nicht auszuwirken vermögen¹²³). Es handelt sich daher um eine Strategie der Deadlockumgehung¹²⁴).

Anhand des Beispiels zweier Fertigungslinien, das vor kurzem eingeführt wurde, lassen sich die beiden Strategien der Deadlockfreiheit und der Deadlockumgehung veranschaulichen. Da bereits ein deadlockbehaftetes Netzmodell vorliegt, interessiert hier aus der Perspektive der Deadlockfreiheit nur die Substrategie der Netzmodifizierung¹²⁵). Mit ihrer Hilfe kann das Netzmodell aus Abb. 59 so verändert werden, daß es sich tatsächlich deadlockfrei verhält. Allerdings muß dazu auf ein besonderes Netzkonstrukt - die Inhibitorkanten - zurückgegriffen werden, das bislang noch nicht eingeführt worden ist. Daher wird das modifizierte Netzmodell erst später präsentiert, wenn das Kernkonzept Synthetischer Netze entsprechend erweitert worden ist.

Die Strategie der Deadlockumgehung kann dagegen mit den bisher definierten Ausdrucksmitteln von Synthetischen Netzen verwirklicht werden. In Abb. 62 wird der Ausschnitt aus dem Erreichbarkeitsgraphen des Netzmodells so erweitert, daß sich aus ihm eine deadlockfreie Prozeßausführung wiederum unmittelbar ablesen läßt. Sie führt von der Ausgangsmarkierung M_0 bis zur Endmarkierung M_{26} , unter der im Zeitpunkt $T=22$ alle vier Vorprodukte in der erwünschten Weise zu Endprodukten verarbeitet worden sind und sich im Endprodukt-Lager befinden. Abb. 63 zeigt das Netzmodell unter dieser intendierten Endmarkierung. Abb. 64 gibt in der Gestalt eines Ablaufdiagramms wieder¹²⁶⁾, wie die Bearbeitungs- und Transportzeiten auf die beiden Fertigungslinien verteilt sind und in welcher Weise hierbei der Transportroboter in Anspruch genommen wird. Es handelt sich nicht nur um eine deadlockumgehende, sondern zugleich auch um eine durchlaufzeitminimale Prozeßkoordinierung. Allerdings setzt die Deadlockumgehung voraus, daß der Koordinierungsträger bei seiner Prozeßkoordinierung zumindest denjenigen Ausschnitt aus dem Erreichbarkeitsgraphen berücksichtigt, der in Abb. 62 wiedergegeben ist. Diese Voraussetzung läßt sich z.B. durch projektive Erreichbarkeitsanalysen mit einem entsprechenden Vorausschauhorizont erfüllen¹²⁷⁾.

Die Problematik des Entdeckens und Vermeidens von unerwünschten Deadlocks, die im voranstehenden Exkurs anhand eines einfachen Beispiels veranschaulicht wurde, findet in der Netzliteratur eine überaus breite Beachtung. Insbesondere führte sie auch zu einer Sammlung paradigmatischer Koordinierungsprobleme¹²⁸⁾. Sie zeichnen sich dadurch aus, daß sie einfach zu erkennende und zugleich unerwünschte Deadlocks besitzen. Für diese Probleme werden verschiedene Koordinationsstrategien untersucht. Sie sollen die deadlockverursachenden Ressourcenzugriffe der involvierten Teilprozesse so "synchronisieren"¹²⁹⁾, daß sich die Deadlockentstehung unterbinden läßt¹³⁰⁾.

Es liegt außerhalb des Erkenntnisinteresses dieser Arbeit, die Vielfalt ressourcenbezogener Deadlockkonfigurationen näher auszuleuchten. Dennoch wurde auf sie voranstehend aus zwei Gründen kurz eingegangen. Erstens wird es für die spätere Eignungsbeurteilung des Petrinetz-Konzepts eine erhebliche Rolle spielen, daß dieses Konzept eine breite Palette von Möglichkeiten besitzt, die Existenz von unerwünschten Deadlocks in Netzen zu untersuchen und gegebenenfalls durch entsprechende Synchronisationsstrategien zu beseitigen. Zweitens wird bei der späteren Gestaltung von Netzmodulen für die Modellierung Flexibler Fertigungssysteme von vornherein auf solche deadlockvermeidenden Synchronisationsstrategien zurückgegriffen. Daher wird es später nicht notwendig sein, für die Netzmodule im einzelnen die Freiheit von unerwünschten Deadlocks nachzuweisen.

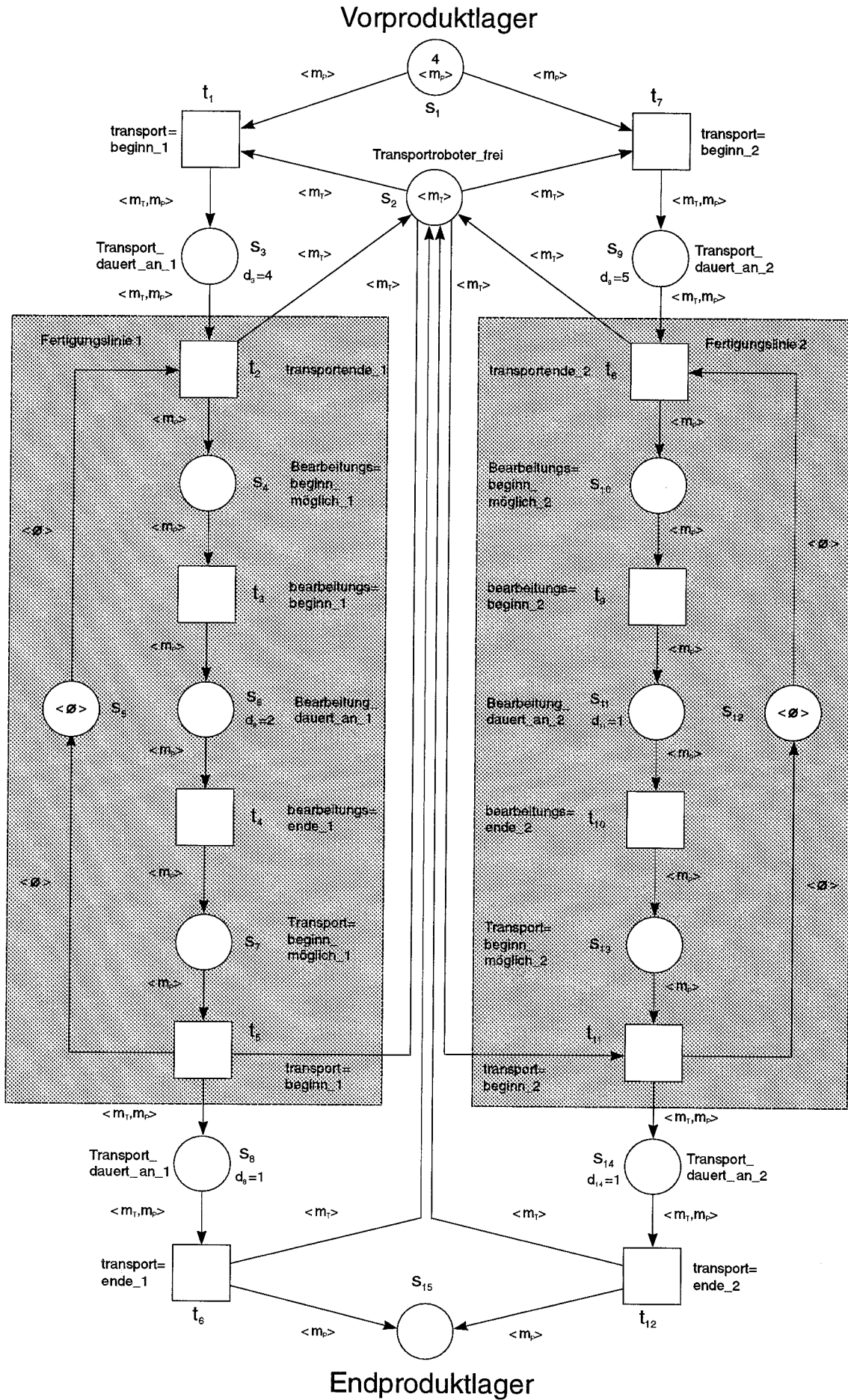


Abb. 59: Netzmodell für zwei Fertigungslinien unter der Ausgangsmarkierung M_0 im Zeitpunkt $T=0$

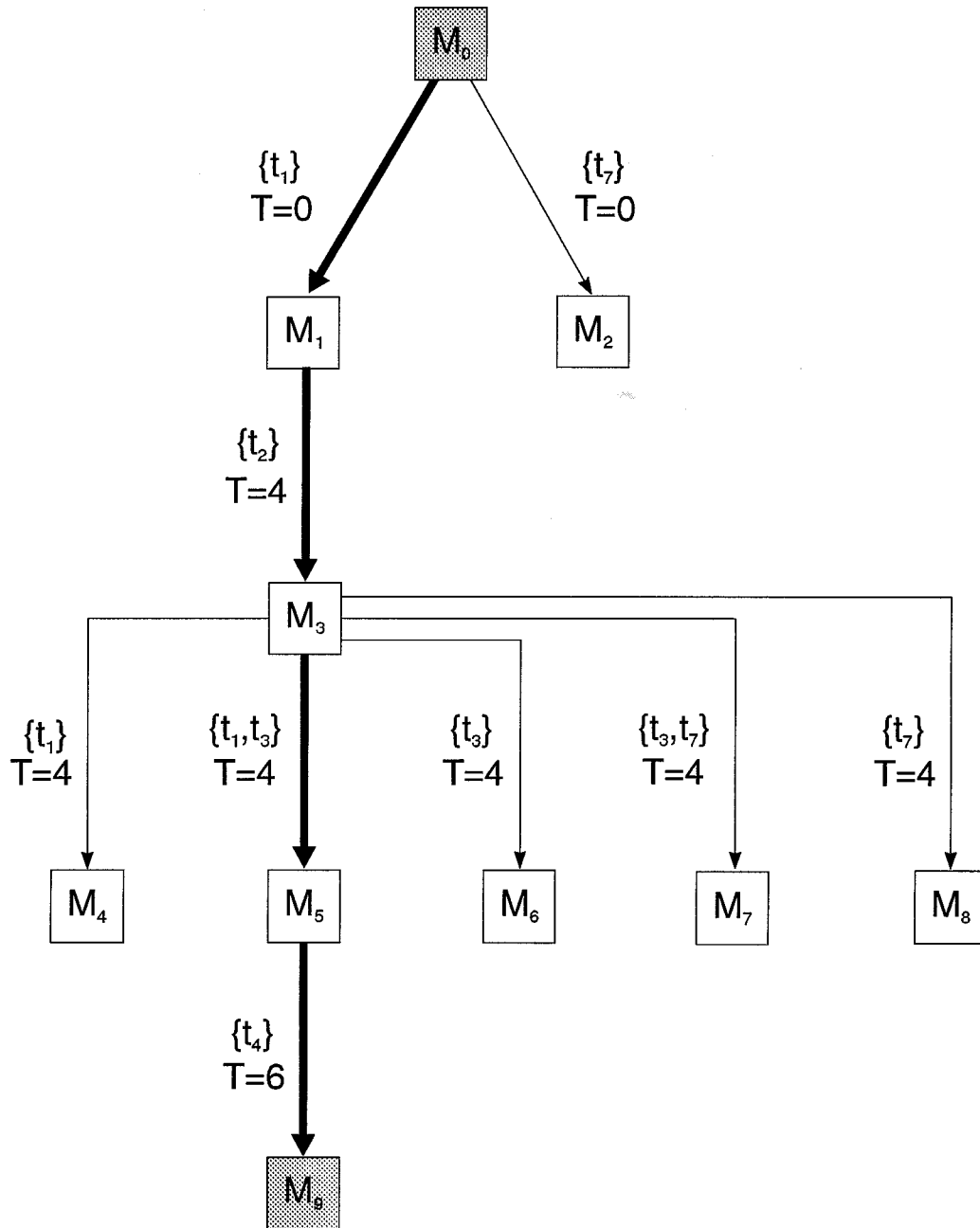


Abb. 60: Erster Ausschnitt aus dem Erreichbarkeitsgraphen für das Netzmodell mit zwei Fertigungslinien aus der Abb. 59

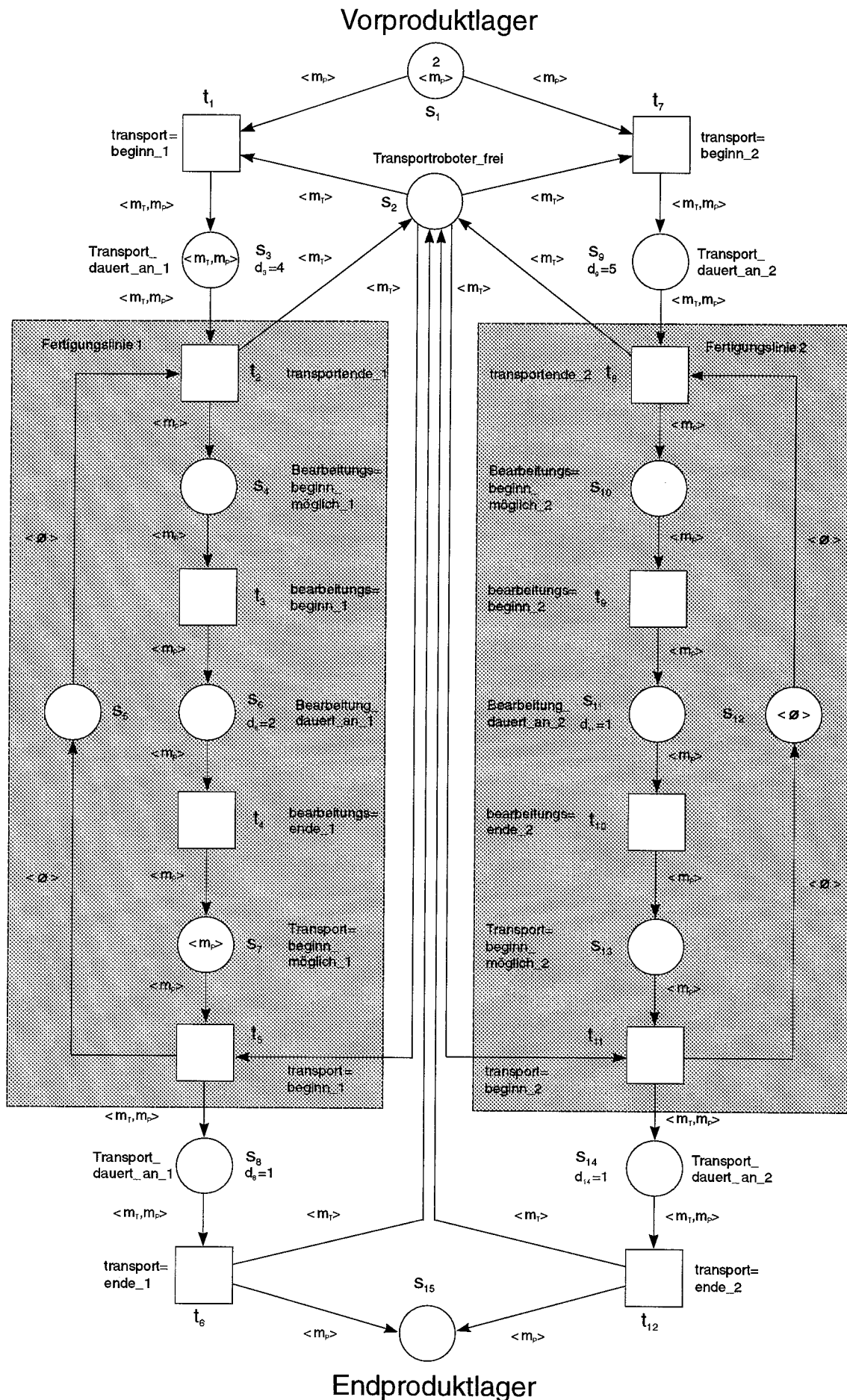


Abb. 61: Netzmodell für zwei Fertigungslinien unter der Deadlockmarkierung M_9 im Zeitpunkt $T \geq 6$

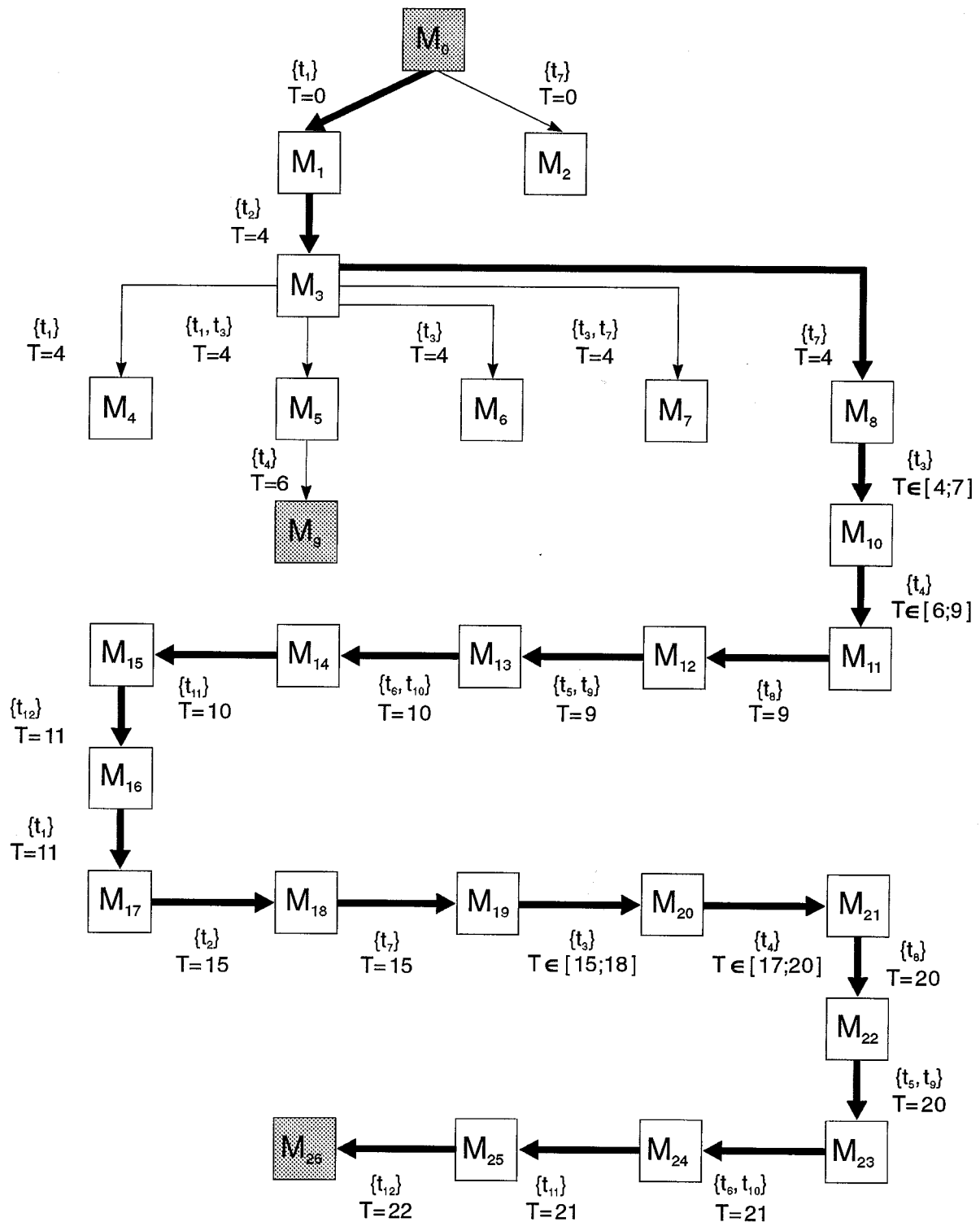


Abb. 62: Zweiter Ausschnitt aus dem Erreichbarkeitsgraphen für das Netzmodell mit zwei Fertigungslinien aus der Abb. 59

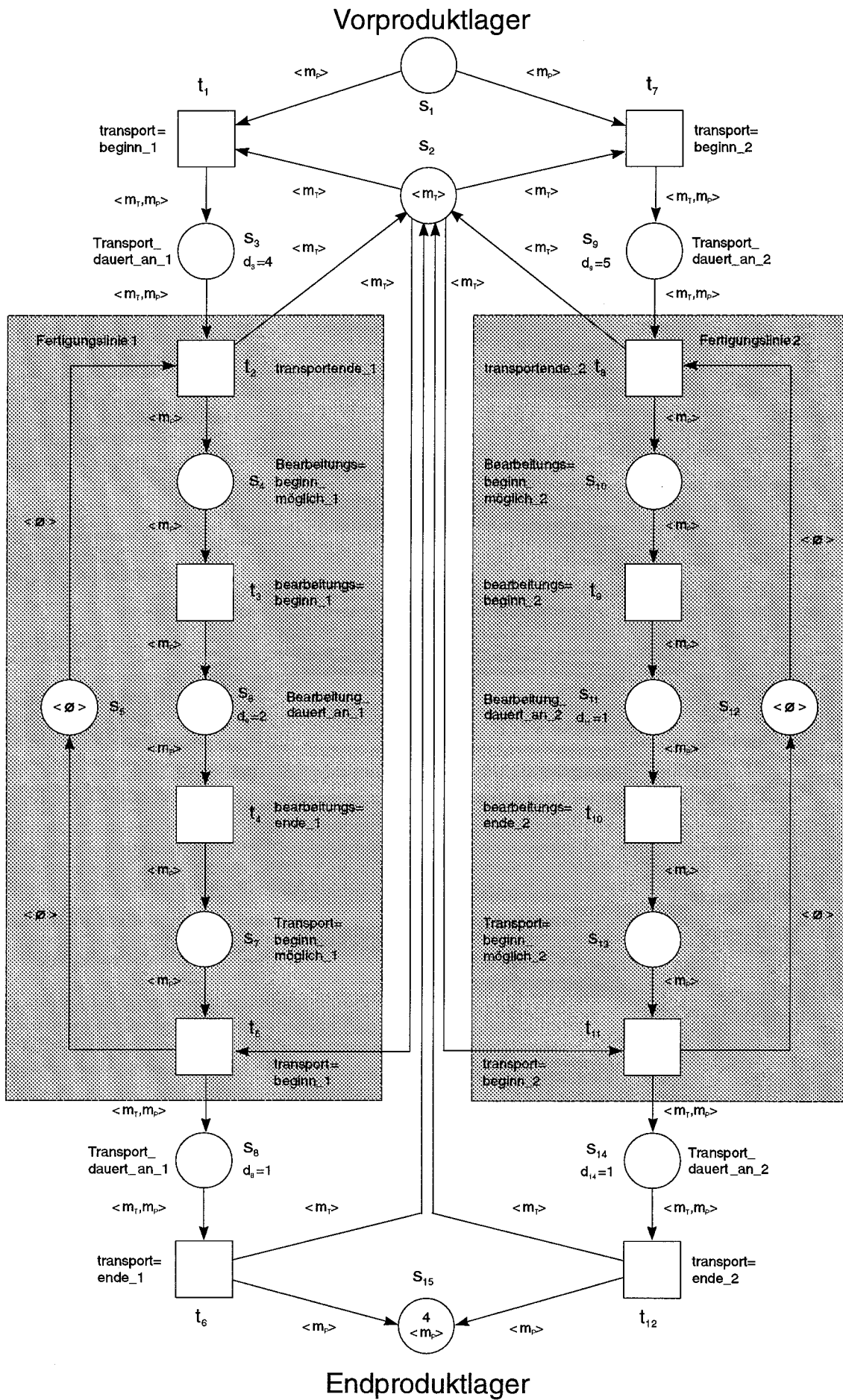


Abb. 63: Netzmodell für zwei Fertigungslinien unter der Endmarkierung M_{26} im Zeitpunkt $T=22$

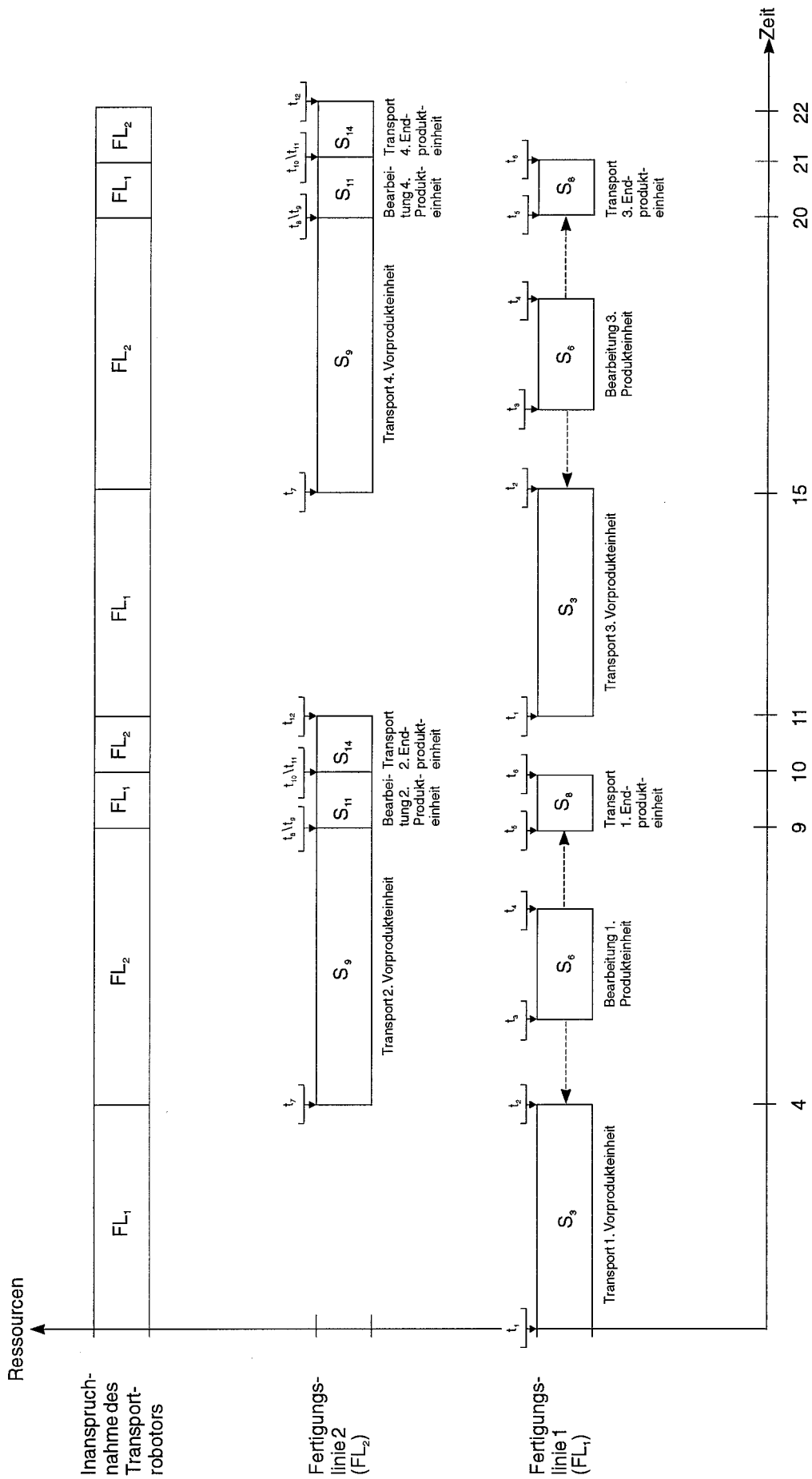


Abb. 64: Ablaufdiagramm für das Netzmodell mit zwei Fertigungslinien aus der Abb. 59

Abschließend wird auf einen Aspekt der dynamischen Struktur von Netzen eingegangen, der sowohl mit der starken Netzlebendigkeit als auch mit Deadlocks eng verwandt ist. Er betrifft die Existenz von Livelocks. Ein Livelock¹³¹⁾ wird hier zunächst als eine besondere Netzmarkierung M_r aufgefaßt. Aus diesem Blickwinkel stellt er das konträre Gegenteil zu einem Deadlock dar. Während in einem Deadlock kein Schaltprozeß mehr fortgeführt werden kann, involviert ein Livelock einen unendlich fortsetzbaren Schaltprozeß. Ein Livelock ist eine Markierung M_r für die mindestens ein endlicher, nicht-entarteter Schaltprozeß $\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L)$ existiert, der diese Markierung sowohl als Startmarkierung M_r als auch als Zielmarkierung $M_f = M_r$ besitzt¹³²⁾. Für das Prädikat $\text{LIVE}(M_r)$, das den Livelockcharakter der Markierung M_r ausdrückt, gilt daher mit Hilfe der Prozeßmenge PROM:

$$\text{LIVE}(M_r) \\ :\Leftrightarrow \exists(L \in \mathcal{N}_+) \exists(\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L) \in \text{PROM}): \text{MF}_L = (M_r, \dots, M_r) \wedge M_f = M_r$$

Der endliche Schaltprozeß, der in der Livelockmarkierung M_r beginnt und auch endet, läßt sich beliebig oft wiederholen. Diese Iterationsmöglichkeit konstituiert einen potentiell unendlichen¹³³⁾, zyklischen Teilprozeß¹³⁴⁾. Er wird als Livelockprozeß oder Endlosschleife¹³⁵⁾ bezeichnet.

Livelockprozesse können durchaus ein erwünschtes zyklisches Netzverhalten darstellen¹³⁶⁾. Ein solches Netzverhalten läßt sich beliebig - potentiell unendlich - oft wiederholen. Auf diese Weise kann z.B. eine Entscheidungsprozedur modelliert werden, die sich bei Bedarf immer wieder erneut aufrufen und dann jeweils in der gleichen Art ausführen läßt. Vor allem im Rahmen der transaktionsorientierten Informationsverarbeitung bietet es sich an, getriggerte Transaktionen als Teilnetze zu modellieren, in denen jeweils ein Livelockprozeß die Transaktionsausführung wiedergibt¹³⁷⁾. Darüber hinaus lassen sich aus einer speziellen Art von Livelockprozessen Erkenntnisse über die inhärente Nebenläufigkeit - d.h. vor allem: der Parallelisierbarkeit - von Prozeßausführungen gewinnen: Es handelt sich um Livelockprozesse, die unabhängig von anderen Teilprozessen innerhalb eines Netzes ausgeführt werden können, sobald die Startbedingungen der Livelockprozesse erfüllt sind. Darüber hinaus dürfen spätere Ausführungen eines Livelockprozesses nicht von den Resultaten der früheren Ausführungen desselben Livelockprozesses abhängen. Falls beide Unabhängigkeitsvoraussetzungen zutreffen, brauchen mehrfache Ausführungen desselben Livelockprozesses nicht zeitlich nacheinander zu erfolgen. Statt dessen ist es auch möglich, sie zur selben Zeit, also parallel abzuwickeln¹³⁸⁾. Schließlich können Livelockprozesse auch erwünscht sein, um Netzmodule jederzeit in ihre wohldefinierten Ausgangszustände zurückzuführen. Darauf wird später zurückgekommen, wenn Netzmodule für die Spezifizierung des Schaltverhaltens von Makrotransitionen erörtert werden.

Daneben ist es aber ebenso möglich, daß Livelocks zum unerwünschten "Verhungern" oder permanenten Blockieren von anderen Teilprozessen¹³⁹⁾ führen¹⁴⁰⁾. Jene Prozesse haben zwar ihre Ausführung begonnen, werden aber durch die Livelockprozesse unendlich lange daran gehindert, beendet zu werden¹⁴¹⁾. Das kann dazu führen, daß sogar in einem stark lebendigen Netz einzelne Teilprozesse verhungern¹⁴²⁾. Denn die Lebendigkeitseigenschaft stellt nur sicher, daß alle Transitionen immer wieder *aktiviert* werden. Ob diese Aktivierungen aber auch tatsächlich *Schaltakte* nach sich ziehen, darüber sagt keine der hier behandelten Lebendigkeitsdefinitionen etwas aus¹⁴³⁾. Daher ist es möglich, daß sich die Transitionen eines alternativen Teilprozesses in einem stark lebendigen Netz zwar unendlich oft aktivieren lassen, aber dennoch innerhalb eines konkreten, potentiell unendlichen Schaltprozesses niemals geschaltet werden¹⁴⁴⁾. Das vermag ein Livelockprozeß zu bewirken, wenn seine Transitionen konfliktionär zu den Transitionen des alternativen Teilprozesses aktiviert sind und in jedem Schaltkonflikt den Vorzug erhalten¹⁴⁵⁾. Die Unerwünschtheit eines solchen Livelockprozesses wird besonders deutlich, falls er alle alternativen

Teilprozesse verhungern läßt, die ihrerseits gestatten würden, erwünschte Netzmarkierungen zu erreichen¹⁴⁶).

Die zuvor skizzierten Schwierigkeiten lassen sich dadurch ausschließen, daß die Livelockfreiheit eines Netzes sichergestellt wird¹⁴⁷). Ein Netz verhält sich livelockfrei, falls seine Erreichbarkeitsmenge keine Livelockmarkierung umfaßt. Die Livelockfreiheit eines Netzes kann daher durch das Prädikat LFR(SN) erfaßt werden, das analog zur Deadlockfreiheit definiert ist:

$$\text{LFR}(\text{SN})$$
$$:\Leftrightarrow \forall (M_t \in \text{RM}(M_0)) : \neg \text{LIVE}(M_t)$$

Ob die Livelockfreiheit eines Netzes eine wünschenswerte Netzeigenschaft darstellt, hängt jedoch von dem jeweils modellierten Objekt und den Modellierungszwecken ab. Dies bestätigt noch einmal die Erläuterungen zur Interpretationsfähigkeit und -bedürftigkeit von Netzeigenschaften.

Anmerkungen zum Kapitel:

1) Die herausragende Bedeutung von Deadlockfreiheit und Lebendigkeit betonen z.B. ausdrücklich GENRICH (1973a), S. 144, der die Netzlebendigkeit zu den "fundamentalen Systemeigenschaften" rechnet; ULLRICH (1976), S. 0-8; SCHESCHONK (1977), S. 7; LAUTENBACH (1979a), S. 99; REISIG (1979d), S. 37, der Netze für praktisch unbrauchbar hält, wenn sie sich nicht mit einer Ausgangsmarkierung versehen lassen, welche die (starke) Netzlebendigkeit sicherstellen; SIFAKIS (1980c), S. 587. Vgl. am Rande auch HÄMMÄINEN (1987), S. 288.

Mittelbar wird das Gewicht von Deadlockfreiheit und Lebendigkeit auch durch eine Vielzahl von Publikationen deutlich, die sich mit diesen Systemeigenschaften - oder den zugrundeliegenden Deadlocks (Verklemmungen) - detailliert auseinandersetzen. Vgl. z.B. HAVENDER (1968), S. 74ff.; HABERMANN (1969), S. 373ff.; SHOSHANI (1970), S. 355ff.; HOLT,R. (1971a), S. 1ff., 8f., 22ff., 61ff., 86ff., 103ff., 135ff. u. 173ff.; COFFMAN (1971), S. 72ff.; KELLER,R. (1972a), S. 32ff.; HOLT,R. (1972b), S. 64ff.; LAUTENBACH (1973), S. 14ff.; COFFMAN (1973), S. 46 u. 51ff.; SCHROFF (1974), S. 1ff.; BYRN (1974), S. II-16ff., II-124ff. u. III-31ff.; MAYR,E. (1975), S. 2ff.; HERZOG,O. (1976c), S. 55ff.; HERZOG,O. (1977), S. 209ff.; BERTHELOT (1977), S. 217ff.; GOSH (1977), S. 13ff.; LAUER (1978a), S. 311ff. u. 319ff.; REISIG (1979d), S. 26, 36ff. u. 71ff.; HERZOG,O. (1979), S. 66 u. 83ff.; GRIESE,W. (1979); FERSTL (1979), S. 105ff.; KERAMIDIS (1980a), S. 265ff.; JANTZEN (1980a), S. 175ff.; SIFAKIS (1980c), S. 587ff.; MÜLLER,B. (1981), S. 201ff.; HACKMANN (1982), S. 51ff.; BERMOND (1983), S. 31ff.; MEMMI (1983b), S. 221ff.; REISIG (1986a), S. 84ff.; VALMARI (1988a), S. 99 u. 106ff.; DEDERICHS (1989b), S. 2ff.; ABEL,D. (1990), S. 2, 14f., 24f., 30ff. u. 107ff.; FEHLING (1990b), S. 3ff.; VON KLEIST-RETZOW (1991), S. 257.

Die vorgenannten Arbeiten bleiben keineswegs auf das Petrinetz-Konzept beschränkt. Statt dessen besitzen Lebendigkeit und Deadlockfreiheit eine derart fundamentale Bedeutung für die Gestaltung und Beurteilung von Prozeßkoordinierungen in dynamischen Systemen, daß sie in verschiedensten konzeptionellen Kontexten thematisiert werden. Dabei steht die Koordinierung von Informationsverarbeitungsprozessen zumeist im Vordergrund. Doch es bereitet keine Schwierigkeiten, die dort gesammelten Erkenntnisse auf die hier interessierende Koordinierung von Produktionsprozessen zu übertragen.

2) Vgl. dazu den Hinweis an anderer Stelle, daß die drei dynamischen Eigenschaften oftmals auch außerhalb des Petrinetz-Konzepts thematisiert werden.

3) Vgl. dazu exemplarisch die Definitionsvarianten, die für unterschiedliche Lebendigkeitsgrade präsentiert wurden, bei KELLER,R. (1972a), S. 38; LAUTENBACH (1973), S. 14; BYRN (1974), S. II-16ff.; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 158; LAUTENBACH (1975a), S. 3f.; SCHESCHONK (1977), S. 38f.; PETERSON,J. (1977), S. 239; BAUMGARTEN (1978), S. 31ff.; YU,S. (1978), S. 176; MEMMI (1979), S. 95; PETERSON,J. (1981), S. 86.

Die Definitionsvielfalt des Lebendigkeitsbegriffs beklagt auch BAUMGARTEN (1978), S. 30. Dort findet sich eine übersichtliche Zuordnung der verschiedenen Lebendigkeitsbegriffe zu ausgewählten Werken der Netzliteratur (S. 31ff.). Vgl. ebenso den Hinweis auf die Vielfalt alternativer Lebendigkeitsbegriffe bei REISIG (1986a), S. 84.

Vgl. ebenso die grundsätzlichen Definitionsoptionen für Deadlocks, die bei SIFAKIS (1980c), S. 587; SIFAKIS (1982), S. 228, vorgestellt werden.

4) Dies zu übersehen, führt mitunter zu Irrtümern. So behandelt beispielsweise PETERSON,J. (1978a), S. 146, die transitionsschwache und die starke Lebendigkeit von Netzen als Synonyme. Tatsächlich unterscheiden sich jedoch beide Lebendigkeitsvarianten erheblich.

5) Dazu gehören vor allem die prozeßorientierten Definitionsansätze. Dort werden in erster Linie Deadlocks (Verklemmungen), seltener Deadlockfreiheit und Lebendigkeit mit der Hilfe von Teilprozessen definiert, die in dynamischen Systemen zur Erfüllung von Aufgaben ausgeführt werden und dabei Ressourcen in Anspruch nehmen. Vgl. dazu z.B. HAVENDER (1968), S. 74 u. 76f.; HABERMANN (1969), S. 373f.; SHOSHANI (1970), S. 355; HOLT,R. (1971a), S. 1ff. u. 22f.; HOLT,R. (1971b), S. 36; COFFMAN (1971), S. 70; HACK,M. (1972), S. 7; HOLT,R. (1972b), S. 64; FANTAO (1972), S. 72; COFFMAN (1973), S. 10 u. 44; SCHROFF (1974), S. 1ff.; WENDT (1974a), S. 209; KOHLER,W. (1976), S. 11; ZIMA (1976), S. 127f. u. 150; MURATA,TA. (1976a), S. 599; BRINCH HANSEN (1977), S. 118f.; HERZOG,O. (1977), S. 209; PETERSON,J. (1978a), S. 146; KWONG (1979), S. 178; MANNA (1979), S. 395; HACKMANN (1982), S. 51; BAUER,F. (1981), S. 418; MÜLLER,B. (1981), S. 201; CARSTENSEN (1982), S. 4; ABEL,D. (1990), S. 24f.

Die markierungs- und die prozeßbezogene Deadlockdefinition stimmen im allgemeinen nicht überein. Die markierungsbezogene Deadlockdefinition betrifft einen Netzzustand, in dem überhaupt kein Prozeß mehr ausgeführt werden kann. Da die Ausführungen von Prozessen in Netzen dem Schalten von Transitionen entsprechen, bedeutet ein markierungsbezogener Deadlock eine Markierung, unter der überhaupt keine Transition geschaltet werden kann. Die prozeßbezogene Deadlockdefinition bezieht sich dagegen nur auf das Blockieren eines einzelnen Teilprozesses. Dabei läßt sich der Teilprozeß mit einer nicht-leeren, echten Teilmenge aus der Transitionenmenge eines Netzes identifizieren. Ein prozeßbezogener Deadlock stellt daher eine Markierung dar, von der aus eine prozeßspezifische Teilmenge von Transitionen nicht mehr geschaltet werden kann.

Die Unterschiedlichkeit von markierungs- und prozeßbezogenen Deadlockdefinitionen wird besonders deutlich bei HOLT,R. (1972b), S. 65; SCHROFF (1974), S. 1; ZIMA (1976), S. 151; KERAMIDIS (1979), S. 29; MONTEL (1983a), S.

295; ABEL,D. (1990), S. 14, 44, 73ff. u. 116. Dort wird zwischen partiellen und totalen Deadlocks oder Verklemmung(szuständ)en differenziert. Dabei werden die totalen Deadlocks der markierungsbezogenen Deadlockdefinition gerecht. Diese Deadlockauffassung wird den Ausführungen dieses Kapitels zunächst zugrundegelegt. Partielle Deadlocks erfüllen dagegen die prozeßbezogene Deadlockdefinition. Vgl. dazu die Behandlung partieller Deadlocks bei KELLER,R. (1972a), S. 31; BYRN (1974), S. II-19; ULLRICH (1976), S. 3/10 (der allerdings von dynamischen Deadlocks spricht); MEKLY (1980), S. 422; MONTEL (1983a), S. 295; VALK (1983b), S. 15. Auf die prozeßbezogene Betrachtungsweise partieller Deadlocks wird später kurz zurückgekommen, nachdem das Konstrukt der M_r -blockierten Transitionen eingeführt worden ist. Zum Schluß dieses Kapitels wird das prozeßbezogene Verständnis von Deadlocks ausführlicher thematisiert. Mit seiner Hilfe lassen sich unerwünschte, ressourcenbezogene Deadlocks anschaulich darstellen.

Daneben werden Deadlocks auch als nicht-leere Teilmengen der Stellenmenge eines Netzes definiert; vgl. z.B. HACK,M. (1972), S. 42; BYRN (1974), S. II-75 u. II-124; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 11; MARIN (1975), S. 2/8; JONES,N. (1977), S. 281; PRINOTH (1977), S. 309; PRINOTH (1978a), S. 16 u. 39; BAUMGARTEN (1978), S. 22f.; GRIESE,W. (1979), S. 5; PAKAS-SKEWES (1979), S. 31; GRIESE,W. (1980), S. 257 u. 262; STARKE (1980), S. 122; JANTZEN (1980a), S. 200; BERMOND (1983), S. 31; BERTHELOT (1983a), S. 44; MEMMI (1983b), S. 221; REISIG (1986a), S. 114f.; ABEL,D. (1990), S. 31f. Dabei wird aber nur auf die Netztopologie Bezug genommen. Daher handelt es sich bei solchen topologischen Deadlocks nicht mehr um eine dynamische, sondern nur noch eine statische Netzeigenschaft. Solche statischen Deadlocks brauchen weder mit markierungs- noch mit prozeßbezogen definierten Deadlocks übereinzustimmen; vgl. STARKE (1980), S. 123 (bezüglich der Deadlockmarkierungen). Sie spielen vor allem dann eine Rolle, wenn die Deadlockfreiheit oder Lebendigkeit von Free-Choice-Netzen untersucht wird. Vgl. dazu die vorgenannten Quellen. Free-Choice-Netzen kommt in dieser Arbeit aber keine weitere Bedeutung zu. Daher werden die topologisch definierten Deadlocks nicht mehr berücksichtigt.

Weitere abweichende Deadlockdefinitionen finden sich bei FERSTL (1979), S. 110 (mittelbar durch Definition der Deadlockfreiheit); SIFAKIS (1980c), S. 588; PETERSON,J. (1981), S. 85f.

Darüber hinaus wird die Lebendigkeit von Netzen oftmals nicht aus der Perspektive lebendiger Transitionen, sondern auf der Basis von lebendigen Markierungen entwickelt. Es gibt sogar Ansätze, die lebendige Stellen definieren; vgl. z.B. O.V. (1982a), S. 8; KIEBLER (o.J.), S. 28.

6) Beispielsweise wird anstelle von Deadlocks und Deadlockfreiheit oftmals von Verklemmungen bzw. Verklemmungsfreiheit gesprochen; vgl. SCHROFF (1974), S. 1ff.; ZIMA (1976), S. 127f.; LOCKEMANN (1978), S. 78; REISIG (1979d), S. 75; KERAMIDIS (1979), passim; CARSTENSEN (1982), S. 4; HERRTWICH (1989b), S. 163ff.; ABEL,D. (1990), S. 14f. u. 24f.; VON KLEIST-RETZOW (1991), S. 257. Seltener werden Deadlocks auch als Haltemarkierungen oder -zustände bezeichnet; vgl. KIEBLER (o.J.), S. 28; O.V. (1982a), S. 9. Mitunter werden Deadlocks auch als Blockierungen angesprochen; vgl. ZACHARIADES (1977), S. 1.3; STARKE (1980), S. 2; BRETSCHNEIDER (1980a), S. 40; ROUCAIROL (1982b), S. 14. Schließlich läßt sich auch auf "hang-ups" Bezug nehmen; vgl. COHEN,E. (1975), S. 186f.; BRUNO,J. (1971), S. 630; TSICHRITZIS (1971), S. 5; KWONG (1977), S. 47; PRIESE (1979), S. 13 u. 48; PRIESE (1983), S. 224. Eine noch weiter führende Variation der terminologischen Einordnung von Deadlockfreiheit findet sich bei DEDERICHS (1989b), S. 2. Dort wird zwischen einer sicherheits- und einer lebendigkeitsorientierten Konzeption der Deadlockfreiheit unterschieden.

Ein ähnliches terminologisches Chaos bietet sich bei den unterschiedlichen Lebendigkeitsbegriffen. Dort ist es besonders beliebt, die Varianten zu numerieren. Solche numerischen Variantenkennzeichnungen erscheinen nicht nur aus mnemotechnischen Gründen recht dürftig. Statt dessen werden sogar inkompatible Numerierungsweisen für transitions- und markierungsbezogene Lebendigkeitsvarianten verfolgt.

In dieser Arbeit werden alle terminologisch variierenden Bezeichnungen unter jeweils einer Bezeichnung zusammengefaßt, sofern sie sich inhaltlich auf denselben Sachverhalt beziehen. Es wird darauf verzichtet, bei den Quellenangaben stets auf ihre unterschiedliche terminologische Verankerung hinzuweisen.

7) Dies ist z.B. dann nicht möglich, wenn dem markierungsbezogenen Deadlockbegriff ein prozeßorientierter Verklemmungsbegriff gegenübersteht.

8) Der "Deadlock" ist ein etablierter Fachbegriff. Um den Anschluß an die Netzliteratur zu wahren, wird er hier nicht übersetzt.

9) Vgl. BYRN (1974), S. II-19 (totaler Deadlock); COTRONIS (1977), S. 198; TORRIGIANI (1977), S. 352; SZLANCO (1978), S. 76; GODBERSEN (1979), S. 251 u. 265; LAUER (1979), S. 113; STARKE (1980), S. 69; AZEMA (1980), S. 602; LAUER (1980), S. 456; MÜLLER,B. (1981), S. 204; STAHLKNECHT (1982), S. 119; MONTEL (1983a), S. 295; VALK (1983b), S. 15 (totaler Deadlock); FINKEL (1987b), S. 500; PAGNONI (1990), S. 143 (stop state); ROSENSTENGEL (1991), S. 39 u. 151.

Die vorgenannten Quellen beziehen sich allerdings nicht auf Schaltschritte, sondern auf das Schalten einzelner Transitionen, das Stattfinden von Ereignissen oder das Ausführen von Operationen (Prozessen). Darüber hinaus reden sie des öfteren nicht von Deadlocks, sondern von Verklemmungen oder ähnlichem.

Schließlich wird mitunter zusätzlich gefordert, daß die Transitionen nicht nur unter der Deadlockmarkierung, sondern ebenso unter allen Folgemarkierungen nicht aktiviert sind. Dieses Postulat kann daraus resultieren, daß der

Deadlockcharakter einer Markierung M_r aus der M_r -Blockierung aller Transitionen hergeleitet wird. Die M_r -Blockierung einer Transition impliziert dann tatsächlich die Zusatzforderung, daß sie unter keiner Folgemarkierung aktiviert ist. Die Generalisierung dieses zusätzlichen Postulats auf *alle* Transitionen eines Netzes führt jedoch zu deren Abundanz. Denn wenn *alle* Transitionen unter der Markierung M_r nicht aktiviert sind, dann ist es unter allen erreichbaren Folgemarkierungen unmöglich, jemals auch nur eine Transition zu aktivieren. Das wird an späterer Stelle näher dargelegt. Dort wird auch aufgezeigt, daß beide Deadlockdefinitionen äquivalent sind.

10) Vgl. WENDT (1974a), S. 209; JUMP (1975), S. 600; ULLRICH (1976), S. 3/10 (bezüglich der dort angeführten "statischen" Deadlocks); KWONG (1977), S. 47; LOCKEMANN (1978), S. 78; AGERWALA (1978a), S. 150; LAUER (1978a), S. 305; AGERWALA (1979), S. 89; PRIESE (1979), S. 13; BRETSCHNEIDER (1980a), S. 40; O.V. (1982a), S. 9; KIEBLER (o.J.), S. 28. Es gilt die gleiche Einschränkung wie in der voranstehenden Anmerkung.

11) Im Gegensatz zu den prozeßorientierten Definitionsansätzen, die in einer anderen Anmerkung angesprochen werden, wird hier ein Deadlock nicht als Eigenschaft von Prozessen definiert. Statt dessen ist das Prozeßenden hier eine Konsequenz der Existenz eines Deadlocks. Sie folgt erst aus der Verknüpfung der markierungsbezogenen Deadlockdefinition mit der Definition von Prozessen in Netzen.

12) Vgl. zum Verklemmen oder Blockieren von Prozessen die Quellen, die bereits in einer früheren Anmerkung angeführt wurden.

13) Dies erklärt den Terminus technicus "Deadlock"; vgl. z.B. BEST, E. (1975a), S. 188; KREOWSKI (1981), S. 312; ABEL, D. (1990), S. 14.

14) Es müssen nicht nur alle nicht-leeren Teilmengen der Transitionenmenge T berücksichtigt werden, sondern ebenso für jede Transitionenteilmenge alle zulässigen Schaltfarbenkombinationen.

15) Diese Äquivalenz läßt sich leicht dadurch indirekt beweisen, daß in beiden Äquivalenzrichtungen jeweils die Inkonsistenz des kontradiktorischen Gegenteils aufgezeigt wird: Erstens ist es widersprüchlich anzunehmen, daß kein Schaltschritt aktiviert ist, aber mindestens eine Transition bezüglich einer beliebigen Schaltfarbe aktiviert ist. Denn dann kann mindestens ein degenerierter Schaltschritt gebildet werden, der aus eben dieser aktivierten Transition und einer ihrer Schaltfarben besteht. Dieser Schaltschritt ist dann per constructionem auch aktiviert. Folglich ist bewiesen: Wenn kein Schaltschritt aktiviert ist, dann kann auch keine Transition bezüglich einer beliebigen Schaltfarbe aktiviert sein. Zweitens führt es in einen Widerspruch zu unterstellen, daß keine Transition mit einer beliebigen Schaltfarbe, wohl aber mindestens ein Schaltschritt aktiviert sei. Denn jeder Schaltschritt enthält mindestens eine Transition. Aus der Definition der Schaltschrittaktivierung folgt, daß alle zugehörigen Transitionen mit jeweils einer Schaltfarbe aktiviert sein müssen. Also muß mindestens eine Transition mit mindestens einer Schaltfarbe aktiviert sein. Daher ist ebenso bewiesen: Wenn keine Transition bezüglich einer beliebigen Schaltfarbe aktiviert ist, dann kann auch kein Schaltschritt aktiviert sein. Aufgrund der voranstehend bewiesenen Äquivalenz gilt also auch: Eine Deadlockmarkierung liegt genau dann vor, wenn unter ihr keine Transition aktiviert ist.

16) Vgl. NIETERS (1978), S. 9; STARKE (1980), S. 119.

17) Vgl. dazu die Rechtfertigung von Nullschaltfolgen. Die Nullschaltfolge widerspricht nicht der Definition von Deadlockmarkierungen, unter denen keine Transition aktiviert sein darf. Denn die Nullschaltfolge $SF_0 = ()$ enthält überhaupt keinen Schaltschritt. A fortiori setzt sie auch nicht die Aktivierung irgendeiner Transition voraus.

18) Vgl. dazu die allgemeine Definition der Erreichbarkeitsrelation $RR(M_0, SR_S)$. Die Bezugnahme auf die Schaltregel SR_S entfällt hier aufgrund der Vereinbarung, sie der Einfachheit halber nicht mehr explizit anzuführen. Dies wurde schon für die Notation der Erreichbarkeitsmenge dargelegt.

19) Falls die Beschriftungsfunktion "bk" durch die Anschriftenmenge BK ersetzt wird, die alle Paare aus Schaltkanten und deren Anschriften durch jeweils ausgeführte Schaltschritte umfaßt, so ist diese Anschriftenmenge leer: $BK = \emptyset$. Zusammen mit der leeren Kantenmenge $KA_{RG} = \emptyset$ und der Knotenmenge $KN_{RG} = M_0$ gilt dann für den Erreichbarkeitsgraphen jedes toten Netzes: $RG(M_0) = (M_0, \emptyset; \emptyset)$.

20) In der Netzliteratur werden im allgemeinen *andere* Definitionen für die Lebendigkeit einer Markierung verwendet. Vgl. dazu die Hinweise auf 1-, 2- und 5-lebendige Netzmarkierungen. Am nächsten kommt dem o.a. Lebendigkeitsprädikat $LEB(M_r)$ noch die Definition der Markierungslebendigkeit durch BAUMGARTEN (1978), S. 31. Dort wird eine Markierung M_r genau dann als lebendig betrachtet, wenn unter ihr mindestens eine Transition t_n lebendig ist. Dabei definiert BAUMGARTEN die Transitionslebendigkeit im Sinne der M_r -Lebendigkeit. Es läßt sich zeigen, daß BAUMGARTEN's Definition mit der des Verf. äquivalent ist. Denn unter einer Markierung M_r ist genau dann mindestens eine Transition t_n aktiviert, wenn bezüglich derselben Markierung mindestens eine Transition t_n M_r -lebendig ist. Einerseits impliziert die Aktivierung einer beliebigen Transition t_n unter einer Markierung M_r , daß zumindest diese Transition $t_n = t_h$ auch M_r -lebendig ist. Das folgt unmittelbar aus dem Sachverhalt, daß die Markierung M_r von sich selbst aus durch die Nullschaltfolge erreichbar ist und qua Voraussetzung zumindest unter dieser "Folge"markierung $M_r = M_r$ aktiviert ist. Andererseits kann unter einer Markierung M_r nur dann eine M_r -lebendige Transition t_h existieren, wenn unter derselben Markierung M_r mindestens eine Transition t_n aktiviert ist. Wäre dies

nicht der Fall, so könnte überhaupt keine Schaltfolge ausgeführt werden, welche von der Markierung M_r aus startete und zur Aktivierung der M_r -lebendigen Transition t_n unter ihrer Zielmarkierung M_r führte. Das Fehlen einer solchen Schaltfolge widerspräche jedoch der vorausgesetzten M_r -Lebendigkeit der Transition t_n . Damit ist bewiesen, daß sich die beiden Definitionen lebendiger Netzmarkierungen durch BAUMGARTEN und durch den Verf. äquivalent verhalten. Darüber hinaus findet sich bei BAUMGARTEN (1978), S. 31, selbst eine Zuordnung zwischen seiner Formulierung, eine Transition t_n sei lebendig bezüglich einer Markierung M_r , und der vom Verf. bevorzugten Variante, eine Transition t_n sei unter derselben Markierung M_r aktiviert.

21) Vgl. CAVARROC (1974), S. 98; VOSS, K. (1979), S. 806; ROUCAIROL (1982b), S. 14; BERTHELOT (1983a), S. 44; PRIESE (1983), S. 224.

Eine weitergehende Folgerung zieht HERZOG, O. (1973), S. 3. Seiner Ansicht nach bedeutet Deadlockfreiheit - in die Terminologie dieser Arbeit übersetzt - sogar, daß ein Netz keine tote Transition besitzt. Dieser doppelten Anforderung, sowohl Deadlockfreiheit als auch Abwesenheit von toten Transitionen zu garantieren, wird erst später durch die Variante der starken Netzlebendigkeit entsprochen. Er wird aber in dieser Arbeit mit der Deadlockfreiheit nicht identifiziert.

22) Vgl. STARKE (1980), S. 122.

23) Diese Konzeptualisierungsperspektive nimmt der Verf. ein, weil sie in besonders anschaulicher Weise gestattet, unterschiedliche Lebendigkeitsvarianten sukzessiv zu entfalten. Dies wird im folgenden demonstriert. Vgl. zu dieser transitionsfundierten Vorgehensweise auch KELLER, R. (1972a), S. 38; BYRN (1974), S. II-16ff.; PETERSON, J. (1981), S. 86.

Allerdings wird in der Netzliteratur des öfteren auch ein markierungsbezogener Ansatz verfolgt, um die Lebendigkeit von Netzen zu definieren. Vgl. dazu die Erläuterungen zu den drei Varianten der Netzlebendigkeit.

24) Vgl. KELLER, R. (1972a), S. 32 u. 38; MASSACHUSETTS COMPUTER ASSOCIATES (1974b), S. 158; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 10; BYRN (1974), S. II-16; MITTERMEIR (1974), S. 10; LIEN (1976a), S. 352; MURATA, T. A. (1976c), S. 37; PETERSON, J. (1977), S. 239; BAUMGARTEN (1978), S. 30; AGERWALA (1978b), S. 309; YU, S. (1978), S. 176; PRINOTH (1978b), S. 3; PRINOTH (1979a), S. 6; GODBERSEN (1979), S. 251 u. 265; MEMMI (1979), S. 95; SURAJ (1980), S. 629; PETERSON, J. (1981), S. 85f.; BERTHELOT (1983a), S. 49.

Die M_r -Lebendigkeit einer Transition wird in den vorgenannten Quellen zumeist als transitionsbezogene 1-Lebendigkeit, mitunter auch als potentielle Schaltbarkeit oder Quasi-Lebendigkeit thematisiert.

25) Falls die M_r -lebendige Transition bereits unter der Markierung M_r selbst aktiviert ist, fällt eine erreichbare Markierung M_p unter der die Transition aktiviert ist, mit der Referenzmarkierung M_r zusammen: $AKT(t_n, M_p)$ und $M_r = M_p$. Abweichender Ansicht ist LIEN (1976a), S. 352, der diesen Sonderfall für Transitionssysteme ausdrücklich verneint. Die Aktivierung einer M_r -lebendigen Transition unter der Referenzmarkierung M_r wird dagegen von MURATA, T. A. (1976c), S. 37; SIFAKIS (1982), S. 249 (ebenfalls im Kontext von Transitionssystemen), zugelassen. Der Verf. schließt sich der letzten Ansicht an. Denn hierdurch läßt sich vermeiden, in der Definition M_r -lebendiger Transitionen das Erreichen der Referenzmarkierung durch die Nullschaltfolge verbieten zu müssen.

Oftmals ist eine M_r -lebendige Transition aber unter der Referenzmarkierung M_r selbst nicht aktiviert. Dann besteht aber immer die Möglichkeit, eine andere Folgemarkierung $M_r \neq M_r$ zu erreichen, unter der die Transition aktiviert ist.

Beide voranstehenden Fälle lassen sich unter die Formulierung subsumieren, daß die M_r -Lebendigkeit einer Transition deren *Aktivierungsmöglichkeit* ausdrückt; vgl. SIFAKIS (1982), S. 249. Diese Charakterisierung ist präziser als die ähnlich lautende Formulierung der potentiellen *Schaltbarkeit*, die in einer früheren Anmerkung angeführt wurde. Denn potentiell schaltbar ist infolge der Permissivität der Schaltregel jede Transition, die unter der Markierung M_r aktiviert ist. Als potentiell aktivierbar erweist sich dagegen nur jede M_r -lebendige Transition.

26) Vgl. HOLT, A. (1970a), S. 25; BAER, J. (1973a), S. 50; BAUMERS (1977), S. 13; ROUCAIROL (1982b), S. 11; FINKEL (1982), S. 114; O. V. (1982a), S. 8; FINKEL (1987b), S. 500; ROSENSTENGEL (1991), S. 39; KIEBLER (o.J.), S. 28.

Die M_0 -Lebendigkeit wird zuweilen als Pseudo- oder Quasi-Lebendigkeit angesprochen.

27) Vgl. zur Unsterblichkeit von Transitionen KELLER, R. (1972a), S. 32 u. 38; HACK, M. (1972), S. 12 u. 40; MASSACHUSETTS COMPUTER ASSOCIATES (1974b), S. 158; BYRN (1974), S. II-16; HACK, M. (1974b), S. 158; HACK, M. (1975a), S. 36f.; KELLER, R. (1975a), S. 17; LAUER (1975b), S. 96; ULLRICH (1976), S. 2/12 u. 3/9; COOPRIDER (1976), S. 7; PETERSON, J. (1977), S. 239; KRIEG, B. (1977), S. 9; GOSH (1977), S. 4; SIFAKIS (1977a), S. 77; YU, S. (1978), S. 176; BAUMGARTEN (1978), S. 31; MEMMI (1978a), S. 128; MEMMI (1978c), S. 506; PRINOTH (1978a), S. 14; GODBERSEN (1979), S. 251 u. 265; MEMMI (1979), S. 95; GRIESE, W. (1979), S. 4; PAKAS-SKEWES (1979), S. 20f.; WINDISCH (1979), S. 17; STARKE (1980), S. 69 u. 119; HEINEMANN (1980), S. 57; JANTZEN (1980a), S. 175 (allerdings nicht in bezug auf die Ausgangsmarkierung M_0 , sondern auf eine beliebige Markierung); BERTHELOT (1980a), S. 278; AZEMA (1980), S. 601; MEKLY (1980), S. 422; SUZUKI, I. (1980a), S. 620; PETERSON, J. (1981), S. 85f.; MÜLLER, B. (1981), S. 204; TÖRN (1981), S. 188; ROUCAIROL (1982b), S. 12; SANDRING (1982), S. 218; STREPPPEL (1982), S. 4; BURKHARD (1982a), S. 86; PRIESE (1983), S. 224 (bei einelementiger Markierungs-

menge); VALK (1983c), S. 357; REISIG (1986a), S. 85; FINKEL (1987b), S. 500; PAGNONI (1990), S. 143; ABEL,D. (1990), S. 14.

Die Unsterblichkeit einer Transition wird in den vorgenannten Quellen überwiegend als Lebendigkeit schlechthin, seltener auch als transitionsbezogene 4-Lebendigkeit angesprochen. Den besonders anschaulichen Begriff der Unsterblichkeit hat der Verf. von KELLER, BAUMGARTEN und GODBERSEN übernommen.

28) Dies übersieht z.B. PAKAS-SKEWES (1979), S. 69, wenn er die Lebendigkeit eines Netzmodells anscheinend damit verknüpft, daß jede seiner Transitionen mit einer positiven "Wahrscheinlichkeit" - treffender wohl: Häufigkeit - geschaltet wird.

29) Vgl. zu M_r -blockierten Transitionen, die zumeist als tot bezüglich der Markierung M_r bezeichnet werden, GENRICH (1971b), S. 5; KELLER,R. (1972a), S. 38; LAUTENBACH (1973), S. 14; GENRICH (1973a), S. 148; BYRN (1974), S. II-16; MITTERMEIR (1974), S. 10; HACK,M. (1974b), S. 158; LAUTENBACH (1975a), S. 3; HACK,M. (1975a), S. 36; BEST,E. (1975a), S. 188; SCHESCHONK (1977), S. 38; COTRONIS (1977), S. 198; PETERSON,J. (1977), S. 239; THIELER-MEVISSEN (1977), S. 9; SCHIFFERS (1977), S. 30; EGGERT (1978), S. 52; NIETERS (1978), S. 9; BAUMGARTEN (1978), S. 8 u. 30; GODBERSEN (1979), S. 251 u. 265; STARKE (1980), S. 118f.; LAUER (1980), S. 456; JANTZEN (1980a), S. 175 u. 177; SURAJ (1980), S. 629; PETERSON,J. (1981), S. 86; KREOWSKI (1981), S. 312; REISIG (1982a), S. 125; VALK (1983b), S. 15 (bei Degenerierung der dort behandelten Transitionen-Teilmengen zu einzelnen Transitionen), S. 24 u. 40; VALK (1983c), S. 357.

30) Vgl. GENRICH (1978a), S. 215; PETERSON,J. (1978a), S. 146; SCHUMACHER (1978), S. 7; MEMMI (1979), S. 95; HACKMANN (1982), S. 17; O.V. (1982a), S. 8; ABEL,D. (1990), S. 14; ROSENSTENGEL (1991), S. 39.

Beispiele für solche toten Transitionen finden sich bei BYRN (1974), S. II-129, Fig. II-D-2; LAUER (1978a), S. 319; PETERSON,J. (1981), S. 87; ROUCAIROL (1982b), S. 15.

31) Mitunter wird eine tote Transition auch unmittelbar dadurch definiert, daß sie unter allen erreichbaren Markierungen M_r -blockiert ist. Beide Definitionsansätze verhalten sich zwar äquivalent. Aber der Verf. bevorzugt den oben gewählten, weil die unmittelbare Identifizierung mit einer M_r -blockierten Transition kompakter und übersichtlicher ausfällt.

32) Vgl. STARKE (1980), S. 119.

33) Darüber hinaus können nunmehr auch die partiellen Deadlocks präzisiert werden, die schon in einer der voranstehenden Anmerkungen angesprochen wurden: Eine erreichbare Markierung M_r stellt einen partiellen Deadlock dar, falls alle Transitionen aus einer nicht-leeren, echten Teilmenge der Transitionenmenge eines Netzes M_r -blockiert sind. Allerdings werden partielle Deadlocks des öfteren mit der Hilfe von blockierten Teilprozessen definiert. Dagegen werden sie hier auf Teilmengen blockierter Transitionen zurückgeführt. Dies spielt jedoch keine Rolle, weil Teilprozesse in Netzen als prozeßspezifische Transitionenteilmengen dargestellt werden können. Auch das wurde schon in einer früheren Anmerkung erwähnt.

34) Vgl. LAUTENBACH (1973), S. 14; MITTERMEIR (1974), S. 10; LAUTENBACH (1975a), S. 3; SCHESCHONK (1977), S. 38; COTRONIS (1977), S. 198; SCHIFFERS (1977), S. 30; BAUMGARTEN (1978), S. 8 u. 31; EGGERT (1978), S. 52; GODBERSEN (1979), S. 251 u. 265; STARKE (1980), S. 118; JANTZEN (1980a), S. 175.

Allerdings wird in den vorgenannten Quellen von toten Markierungen und Transitionen gesprochen. Ungeachtet dieser terminologischen Variationen wird daraus die Bezeichnung von Deadlocks als Verklemmungen oder Blockierungen besonders deutlich.

35) Gleiches gilt bei Bezugnahme auf eine *echte*, nicht-leere Teilmenge der Transitionenmenge eines Netzes.

36) Einerseits folgt aus der Blockierung einer Transition unter der Markierung M_r , daß sie unter allen Markierungen M_r mit $M_r \in RM(M_r)$ nicht aktiviert sein kann. Da jede Markierung wegen $M_r \in RM(M_r)$ von sich selbst aus durch Ausführen der Nullschaltfolge erreichbar ist, sind a fortiori alle M_r -blockierten Transitionen unter der Referenzmarkierung M_r nicht aktiviert. Daher gilt bei universeller Betrachtung *aller* Transitionen eines Netzes: $BLO(t_n, M_r) \Rightarrow \neg AKT(t_n, M_r)$. Andererseits impliziert die Nichtaktivierung aller Transitionen unter derselben Markierung M_r , daß im Netz unter dieser Markierung überhaupt kein Schaltschritt mehr ausgeführt werden kann. Alle Schaltprozesse kommen notwendig zum Erliegen. Daher schrumpft die Menge aller Folgemarkierungen M_p , die sich von der Referenzmarkierung M_r aus erreichen lassen, auf die einelementige Menge $RM(M_r) = M_r$ zusammen. Es läßt sich nur noch die "Folge"markierung $M_r = M_r$ erreichen, indem die Nullschaltfolge zur Referenzmarkierung M_r zurückführt. Da laut Voraussetzung schon unter der Markierung M_r alle Transitionen nicht aktiviert sind, bleiben sie auch unter "allen erreichbaren" Folgemarkierungen - nämlich der genau einen Markierung $M_r = M_r$ - nicht aktiviert. Damit erfüllen alle Transitionen die Definition der M_r -Blockiertheit. Also gilt bei universeller Betrachtung *aller* Transitionen eines Netzes ebenso: $\neg AKT(t_n, M_r) \Rightarrow BLO(t_n, M_r)$.

37) Die Äquivalenz der beiden Definitionsvarianten für Deadlocks geht erst dann verloren, wenn auch partielle Deadlocks betrachtet werden. Auf solche Deadlocks wurde bereits hingewiesen. Sie werden in dieser Arbeit jedoch nicht weiter thematisiert.

38) Die Äquivalenz läßt sich ohne Schwierigkeiten aufzeigen: Einerseits folgt aus der Definition eines toten Netzes durch die Deadlockeigenschaft seiner Ausgangsmarkierung M_0 , daß unter dieser Markierung M_0 alle Transitionen nicht aktiviert sind. Falls unter derselben Markierung eines Netzes überhaupt keine Transition aktiviert ist, sind alle Transitionen blockiert. Dies wurde schon in einer früheren Anmerkung für eine beliebige Deadlockmarkierung aufgezeigt. Da hier alle Transitionen bezüglich der Ausgangsmarkierung M_0 blockiert sind, sind sie per definitionem tot. Andererseits impliziert die Definition der Totheit eines Netzes durch die Totheit aller seiner Transitionen, daß keine dieser Transitionen unter der Ausgangsmarkierung aktiviert ist. Eine Markierung, unter der keine Transition aktiviert ist, stellt per definitionem einen Deadlock dar. Also ist die Ausgangsmarkierung eine Deadlockmarkierung.

39) Vgl. dazu auch ROUCAIROL (1982b), S. 11.

40) Der Übersichtlichkeit halber beschränken sich die anschließenden Exempel zunächst auf den Blockierungsaspekt. Die korrespondierenden Ausführungen zur Lebendigkeit finden sich in den jeweils zugehörigen Anmerkungen.

41) Die Existenz toter Transitionen wird ebenfalls als ein Indiz fehlerhafter Netze betrachtet von BYRN (1974), S. III-45; vgl. darüber hinaus die Quellen aus den anschließenden Anmerkungen, die sich jeweils auf spezielle Fehleraspekte konzentrieren.

42) Vgl. O.V. (1982a), S. 8; KIEBLER (o.J.), S. 28. Dort wird von unvollständigen Systementwürfen gesprochen. Die M_0 -Lebendigkeit und die Unsterblichkeit derselben Transition ließen dagegen erkennen, daß die Koordinierung der Objektaktivitäten zumindest partiell korrekt ist: Sie läßt die Erfüllung der erwünschten Funktion durch Aktivitäten der abgebildeten Objektkomponente mindestens einmal bzw. immer wieder zu.

43) Vgl. GRILL (1982), S. 41. Er behandelt den voranstehend skizzierten Fall in bezug auf den erwünschten Output eines Automatischen Informationsverarbeitungssystems.

44) Vgl. zur Differenzierung zwischen Redundanz- und Abundanzbegriff.

45) Vgl. ULLRICH (1976), S. 3/10; KRÄMER (1981), S. 470; HACKMANN (1982), S. 18; ROUCAIROL (1982b), S. 11; O.V. (1982a), S. 8; KIEBLER (o.J.), S. 28.

Umgekehrt würde die M_0 -Lebendigkeit einer Transition bedeuten, daß die zugehörige Objektkomponente nicht überflüssig wäre. Vielmehr würde sie in mindestens einem zulässigen Objektzustand "r" ihre Funktion erfüllen können. Dies wird im Netzmodell dadurch abgebildet, daß die Transition unter mindestens einer erreichbaren Markierung M_r aktiviert ist. Die Unsterblichkeit derselben Transition zeigte sogar an, daß die Objektkomponente immer wieder ihre funktionserfüllende Aktivität entfalten könnte.

46) Dann würde bereits die M_0 -Lebendigkeit der Transition bedeuten, daß die korrekte Funktionsweise des Objekts in mindestens einer Situation verletzt wird. Bei Lebendigkeit derselben Transition wäre die Objektfunktion sogar immer wieder gestört.

47) Gleiches gilt für die Lebendigkeit, vgl. dazu die entsprechenden voranstehenden Anmerkungen.

48) Gemeint sind damit die Techniken zur Untersuchung von Netzeigenschaften, die im Abschnitt 5.2.3 vorgestellt werden, insbesondere die im Abschnitt 5.2.3.5 ausführlicher behandelte Technik der Erreichbarkeitsanalyse.

49) Vgl. dazu die Ausführungen zur Problematik intensionaler Konzepte, die im Gegensatz zu extensionalen Konzepten versuchen, nicht nur formal definierte, sondern auch materielle Konstruktbedeutungen zu berücksichtigen.

50) Vgl. dazu die Anmerkungen zu den tendenziellen Effizienzvorteilen formaler - d.h. rein syntaktisch definierter - Konzepte.

51) Vgl. zu inhaltlich äquivalenten, aber zumeist formal anders präsentierten Definitionen der transitionsschwachen Netzlebendigkeit GENRICH (1973a), S. 147f.; WENDT (1974a), S. 209; LAUER (1975a), S. 331; MAYR, E. (1975), S. 58; ULLRICH (1976), S. 3/9; TORRIGIANI (1977), S. 352; PETERSON, J. (1978a), S. 146; SZLANCO (1978), S. 76; MICHEL, M. (1978), S. 180; LOCKEMANN (1978), S. 78 u. 80f.; ROSENSTENGEL (1979), S. 14f.; LAUER (1980), S. 503; GIRAULT (1982c), S. 14; vgl. ansatzweise auch HERZOG, O. (1973), S. 11.

Die transitionsschwache Netzlebendigkeit basiert auf dem Konzept der 1-lebendigen Markierungen. Eine Markierung M_r heißt genau dann 1-lebendig, wenn von ihr aus keine Deadlockmarkierung erreicht werden kann; vgl. LAUTENBACH (1973), S. 14; LAUTENBACH (1974a), S. 188; BEST, E. (1974), S. 11; MITTERMEIR (1974), S. 11; LAUTENBACH (1975a), S. 3; BEST, E. (1975a), S. 188; SCHESCHONK (1977), S. 38; SCHIFFERS (1977), S. 30f.; BAUMGARTEN (1978), S. 8 u. 31, der allerdings von unsterblichen Markierungen redet; EGGERT (1978), S. 53; GODBERSEN (1978), S. 47; LAUER (1980), S. 502; KREOWSKI (1981), S. 312.

Falls mit $M_r = M_0$ die Ausgangsmarkierung M_0 eines Netzes 1-lebendig ist, dann läßt sich von der Ausgangsmarkierung aus keine Deadlockmarkierung erreichen. Folglich handelt es sich bei allen Markierungen M_r aus der Erreichbarkeitsmenge eines Netzes mit 1-lebendiger Ausgangsmarkierung jeweils um lebendige Markierungen im Sinne des Prädikats $LEB(M_r)$. Daher wird die o.a. Definition der transitionsschwachen Netzlebendigkeit erfüllt.

Die transitionsschwache Netzlebendigkeit entspricht der "Sicherheit" von Netzen und anderen dynamischen Systemen, die beschrieben wird von HOLT,R. (1971a), S. 24; COFFMAN (1971), S. 75; GOSTELOW (1971), S. 49 u. 158; HOLT,R. (1972b), S. 65; COHEN,E. (1975), S. 91; ZIMA (1976), S. 214. Dieser lebendigkeitsäquivalente Sicherheitsbegriff ist nicht zu verwechseln mit der o.a. Eigenschaft der Netzsicherheit von 1-beschränkten Netzen. Abweichender Ansicht ist KELLER,R. (1972a), S. 33, der die o.a. Netzsicherheit mit der markierungsschwachen Lebendigkeit gleichsetzt, die in dieser Arbeit an anderer Stelle vorgestellt wird.

52) Eine abweichende Einstufung findet sich bei LAUTENBACH (1973), S. 21; LAUTENBACH (1975a), S. 4 u. 13. Dort werden die Netzeigenschaften, die hier als transitionsschwache und markierungsschwache Netzlebendigkeit bezeichnet werden, als starke bzw. schwache Lebendigkeitsvarianten betrachtet. Wie LAUTENBACH zu seiner Ansicht gelangt, vermag der Verf. allerdings nicht nachzuvollziehen. Dagegen klingen bei BYRN (1974), S. III-29, und BAUMGARTEN (1978), S. 31, Beurteilungen der Strenge von Lebendigkeitsdefinition an, die sich mit der des Verf. decken.

53) Vgl. zur Verknüpfung von Deadlockfreiheit (Verklemmungsfreiheit u.ä.) und Lebendigkeit (Sicherheit) COFFMAN (1971), S. 75; HACK,M. (1972), S. 7 u. 35; HERZOG,O. (1973), S. 3 u. 10f.; NOE (1973a), S. 718; BEST,E. (1974), S. 1; ULLRICH (1976), S. 0-8; BERTHELOT (1976), S. 202; AZEMA (1976b), S. 111; HERZOG,O. (1976b), S. 3; HERZOG,O. (1976c), S. 57; COOPRIDER (1976), S. 7 (der sich allerdings inhaltlich auf die starke Netzlebendigkeit bezieht); SCHESCHONK (1977), S. 38; THIELER-MEVISSSEN (1977), S. 2; BERTHELOT (1977), S. 217; MURATA,TA. (1977a), S. 415; EGGERT (1978), S. 52f.; PETERSON,J. (1978a), S. 146; GODBERSEN (1978), S. 47; MICHEL,M. (1978), S. 180; KERAMIDIS (1979), S. 28 u. 31; MURATA,TA. (1979a), S. 807; AGERWALA (1979), S. 92; JOHNSONBAUGH (1979), S. 387; VALETTE (1979b), S. 157; LAUER (1980), S. 477; MURATA,TA. (1980a), S. 525; SUZUKI,I. (1980a), S. 620; MEKLY (1980), S. 422; BAER,J. (1982), S. 77; VALETTE (1982c), S. 5; MONTEL (1983a), S. 295.

Vgl. des weiteren zur äquivalenten Kontraposition, daß die Existenz mindestens eines Deadlocks die Lebendigkeit eines Netzes vereitelt, BEST,E. (1974), S. 82; SURAJ (1980), S. 629; o.V. (1983a), S. 6.

54) Diese Umkehrung folgt, wie anschließend aufgezeigt wird, unmittelbar aus der Definition der transitionsschwachen Netzlebendigkeit und der Deadlockdefinition. Allerdings trifft dies nur für diese Variante der transitionsschwachen Lebendigkeit zu. Bei anderen, stärkeren Definitionsvarianten liegen andere Verhältnisse vor. Darauf wird an anderer Stelle zurückgekommen. Vgl. dazu auch die Hinweise auf Netze, die sich trotz ihrer Deadlockfreiheit nicht mehr lebendig verhalten, bei EGGERT (1978), S. 52; ROUCAIROL (1982b), S. 14f.; ROSENSTENGEL (1991), S. 39.

55) Dies folgt aus der früher gerechtfertigten Prämisse, von extensionalen Begriffsdefinitionen auszugehen. Denn äquivalente Prädikate besitzen gleiche Mengen prädikaterfüllender Argumente als Extensionen. Extensionsgleiche Begriffe werden als bedeutungsgleich (synonym) behandelt.

56) Dieser Schaltschritt SS_a umfaßt die eine Transition t_n , die aufgrund der Definition des Lebendigkeitsprädikats $LEB_s(SN)$ unter der betrachteten erreichbaren Markierung M_i aktiviert sein muß: $SS_a = \{t_n\}$.

57) Dies verdeutlicht die Eigenschaftsbezeichnung "lebendig". Vgl. zu diesem Aspekt der Netzlebendigkeit auch HERZOG,O. (1973), S. 11.

58) Für die Bedeutung der Netzlebendigkeit in unterschiedlichen Modellierungskontexten gelten die gleichen Hinweise, die bereits oben zur äquivalenten Netzeigenschaft der Deadlockfreiheit erfolgten.

59) Vgl. BEST,E. (1975a), S. 188; BAUMGARTEN (1978), S. 8; GIRAULT (1982c), S. 14ff.

60) In dieser Arbeit wird unter der Lebendigkeit eines Netzes - ohne präzisierenden Zusatz - stets die zunächst präzentrierte transitionsschwache Definitionsvariante verstanden. Wenn sie von den später ergänzten markierungsschwachen und starken Lebendigkeitsvarianten explizit abgegrenzt werden soll, wird sie als transitionsschwache Netzlebendigkeit angesprochen.

61) Falls ein Netzmodell keine toten Transitionen enthält, müssen sie alle M_0 -lebendig sein. M_0 -lebendige Transitionen können Komponenten eines modellierten Objekts abbilden, die weder abundant noch redundant sind. Daher besitzt das modellierte Objekt keine überflüssigen Komponenten, wenn sein Netzmodell frei von toten Transitionen ist. Auf diesen Aspekt der Lebendigkeit von Netzmodellen für reale Objekte weist auch AGERWALA (1978a), S. 152, hinsichtlich des netzgestützten Entwurfs elektronischer Schaltkreise hin.

62) Vgl. zu inhaltlich äquivalenten, aber zumeist formal anders aufbereiteten Definitionen der markierungsschwachen Netzlebendigkeit GENRICH (1971b), S. 5; BAER,J. (1973a), S. 50; GENRICH (1973a), S. 147; HERZOG,O. (1973), S. 3 (in abweichender Terminologie); CAVARROC (1974), S. 98; MURATA,TA. (1977c), S. 3; ROSENSTENGEL (1979), S. 14f.; TOULOTTE (1979), S. 224 (dort wird von Pseudo-Lebendigkeit gesprochen); ROUCAIROL (1982b), S. 11; FINKEL (1982), S. 114; FINKEL (1987b), S. 500; ROSENSTENGEL (1991), S. 39 u. 151 (auf S. 151 allerdings mit einem fehlerhaften Bezug auf Stellen anstatt auf Transitionen).

Die markierungsschwache Lebendigkeit beruht auf dem Konzept der 2-lebendigen Markierungen. Eine Markierung M_r heißt genau dann 2-lebendig, wenn unter ihr alle Transitionen M_r -lebendig sind, d.h. wenn unter ihr keine Transition M_r -blockiert ist; vgl. GENRICH (1971b), S. 5; LAUTENBACH (1973), S. 14; GENRICH (1973a), S. 148 (dort allerdings als 1-Lebendigkeit angesprochen); BEST,E. (1974), S. 11; BYRN (1974), S. II-19; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 14; LAUTENBACH (1975a), S. 4; MURATA,TA. (1976c), S. 37; SCHESCHONK (1977), S. 38; COTRONIS (1977), S. 198; GODBERSEN (1978), S. 47; AZEMA (1978), S. 169; BAUMGARTEN (1978), S. 9 u. 31, der sich jedoch auf "überall lebendige" Markierungen bezieht; BERTHELOT (1983a), S. 49; REISIG (1986a), S. 85 (er spricht nur von Lebendigkeit).

Falls mit $M_r=M_0$ die Ausgangsmarkierung M_0 eines Netzes 2-lebendig ist, wird dadurch die o.a. Definition der markierungsschwachen Netzlebendigkeit erfüllt. Denn in einem Netz mit 2-lebendiger Ausgangsmarkierung M_0 sind per definitionem alle Transitionen M_0 -lebendig.

63) Dies folgt unmittelbar aus der voranstehenden Anmerkung: Da keine Transition eines markierungsschwach lebendigen Netzes M_0 -blockiert ist und nur M_0 -blockierte Transitionen tot wären, können diese Netze keine toten Transitionen besitzen. Dies scheint auch PAKAS-SKEWES (1979), S. 69, zu meinen, wenn er davon spricht, daß aus der Lebendigkeit eines Netzmodells folge, im jeweils abgebildeten System führe jede Komponente ihre Aktivität mit einer positiven "Wahrscheinlichkeit" - treffender wohl: Häufigkeit - aus. (Vgl. aber auch den Hinweis auf die unzulässige Verquickung von tatsächlichem Schalten und erreichbarer Aktivierung.)

64) Vgl. HERZOG,O. (1973), S. 10.

65) Vgl. zu inhaltlich äquivalenten, aber zumeist formal abweichend dargestellten Definitionen der starken Netzlebendigkeit HACK,M. (1972), S. 12; YOELI (1973), S. 7; HERZOG,O. (1973), S. 11 (undeutlich); BYRN (1974), S. II-62 u. 3-29; LÖHR (1974), S. 64; HACK,M. (1975a), S. 36; LAUER (1975b), S. 96; BERTHELOT (1976), S. 203; COOPRIDER (1976), S. 7; AZEMA (1976b), S. 110; VALETTE (1977a), S. 67; BERTHELOT (1977), S. 217; CRESPIREGHIZZI (1977), S. 180; PRINOTH (1977), S. 309; SIFAKIS (1977a), S. 77; MURATA,TA. (1977c), S. 3; GOSH (1977), S. 4; SIFAKIS (1978), S. 475; KOTOV (1978), S. 41; PETERSON,J. (1978a), S. 146; VALETTE (1978b), S. 189; MEMMI (1978a), S. 128; MEMMI (1978c), S. 506; BOUSSIN (1978), S. 1529; DEVY (1979), S. 50; CONRY (1979a), S. 254; AYACHE (1979a), S. 1052; TOULOTTE (1979), S. 224; GRIESE,W. (1979), S. 4; WINDISCH (1979), S. 17f.; STARKE (1980), S. 69; JANTZEN (1980a), S. 175; AUGIN (1980), S. 57; BERTHELOT (1980a), S. 278 (mit redundanter Bezugnahme auf alle Zustände eines Transitionssystems, die jeweils den erreichbaren Markierungen eines Netzes entsprechen); LAUER (1980), S. 503; GENRICH (1980b), S. 529; SUZUKI,I. (1980a), S. 620; MÜLLER,B. (1981), S. 204; REISIG (1982a), S. 125; ROUCAIROL (1982b), S. 13f.; BURKHARD (1982a), S. 86; VALK (1983b), S. 35; VALK (1983c), S. 357; REISIG (1986a), S. 26 u. 85; FINKEL (1987b), S. 500; PAGNONI (1990), S. 143.

Der starken Netzlebendigkeit liegt das Konzept der 5-lebendigen Markierungen zugrunde. Eine Markierung M_r heißt genau dann 5-lebendig, wenn unter allen ihren Folgemarkierungen M_f mit $M_f \in RM(M_r)$ jeweils alle Transitionen M_f -lebendig sind, d.h. wenn keine von diesen Transitionen M_f -blockiert ist; vgl. HOLT,A. (1970a), S. 24; LAUTENBACH (1975a), S. 4; SHAPIRO,R. (1975b), S. 144; ULLRICH (1976), S. 2/12; SCHESCHONK (1977), S. 39; BAUMGARTEN (1978), S. 9 u. 31, allerdings unter Bezugnahme auf "überall unsterbliche" Markierungen; LAUER (1978a), S. 305; STARKE (1980), S. 69; KREOWSKI (1981), S. 312; PRIESE (1983), S. 224 (bei einelementiger Markierungsmenge); THIAGARAJAN (1983b), S. 326.

Falls mit $M_r=M_0$ die Ausgangsmarkierung M_0 eines Netzes 5-lebendig ist, erweisen sich mit $M_r=M_0$ alle Transitionen per definitionem als M_r -lebendig bezüglich aller erreichbaren Markierungen M_f mit $M_f \in RM(M_0)$. Derart lebendige Transitionen sind wiederum per definitionem unsterblich. Daher erfüllt jedes Netz mit 5-lebendiger Ausgangsmarkierung M_0 die o.a. Definition der starken Netzlebendigkeit.

Der starken Lebendigkeit entspricht auch der Adäquatheitsbegriff von LAUER (1978a), S. 311; SHIELDS (1979a), S. 249ff.; SHIELDS (1979b), S. 3 u. 16; LAUER (1979), S. 113; LAUER (1980), S. 456 u. 477; BEST,E. (1980c), S. 292 und - insbesondere - S. 298; BEST,E. (1982d), S. 149 u. 155f.

66) Da auch die Ausgangsmarkierung M_0 eine erreichbare Markierung darstellt, gibt es a fortiori auch keine M_0 -blockierten Transitionen. Folglich zeichnen sich stark lebendige Netze ebenso wie markierungsschwach lebendige Netze dadurch aus, daß sie keine toten Transitionen besitzen können. Stark lebendige Netze unterscheiden sich aber von markierungsschwach lebendigen Netzen durch ihre weitergehende Blockierungsfreiheit. Denn in markierungsschwach lebendigen Netzen können lediglich keine M_0 -blockierten Transitionen existieren. Das schließt keineswegs aus, daß eine Transition zwar zunächst ein- oder mehrmals geschaltet wird, aber schließlich eine Markierung M_r erreicht wird, unter der diese Transition M_r -blockiert ist. Solche M_r -blockierten Transitionen sind trotz markierungsschwacher Netzlebendigkeit für alle erreichbaren Markierungen M_f mit $M_f \in (RM(M_0) - \{M_0\})$ möglich. Erst in einem stark lebendigen Netz ist sichergestellt, daß auch keine M_r -blockierten Transitionen existieren können.

In einer früheren Anmerkung wurden totale und partielle Deadlocks als markierungs- bzw. prozeßbezogene Deadlockdefinitionen eingeführt. Ein partieller Deadlock liegt in einem Netz unter einer Markierung M_r vor, wenn alle Transitionen aus einer Transitionenteilmenge, die einen Teilprozeß charakterisiert, niemals mehr geschaltet werden können. Diese Transitionen sind dann M_r -blockiert. Da in einem stark lebendigen Netz keine M_r -blockierten Transitionen existieren können, sind in einem solchen Netz keine partiellen Deadlocks möglich. Nun ist ein stark leben-

diges Netz auf jeden Fall auch transitionsschwach lebendig. Für transitionsschwach lebendige Netze wurde bereits aufgezeigt, daß es in ihnen per definitionem keine totalen Deadlocks geben kann. Daraus folgt: Wenn ein Netz stark lebendig ist, dann sind alle totalen und auch alle partiellen Deadlocks ausgeschlossen. Es gilt ebenso die Umkehrung. Denn erst die starke Netzlebendigkeit garantiert die Nichtexistenz von M_r -blockierten Transitionen und somit auch die Nichtexistenz aller partiellen Deadlocks. Folglich läßt sich die starke Netzlebendigkeit in bezug auf Deadlocks durch folgende Äquivalenz charakterisieren: Ein Netz ist genau dann stark lebendig, wenn in ihm weder totale noch partielle Deadlocks vorkommen können. Vgl. zu dieser Äquivalenzfeststellung ABEL, D. (1990), S. 15 u. 44 (ohne die Präzisierung "stark").

67) Vgl. VALETTE (1977a), S. 67; MURATA, T. (1977c), S. 3; MEMMI (1978a), S. 125 i.V.m. S. 128; LAUER (1979), S. 113; SHIELDS (1979a), S. 254; AYACHE (1979a), S. 1052; SHIELDS (1979b), S. 3 u. 16; DEVY (1979), S. 50; LAUER (1980), S. 456; BEST, E. (1982d), S. 156.

68) Die Transitionen können nicht bezüglich der Ausgangsmarkierung M_0 blockiert sein. Dies widerspräche der Definition der markierungsschwachen Netzlebendigkeit. Daher dürfen die M_r -blockierten Transitionen sich nur auf Markierungen M_r mit $M_r \in (RM(M_0) - \{M_0\})$ beziehen.

69) Vgl. BAUMGARTEN (1978), S. 9, mit einem anschaulichen Beispiel.

70) Ein ähnlicher Ansatz findet sich bei BAUMGARTEN (1978), S. 32f., der sich allerdings auf anders definierte Prädikate stützt.

71) Am Rande sei auch auf die epistemische Befruchtungsmöglichkeit verwiesen, die von dieser graphisch visualisierten Wissensrepräsentation ausgeht. Denn das Definitionsnetz aus Abb. 57 stellt nur das Endprodukt eines längeren Reifungsprozesses dar, in dem die präsentierten Netzeigenschaften entwickelt und zueinander in Beziehung gesetzt wurden. Dabei erwies sich die Netzvisualisierung besonders hilfreich bei dem Bemühen, einen kohärenten und in sich abgeschlossenen Eigenschaftskatalog zu entwerfen. Es steht hier nicht der Raum zur Verfügung, die Vielfalt der Anregungen zu dokumentieren, die von der vernetzten Repräsentation der Netzeigenschaften ausgingen. Sie wird nur anhand zweier Beispiele verdeutlicht.

Erstens wurde von dem Gestaltungsansatz ausgegangen, daß sich die Kohärenz eines Katalogs von Prädikaten, die dynamische Netzeigenschaften definieren, in einem Definitionsnetz, das den logischen Zusammenhang der Prädikatsdefinitionen repräsentiert, durch eine entsprechende Netzkohärenz niederschlagen muß. Der Kohärenzbegriff war dabei aber inhaltlich noch unbestimmt (vgl. dazu die Kritik an der Vagheit des allgemeinen Kohärenzbegriffs). Er wurde dadurch präzisiert, daß ein Definitionsnetz genau dann als kohärent gilt, wenn seine graphische Repräsentation einen zusammenhängenden Graphen bildet. In einem solchen Graphen sind isolierte Knoten und unzusammenhängende Teilgraphen ausgeschlossen. Isolierte Knoten würden Eigenschaftsdefinitionen entsprechen, die sich aus anderen Definitionen nicht ableiten lassen. Unzusammenhängende Teilgraphen würden mit Definitionsgruppen korrespondieren, die untereinander nicht zusammenhängen. Beides widerspräche dem intuitiven Kohärenzverständnis, das an früherer Stelle umrissen wurde.

Der Verf. wurde angeregt, nach einem Definitionskonzept Ausschau zu halten, das die derart inhaltlich konkretisierte und graphentheoretisch präzisierte Netzkohärenz gewährleistet. Er fand es in der folgenden netzbasierten, hierarchisch strukturierten Systematik für den Definitionszusammenhang: Alle eigenschaftsdefinierenden Prädikate werden aus der originären Definition des Aktivierungsprädikats $AKT(t_p, M_r)$ abgeleitet. Jede Stelle des Definitionsnetzes wird mit einem derart originär oder derivativ definierten Prädikat beschriftet. Jeder Ableitungsschritt wird durch eine Transition repräsentiert. In die Ableitung eines neuen Prädikats dürfen keine anderen als die bereits definierten Prädikate einfließen. Dies bedeutet, daß im Definitionsnetz jede ableitungsrepräsentierende Transition mit mindestens einer Eingangsstelle verknüpft sein muß, die mit einem bereits definierten Prädikat beschriftet ist. Die Neuartigkeit eines abgeleiteten Prädikats wird durch zusätzliche Definitionsbestandteile konstituiert, die in den unmittelbar vorangehenden Prädikaten noch nicht enthalten sind. Die Transition, die einen solchen Ableitungsschritt repräsentiert, wird mit diesen Definitionserweiterungen beschriftet. Jede Stelle, die ein derivativ definiertes Prädikat definiert, ist per constructionem über eine gleichsinnig gerichtete Folge von Kanten, Transitionen und - gegebenenfalls - Stellen mit der einen Stelle des originären Aktivierungsprädikats verbunden. Dadurch wird die Kohärenz des gesamten Definitionsnetzes sichergestellt. Es besitzt einen zusammenhängenden Graphen mit genau einem stellenartigen Knoten ohne Eingangskante (Wurzel), der das originäre Aktivierungsprädikat $AKT(t_p, M_r)$ repräsentiert. Der Graph des Definitionsnetzes wird in der Richtung der Ableitungsschritte durch mehrere stellenartige Knoten ohne Ausgangskanten (Blätter) begrenzt. Sie stellen jeweils eigenschaftsdefinierende Prädikate dar, aus denen keine weiteren Prädikate mehr abgeleitet worden sind. Jeder transitionsartige Knoten besitzt mindestens eine Eingangskante für diejenigen Prädikate, die in die Definition eines neuen Prädikats eingeflossen sind, und mindestens eine Ausgangskante für das jeweils abgeleitete Prädikat. Es werden mehrere Ausgangskanten für den Fall zugelassen, daß äquivalente Prädikate für die Definition derselben Netzeigenschaft abgeleitet werden. Dies betrifft die äquivalenten Prädikate für die Deadlockfreiheit und die transitionsschwache Netzlebendigkeit. Ebenso sind mehrere Eingangskanten für einen transitionsartigen Knoten möglich. Dieser Fall kann auf zwei Ursachen beruhen. Entweder fließen in die Definition eines neuen Prädikats mehrere bereits definierte Prädikate ein. Dann gelten die betroffenen Eingangskanten als konjunktiv verknüpft. Dieser Fall liegt in dem hier vorgestellten Definitionsnetz nicht vor. Oder das

neue Prädikat läßt sich auf alternative Weisen aus jeweils einem alten Prädikat herleiten. Entsprechend werden die Eingangskanten des transitionsartigen Knotens als disjunktiv verknüpft betrachtet. Dies trifft auf die Definition von Deadlockfreiheit und transitionsschwacher Netzlebendigkeit zu. Sie kann sowohl nur aus dem Deadlock- als auch nur aus dem markierungsbezogenen Lebendigkeitsprädikat abgeleitet werden. Die zugehörige disjunktive Kantenverknüpfung wird in visualisierten Graphen des Definitionsnetzes in der gleichen Weise verdeutlicht, die bereits für Kanal/Instanz-Netze eingeführt wurde.

Durch diese Konstruktion eines kohärenten Definitionsnetzes wurde vermieden, eine Vielzahl inkohärenter Definitionsansätze einfließen zu lassen, die in der Netzliteratur weit verbreitet sind. Beispielsweise wird dort der gleiche Sachverhalt - die Erreichbarkeit einer Folgemarkierung durch eine endliche Schaltfolge - durch zwei unterschiedliche Notationen erfaßt. Dabei handelt es sich einerseits um die Bezugnahme auf Erreichbarkeitsmengen ($M_f \in RM(M_f)$), die auch vom Verf. benutzt wurde, und andererseits um die Verwendung von Schaltfolgen aus dem freien Monoid der Transitionenmenge. Das Hin- und Herspringen zwischen den beiden Notationsvarianten beeinträchtigt zunächst die Transparenz - insbesondere die Vergleichsfreundlichkeit - der verschiedenen Definitionsansätze. Darüber hinaus verhalten sich die Notationsalternativen noch nicht einmal äquivalent, sobald Höhere Netze betrachtet werden, die auch Schaltfarben für das Schalten von Transitionen zulassen. Dies kann durch die erstgenannte Notation berücksichtigt werden, nicht aber durch die zweite. (Daher hat der Verf. die erste bevorzugt.) Wegen dieser Schwierigkeiten sollte in einem kohärenten Definitionskonzept nur eine der beiden Notationsvarianten verwendet werden. Die Beschriftung der Netztransitionen und die kompakte Netzvisualisierung gestatten es, die Einhaltung dieser Kohärenznorm im Definitionsnetz rasch zu überprüfen. Auf diese Weise wurde der Verf. zu mehrfachen Überarbeitungen seiner Eigenschaftsdefinitionen angeregt. Er hatte sie zunächst an die vorherrschenden Definitionsansätze der Netzliteratur angelehnt, woraus mehrere Verstöße gegen die zuvor erwähnte Kohärenznorm resultierten. Mit der Konstruktion des Definitionsnetzes ließen sich diese Inkohärenzen aber sofort aufdecken. Die Einhaltung der o.a. Ableitungssystematik des Definitionsnetzes sorgte dann für ein in sich kohärentes System von Eigenschaftsdefinitionen.

Zweitens zeigte sich im Definitionsnetz, daß eine vorläufige Entwicklungsstufe des Eigenschaftskatalogs nicht in sich abgeschlossen war. Es enthielt eine Stelle für das Prädikat $TOT(t_n)$ mit leerem Nachbereich. Dies irritierte, weil sich das Definitionsnetz im übrigen durch eine klare Struktur des logischen Zusammenhangs aller eigenschaftsdefinierenden Prädikate auszeichnete: Der netzrepräsentierende Graph endete nur in solchen Blättern, deren stellenartige Knoten Prädikate für *globale* Eigenschaften repräsentieren, die einem Netz SN als Ganzheit zukommen. Davon wich der eine stellenartige Knoten ab, der das Prädikat $TOT(t_n)$ im Graphen des Definitionsnetzes als ein Blatt repräsentierte. Dieses Prädikat besaß nur *lokale* Gültigkeit für eine einzelne Transition. Das visualisierte Definitionsnetz ließ diese Diskrepanz in seiner topologischen, graphisch repräsentierten Struktur sofort ins Auge fallen. Sie veranlaßte den Verf., durch einen Generalisierungsschritt das Prädikat $TOT(t_n)$ auch für die Gesamtheit aller Transitionen eines Netzes zu definieren. Daraus resultierte eine neue Ableitung für das globale Prädikat $TOT(SN)$. Das derart modifizierte Definitionsnetz beseitigte die oben angesprochene Irritation, da nunmehr alle Blätter seines Graphen Prädikate für globale Netzeigenschaften repräsentierten. Durch die Ergänzung des generalisierenden Ableitungsschritts konnte das Definitionsnetz in sich abgeschlossen werden.

Als interessant erweist sich auch die "Reflexivität" oder "Selbstbezüglichkeit" des Petrinetz-Konzepts, die sich in den voranstehenden Erläuterungen offenbart: Die Ausdruckskraft dieses Konzepts ist so groß, daß sich Erkenntnisse über den logischen Zusammenhang von Netzeigenschaften wiederum mit der Hilfe eines Netzes darstellen - und sogar systematisieren - lassen. Dadurch wird das Netzkonzept nicht nur auf der Objektebene für die Repräsentation von Wissen über Modellierungsobjekte eingesetzt. Vielmehr wird das Netzkonzept auf der Metaebene auch auf sich selbst angewendet. Dabei werden in metasprachlicher Weise Beziehungen zwischen objektsprachlich definierten Netzeigenschaften ausgedrückt. Die Metasprache ist eine Formalsprache mit wiederum netzartigem Charakter.

72) Vgl. zu weiteren Eigenschaftsdefinitionen, die zum Bedeutungsumfeld von Deadlockfreiheit und Lebendigkeit gehören, sowie zu Definitionskomplexen, die solche alternativen Eigenschaftsbestimmungen enthalten, KELLER, R. (1972a), S. 38; LAUTENBACH (1973), S. 14; BYRN (1974), S. II-16ff.; MASSACHUSETTS COMPUTER ASSOCIATES (1974b), S. 158; LAUTENBACH (1975a), S. 4; SCHESCHONK (1977), S. 39; SCHIFFERS (1977), S. 31; BAUMERS (1977), S. 13; BAUMGARTEN (1978), S. 31ff.; YU, S. (1978), S. 176; SCHUMACHER (1978), S. 7; GODBERSEN (1979), S. 251 u. 265; MEMMI (1979), S. 95; FERSTL (1979), S. 110; SURAJ (1980), S. 629; PETERSON, J. (1981), S. 86; VALK (1983b), S. 3, 15 u. 40; VALK (1983c), S. 343, 353 u. 357; REISIG (1986a), S. 85.

In den vorgenannten Quellen werden u.a. auch die 2- und 3-Lebendigkeiten behandelt, die zwischen den transitionsbezogenen 1- und 4-Lebendigkeiten liegen. 2- und 3-Lebendigkeit erstrecken sich auf untere Schranken für die Schaltanzahlen von Transitionen bzw. auf unendlich oft wiederholtes Schalten von Transitionen. Beide Aspekte interessieren in dieser Arbeit aber nicht.

In den o.a. Quellen werden ebenfalls die 3- und 4-Lebendigkeiten vorgestellt, die zwischen jenen markierungsbezogenen 1-, 2- und 5-Lebendigkeiten fehlen, die bereits in früheren Anmerkungen angesprochen wurden. Diese 3- und 4-Lebendigkeiten betreffen die Reproduzierbarkeit von Markierungen, die in dieser Arbeit überhaupt keine Berücksichtigung erfährt.

Einfache, aber infolgedessen besonders übersichtliche Netzbeispiele für die unterschiedlichen Lebendigkeitsvarianten finden sich bei BYRN (1974), S. II-17ff.; MASSACHUSETTS COMPUTER ASSOCIATES (1974b), S. 159; SCHE-

SCHONK (1977), S. 38ff.; MURATA,TA. (1977c), S. 3; BAUMGARTEN (1978), S. 9; EGGERT (1978), S. 53; LAUER (1978a), S. 319; MEMMI (1979), S. 96; PETERSON,J. (1981), S. 87.

Die logischen Implikations- und Äquivalenzbeziehungen, die zwischen den Lebendigkeitsvarianten im allgemeinen oder in speziellen Netzklassen herrschen, werden behandelt von KELLER,R. (1972a), S. 35 u. 39ff.; LAUTENBACH (1973), passim, z.B. S. 94, 99, 105f. u. 120; MASSACHUSETTS COMPUTER ASSOCIATES (1974b), S. 158f.; LAUTENBACH (1975a), S. 4 u. 13; BAUMGARTEN (1978), S. 32ff.; LAUER (1978a), S. 319; MEMMI (1979), S. 95.

73) Die herausragende Bedeutung, die der Deadlockfreiheit und (den Varianten) der Lebendigkeit für die Gestaltung oder Beurteilung von dynamischen Systemen zukommt, wurde schon zu Beginn dieses Kapitels erwähnt. Vgl. dazu vor allem die zahlreichen Quellen aus der zugehörigen Anmerkung.

74) Das wird besonders deutlich, wenn lebendige Netze benutzt werden, um Systeme mit einer zyklischen Verhaltensweise zu modellieren, die sich unbegrenzt fortsetzen läßt. Vgl. zur Verknüpfung der Netzlebendigkeit mit zyklischem Systemverhalten z.B. HUEN (1975), S. 60.

75) Beispielsweise kritisieren LOCKEMANN (1978), S. 78; VALETTE (1978b), S. 189; BRETSCHNEIDER (1980a), S. 40; REISIG (1979d), S. 53, sowie ABEL,D. (1990), S. 44, 75 u. 109, Deadlocks, totale Verklemmungen und fehlende Netzlebendigkeit als Funktions-, Entwurfs- oder Darstellungsfehler von dynamischen Systemen. HOLT,R. (1971a), S. 8, führt Deadlocks auf mangelhafte Prozeßkoordinierungen in Systemen zurück. SIFAKIS (1978), S. 483, sieht Netze als "pathologisch" an, wenn sie unter keiner zulässigen Ausgangsmarkierung lebendig sind.

SHOSHANI (1970), S. 355; HOLT,R. (1971a), S. 8, 25, 29 u. 103; COFFMAN (1971), S. 72ff.; COFFMAN (1973), S. 46; ZIMA (1976), S. 130; LAUER (1978a), S. 311; empfehlen das Vermeiden, Entdecken und Beseitigen von Deadlocks. Ebenso wird es gefordert, Systeme so zu modellieren, daß ihre Netzmodelle deadlockfrei sind; vgl. RAUBOLD (1972), S. 207; HOLT,A. (1975d), S. 158; LOCKEMANN (1975), S. 5; MOALLA (1977), S. 90; BILLINGTON (1981a), S. 18f.; ABEL,D. (1990), S. 43 (Verklemmungsfreiheit als Korrektheitsnachweis). In analoger Weise wird die Gestaltung lebendiger Systemmodelle empfohlen; vgl. z.B. AGERWALA (1978a), S. 152.

76) Dabei wird vorausgesetzt, daß es sich um die Inanspruchnahme einer knappen Ressource handelt. Daher wäre es widersinnig, alternative und nutzenstiftende Einsatzmöglichkeiten des Ressourceneinsatzes dadurch zu verhindern, daß eine funktionslose Ressourcenbindung für die Darstellung einer toten Transition erfolgt.

77) Vgl. NIETERS (1978), S. 32; GRABOWSKI,J. (1980c), S. 19; PETERSON,J. (1981), S. 90; KRÄMER (1981), S. 470 (Eliminierung toter Transitionen); ABEL,D. (1990), S. 44.

78) Vgl. FREEDMAN (1988b), S. 334.

79) Es wird aber später ein einfaches 1-beschränktes Zählernetz vorgelegt werden. Seine Erreichbarkeitsanalyse zeigt, daß es dennoch ein unendliches, nicht-zyklisches Schaltverhalten mit unendlichem Erreichbarkeitsgraphen besitzt. Solche "exotischen" Netze kommen jedoch bei der praktischen Anwendung von Netzmodellen kaum vor. Daher werden sie hier nicht zum oben erwähnten "Regelfall" gerechnet.

80) Ein Schaltprozeß, der in einem beschränkten Netz beliebig fortgesetzt werden kann, korrespondiert im zugehörigen endlichen Erreichbarkeitsgraphen mit einem zyklischen Schaltweg.

81) Leider kann den Quellen nicht immer klar entnommen werden, welches Lebendigkeitsverständnis sie verfolgen. Sofern sie die transitionsschwache Netzlebendigkeit meinen, kann auf ihre redundante Angabe verzichtet werden, weil sie mit der Deadlockfreiheit äquivalent ist. Falls dagegen eine der anderen, stärkeren Lebendigkeitsvarianten zugrundeliegt, so geschieht eine echte Eingrenzung der Klassen wohlgeformter oder adäquater Netze. Beispielsweise erlaubt der Bezug auf markierungsschwach lebendige Netze den Ausschluß toter Transitionen. Die Unterstellung des starken Lebendigkeitsbegriffs würde sogar die Wohlgeformtheit bzw. Adäquanz eines Netzes an die Unsterblichkeit aller seiner Transitionen knüpfen.

82) Vgl. AUGIN (1978a), S. 864; PRIESE (1979), S. 12; DÖPP (1983a), S. 3. Oftmals wird der Begriff der Wohlgeformtheit auch enger definiert, indem zusätzlich zur Beschränktheit, Lebendigkeit und Deadlockfreiheit weitere Netzeigenschaften gefordert werden. Dazu gehören vor allem die nachstehend erörterten Eigenschaften der Finalität und Reversibilität. Vgl. zu solchen engeren Auffassungen der Netzwohlgeformtheit PETERSON,J. (1973a), S. 141ff.; MERLIN,P. (1976a), S. 618; HERZOG,O. (1976b), S. 3; LIEN (1976a), S. 354f. (allerdings in bezug auf Transitionssysteme); VALETTE (1977b), S. 105; VALETTE (1978b), S. 189f.; AGERWALA (1978a), S. 152; BOUSSIN (1978), S. 1529; BARZILAI (1978b), S. 10; VALETTE (1979a), S. 38; AGERWALA (1979), S. 93; YOELI (1980a), S. 73f.; MONTEL (1983a), S. 295; ABEL,D. (1990), S. 56f.

83) Vgl. COTRONIS (1977), S. 197; LAUER (1978a), S. 301 u. 311.

84) Vgl. SILVA,M. (1981b), S. 336.

85) Ähnlich werden in FREEDMAN (1988b), S. 334, nur solche Netze als "sinnvoll" (meaningful) zugelassen, die zugleich sicher - also 1-beschränkt - und deadlockfrei sind.

86) Anschauliche Beispiele dafür bietet ABEL, D. (1990), S. 75f. Die anschließenden Erläuterungen lehnen sich an die Fälle a) und b) von ABEL an.

87) Ein anders gelagertes Interesse an nicht-lebendigen Netzen bekundet ABEL, D. (1990), S. 44. Er möchte zuweilen Netze mit partiellen Verklemmungen zulassen. Solche partiellen Verklemmungen bedeuten die Existenz einer nicht-leeren Menge von M_T -blockierten Transitionen. Daher verletzen diese Netze die Anforderung der starken Netzlebendigkeit. Allerdings kann es sich weiterhin um deadlockfreie Netze handeln, sofern der Deadlockbegriff der hier definierten Deadlockmarkierungen (totale Deadlocks) zugrundeliegt.

88) Die Möglichkeit, daß Deadlocks durchaus erwünscht sein können, sprechen ebenfalls an: BYRN (1974), S. III-29 u. III-42; PAGNONI (1990), S. 143.

89) Diese Voraussetzungen werden dagegen verletzt von unkonventionellen Aufgaben der Automatischen Informationsverarbeitung, die ohne definierten Abschluß permanent erfüllt werden sollen. Dazu gehören z.B. die Überwachung von technischen Systemen unter Realzeitbedingungen oder die Transaktionsverarbeitung in offenen Informationssystemen (etwa Reservierungssystemen).

90) Vgl. KELLER, R. (1976), S. 379.

91) KIEBLER (o.J.), S. 28, reserviert den Deadlockbegriff für solche unerwünschten Deadlocks. Bei erwünschten Deadlocks spricht er dagegen von Haltezuständen. Vgl. zur Einschränkung des Deadlockbegriffs auf unerwünschte Deadlocks auch ZELEWSKI (1986a), S. 242 u. 284.

92) Dies umschließt auch jene endlichen Prozesse, deren Ausführung zwar erwünscht ist, die aber zum Eintritt eines Deadlocks überhaupt noch nicht begonnen wurden.

93) Beispielsweise läßt sich das Stelle/Transition-Netz aus Abb. 16 als Modell eines Planungsproblems für die Belegung von zwei Maschinen interpretieren. Auf diesen soll aus drei Vorprodukten, von denen jeweils vier Einheiten in Eingangslagern - modelliert als je vier Marken auf den Stellen s_1 , s_2 und s_3 - vorgehalten werden, sowie aus zwei Zwischenprodukten mindestens eine Einheit eines Endprodukts gefertigt werden. Das Vorliegen einer solchen Endprodukteinheit würde durch eine Marke auf der Stelle s_{10} angezeigt werden.

Aus dem Erreichbarkeitsgraphen der Abb. 17 für dieses Netz läßt sich erkennen, daß dieses Planungsproblem nicht gelöst werden kann. Es existiert kein einziger Prozeß, der zu einer erwünschten Endmarkierung $M_{E,e}$ mit $M_{E,e}(s_{10}) \geq 1$ führt, weil es keine solche Markierung im Erreichbarkeitsgraphen gibt. Statt dessen enden alle möglichen Prozesse in einem unerwünschten Endzustand des Fertigungssystems, in dem Vorprodukte teilweise verbraucht und Zwischenprodukte teilweise hergestellt vorliegen, aber keine Endprodukteinheit gefertigt wurde. Es handelt sich hierbei um die Deadlock-Markierung M_{11} , in der alle Prozesse, die vom Ausgangszustand starten, einmünden. Diese Markierung repräsentiert einen typischen unerwünschten Deadlock.

94) Vgl. zu dieser (teil)prozeßbezogenen Thematisierung von Deadlocks die Quellen, die in einer Anmerkung zu Beginn dieses Kapitels angeführt wurden. Besonders anschaulich erfolgt die Betrachtung von Teilprozessen z.B. bei SCHROFF (1974), S. 1ff.

Die Teilprozesse eines Netzes sind deutlich von dem Schaltprozeß desselben Netzes zu unterscheiden. Denn ein Schaltprozeß $PRO_r(SF_L, MF_L)$ definiert für jede Markierung $M_{a(i)}$ aus seiner Markierungsfolge MF_L genau einen Schaltschritt $SS_{a(i)}$ mit $i \in \{1, \dots, L\}$, der jeweils *alle* Transitionen umfaßt, die unter der unmittelbar vorangehenden Markierung $M_{a(i)}$ des Netzes aktiviert sind *und* auch tatsächlich geschaltet werden. In einem Netz kann daher immer nur *ein* Schaltprozeß ablaufen. Es sind zwar oftmals viele unterschiedliche Schaltprozesse zulässig, aber für die Realisierung eines Netzverhaltens muß daraus immer genau ein Schaltprozeß ausgewählt werden.

Ein Teilprozeß $TPRO_p$ ist dagegen auf eine nicht-leere Teilmenge der Transitionenmenge eines Netzes beschränkt. Daher ist es durchaus möglich, für ein Netz *mehrere* Teilprozesse durch Auszeichnung von (teil)prozeßspezifischen Transitionenteilmengen zu definieren. Dabei wird vorausgesetzt, daß die prozeßspezifischen Transitionenteilmengen disjunkt sind. Jeder Teilprozeß wird als eine Schaltfolge $SF_p = (SS_{a(1)}, \dots, SS_{a(L_p)})$ definiert, in der nur jene Transitionen geschaltet werden können, die in den prozeßspezifischen Transitionenteilmengen enthalten sind. Diese teilprozeßspezifische Schaltfolge braucht keineswegs sequentiellen Charakter zu besitzen. Statt dessen kann sie - ebenso wie die Schaltfolge eines Schaltprozesses - auch eine nebenläufige Schaltfolge darstellen. Allerdings wird ein Teilprozeß *nur* durch seine Schaltfolge konstituiert: $TPRO_p = SF_p = (SS_{a(1)}, \dots, SS_{a(L_p)})$. Im Gegensatz zum Schaltprozeß ist es nicht möglich, eine prozeßspezifische Markierungsfolge mit der Schaltfolge alternierend zu verschränken. Denn jede Markierungsfolge besteht aus Netzmarkierungen, die jeweils für ein *gesamtes* Netz definiert sind. Teilprozesse wirken sich dagegen nur auf die Markierungen derjenigen Stellen aus, die zur Nachbarschaft der prozeßspezifischen Transitionenteilmengen gehören. Diese prozeßbetroffenen Stellen überdecken aber im allgemeinen nicht die gesamte Stellenmenge eines Netzes. Daher korrespondiert mit der Schaltfolge eines Teilprozesses in der Regel keine wohldefinierte Markierungsfolge des Netzes. Im selben Netz können mehrere Teilprozesse nebenläufig ausgeführt werden. Der eine Schaltprozeß des Netzes spezifiziert dann, wie diese nebenläufige Ausführung mehrerer Teilprozesse konkret erfolgt. Die Transitionen, die in einem Schaltschritt des Schaltprozesses enthalten sind, gehören jeweils zu genau einem Teilprozeß. Dies garantiert die vorausgesetzte Disjunktheit der prozeßspezifischen Transi-

tionenteilmengen. Daher werden in jedem Schaltschritt aus dem Schaltprozeß des Netzes genau jene Teilprozesse begonnen oder fortgesetzt, deren prozeßspezifische Transitionen im Schaltschritt enthalten sind.

Teilprozesse eignen sich, um im Netzmodell eines realen Modellierungsobjekts mehrere Aktivitäten abzubilden, die im modellierten Objekt stattfinden können. Um ihre Verschiedenartigkeit gegenüber dem einen Schaltprozeß des Netzmodells zu betonen, können die Teilprozesse auch als Realprozesse angesprochen werden. Nur in dem Grenzfall, daß im modellierten realen Objekt nur eine Aktivität möglich ist, fällt der korrespondierende Realprozeß mit dem Schaltprozeß des Netzmodells zusammen.

95) Ein Ressourcenverbrauch geschieht, falls die schaltende Transition mindestens eine Marke von einer ihrer Eingangsstellen abzieht, sie aber auf keiner ihrer Ausgangsstellen ablegt. Ebenso liegt ein Ressourcenverbrauch vor, wenn die Transition mindestens eine Marke auf einer ihrer Ausgangsstellen ablegt und die betroffene Ausgangsstelle eine beschränkte Markenkapazität besitzt. Dann wird die Ressource "freie Markenkapazität" verbraucht.

96) Eine Ressourcenfreigabe findet dann statt, wenn die schaltende Transition mindestens eine Marke von einer Eingangsstelle abzieht, sie aber auch wieder auf einer Ausgangsstelle ablegt.

97) Weder ein Ressourcenverbrauch noch eine Ressourcenfreigabe erfolgt dagegen dann, wenn die schaltende Transition zwar zur Erfüllung ihrer Aktivierungsbedingung auf eine Ressource zugreift, sie aber nicht bindet. Dies kann für alle Marken zutreffen, die sich auf einer Informationsstelle der schaltenden Transition befinden. Die Transition muß zwar auf diese Marken in dem Ausmaß zugreifen, wie es zur Erfüllung ihrer Aktivierungsbedingung notwendig ist. Aber dabei werden die Marken nicht gebunden, sondern nur hinsichtlich ihres Informationsgehalts zur Kenntnis genommen.

Darüber hinaus ist es sogar möglich, daß eine schaltende Transition neue Ressourcen schafft. Dies ist immer dann der Fall, wenn sie mindestens eine Marke auf einer Ausgangsstelle ablegt, die nicht zuvor von einer Eingangsstelle abgezogen wurde. Hiermit wird die Persistenzprämisse für die Modellierung realer Objekte nicht verletzt. Denn sie erfordert nur, daß das formale Abbild eines realen Objekts im Modell fortbestehen soll, solange das zugrundeliegende Objekt in der Realität weiter existiert. Das spricht nicht dagegen, daß neue Ressourcen entstehen, indem in den modellierten Realitätsausschnitt neue reale Objekte eintreten. Darüber hinaus kann auch die unstrukturierte Basismarke benutzt werden, um Ressourcen abzubilden. Sie läßt sich durch das Schalten einer Transition beliebig vermehren. Die Persistenzprämisse ist hiervon nicht betroffen, weil sie sich nur auf strukturierte Marken bezieht.

Beide voranstehend angesprochenen Aspekte des bindungsfreien Ressourcenzugriffs und der Ressourcenerzeugung werden bei der Betrachtung von Deadlocks nicht weiter berücksichtigt. Sie spielen dort keine Rolle.

98) Dies entspricht der Prozeßblockierung bei HOLT,R. (1972b), S. 64; ZIMA (1976), S. 127 u. 150. Der Blockierungsbegriff wird in dieser Arbeit in einem engeren Verständnis benutzt, das nachfolgend definiert wird. Die o.a. Prozeßunterbrechung korrespondiert auch mit dem Konzept der Interlocks, die von HAVENDER (1968), S. 75, beschrieben werden.

99) Das Ausführungsende eines endlichen Teilprozesses läßt sich durch das Markieren einer prozeßspezifischen Stelle definieren, die solange unmarkiert bleibt, wie der Prozeß nicht erfolgreich beendet worden ist. Vgl. dazu die Konstruktion finaler Stellen.

100) Es kann sich um mehrere Transitionen handeln, die in einem Schaltschritt des Teilprozesses nebenläufig zu schalten sind.

101) Es werden hier nur unerwünschte Deadlocks betrachtet. Bei ebenso unwillkommenen Livelocks würden dagegen zwar genügend Ressourcen bereitstehen, aber jeweils durch einen anderen Teilprozeß gebunden. Da dieser Fall des Prozeßverhungerns bereits für Livelocks behandelt wurde, findet er hier keine weitere Berücksichtigung.

102) Von HOLT,R. (1972b), S. 64f., und ZIMA (1976), S. 127 u. 150, wird in diesem Fall von einem Deadlock bzw. einer Prozeßverklemmung gesprochen. Beide entsprechen den partiellen Deadlocks, die in einer früheren Anmerkung eingeführt wurden.

103) In Ausnahmefällen kann ein unerwünschter Deadlock auch andere Ursachen besitzen als die nachfolgend erläuterte. Dazu gehört beispielsweise der extreme Fall, daß in einem Netz noch nicht einmal diejenigen Ressourcen bereitstehen, die notwendig wären, um einen einzelnen Teilprozeß auszuführen. Dieser Sonderfall erlangt jedoch nur selten Bedeutung. Denn Netze werden im allgemeinen so gestaltet, daß genügend Ressourcen für die Ausführung einzelner Teilprozesse zur Verfügung stehen. Vgl. auch zu Deadlockdefinitionen, die sich jeweils nur auf einen Teilprozeß beziehen, HOLT,R. (1971a), S. 1; ZIMA (1976), S. 127 i.V.m. S. 150.

Dagegen beruhen typische unerwünschte Deadlocks auf einem anderen Phänomen. Dabei reichen die Ressourcen für die Ausführung einzelner Teilprozesse zwar aus, aber nicht für die nebenläufige Ausführung aller im Netz überhaupt zulässigen Teilprozesse. Dieser Regelfall wird anschließend näher ausgeführt. Er wurde von HOLT,R. (1971a), S. 1ff.; HOLT,R. (1972b), S. 66ff., durch sein Konzept der Ressourcensysteme präzise charakterisiert. Vgl. vor allem die Blockierungsbedingungen bei HOLT,R. (1971a), S. 49, 125 u. 161; HOLT,R. (1972b), S. 66, die den Aspekt des mehrfachen Zugriffs auf eine gemeinsame und insgesamt knappe Ressource besonders deutlich werden läßt.

Einen *prima facie* allgemeineren Standpunkt bei der Definition unerwünschter Deadlocks nimmt GOSTELOW (1975) ein. Er versteht einen Deadlock als jedes Phänomen, das ein dynamisches System daran hindert, sich "angemessen" (*properly*) zu verhalten. Dabei können Ressourcen, von denen die Prozeßausführung im System abhängt, eine Rolle spielen, müssen es aber keinesfalls. Statt dessen können z.B. auch logische Bedingungen wichtig werden, von deren Erfüllung es abhängt, ob sich die jeweils betrachteten Teilprozesse fortsetzen lassen. Einen besonderen Fall der logisch bedingten Deadlockentstehung führt FERSTL (1979), S. 107f., vor: Er zeigt, daß die Modellierung eines speziellen ökonomischen Gleichungssystems durch ein Petrinetz in einen unerwünschten Deadlock führt.

Der allgemeiner anmutende, nicht auf Ressourcen bezogene Ansatz der Deadlockbetrachtung wird durch die hier beschriebenen ressourcenbedingten Deadlocks mit erfaßt. Denn der Verf. hat sich nicht darauf festgelegt, den Ressourcenbegriff auf reale Objekte zu beschränken. Vielmehr wurden Ressourcen mit Marken und freien Markenkapazitäten in Netzen identifiziert. In Synthetischen Netzen werden aber die Erfüllung und Verletzung von logischen Bedingungen durch die entsprechende Markierung von Stellen erfaßt, die jeweils mit den Namen der bedingungs- ausdrückenden Prädikatssymbole beschriftet sind. Dabei repräsentieren die Stellenmarkierungen jeweils Multimengen von gültigen Prädikatsvorkommnissen (Fakten). Daher lassen sich durch die *abstrakte* Ressource "Marke" auch solche Deadlocks in ressourcenbezogener Weise darstellen, die auf der Prozeßbeeinflussung durch logische Bedingungen beruhen. Allgemein gilt: In Synthetischen Netzen werden alle Deadlocks, die überhaupt durch prädi- katenlogische Ausdrucksmittel beschrieben werden können, mit der Hilfe von Marken erfaßt. Da diese Marken in Netzen die Qualität von Ressourcen besitzen, können jene Deadlocks im Rahmen des Petrinetz-Konzepts immer in ressourcenbezogener Weise konzeptualisiert werden. Daher führt die Beschränkung von unerwünschten Deadlocks auf ressourcenbezogene Deadlocks zu keiner nennenswerten Verengung der Betrachtungsperspektive.

104) Vgl. zur Deadlockverursachung durch wechselseitiges Blockieren mehrerer Teilprozesse, die auf gemeinsam benötigte knappe Ressourcen zugreifen, HAVENDER (1968), S. 74 u. 76f.; HABERMANN (1969), S. 373f.; SHOSHANI (1970), S. 355 u. 357; HOLT,R. (1971b), S. 36; COFFMAN (1971), S. 70; COFFMAN (1973), S. 10 u. 44; SCHROFF (1974), S. 1; ULLRICH (1976), S. 3/16; ZIMA (1976), S. 128; MURATA,TA. (1976a), S. 599; HERZOG,O. (1977), S. 209; BRINCH HANSEN (1977), S. 118; PETERSON,J. (1978a), S. 146; FERSTL (1979), S. 105 u. 109f.; MANNA (1979), S. 395; MÜLLER,B. (1981), S. 201; BAUER,F. (1981), S. 418; HACKMANN (1982), S. 51; ALAGIC (1986), S. 186f. (für Transaktionen als besondere Ausprägungen von Teilprozessen).

105) Es wurde bereits GOSTELOW's Ansicht erörtert, daß die ressourcenbezogenen Deadlocks keineswegs die einzigen interessanten Exemplare unerwünschter Deadlocks darstellten.

106) Es lassen sich allerdings auch Sonderfälle vorstellen, in denen ein wechselseitiges Blockieren von Teilprozessen erwünscht ist. Ein Beispiel dafür aus dem Bereich von Datenbanksystemen findet sich bei MÜLLER,B. (1981), S. 202.

107) Die Bedeutung von ressourcenbezogenen, unerwünschten Deadlocks wird explizit herausgestellt von KELLER,R. (1976), S. 379. Die Wichtigkeit intensiver Erforschung von Deadlocks unterstreicht auch SIFAKIS (1980c), S. 587.

Die intensive Auseinandersetzung mit solchen Deadlocks betrifft auch die bereits angeführte Aufmerksamkeit, die der Untersuchung der Deadlockfreiheit und Lebendigkeit von Netzen zuteil wird. Vgl. dazu die Quellen, die in einer früheren Anmerkung die Forderung nach Deadlockfreiheit und Lebendigkeit belegten. Ihnen liegt zumeist das Studium von ressourcenbezogenen Deadlocks und deren Verhinderung zugrunde. Vgl. darüber hinaus die Quellen, die in den nachfolgenden Anmerkungen aufgeführt werden. Dazu zählen insbesondere auch die Arbeiten, die in einer früheren Anmerkung zu paradigmatischen, jeweils deadlockbelasteten Koordinierungsproblemen angeführt werden.

108) Ein konkretes Beispiel für die produktionswirtschaftliche Relevanz von ressourcenbezogenen Deadlocks wird in einem Exkurs dargelegt, der sich unmittelbar anschließt. Im Interesse der Übersichtlichkeit wird dieses Exempel recht einfach gehalten. Anspruchsvollere Anwendungen für die Deadlockanalyse von Netzmodellen lassen sich aber ebenso vorstellen. Beispielsweise kann daran gedacht werden, bei Flexiblen Fertigungssystemen einzelne Produktionsschichten - z.B. über Nacht oder an Feiertagen - vollautomatisch auszuführen. Die Automatisierung läßt sich so weit treiben, daß selbst die Koordinierungsentscheidungen, die während der "Geisterschicht" erforderlich werden, von einem Automatischen Informationsverarbeitungssystem getroffen werden. Solche Entscheidungen können z.B. erforderlich werden, wenn vom vorbereiteten Produktionsausführungsplan abgewichen werden muß, weil unvorhergesehene Produktionsstörungen eintreten sind. In diesem Fall kommt der Einsatz einfacher Umplanungsheuristiken in Betracht, die auf der Verwendung von Prioritätsregeln beruhen. Der lokale Entscheidungscharakter von Prioritätsregeln hilft zwar, die Koordinierungskomplexität niedrig zu halten. Zugleich birgt er aber auch die Gefahr in sich, daß solche lokalen, prioritätsbasierten Koordinierungsentscheidungen in eine unerwünschte Deadlocksituation hineinführen. An diesem Punkt kann auf die Netzeigenschaft der Deadlockfreiheit zurückgegriffen werden: Das Produktionssystem wird einschließlich der Prioritätsregeln, die zu seiner Koordinierung eingesetzt werden sollen, durch ein Netzmodell repräsentiert. Wenn es gelingt, die Deadlockfreiheit des Netzmodells zu beweisen, lassen sich die Prioritätsregeln zur Umplanung von Produktionsprozessen während einer vollautomatischen Produktionsschicht ohne Bedenken einsetzen. Andernfalls - wenn drohende Deadlocks im Netzmodell erkannt werden - lassen sich die

Umplanungsstrategien so lange modifizieren, bis die erwünschte Deadlockfreiheit des Netzmodells eintritt. Auf welche Weise für die Deadlockfreiheit eines Netzmodells gesorgt werden kann, wird in Kürze dargelegt.

109) Zu den seltenen Ausnahmen gehört der Hinweis von KNOOP (1987), S. 55, daß in seinem Simulationsmodell für Flexible Fertigungssysteme "die Blockierung ... der Stationen vermieden werden" könne. Allerdings läßt KNOOP offen, ob er nur eine vorübergehende Stationsblockierung oder einen dauerhaften Deadlock meint. Selbst wenn eine Bezugnahme auf Deadlocks angenommen wird, so läßt sich kein Ansatz erkennen, wie KNOOP die Deadlockfreiheit seiner Modellierungen zu untersuchen - oder gar zu beweisen - gedenkt. Es scheint sich hier mehr um einen programmatischen Anspruch als um eine überprüfbare Modelleigenschaft zu handeln. Aber immerhin ist bemerkenswert, daß er durch seine Äußerung die Relevanz von Blockierungen für die Modellierung Flexibler Fertigungssysteme anerkennt. Auch bei HELBERG (1987), S. 199, findet sich ein Hinweis auf "Blockierungen". Dort ist jedoch offensichtlich, daß lediglich vorübergehende Blockierungen betrachtet werden. Sie "können dadurch auftreten, daß ein bearbeitetes Werkstück eine Maschine nicht verlassen kann, weil der Ausgangspuffer *noch* durch das vorherige Werkstück belegt ist." (S. 199, kursive Hervorhebung durch den Verf.). Das spätere Freiwerden des Ausgangspuffers wird also grundsätzlich für möglich gehalten. Infolgedessen wird die Gefahr dauerhafter Deadlocks überhaupt nicht gesehen. Vgl. zu weiteren spärlichen Bezugnahmen auf Blockierungen in produktionswirtschaftlichen Modellierungen VILLA (1988c), S. 358; ROSENBERG, O. (1990), S. 13f.

110) Nur an einer Stelle ist dem Verf. bisher in der produktionswirtschaftlichen Literatur - außerhalb von Beiträgen zum Petrinetz-Konzept - die explizite und zweifelsfreie Thematisierung einer Deadlocksituation aufgefallen. Es handelt sich um die Passage bei KOCHAN, D. (1986), S. 140 i.V.m. Abb. 77 auf S. 139. Allerdings wird auch dort kein typisches OR-Programm behandelt. Statt dessen befaßt sich KOCHAN mit der Simulation von Produktionsprozessen in Flexiblen Fertigungssystemen.

Ebenso könnte von KNOOP (1986), S. 166 u. 168, auf einen Deadlock angespielt werden. Allerdings spricht er nur von einer "Systemblockierung". Darüber hinaus hegt der Verf. Zweifel, ob KNOOP tatsächlich einen Deadlock im hier definierten Sinn gemeint hat. Denn aus den dürftigen Erläuterungen der vorgenannten Quelle (vgl. S. 166, Fn. 2) läßt sich der Schluß ziehen, daß die beschriebene System"blockierung" nur vorübergehender Natur ist. KNOOP zufolge läßt sie sich dadurch aufheben, daß Spannvorrichtungen, deren Fehlen zunächst zu einer Systemblockierung geführt hat, nach Fertigstellung von Aufträgen im Flexiblen Fertigungssystem frei werden, zu den Spannstationen zurückgelangen, dort zum Aufspannen der wartenden - "blockierten" - Aufträge benutzt werden und so die angebliche Systemblockierung wieder auflösen. Eine derart temporäre Blockierung verletzt aber die Eigenart von Deadlocks, daß nach ihrem Eintreten *überhaupt kein* Systemverhalten - außer dem vollkommenen Systemstillstand - mehr möglich ist.

111) Einen präzisen Beweis dieser Unmöglichkeitsthese strebt der Verf. in dieser Arbeit nicht an, weil die Analyse des Ausdruckspotentials von OR-Programmen außerhalb ihres Erkenntnisinteresses liegt. Aber die Schwierigkeit, in der einschlägigen OR-Literatur Deadlockuntersuchungen auf dem Fundament von OR-Programmen aufzufinden, liefert immerhin ein starkes Indiz zugunsten der vorgenannten Unmöglichkeitsbehauptung.

112) Weitere Beispiele für ressourcenbedingte unerwünschte Deadlocks in Netzen finden sich bei HOLT, R. (1971a), S. 1ff.; ULLRICH (1976), S. 3/31; COTRONIS (1977), S. 200; FERSTL (1979), S. 105ff.; MÜLLER, B. (1981), S. 201; PETERSON, J. (1981), S. 85f.; ABEL, D. (1986a), S. 62 (mit speziellem Bezug auf Flexible Fertigungssysteme); ABEL, D. (1990), S. 107ff.

113) Ein Analogon zu dem hier betrachteten produktionswirtschaftlichen Beispiel findet sich bei REISIG (1989b), S. 3ff.; ABEL, D. (1990), S. 105ff. Dort wird die Koordinierung der Containerverladung durch Portalkrane modelliert, die zwei parallele Bahnlinien auf einem Umladebahnhof bedienen.

114) Ein- und Ausgangspuffer finden in dem hier gewählten einfachen Beispiel noch keine Berücksichtigung. Diese realitätsferne Simplifizierung wird später im Rahmen der ausführlichen Fallstudie beseitigt. Die knappe Kapazität jeder Fertigungslinie wird im Netzmodell der Abb. 59 durch die jeweils rückwärts gerichtete Teilschleife erfaßt: Solange sich eine Kopie der Basismarke auf der Stelle s_5 (s_{12}) befindet, kann die Fertigungslinie eine Produkteinheit zur Bearbeitung aufnehmen. Sobald dies geschehen ist, verschwindet die Markenkopie so lange, bis die bearbeitete Produkteinheit durch den Transportroboter von der Fertigungslinie entnommen und zum Endprodukt-Lager gebracht wird. Daher repräsentiert die Kopie der Basismarke auf der Stelle s_5 (s_{12}) die jeweils freie Bearbeitungskapazität einer Fertigungslinie.

115) Es handelt sich um die Transportdauern d_3 , d_8 , d_9 und d_{14} sowie um die Bearbeitungsdauern d_6 und d_{11} , mit denen die Stellen s_3 , s_8 , s_9 und s_{14} bzw. s_6 und s_{11} beschriftet sind.

116) Die Berücksichtigung solcher Zeitgrößen ist in den bisher entfalteten Synthetischen Netzen strenggenommen noch nicht möglich. Denn es fehlt noch eine entsprechende Einbindung von temporalen Einflußgrößen in das Schaltverhalten dieser Netze. Insofern wird den späteren Ausführungen zur Erweiterung von Synthetischen Netzen um Zeitaspekte vorgegriffen. Für das Verständnis des hier betrachteten Beispiels reicht jedoch zunächst eine intuitive Annäherung aus: Wenn eine Stelle s_m mit einer Transport- oder Bearbeitungsdauer d_m beschriftet ist, so darf

eine Marke, die dort im Zeitpunkt T abgelegt worden ist, durch eine Ausgangstransition der Stelle s_m frühestens im Zeitpunkt $T+d_m$ abgezogen werden. Solche "Markenbindungsdauern" werden an anderer Stelle durch temporale Markenattribute und korrespondierende zeitbezogene Schaltrestriktionen für Transitionen präzisiert.

117) Die Darstellung des Erreichbarkeitsgraphen wurde so vereinfacht und zugleich ergänzt, daß die wesentlichen Aspekte der hier behandelten Deadlockproblematik plastisch hervortreten. Einerseits wurden die Schaltkanten des Erreichbarkeitsgraphen nicht mit den zugehörigen Schaltschritten, sondern nur mit den Transitionenmengen dieser Schaltschritte beschriftet. Daher wurde auf die Angabe der jeweils betroffenen Schaltfarbe, also der fortgeschalteten Marken, der Übersichtlichkeit halber verzichtet. Andererseits wird der Erreichbarkeitsgraph um die Angabe der Zeitpunkte "T" ergänzt, in denen die Schaltschritte jeweils ausgeführt werden.

Darüber hinaus ist anzumerken, daß der Erreichbarkeitsgraph des Netzmodells weitaus umfangreicher ausfällt, als es die beiden Ausschnitte der Abb. 60 u. 62 vielleicht erscheinen lassen. Denn die gleichen Schaltschritte lassen sich oftmals nicht nur im Zeitpunkt ihrer erstmaligen Aktivierung ausführen. Statt dessen ist es wegen der charakteristischen Permissivität der Schaltregel häufig ebenso möglich, mit dem Ausführen eines aktivierten Schaltschritts bis zu einem späteren Zeitpunkt abzuwarten. Da die Schaltkanten im Erreichbarkeitsgraphen nicht nur schaltschritt-, sondern auch schaltzeitpunktspezifisch sind, führt jede Ausführungsverzögerung eines aktivierten Schaltschritts zu einer neuen Schaltkante. Entsprechend wird der Erreichbarkeitsgraph aufgebläht. Auf diese Verzögerungsmöglichkeiten wurde im hier betrachteten Netzmodell jedoch weitgehend verzichtet. Denn es wurde das Koordinierungsziel zugrundegelegt, die Transformation aller Vor- in Endprodukte mit einer möglichst geringen Durchlaufzeit zu verwirklichen. Allerdings wurde die Option der Ausführungsverzögerung im erweiterten Erreichbarkeitsgraphen der Abb. 62 in vier Fällen beibehalten, in denen sie zu keiner unerwünschten Verlängerung der Durchlaufzeit führt. Statt dessen bewirkt sie beim Ausführen der Schaltschritte, welche die Markierungen M_8 und M_{10} , M_{10} und M_{11} , M_{19} und M_{20} sowie M_{20} und M_{21} ineinander transformieren, zu entsprechenden Pufferzeiten. Diese Pufferzeiten äußern sich im Erreichbarkeitsgraphen darin, daß für die Ausführungen der Schaltschritte keine Zeitpunkte, sondern endliche Zeitintervalle angegeben werden. Im korrespondierenden Ablaufdiagramm der Abb. 64 treten die Pufferzeiten als vor- und rückwärtsgerichtete Verschiebungsmöglichkeiten für die Ausführungen der Bearbeitungen der ersten und der dritten Produkteinheit aus.

118) Vgl. zu dieser Kompositionsstrategie LAUER (1978a), S. 311ff.; KERAMIDIS (1980a), S. 266ff.; STREPPPEL (1982), S. 7.

Für die Kompositionsstrategie spricht einerseits, daß aufwendige Deadlockanalysen und - gegebenenfalls - anschließende Netzmodifizierungen unterbleiben können. Andererseits wird durch die Regeln für die Komposition zusammengesetzter Netze das Ausdruckspotential des Petrinetz-Konzepts erheblich eingeschränkt. Hinzu kommt als weiterer Nachteil, daß solche Kompositionsregeln für garantiert deadlockfreie Höhere Netze bislang noch nicht vorgelegt wurden.

119) Die Kompositionsregeln sind notwendig, weil die Zusammenfassung von deadlockfreien Teilnetzen keineswegs immer zu ebenso deadlockfreien Gesamtnetzen führen muß. Ein Beispiel für den zusammensetzungsinduzierten Verlust der Deadlockfreiheit findet sich bei JANTZEN (1980a), S. 198f. Zwar wird dort explizit nur die Einbuße der markierungsschwachen Netzlebendigkeit thematisiert. Aber für das vorgestellte Beispiel gilt ebenso: Das Gesamtnetz, das aus zwei deadlockfreien Teilnetzen zusammengesetzt wird, besitzt Deadlockmarkierungen. So endet etwa die Schaltfolge aus den Transitionen t_1 , t_2 und t_7 im zusammengesetzten Netz in einer Deadlockmarkierung.

120) Vgl. zu dieser Modifizierungsstrategie ULLRICH (1976), S. 3/31; MÜLLER, B. (1981), S. 201f.; HACKMANN (1982), S. 55f.; ABEL, D. (1986a), S. 61f.; ABEL, D. (1990), S. 107ff., insbesondere S. 108; FEHLING (1990b), S. 4ff. Die vorgenannten Quellen enthalten anschauliche Beispiele für deadlockbeseitigende Netzmodifizierungen.

121) Bislang ist kein Modifizierungsalgorithmus bekannt, der eine vollständige Deadlockbeseitigung sicherzustellen vermag.

122) Beispielsweise reicht bei ABEL, D. (1990), S. 108, die Einführung zweier zusätzlicher Kanten (zwischen der siebten Stelle und der ersten Transition sowie zwischen der achten Stelle und der dritten Transition) aus, um ein deadlockbehaftetes in ein deadlockfreies Netz zu transformieren.

123) Zu diesem Zweck muß die Schaltregel eines Netzes durch entsprechende Koordinierungsregeln ("Synchronisationsregeln") ergänzt werden, die das Erreichen einer Deadlockmarkierung ausschließen. Zu diesen deadlockfreien Koordinierungsstrategien gehört vor allem das Konzept verklemmungsfreier Führung von Schaltprozessen. Vgl. dazu SCHROFF (1974), S. 3ff., insbesondere S. 3, 22 u. 135f., und MAYR, E. (1975), insbesondere S. 13ff. u. 57f.

Einen exotischen Sonderfall stellt die Strategievariante des vorsorglichen Prozeßabbruchs dar. Dabei werden die Ressourcenzugriffe von Teilprozessen dadurch "koordiniert", daß die Prozeßausführungen endgültig abgebrochen werden, falls ihre Ressourcenanforderungen nicht innerhalb strategiespezifischer Wartefristen erfüllt werden können. Daher kann es niemals zu wechselseitigen Blockierungen von Teilprozessen kommen; vgl. HOLT, R. (1971a), S. 8f. (halbstündige Wartefrist); BYRN (1974), S. III-32 (sofortiger Ausführungsabbruch).

Vgl. zu weiteren Ausprägungen der Strategie, in dynamischen Systemen Deadlocks zwar zuzulassen, aber das Systemverhalten am Erreichen solcher Deadlocks zu hindern, COFFMAN (1971), S. 74ff.; COFFMAN (1973), S. 46 u. 53ff.

124) Vgl. zur strategischen Stoßrichtung der Deadlockumgehung BAKER, H. (1972), S. 3; vgl. ebenso die Quellen aus der voranstehenden Anmerkung.

125) Aber auch wenn kein deadlockbehaftetes Netzmodell vorgegeben wäre, sieht der Verf. keinen Ansatz für einen erfolgreichen Einsatz der Kompositionsstrategie. Denn das deadlockfreie Netzmodell, das später präsentiert wird, beruht im wesentlichen auf der Ergänzung von Inhibitoranten. Solche Inhibitoranten sind dem Verf. aus den Beiträgen, die sich mit der deadlockfreien Komposition von Netzmodulen befassen, nicht bekannt. Der Verf. sieht auch keinen anderen Weg, um im Rahmen der Kompositionsstrategie für das hier beleuchtete Koordinierungsbeispiel ein deadlockfreies Netzmodell aus Netzmodulen zusammenzustellen. Allerdings läßt sich der Verf. gern vom Gegenteil überzeugen.

126) Die Zeitbalken der beiden Fertigungslinien geben jeweils an, wie lange die Marken, die den Transportroboter oder Produkteinheiten repräsentieren, auf den Stellen s_3 , s_6 , s_8 , s_9 , s_{11} , und s_{14} des Netzmodells während der Ausführung von Transport- oder Bearbeitungsprozessen gebunden bleiben. Die Schaltzeitpunkte von Transitionen, die solche Markenbindungen einleiten oder beenden können (aber nicht müssen), sind ebenfalls festgehalten. Allerdings ist es im Ablaufdiagramm nicht möglich, bei Transitionen mit identischen Schaltzeitpunkten zwischen zwei verschiedenen Fällen zu differenzieren: Es kann sich entweder um nebenläufiges Schalten von Transitionen handeln, die zum selben Schaltschritt gehören. Statt dessen können die Transitionen aber auch zwar zeitgleich, jedoch logisch nacheinander geschaltet werden. Im Ablaufdiagramm können nur zeitliche Beziehungen zwischen Schaltakten, nicht aber alle logischen Schaltbeziehungen ausgedrückt werden. Die zugrundeliegenden logischen Schaltbeziehungen des Netzmodells und der darin ausgeführten Schaltprozesse werden dagegen durch den Erreichbarkeitsgraphen vollständig expliziert. Beispielsweise läßt sich aus dem Ausschnitt des Erreichbarkeitsgraphen in Abb. 62 erkennen, daß die Transition t_8 logisch vor den beiden - in einem Schaltschritt nebenläufig geschalteten - Transitionen t_5 und t_9 geschaltet werden muß, obwohl alle drei Transitionen im selben Zeitpunkt $T=9$ geschaltet werden. Anhand dieses Beispiels läßt sich erkennen, daß der Informationsgehalt, den das Petrinetz-Konzept über dynamische Modellaspekte mit Hilfe seiner Erreichbarkeitsgraphen anzubieten vermag, deutlich den Informationsgehalt konventioneller Ablaufdiagramme übersteigt.

127) Für das hier diskutierte Beispiel kann eine Prozeßkoordinierung eingerichtet werden, die auf einer projektiven Erreichbarkeitsanalyse mit einem Vorausschauhorizont ("look ahead") von mindestens zwei Schaltschritten beruht. Wenn die Prozeßkoordinierung im Erreichbarkeitsgraphen der Abb. 62 an der Markierung M_3 anlangt, könnte sie versucht sein, den Schaltschritt $\{t_1, t_3\}$ auszuführen. Durch die Projektion der Ausführungskonsequenzen würde jedoch erkannt werden, daß schon nach dem darauf folgenden Schaltschritt $\{t_4\}$ die Deadlockmarkierung M_9 erreicht wäre. Aufgrund dieses Projektionsergebnisses vermeidet die Prozeßkoordinierung den kritischen Schaltschritt $\{t_1, t_3\}$. Die Konsequenzenprojektion des alternativen Schaltschritts $\{t_7\}$ zeigt dagegen - innerhalb des Vorausschauhorizonts von 2 Schaltschritten - keine Deadlockgefahr. Daher läßt die Prozeßkoordinierung zu, den Schaltprozeß entlang des fett eingezeichneten Weges im Erreichbarkeitsgraphen der Abb. 62 fortzusetzen. Dadurch wird die Deadlockmarkierung M_9 umgangen.

Die voranstehende Skizze läßt allerdings erkennen, daß projektive Erreichbarkeitsanalysen keineswegs unter allen Umständen sicherstellen können, alle Deadlocks zu umgehen. Denn die Deadlockumgehung ist nur so lange garantiert, wie folgende Bedingung erfüllt ist: Die Anzahl der Schaltschritte, die im Vorausschauhorizont untersucht werden, muß mindestens so groß sein wie die Schaltschrittzahlen aller deadlockerzwingenden Schaltprozesse im untersuchten Netz. Dabei wird ein Schaltprozeß als deadlockerzwingend bezeichnet, wenn sich nach dem Ausführen des ersten prozeßzugehörigen Schaltschritts nicht mehr verhindern läßt, bei fortgesetzten Schaltschrittausführungen schließlich eine Deadlockmarkierung zu erreichen. Eine Deadlockvermeidung wäre dann nur noch möglich, indem - vor dem Erreichen der Deadlockmarkierung - durch Ausnutzen der Schaltregelpermissivität überhaupt kein Schaltschritt ausgeführt würde. Aber dies wäre lediglich ein "freiwilliges" Antizipieren der - ansonsten unvermeidlichen - Deadlockmarkierung. Im Erreichbarkeitsgraphen der Abb. 62 führt ein solcher deadlockerzwingender Schaltprozeß von der Markierung M_3 durch das Ausführen des Schaltschritts $\{t_1, t_3\}$ über die Markierung M_5 und den Schaltschritt $\{t_4\}$ zur Deadlockmarkierung M_9 . Nachdem der Schaltschritt $\{t_1, t_3\}$ ausgeführt worden ist, könnte die Deadlockmarkierung M_9 nur noch dadurch "verhindert" werden, daß unter der Markierung M_5 auf das Ausführen des einzig aktivierten Schaltschritts $\{t_4\}$ verzichtet würde.

Die Problematik von deadlockumgehenden Koordinierungsstrategien besteht darin, daß Netzmodelle für realistische Produktionssysteme überaus umfangreich ausfallen (vgl. zur Verdeutlichung die Fallstudie). Für solche Netzmodelle scheidet die vollständige Konstruktion ihrer Erreichbarkeitsgraphen aufgrund begrenzter Informationsverarbeitungsressourcen im allgemeinen aus. Wenn der Erreichbarkeitsgraph eines Netzmodells nicht vollständig bekannt ist, besteht auch keine Kenntnis darüber, ob in den noch nicht konstruierten Bereichen des Erreichbarkeitsgraphen Deadlockmarkierungen vorkommen. Folglich ist es unmöglich, für die Schaltschrittzahlen aller deadlockerzwingenden Schaltprozesse eine obere Schranke mit Sicherheit anzugeben. Statt dessen kann nur eine obere Schranke für den

bisher konstruierten Ausschnitt des Erreichbarkeitsgraphen angegeben werden. Deshalb lassen es unvollständig konstruierte Erreichbarkeitsgraphen nicht zu, die Erfüllung der o.a. Bedingung für eine garantierte Deadlockumgehung festzustellen. Lediglich die Bedingungsverletzung läßt sich erkennen. Dies ist genau dann der Fall, wenn im bereits konstruierten Ausschnitt des Erreichbarkeitsgraphen mindestens ein deadlockerzwingender Schaltprozeß existiert, dessen Schrittzahl den Vorausschauhorizont der Koordinierungsstrategie übersteigt.

Trotz der vorgenannten Schwierigkeiten ist die deadlockumgehende Koordinierungsstrategie im wesentlichen aus zwei Gründen keineswegs abwegig. Erstens entsprechen Koordinierungsstrategien mit begrenzten Vorausschauhorizonten durchaus produktionswirtschaftlichen Gepflogenheiten. Vgl. dazu den Hinweis auf "look ahead"-Regeln. Daher spricht der allgemeine Kohärenzgedanke zugunsten der deadlockumgehenden Koordinierungsstrategie. Zweitens stößt die Strategie der Deadlockumgehung erst dann auf Schwierigkeiten, wenn mit unvollständig konstruierten Erreichbarkeitsgraphen gearbeitet wird. In diesem Fall versagt aber auch die Modifizierungsstrategie zur Sicherung der Deadlockfreiheit. Denn die Deadlockfreiheit eines Netzmodells läßt sich - im hier diskutierten Kontext der Erreichbarkeitsanalyse - ohne einen vollständigen Erreichbarkeitsgraphen nicht nachweisen. Lediglich die Kompositionsstrategie könnte in betracht gezogen werden, um die Deadlockfreiheit eines Netzmodells zu garantieren. Es wird aber an anderer Stelle dargelegt, daß die Netzmodule und Kompositionsregeln, die im Rahmen dieser Kompositionsstrategie angeboten werden, unter einer gravierenden Einschränkung ihrer Modellierungsmächtigkeit leiden. Daher scheidet auch diese Strategie für die hier interessierende Modellierung realer Produktionssysteme aus. Folglich sind nur dann Schwierigkeiten der deadlockumgehenden Koordinierungsstrategie zu beklagen, wenn auch die beiden Varianten der Strategie der Deadlockfreiheit unter Unzulänglichkeiten leiden. Deshalb läßt sich die Strategie der Deadlockumgehung durch ein "faute de mieux"-Argument verteidigen.

Zwar kann für das hier diskutierte, extrem vereinfachte Beispiel zweier Fertigungslinien ein modifiziertes Netzmodell vorgelegt werden, das deadlockfrei ist. Aber selbst in diesem realitätsfernen Fall läßt sich die Modifizierungsstrategie nur dadurch zum Erfolg führen, daß auf das ausdrucksstarke Konstrukt der Inhibitorkanten zurückgegriffen wird. Dies ist mit den Ausdrucksmitteln der Synthetischen Netze bislang noch nicht möglich. Gleiches gilt für zahlreiche andere Netzklassen, wie z.B. Stelle/Transition-Netze und Prädikat/Transition-Netze.

128) Zu den Koordinierungsproblemen, die im Rahmen des Petrinetz-Konzepts besonders häufig diskutiert werden, gehören:

- das Problem der speisenden Philosophen; vgl. RAZOUK (1985c), S. 12ff.; VALMARI (1988a), S. 101ff. u. 111; vgl. darüber hinaus auch die Beiträge zum selben Koordinierungsproblem, die jedoch nicht zum Petrinetz-Konzept gehören, DIJKSTRA (1972) (als geistiger Urheber der Problemformulierung); CHANDY (1984); PEARL (1987), S. 255f.;
- das Problem der Zigarettenraucher;
- das Problem des gemeinsamen Zugriffs von zwei Teilprozessen auf zwei Ressourcen; vgl. ULLRICH (1976), S. 3/31; MÜLLER, B. (1981), S. 201; PETERSON, J. (1981), S. 85f.; ABEL, D. (1990), S. 24f.

129) In der Netzliteratur hat sich der Terminus technicus "Synchronisation" durchgesetzt. Inhaltlich stimmt er mit dem Begriff der Prozeßkoordinierung überein, den der Verf. bevorzugt.

130) Vgl. zu solchen Synchronisationsstrategien BYRN (1974), S. III-32ff. (mit einer ausführlichen Erörterung verschiedenster Strategieansätze); ABEL, D. (1990), S. 26f. (Koordinierung durch gemeinsam geteilte Stellen versus Synchronisation durch gemeinsam geteilte Transitionen).

Die Synchronisationsstrategien bestehen zumeist darin, die deadlockbelasteten Netze so umzugestalten, daß die Deadlockfreiheit der modifizierten Netze sichergestellt ist. Eine seltener verfolgte Synchronisationsstrategie besteht darin, von einfachen, als deadlockfrei bekannten Netzmodulen auszugehen. Diese werden auf der Grundlage eines strategiespezifischen Regelsatzes so zusammengesetzt, daß sich auch die Deadlockfreiheit der resultierenden komplexeren Netze garantieren läßt. Schließlich wird mitunter auch vorgeschlagen, die Deadlockfreiheit von Netzen nicht unbedingt vorauszusetzen. Statt dessen soll das Schalten von Transitionen in Netzen gezielt so ausgestaltet werden, daß es niemals zu einem unerwünschten Deadlock kommen kann. Alle drei vorgenannten Strategievarianten wurden schon kurz zuvor als Modifizierungs-, Kompositions- und Umgehungsstrategie thematisiert.

Darüber hinaus werden mitunter auch von vornherein Synchronisationskonzepte verwandt, die eine deadlockfreie Koordinierung von Teilprozessen garantieren. Dabei erfolgt aber keine Komposition von umfangreicheren Netzen aus einfachen Netzmodulen. Vielmehr werden die Synchronisationskonzepte in "monolithischen" Netzkonstruktionen angewendet. Zu diesen Synchronisationskonzepten gehören z.B. das Semaphoren-Konzept und das Rendezvous-Konzept. Nähere Erläuterungen dieser Synchronisationskonzepte und der Gestaltungsstrategien, die daraus für deadlockfreie Netze abgeleitet werden, finden sich bei HOLT, R. (1971a), S. 29, 103ff., 172f. u. 179; COFFMAN (1971), S. 72f.; COFFMAN (1973), S. 46 u. 51ff.

131) Vgl. zu Livelocks KWONG (1979), S. 172f., 178 u. 182ff.; SIFAKIS (1980c), S. 588; PRIESE (1983), S. 237; FEHLING (1990b), S. 3f.

Die Quellen unterscheiden sich in ihren Livelock-Auffassungen. Sie stimmen allerdings in der Konsequenz überein, zum Phänomen des "Verhungerns" von Prozessen führen zu können, das an anderer Stelle skizziert wird. Die Livelockdefinition wird hier vorrangig auf diese Konsequenz hin ausgerichtet.

132) Falls ein solcher Schaltprozeß existiert, erfüllen alle Markierungen $M_{r(t)}$ aus einer Markierungsfolge MF_L mit $l \in \{1, \dots, L\}$ und $L \in \mathcal{N}_L$ die Definition der Livelockmarkierung.

133) Aktuell unendliche Prozesse vertragen sich nicht mit der finitistischen Ausrichtung dieser Arbeit. Der Einfachheit halber werden potentiell unendliche Schaltprozesse auch nur als unendliche Prozesse angesprochen.

134) Vgl. die Erläuterung von Teilprozessen in Netzen.

135) Vgl. zu Interpretation von Livelocks durch Endlosschleifen PRIESE (1983), S. 237.

Solche Endlosschleifen scheint auch HÄMMÄINEN (1987), S. 288, zu meinen, obwohl dort kein expliziter Bezug auf Livelocks erfolgt.

136) Vgl. dazu den Hinweis, daß die Ausgangsmarkierung M_0 jedes M_0 -reversiblen Netzes eine Livelockmarkierung ist.

137) Vgl. dazu auch die Modellierung der Transaktion "Einlastungsentscheidung".

138) Dieser Gedanke findet sich bereits bei HOLT, A. (1968), S. 294, bezüglich der Möglichkeit, Informationsverarbeitungsprozesse parallel auszuführen. Er läßt sich aber auch auf produktionswirtschaftliche Sachverhalte übertragen. Dies betrifft vor allem losweise Produktionen. Denn die Herstellung eines nicht-degenerierten Loses bedeutet, daß an jeder betroffenen Bearbeitungsstation Arbeitsgänge wiederholt nacheinander ausgeführt werden, welche die beiden o.a. Unabhängigkeitsvoraussetzungen erfüllen. Daraus folgt die Erkenntnis, daß die Arbeitsgänge ebenso parallel auf mehreren Bearbeitungsstationen ausgeführt werden können. Die Realisierung dieser Erkenntnis bedeutet ein Auflösen der fest vorgegebenen Produktionslose. An ihre Stelle tritt ein "individueller" Durchlauf der Werkstücke, an denen die Arbeitsgänge vollzogen werden müssen, durch das modellierte Produktionssystem. Darauf beruht auch die Konzeptualisierung der Werkstückbearbeitung, die für Flexible Fertigungssysteme bereits skizziert wurde und später in der Fallstudie umgesetzt wird.

139) Die Teilprozesse können auch zu Schaltakten einzelner Transitionen degenerieren. Das ist z.B. bei dem Schaltproblem der Fall, das FEHLING (1990b), S. 3f., anspricht.

140) Vgl. zum "Verhungern" (starvation) oder - synonym - permanenten (kritischen) Blockieren von Prozessen in dynamischen Systemen HOLT, R. (1971b), S. 36; BYRN (1974), S. I-32f.; COHEN, E. (1975), S. 91; ELLIS, C.A. (1977), S. 68; TORRIGIANI (1977), S. 352; TORRIGIANI (1978), S. 573; DEVILLERS (1978), S. 157; KWONG (1979), S. 172f., 182 u. 184; LAUER (1979), S. 113; CZAJA (1980), S. 234 u. 239; LAUER (1980), S. 456; CARSTENSEN (1982), S. 4; VILLA (1988c), S. 358 (im Kontext Flexibler Fertigungssysteme, allerdings nicht auf Prozesse, sondern auf "starving situations" bezogen).

141) Ein Teilprozeß $TPRO_p$ mit der prozeßspezifischen Schaltfolge $SF_p = (SS_{a(1)}, \dots, SS_{a(L_p)})$ verhungert in einem Schaltschritt $SS_{a(t)}$ mit $l \in \{1, \dots, L_p\}$ und $L_p \in \mathcal{N}_L$, falls folgende Bedingungen vorliegen:

- Der Teilprozeß könnte zwar durch Ausführen seines ersten Schaltschritts $SS_{a(t)}$ mit $l=1$ gestartet oder durch Ausführen seines Schaltschritts $SS_{a(t)}$ mit $l \in \{2, \dots, L\}$ fortgesetzt werden. Aber er ist noch nicht beendet, weil noch nicht sein letzter Schaltschritt $SS_{a(L_p)}$ ausgeführt worden ist.
- Der Schaltschritt $SS_{a(t)}$ ist zwar unter unendlich vielen Markierungen aktiviert, steht dabei aber jeweils in Konflikt mit einem Schaltschritt aus einem parallel ablaufenden Livelockprozeß.
- Unter jeder Markierung, unter welcher der Schaltschritt $SS_{a(t)}$ konfliktionär zu einem Livelock-Schaltschritt aktiviert ist, wird der Schaltkonflikt zugunsten des Livelockprozesses aufgelöst.
- Der Teilprozeß verharrt daher in seinem Schaltschritt $SS_{a(t)}$ unendlich oft aktiviert, ohne jemals beendet zu werden.

Ein solcher verhungerner Teilprozeß bedeutet, daß das gesamte Netz unfair ist. Denn die Transitionen, die zum Schaltschritt $SS_{a(t)}$ gehören, sind per constructionem unendlich oft aktiviert, ohne jemals geschaltet zu werden. Vgl. zur Unfairness von Netzen mit verhungern den Teilprozessen CZAJA (1980), S. 234; CARSTENSEN (1982), S. 4. Umgekehrt ist es in fairen Netzen unmöglich, Teilprozesse in der oben skizzierten Weise verhungern zu lassen; vgl. BYRN (1974), S. I-33. Denn in fairen Netzen ist es per definitionem ausgeschlossen, daß unendlich oft aktivierte Transitionen nicht ebenso unendlich oft geschaltet werden. Die Fairness von Netzen wurde bereits an früherer Stelle definiert.

142) Vgl. KELLER, R. (1972a), S. 45.

143) Vgl. ULLRICH (1976), S. 3/9.

144) Dead- und Livelocks stimmen also zunächst darin überein, daß die Transitionen dauerhaft nicht geschaltet werden. Dieses Nichtschalten hat jedoch zwei vollkommen unterschiedliche Ursachen. Bei Deadlocks sind die Transitionen blockiert, d.h. überhaupt nicht aktiviert. Bei Livelocks sind dagegen die Transitionen eines verhungern den Teilprozesses immer wieder aktiviert, werden aber nicht geschaltet, weil die Transitionen eines konkurrierenden Livelockprozesses permanent bevorzugt werden. Vgl. dazu auch die Gegenüberstellung von Deadlocks und

permanentem Blockieren (Livelocks) bei HOLT,R. (1971b), S. 36. Darüber hinaus setzen Livelocks immer die Betrachtung von mehreren Teilprozessen voraus, während Deadlocks zunächst ohne Bezugnahme auf Teilprozesse definiert wurden.

145) Aus dieser Perspektive lassen sich die Transitionen des permanent blockierten Teilprozesses als Livelock-betroffene Transitionen - oder kurz: Livelocktransitionen - bezeichnen.

146) Auf solche erwünschten (End-)Markierungen eines Netzmodells wird im folgenden Kapitel näher eingegangen.

147) Die Forderung nach Livelockfreiheit vertritt z.B. BILLINGTON (1981a), S. 18f.; mittelbar auch VILLA (1988c), S. 358, durch den Wunsch, "starving situations" zu vermeiden.

5.2.2.2.4 Finalität, Optimalität und Terminieren

Bei der Finalität von Netzen handelt es sich um eine vielschichtige Netzeigenschaft. Sie wirkt sich auf allen drei semiotischen Betrachtungsebenen aus. Aus syntaktischer Perspektive wird die konventionelle Definition von Netzen um ein Komplement zur Ausgangsmarkierung - die Endmarkierungsmenge - ergänzt. Die Semantik von Netzen kann in operationaler Hinsicht um den Aspekt des Terminierens bereichert werden. Auf der pragmatischen Ebene läßt sich das Zielsystem des Modellierungsträgers einbeziehen.

Notwendige Voraussetzung für die Finalität eines Netzes ist, daß eine nicht-leere Menge von Endmarkierungen $M_{E/e}$ ¹⁾ mit $e \in \{1, \dots, E\}$ und $E \in \mathcal{N}_+$ bekannt ist²⁾. Unter einer Endmarkierung wird eine zulässige Netzmarkierung verstanden, die vom Modellierungsträger als erwünschte Markierung ausgezeichnet wird³⁾. Im Regelfall unterscheiden sich die Endmarkierungen von der Ausgangsmarkierung eines Netzes. Dies ist jedoch keineswegs notwendig⁴⁾. Die Gesamtheit aller derart festgelegten Markierungen ist die Endmarkierungsmenge ME mit $ME = \{M_{E/e}: e = 1, \dots, E\}$.

Aufgrund ihres intentionalen Charakters knüpfen Endmarkierungen unmittelbar an das Zielsystem eines Modellierungsträgers⁵⁾ an. Da Endmarkierungen eines Netzmodells jeweils einzelne erwünschte Modellzustände darstellen, können sie sich nicht auf die Formalziele dieses Zielsystems beziehen. Denn die Formalziele betreffen stets die Art und Weise, in der jene intendierten Modellzustände realisiert werden. Statt dessen vermag aber jede Endmarkierung einen Modellzustand zu repräsentieren, in dem alle Sachziele des Modellierungsträgers erfüllt werden. Fortan werden Endmarkierungen nur zu diesem Modellierungszweck verwendet. Daher bildet die Endmarkierungsmenge die Gesamtheit aller Modellzustände ab, die einem vorliegenden Sachzielsystem genügen.

Ein Netz, das eine Endmarkierungsmenge besitzt, verhält sich genau dann final, wenn es mindestens einen endlichen Schaltprozeß zuläßt, der von der Ausgangsmarkierung des Netzes zu mindestens einer seiner Endmarkierungen führt. Ein endlicher Schaltprozeß $PRO_{r,f}(SF_L, MF_L)$ ist eine Folge von alternierenden Netzmarkierungen und Schaltschritten, die in der Markierung M_r beginnt und nach Ausführen der endlichen Schaltfolge SF_L mit $L \in \mathcal{N}_0$ in der Markierung M_f mit $M_f \in RM(M_r)$ endet. Daher gilt für die Finalität eines Netzes SN, die durch das Prädikat $FIN(SN)$ ausgedrückt wird:

$$\begin{aligned} & FIN(SN) \\ :\Leftrightarrow & \exists(M_{E/e} \in ME): M_{E/e} \in RM(M_0) \end{aligned}$$

Mit Hilfe der früher eingeführten Prozeßmenge PROM läßt sich die Netzfinalität äquivalent ausdrücken als⁶⁾:

$$\begin{aligned} & FIN(SN) \\ \Leftrightarrow & \exists(M_{E/e} \in ME) \exists(L \in \mathcal{N}_0) \exists(PRO_{r,f}(SF_L, MF_L) \in PROM): \dots \\ & MF_L = (M_r, \dots, M_f) \wedge M_r = M_0 \wedge M_f = M_{E/e} \end{aligned}$$

Die endlichen Schaltprozesse $PRO_{r,f}(SF_L, MF_L)$, die wegen $M_r = M_0$ und $M_f = M_{E/e}$ jeweils unter der Ausgangsmarkierung M_0 starten und in einer der Endmarkierungen $M_{E/e}$ enden, werden als finale Schaltprozesse bezeichnet und als $PRO_{0,E/e}(SF_L, MF_L)$ notiert. Sie werden in der Finalprozeßmenge FPROM zusammengefaßt, für die gilt:

$$\text{FPROM} = \{ \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L) : \exists (M_{E/e} \in \text{ME}) \exists (L \in \mathcal{N}_0) : \dots \\ \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L) \in \text{PROM} \wedge \text{MF}_L = (M_0, \dots, M_{E/e}) \}$$

Falls ein Netz eine nicht-leere Menge erwünschter Endmarkierungen besitzt, aber nicht final ist, dann läßt sich das Sachzielsystem des Modellierungsträgers unter keinen Umständen erfüllen. Daher wird ein solches Netz auch als sachlich inkonsistent bezeichnet. Alle Schaltprozesse, die unter der Ausgangsmarkierung gestartet werden, geraten in einem solchen Netz in eine von zwei folgenden Fällen⁷⁾: Entweder enden die Prozesse in einer Deadlock-Markierung, noch bevor sie eine der intendierten Endmarkierungen erreichen können. Oder sie werden unendlich lange fortgesetzt, ohne dabei eine der Endmarkierungen zu erreichen, weil sie sich in Livelocks verfangen. Die Finalprozeßmenge eines solchen Netzes ist notwendig leer: $\text{FPROM} = \emptyset$. Andernfalls, wenn sich ein Netz mit nicht-leerer Menge erwünschter Endmarkierungen final verhält, wird von einem sachlich konsistenten Netz gesprochen. Seine Finalprozeßmenge FPROM umfaßt dann mindestens einen finalen Schaltprozeß.

Die Netzfinalität berücksichtigt durch ihre Bezugnahme auf erreichbare Endmarkierungen zunächst nur das Sachzielsystem eines Entscheidungsträgers. Sie kann aber auch um den Aspekt des Formalzielsystems bereichert werden. Zu diesem Zweck wird jeder Transition t_n und jeder Schaltfarbe c_n der Schaltwert $\underline{\text{sw}}(t_n, c_n)$ zugeordnet⁸⁾. Der Schaltwert ist für ein G -dimensionales Formalzielsystem FZ , das die Formalziele fz_g mit $g \in \{1, \dots, G\}$ und $G \in \mathcal{N}_+$ umfaßt, ein Zeilenvektor⁹⁾ $\underline{\text{sw}}(t_n, c_n) = (\text{sw}_1(t_n, c_n), \dots, \text{sw}_G(t_n, c_n))$ ¹⁰⁾. Jede Komponente $\text{sw}_g(t_n, c_n)$ des Schaltvektors gibt für ein Formalziel fz_g an, um welchen Betrag die Zielfunktion zf_g des Formalziels fz_g durch das Schalten der Transition t_n mit der Schaltfarbe c_n verändert wird¹¹⁾. Der Schaltwert $\underline{\text{sw}}(\text{SS}_a) = (\text{sw}_1(\text{SS}_a), \dots, \text{sw}_G(\text{SS}_a))$ eines Schaltschritts SS_a ergibt sich unmittelbar als Vektorsumme der Schaltwerte $\underline{\text{sw}}(t_{n(w)}, c_{n(w)})$ aller Transitionen $t_{n(w)}$, die in den Paaren $(t_{n(w)}, c_{n(w)})$ des Schaltschritts SS_a enthalten sind:

$$\begin{aligned} \text{SS}_a &= \{ (t_{n(w)}, c_{n(w)}) : w = 1, \dots, W_a \} \\ \Rightarrow \underline{\text{sw}}(\text{SS}_a) &= (\text{sw}_1(\text{SS}_a), \dots, \text{sw}_G(\text{SS}_a)) \\ &= (\sum_{w \in \{1, \dots, W_a\}} : \text{sw}_1(t_{n(w)}, c_{n(w)}), \dots, \sum_{w \in \{1, \dots, W_a\}} : \text{sw}_G(t_{n(w)}, c_{n(w)})) \\ \Rightarrow \underline{\text{sw}}(\text{SS}_a) &= \sum_{w \in \{1, \dots, W_a\}} : \underline{\text{sw}}(t_{n(w)}, c_{n(w)}) \end{aligned}$$

Der Schaltwert $\underline{\text{sw}}(\text{PRO}_{r.f}(\text{SF}_L, \text{MF}_L))$ eines endlichen Schaltprozesses $\text{PRO}_{r.f}(\text{SF}_L, \text{MF}_L)$ ist die Vektorsumme der Schaltwerte $\underline{\text{sw}}(\text{SS}_{a(l)})$ aller Schaltschritte $\text{SS}_{a(l)}$, die mit $l \in \{1, \dots, L\}$ in der Schaltfolge SF_L des Schaltprozesses $\text{PRO}_{r.f}(\text{SF}_L, \text{MF}_L)$ enthalten sind. Der Übersichtlichkeit halber wird der betroffene Schaltprozeß $\text{PRO}_{r.f}(\text{SF}_L, \text{MF}_L)$ vereinfacht als $\text{PRO}_{r.f.p}$ mit $p \in \{1, \dots, P_{r.f}\}$ und $P_{r.f} \in \mathcal{N}_+(\omega)$ ¹²⁾ notiert. Dann gilt für den Schaltwert $\underline{\text{sw}}(\text{PRO}_{r.f.p}) = \underline{\text{sw}}_{r.f.p}$ dieses Schaltprozesses:

$$\begin{aligned} \text{PRO}_{r.f.p} &= \text{PRO}_{r.f}(\text{SF}_L, \text{MF}_L) \wedge \text{SF}_L = (\text{SS}_{a(1)}, \dots, \text{SS}_{a(L)}) \\ \Rightarrow \underline{\text{sw}}(\text{PRO}_{r.f.p}) &= (\text{sw}_1(\text{PRO}_{r.f.p}), \dots, \text{sw}_G(\text{PRO}_{r.f.p})) \\ &= \sum_{l \in \{1, \dots, L\}} : \underline{\text{sw}}(\text{SS}_{a(l)}) \\ &= (\sum_{l \in \{1, \dots, L\}} : \text{sw}_1(\text{SS}_{a(l)}), \dots, \sum_{l \in \{1, \dots, L\}} : \text{sw}_G(\text{SS}_{a(l)})) \end{aligned}$$

Fortan wird als vereinfachte Notation des Schaltwerts $\underline{\text{sw}}(\text{PRO}_{r.f.p})$ eines Schaltprozesses $\text{PRO}_{r.f.p}$ auch der Zeilenvektor $\underline{\text{sw}}_{r.f.p}$ mit $\underline{\text{sw}}_{r.f.p} = (\text{sw}_{1.r.f.p}, \dots, \text{sw}_{G.r.f.p})$ und $\text{sw}_{g.r.f.p} = \sum_{l \in \{1, \dots, L\}} : \text{sw}_g(\text{SS}_{a(l)})$ für alle $g \in \{1, \dots, G\}$ zugelassen.

Auf diese Weise ist auch der Schaltwert $\underline{sw}(\text{PRO}_{0.E/e.p}) = \underline{sw}_{0.E/e.p}$ jedes finalen Schaltprozesses $\text{PRO}_{0.E/e.p} = \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L)$ eindeutig determiniert. Auf dieser Grundlage läßt sich untersuchen, welche Schaltprozesse in einem Netz das kombinierte Sach- und Formalzielsystem eines Entscheidungsträgers bestmöglich (optimal) erfüllen. Dadurch wird die Erreichbarkeitsanalyse von Netzmodellen dem betriebswirtschaftlich besonders interessanten Bereich der Optimierungsmodelle erschlossen.

Ein finaler Schaltprozeß $\text{PRO}_{0.E/e.p}$ erfüllt per definitionem das Sachzielsystem eines Entscheidungsträgers. Er wird daher auch als sachlich zulässiger Schaltprozeß bezeichnet. Ein solcher Schaltprozeß heißt darüber hinaus auch formal zulässig, wenn sein Schaltwert $\underline{sw}(\text{PRO}_{0.E/e.p})$ alle Restriktionen des Formalzielsystems - also alle Satisfizierungsziele - erfüllt¹³). Andernfalls heißt der finale Schaltprozeß formal unzulässig¹⁴). Alle finalen Schaltprozesse, die formal - und somit auch sachlich - zulässig sind, werden fortan auch kurz als zulässige finale Schaltprozesse bezeichnet. Die Menge aller zulässigen finalen Schaltprozesse eines Netzes ist eine Teilmenge der bereits eingeführten Finalprozeßmenge FPRM. Diese Teilmenge wird als zulässige Finalprozeßmenge ZFPRM angesprochen. Mit $\text{SAT}(\text{PRO}_{0.E/e.p})$ als Prädikat, das die Erfüllung aller Satisfizierungsziele durch einen finalen Schaltprozeß $\text{PRO}_{0.E/e.p}$ mit $\text{PRO}_{0.E/e.p} = \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L)$ ausdrückt, gilt für diese Prozeßmenge ZFPRM:

$$\begin{aligned} \text{ZFPRM} = \{ & \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L) : \dots \\ & \exists (M_{E/e} \in \text{ME}) \exists (L \in \mathcal{N}_0) : \text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L) \in \text{PRM} \\ & \wedge \text{MF}_L = (M_0, \dots, M_{E/e}) \wedge \text{SAT}(\text{PRO}_{0.E/e}(\text{SF}_L, \text{MF}_L)) \} \end{aligned}$$

Wenn in einem Netz mindestens ein zulässiger finaler Schaltprozeß existiert, heißt es formal konsistent. Andernfalls wird von einem formal inkonsistenten Netz geredet, wenn es zwar sachlich konsistent ist, aber alle seine finalen Schaltprozesse unzulässig sind¹⁵). Dann fällt die zulässige Finalprozeßmenge ZFPRM mit der leeren Menge zusammen, obwohl die Finalprozeßmenge FPRM wegen der vorausgesetzten sachlichen Netzkonsistenz nicht leer ist.

Ein formal zulässiger Schaltprozeß wird durch das Dominanzkonzept der Vektoroptimierung¹⁶) genau dann als nicht-dominierter (effizienter¹⁷) Schaltprozeß ausgezeichnet, wenn kein anderer formal zulässiger Schaltprozeß existiert, der den erstgenannten Schaltprozeß durch einen echt besserstellenden Schaltwert dominiert. Dies wird durch das Prädikat $\text{NDOM}(\text{PRO}_{0.E/e.p})$ ausgedrückt¹⁸). Ein Netz heißt aufgrund dieser Vereinbarungen genau dann \underline{sw}^* -final, wenn folgende Bedingungen erfüllt sind:

- Das Netz ist final.
- $\underline{sw}^* = (\underline{sw}_1^*, \dots, \underline{sw}_G^*)$ ist der Schaltwert von mindestens einem finalen Schaltprozeß $\text{PRO}_{0.E/e.p}$ mit $\underline{sw}_g^* = \underline{sw}_{g,0.E/e.p}$ für alle $g \in \{1, \dots, G\}$.
- Der mindestens eine finale Schaltprozeß $\text{PRO}_{0.E/e.p}$ mit dem Schaltwert \underline{sw}^* ist formal zulässig, d.h. es gilt $\text{PRO}_{0.E/e.p} \in \text{ZFPRM}$.
- Der mindestens eine finale, formal zulässige Schaltprozeß $\text{PRO}_{0.E/e.p}$ mit dem Schaltwert \underline{sw}^* wird von keinem anderen finalen, ebenso formal zulässigen Schaltprozeß dominiert: $\text{NDOM}(\text{PRO}_{0.E/e.p})$.

Der nicht-dominierte Schaltwert \underline{sw}^* eines finalen (und formal zulässigen¹⁹) Schaltprozesses gilt nicht nur für diesen Schaltprozeß selbst. Vielmehr stellt er eine dynamische Eigenschaft dar, die für ein \underline{sw}^* -finales Netz als Ganzes zutrifft²⁰). Er wird daher auch kurz als optimaler Schaltwert \underline{sw}^* des \underline{sw}^* -finalen Netzes bezeichnet. Aus dem Dominanzkonzept der Vektoroptimierung folgt, daß jeder nicht-dominierte Schaltwert \underline{sw}^* einen der *bestmöglichen* Schaltwerte darstellt, die durch *alle* finalen und formal zulässigen Schaltprozesse des Netzes erreicht werden können. Jeder nicht-dominierte Schaltwert \underline{sw}^* des Netzes erfüllt daher das zugrundeliegende Sach- und

Formalzielsystem in optimaler Weise. Deshalb wird er fortan auch als optimaler Schaltwert \underline{sw}^* bezeichnet. Aus dem gleichen Grund läßt sich ein \underline{sw}^* -finales Netz als ein \underline{sw}^* -optimales Netz ansprechen. Aus demselben Grund läßt sich jeder nicht-dominierte Schaltwert \underline{sw}^* eines Netzes auch als ein Netzoptimum ansprechen. Der zugehörige Schaltprozeß, der den optimalen Schaltwert \underline{sw}^* aufweist, heißt dementsprechend ein optimaler Schaltprozeß.

Im allgemeinen ist das Netzoptimum nicht eindeutig bestimmt. Denn für dasselbe Netz können mehrere zulässige finale Schaltprozesse mit verschiedenen, jeweils optimalen Schaltwerten \underline{sw}^* existieren. Ebenso ist es möglich, daß ein finales Netz überhaupt kein Netzoptimum besitzt, weil alle seine finalen Schaltprozesse formal unzulässig sind. Dann liegt ein formal inkonsistentes Netz mit $ZFPROM = \emptyset$ vor. Existenz und Mehrdeutigkeit von Netzoptima lassen sich dadurch formal präzise erfassen, daß jedes existierende Netzoptimum durch einen Index "o" mit $o \in \{1, \dots, O\}$ und $O \in \mathcal{N}_+$ identifiziert und als \underline{sw}_o^* notiert wird. Falls kein Netzoptimum existiert, weil alle finalen Schaltprozesse formal unzulässig sind, gilt $SW^* = \emptyset$ mit $O = 0$. Daher legt die Menge $SW^* = \{\underline{sw}_o^* : o = 1, \dots, O\}$ mit $O \in \mathcal{N}_0$ die Gesamtheit aller Netzoptima eines finalen Netzes eindeutig fest. Sie wird auch als Optimierungsmenge des jeweils untersuchten Netzes bezeichnet.

Analog zur oben eingeführten \underline{sw}^* -Finalität wird ein finales Netz fortan als SW^* -final bezeichnet, falls es die Optimierungsmenge SW^* besitzt. Ein finales, aber formal inkonsistentes Netz ist wegen $SW = \emptyset$ ein \emptyset -finales Netz. Wenn die Optimierungsmenge SW^* mindestens ein Netzoptimum umfaßt ($SW^* \neq \emptyset$), wird auch von einem SW^* -optimalen Netz gesprochen. Falls das zugrundeliegende Formalzielsystem mit $G=1$ aus nur genau einem Ziel besteht, liegt für jedes formal konsistente Netz sein Netzoptimum \underline{sw}^* mit $O=1$ und $SW^* = \{\underline{sw}_1^*\}$ eindeutig fest²¹⁾. Die voranstehend entfalteten Aspekte der Netzoptimalität werden abschließend durch das Prädikat $SW^*(SN)$ präzisiert²²⁾:

$$\begin{aligned} & SW^*(SN) \\ :\Leftrightarrow & SW^* = \emptyset \vee \dots \\ & (\exists (O \in \mathcal{N}_+) : SW^* = \{\underline{sw}_o^* : o = 1, \dots, O \wedge \dots \\ & \quad (\exists (M_{E/e} \in ME) \exists (L \in \mathcal{N}_0) \exists (PRO_{0,E/e}(SF_L, MF_L) \in ZFPROM) : \dots \\ & \quad \underline{sw}_o^* = \underline{sw}(PRO_{0,E/e}(SF_L, MF_L)) \wedge NDOM(PRO_{0,E/e}))) \end{aligned}$$

Die Eigenschaft der Netzfinalität wird verschärft durch die Anforderung der Netzterminierung²³⁾. Sie ist erfüllt, wenn von jeder erreichbaren Netzmarkierung aus mindestens eine Endmarkierung durch mindestens einen endlichen Schaltprozeß erreicht werden kann²⁴⁾. Wenn das Prädikat $TER(SN)$ das Terminieren eines Netzes SN ausdrückt, dann gilt²⁵⁾:

$$\begin{aligned} & TER(SN) \\ :\Leftrightarrow & \forall (M_r \in RM(M_0)) \exists (M_{E/e} \in ME) : M_{E/e} \in RM(M_r) \end{aligned}$$

Sofern die Endmarkierungsmenge genau eine Endmarkierung umfaßt, stellt diese einen "Heimzustand"²⁶⁾ dar. Der Heimzustand der Endmarkierung eines terminierenden Netzes kann von allen erreichbaren Markierungen aus durch jeweils einen endlichen Schaltprozeß erreicht werden²⁷⁾. In einem terminierenden Netz können allenfalls die Endmarkierungen Deadlocks darstellen²⁸⁾. Dies muß jedoch nicht der Fall sein. Beispielsweise kann ein terminierendes Netz so konstruiert sein, daß unter jeder seiner Endmarkierungen ein Schaltschritt aktiviert ist, dessen Ausführen jeweils die Ausgangsmarkierung hervorbringt²⁹⁾. Ein Schaltprozeß, der eine der Endmarkierungen erreicht, braucht daher dort nicht zu enden, sondern läßt sich so fortsetzen, daß er in zyklischer Weise zur Ausgangsmarkierung zurückführt. Das Netz ist dann zugleich terminierend und deadlockfrei³⁰⁾. Es besitzt zumindest die Ausgangsmarkierung und alle Endmarkierung

gen als Livelocks. Ein terminierendes Netz heißt dagegen echt terminierend³¹⁾, wenn drei zusätzliche Anforderungen erfüllt werden:

- Alle Endmarkierungen stellen erwünschte Deadlocks dar³²⁾.
- Alle erreichbaren Markierungen, bei denen es sich um keine Endmarkierungen handelt, sind lebendig³³⁾.
- Es ist unmöglich, im Netz potentiell unendliche Schaltprozesse auszuführen³⁴⁾.

In einem echt terminierenden Netz führen alle Schaltprozesse, die jeweils von einer erreichbaren Markierung aus gestartet werden, mit Sicherheit nach endlich vielen Schrittausführungen³⁵⁾ zu einer der Endmarkierungen, wenn sie nur lange genug fortgesetzt werden. Sobald diese Schaltprozesse jeweils eine Endmarkierung erreicht haben, enden sie dort auch notwendig, weil alle Endmarkierungen Deadlocks darstellen. Derart erwünscht endende Schaltprozesse werden als terminierende Schaltprozesse bezeichnet.

Anmerkungen zum Kapitel:

1) Falls wegen $E=1$ nur genau eine Endmarkierung M_{E_e} vorgesehen ist, läßt sie sich auch vereinfacht als M_E notieren.

2) Vgl. zur Verwendung von Endmarkierungen GOSTELOW (1971), S. 22 u. 113; GOSTELOW (1972), S. 20; CERF (1972), S. XVIII u. 144; BAER, J. (1973b), S. 19f.; BYRN (1974), S. III-42; GOSTELOW (1975), S. 349; PETERSON, J. (1976), S. 2 u. 4f.; ARAKI (1976b), S. 7f.; HERZOG, O. (1976b), S. 8; RAMMIG (1978), S. 43; BARZILAI (1979), S. 11; CONRY (1979a), S. 254; YOELI (1980a), S. 73; ROUCAIROL (1982b), S. 15ff.; PASSINO (1988a), S. 629f.

3) Die besondere Qualität von Endmarkierungen läßt sich in einem Netzmodell dadurch hervorheben, daß eine zusätzliche finale Stelle s_E eingeführt wird, für die gilt: Die Stelle s_E trägt unter einer erreichbaren Markierung M_i genau dann eine (keine) Marke, wenn diese Markierung eine Endmarkierung darstellt. Unter Vorgriff auf die nachfolgend eingeführte Endmarkierungsmenge ME gilt hierfür:

$$\forall (M_i \in \text{RM}(M_0)): (M_i(s_E) = 1 \leftrightarrow M_i \in \text{ME}) \wedge (M_i(s_E) = 0 \leftrightarrow M_i \notin \text{ME})$$

Vgl. zur Konstruktion einer solchen finalen Stelle YOELI (1980a), S. 73. Allerdings wird dort eine weniger strenge Konstruktionsvariante vorgenommen, die - nach Anpassung an die o.a. Notation - lediglich das Subjugat $M_i \in \text{ME} \rightarrow M_i(s_E) = 1$ umfaßt. Eine analoge Konstruktion, die allerdings das umgekehrte Subjugat $M_i(s_E) = 1 \rightarrow M_i \in \text{ME}$ liefert, findet sich bei BARZILAI (1979), S. 11. Mit mehreren finalen Stellen arbeitet PETERSON, J. (1976), S. 2 u. 4f. Vgl. zu weiteren Verwendungen finaler Stellen in Netzen ROUCAIROL (1982b), S. 15f.

4) Daher ist es möglich, bei der späteren Betrachtung zweier Netztheoreme die Ausgangsmarkierung als eine Endmarkierung auszuzeichnen, deren Reproduktion erwünscht wird.

5) Wiederum werden Modellierungs- und Entscheidungsträger miteinander identifiziert.

6) Diese zweite Ausdrucksweise wird jedoch nicht zur Definition der Netzfinalität herangezogen, weil die Prozeßmenge nur auf informale Weise eingeführt wurde. Die voranstehende erste Ausdrucksweise definiert die Finalität eines Netzes dagegen ausschließlich mit solchen Konstrukten, die sich aus den Definitionen von Stelle/Transition-Netzen und von Synthetischen Netzen in vollständig formalisierter Weise herleiten lassen.

7) In beiden Fällen gehört keine der Endmarkierungen zur Erreichbarkeitsmenge der Ausgangsmarkierung M_0 . Folglich gilt $M_{E_e} \notin \text{RM}(M_0)$ für alle $M_{E_e} \in \text{ME}$.

8) Schaltwerte von Transitionen finden sich ebenfalls bei PASSINO (1988a), S. 626f. (als Transitions-Kostenfunktion). Dabei bleiben allerdings die oben ebenso berücksichtigten Schaltfarben außer Betracht, weil dort nur Stelle/Transition-Netze betrachtet werden.

Es könnte auch in Betracht gezogen werden, den Stellen eines Netzes einen analogen Markierungswert zuzuweisen. Davon wird jedoch in dieser Arbeit aus zwei Gründen grundsätzlich abgesehen.

Erstens lassen sich Erreichbarkeitsgraphen am leichtesten mit bereits vorhandenen Auswertungsmethoden aus den Bereichen der Graphentheorie und der Künstlichen Intelligenz untersuchen, wenn nur die Kanten, nicht aber die Knoten der Erreichbarkeitsgraphen bewertet sind. Die Kantenbewertungen eines Erreichbarkeitsgraphen folgen aus den Schaltwerten der Transitionen aus den jeweils zugrundeliegenden Netzen. Dies wird in Kürze näher dargelegt. Darüber hinaus existieren spezielle Auswertungsmethoden für die Klasse der Synchronisationsnetze, die sich nur dann anwenden lassen, falls ihre Transitionen mit (skalaren) Schaltwerten versehen sind und ihre Stellen keine Markierungswerte besitzen. Ein Beispiel wird später anlässlich der Transformation einfacher Netzpläne in Synchronisationsnetze vorgestellt.

Zweitens besteht bei der Verwendung von Schalt- und Markierungswerten die Gefahr der Mehrfachverrechnung (double accounting) desselben Zielbeitrags. Falls z.B. das Ziel einer möglichst geringen Kapitalbindung betrachtet wird, so lassen sich entsprechende Zielbeiträge einerseits durch Marken erfassen, die kapitalbindende Objekte repräsentieren. Andererseits kommen dafür ebenso Transitionen in Betracht, deren Schaltakte kapitalbindende Auszahlungsakte darstellen. Im zweiten Fall richtet sich die Höhe der schaltaktbedingten Kapitalbindung nach der jeweils zugrundegelegten Schaltfarbe. Denn diese Schaltfarbe determiniert eindeutig, welche Marken am Schaltakt einer Transition teilnehmen. Diese Marken bilden aber - wie bereits erwähnt - jeweils die kapitalbindenden Objekte ab. Es müßte dann bei der Formulierung eines Netzmodells genau darauf geachtet werden, daß eine nur einmal anfallende Kapitalbindung nicht doppelt verrechnet wird. Dies wäre der Fall, wenn sowohl der Schaltwert eines Paares aus einer Transition und einer Schaltfarbe als auch Beitrag einer geschalteten Markenmenge zum Markierungswert einer Stelle berücksichtigt würde. Diese Gefahr wird jedoch von vornherein ausgeschlossen, wenn entsprechende Zielbeiträge entweder nur dem Schalten von Transitionen mit ausgezeichneten Schaltfarben oder aber nur den Markierungen von Stellen zugeordnet werden. Die Zuordnungsentscheidung ist in dieser Arbeit zugunsten der Transitionen und Schaltfarben gefallen, weil sich nur auf diese Weise der Aspekt, Mehrfachverrechnungen zu vermeiden, mit dem Aspekt der Ausnutzung vorhandener Auswertungsmethoden kohärent vereinbaren läßt. Denn die oben

erwähnten Analysemethoden für Erreichbarkeitsgraphen und Synchronisationsnetze setzen jeweils voraus, daß die Schaltakte von Transitionen bewertet sind und keine Markierungswerte der Stellen vorkommen.

9) Der Zeilenvektor könnte ebenso als ein Spaltenvektor festgelegt werden. Dies erfordert aber oftmals, auf die unständlichere Notation seiner transponierten Zeilendarstellung zurückgreifen zu müssen. Der rechentechnische Vorteil von Spaltenvektoren, daß sich mit ihrer Hilfe Gleichungssysteme übersichtlicher darstellen und lösen lassen, entfällt dagegen hier, da der Vektor des Schaltwerts einer Transition in keine Gleichungssysteme eingeht. Daher wird von vornherein auf die Notation des Zeilenvektors zurückgegriffen, die sich in natürlichsprachlichen Texten leichter handhaben läßt.

10) Wenn das Formalzielsystem wegen $G=1$ nur aus genau einem Formalziel $fz_1=fz_G=fz$ besteht, besitzt der Schaltwertvektor $\underline{sw}(t_n, c_n)$ nur genau eine Komponente: $\underline{sw}(t_n, c_n) = (sw_1(t_n, c_n))$. Für diesen degenerierten Fall kann der Schaltwert in formal abweichender, aber materiell äquivalenter Weise ebenso als Schaltwertskalar $sw(t_n, c_n) = sw_1(t_n, c_n)$ notiert werden. Solche skalaren Schaltwerte werden fortan für alle degenerierten Formalzielsysteme mit $G=1$ zugelassen, ohne jeweils ausdrücklich darauf hinzuweisen. Für Stelle/Transition-Netze, die keine Schaltfarben c_n vorsehen, nimmt der Schaltwertskalar die noch einfachere Form $sw(t_n)$ an.

11) Falls das Erfüllungsniveau eines Formalziels fz_g vom Schalten der Transition t_n nicht betroffen ist, wird der Änderungsbetrag als $sw_g(t_n, c_n) = 0$ festgesetzt.

12) Der Teilindex "p" identifiziert jeweils einen der insgesamt $P_{r,f}$ endlichen Schaltprozesse $PRO_{r,f}(SF_L, MF_L)$, welche dieselbe Startmarkierung M_r in dieselbe Zielmarkierung M_f transformieren, sich aber untereinander durch verschiedene Schalt- und Markierungsfolgen SF_L bzw. MF_L unterscheiden. Wegen der vorausgesetzten Endlichkeit der untersuchten Erreichbarkeitsgraphen sind unendliche azyklische Schaltprozesse ausgeschlossen. Endliche Erreichbarkeitsgraphen können aber zyklische Schaltwege enthalten, die von der Markierung M_r zur Markierung $M_f = M_r$ zurückführen. Solche Schaltwege entsprechen zyklischen Schaltprozessen, die sich potentiell unendlich oft wiederholt ausführen lassen. Jede endliche Wiederholungsanzahl stellt einen eigenständigen endlichen Schaltprozeß dar. Daher kann es in einem endlichen Erreichbarkeitsgraphen durchaus unendlich viele verschiedene endliche Schaltprozesse geben, die von der Startmarkierung M_r zur Zielmarkierung M_f führen. Daher erfüllt die Prozeßanzahl $P_{r,f}$ im allgemeinen die Beziehung $P_{r,f} \in \mathcal{N}_{\omega} \cup \{\omega\}$. Nur wenn im untersuchten Erreichbarkeitsgraphen zwischen der Startmarkierung M_r und der Zielmarkierung M_f kein zyklischer Schaltweg existiert, gilt $P_{r,f} \in \mathcal{N}_{\omega}$.

13) Aufgrund dieser Festlegung ist jeder formal zulässige notwendig auch ein sachlich zulässiger Schaltprozeß. Die Umkehrung gilt dagegen nicht, weil durchaus sachlich zulässige, aber formal unzulässige Schaltprozesse existieren können.

14) Dies berührt jedoch nicht die sachliche Zulässigkeit des Schaltprozesses, da jeder finale Schaltprozeß per definitionem sachlich zulässig ist. Jeder finale, aber formal unzulässige Schaltprozeß verletzt mindestens eine Restriktion des Formalzielsystems. Dies folgt unmittelbar aus der Negation der Definition formal zulässiger Schaltprozesse.

15) Da ein finales Netz vorausgesetzt wird, muß das Netz sachlich konsistent sein und mindestens einen finalen Schaltprozeß besitzen. Das Fehlen jedes Netzoptimums kann in einem finalen Netz daher niemals auf dessen - qua Voraussetzung niemals vorhandene - sachliche Inkonsistenz zurückgeführt werden. Die Nichtexistenz von Netzoptima wird dagegen von den Restriktionen (Satisfizierungszielen) des Formalzielsystems verursacht. Sie sind so eng formuliert, daß kein finaler, das Sachzielsystem erfüllender Schaltprozeß existiert, der allen Restriktionen zugleich gerecht würde. Die Restriktionen des Formalzielsystems widersprechen sich daher in ihrer Gesamtheit. Aus diesem Grunde wird nachfolgend auch von einer formalen Inkonsistenz des finalen Netzes gesprochen.

16) Vgl. zur Vektoroptimierung und ihrem Dominanzkonzept FANDEL (1972), S. 12; PFOHL (1981), S. 172ff.; DINKELBACH (1982), S. 159ff.; VON NEUMANN-COSEL (1983), S. 28; FANDEL (1991a), S. 48ff., insbesondere S. 50; vgl. auch am Rande KISTNER (1981c), S. 74f.; VINCKE (1986), S. 161f.; RAJURKAR (1987), S. 513f.; ZIMMERMANN, H. (1987b), S. 26f.; SIEBEN (1990), S. 29f. u. 48f.; KNOBLOCH (1990), S. 10; ZIONTS (1991), S. 378; FANDEL (1991b), S. 234.

In den voranstehenden Quellen wird oftmals nicht auf das Dominanz-, sondern auf ein Effizienzkonzept Bezug genommen. Ein Grund für diese terminologische Abweichung könnte darin liegen, daß das Dominanzkonzept ein negatorisch gehandhabtes Optimierungskonzept darstellt. Denn es wird stets nach *nicht*-dominierten Problemlösungen gesucht. Zwar mag der Hang zu affirmatorischen Konzeptbenennungen nahelegen, lieber von effizienten Lösungen als von nicht-dominierten Lösungen zu sprechen; vgl. z.B. KISTNER (1981c), S. 75, oder SIEBEN (1990), S. 29f. Der Verf. hält jedoch an der Bezeichnung nicht-dominierter Problemlösungen fest, weil der Effizienzbegriff wegen seiner vielfältig variierenden Ausdeutungen rasch zu Mißverständnissen Anlaß geben kann. Auch in dieser Arbeit wird der Effizienzbegriff in abweichender, nicht notwendig mit der Vektoroptimierung verträglicher Weise verwendet.

17) Vgl. die voranstehende Anmerkung.

18) Zur formalen Präzisierung von echt besserstellenden und nicht-dominierten Schaltwerten wird der Einfachheit halber ein Formalzielsystem $FZ_g = \{fz_1, \dots, fz_g\}$ betrachtet, für dessen Formalziele fz_g mit $g \in \{1, \dots, G\}$ gilt: Sie werden

um so besser erfüllt, je niedriger der Zielbeitrag $sw_g(t_n, c_n)$ einer Transition t_n ausfällt, die mit der Schaltfarbe c_n geschaltet wird. Unter dieser Voraussetzung stellt der Schaltwert $\underline{sw}(\text{PRO}_{0,E/e_1,p_1}) = (sw_{1,0,E/e_1,p_1}, \dots, sw_{G,0,E/e_1,p_1})$ eines formal zulässigen Schaltprozesses $\text{PRO}_{0,E/e_1,p_1}$ einen Entscheidungsträger bezüglich seines Formalzielsystems $FZ_\leq = \{fz_1, \dots, fz_G\}$ genau dann echt besser als der Schaltwert $\underline{sw}(\text{PRO}_{0,E/e_2,p_2}) = (sw_{1,0,E/e_2,p_2}, \dots, sw_{G,0,E/e_2,p_2})$ eines anderen, ebenso formal zulässigen Schaltprozesses $\text{PRO}_{0,E/e_2,p_2}$ mit $(e_1, p_1) \neq (e_2, p_2)$, falls das Prädikat $\text{BES}_\leq(\text{PRO}_{0,E/e_1,p_1}, \text{PRO}_{0,E/e_2,p_2})$ gültig ist. Dieses Prädikat wird mit Hilfe der Menge ZFPROM aller formal (und sachlich) zulässigen Schaltprozesse definiert durch:

$$\begin{aligned} & \text{BES}_\leq(\text{PRO}_{0,E/e_1,p_1}, \text{PRO}_{0,E/e_2,p_2}) \\ \Leftrightarrow & \text{PRO}_{0,E/e_1,p_1} \in \text{ZFPROM} \wedge \text{PRO}_{0,E/e_2,p_2} \in \text{ZFPROM} \\ & \wedge (\forall (g \in \{1, \dots, G\}): sw_{g,0,E/e_1,p_1} \leq sw_{g,0,E/e_2,p_2}) \\ & \wedge (\exists (g \in \{1, \dots, G\}): sw_{g,0,E/e_1,p_1} < sw_{g,0,E/e_2,p_2}) \end{aligned}$$

Diese Formulierung unterliegt keiner grundsätzlichen Einschränkung ihrer Allgemeingültigkeit. Denn sie erfaßt bereits alle Meliorisierungs- und Extremierungsziele fz_g , die eine relative Verringerung bzw. eine absolute Minimierung der jeweils zugehörigen Zielfunktionen zf_g vorschreiben (Negativziele). Die komplementären Meliorisierungs- und Extremierungsziele fz_g , die eine relative Vergrößerung bzw. eine absolute Maximierung der jeweils zugehörigen Zielfunktionen zf_g erfordern (Positivziele), lassen sich analog behandeln. Dazu brauchen lediglich die Relatoren " \leq " und " $<$ " in den o.a. Ungleichungsrelationen durch die Relatoren " \geq " bzw. " $>$ " angepaßt zu werden. Falls innerhalb eines Zielsystems sowohl Negativ- als auch Positivziele vorkommen, ist das Zielsystem in zwei Teilsysteme aufzuspalten, von denen das eine ausschließlich Negativ- und das andere ausschließlich Positivziele umfaßt. Beide Zielsysteme müssen die Ungleichungsrelationen der ersten Art mit dem Relator " \leq " bzw. " \geq " erfüllen. Dagegen braucht nur eines der beiden Zielsysteme die Ungleichungsrelationen der zweiten Art mit dem Relator " $<$ " bzw. " $>$ " gerecht zu werden. Schließlich können Satisfizierungsziele von vornherein vernachlässigt werden. Denn durch die Voraussetzung formal zulässiger Schaltprozesse werden sie ohnehin schon erfüllt.

Ein formal zulässiger Schaltprozeß dominiert einen anderen, ebenso formal zulässigen Schaltprozeß genau dann, wenn der erste Schaltprozeß einen Entscheidungsträger echt besser stellt als der zweite Schaltprozeß. Der zweite Schaltprozeß heißt dann ein dominierter (ineffizienter oder inferiorer) Schaltprozeß. Ein formal zulässiger Schaltprozeß $\text{PRO}_{0,E/e,p}$ verhält sich dagegen nicht-dominiert (effizient oder superior), falls unter denselben Voraussetzungen kein echt besserstellender anderer formal zulässiger Schaltprozeß $\text{PRO}_{0,E/e',p'}$ existiert. Dieser Sachverhalt wird durch das Prädikat $\text{NDOM} \leq (\text{PRO}_{0,E/e,p})$ ausgedrückt, für das gilt:

$$\begin{aligned} & \text{NDOM} \leq (\text{PRO}_{0,E/e,p}) \\ \Leftrightarrow & \text{PRO}_{0,E/e,p} \in \text{ZFPROM} \wedge (\forall (\text{PRO}_{0,E/e',p'} \in \text{ZFPROM}): \neg \text{BES}_\leq(\text{PRO}_{0,E/e',p'}, \text{PRO}_{0,E/e,p})) \end{aligned}$$

Vgl. dazu FANDEL (1991a), S. 50, und FANDEL (1991b), S. 234 (allerdings in bezug auf Technologien); am Rande auch ZELEWSKI (1986a), S. 685. Wenn die oben erläuterte Differenzierung zwischen Negativ- und Positivzielen nicht erheblich ist, kann das Prädikat der Nichtdominanz auch als $\text{NDOM}(\text{PRO}_{0,E/e,p})$ notiert werden.

19) Die formale Zulässigkeit finaler Schaltprozesse wird fortan - wenn keine ausdrücklich abweichenden Festlegungen erfolgen - unterstellt. Daher braucht auf diese Prozeßeigenschaft nicht mehr explizit hingewiesen zu werden. Es kann aber weiterhin geschehen, wenn es der Deutlichkeit halber angeraten erscheint.

20) Dies folgt daraus, daß die Eigenschaft der Nichtdominanz in der voranstehenden Anmerkung in bezug auf *alle* formal (und sachlich) zulässigen finalen Schaltprozesse eines Netzes definiert wurde.

21) Aus dem Dominanzkonzept der Vektoroptimierung folgt für jedes degenerierte Formalzielsystem mit nur genau einem Extremierungsziel $fz_1 = fz_G$, daß jeder optimale Schaltwert \underline{sw}^* dieses eine Formalziel mit $\underline{sw}^* = (sw_1^*)$ - je nach erwünschter Extremierungsrichtung des Formalziels - entweder minimal oder maximal erfüllt. Für ein Extremierungsziel kann aber immer nur (höchstens) ein Minimum bzw. Maximum existieren. Deshalb ist das Netzoptimum \underline{sw}^* in diesem Fall das eindeutig determinierte Extremum des einen Formalziels. Die Eindeutigkeit gilt allerdings nur für das Netzoptimum \underline{sw}^* selbst. Es ist nicht ausgeschlossen, daß mehrere verschiedene finale Schaltprozesse $\text{PRO}_{0,E/e,p}$ existieren, die denselben optimalen Schaltwert $\underline{sw}(\text{PRO}_{0,E/e,p}) = \underline{sw}^*$ besitzen.

22) Da in der nachfolgenden Definitionsformel Existenz- anstelle von Einsquantoren verwendet werden, kann auf den identifizierenden Teilindex "p" für die Schaltprozesse verzichtet werden. Denn es ist für die Definition der Netzoptimalität unerheblich, ob ein Netzoptimum \underline{sw}^* als Schaltwert von nur einem oder aber mehreren optimalen Schaltprozessen $\text{PRO}_{0,E/e}(\text{SF}_1, \text{MF}_1)$ realisiert wird.

23) Vgl. zum hier zugrundegelegten Verständnis des Terminierens von Netzen GOSTELOW (1975), S. 349; HERZOG, O. (1976c), S. 64; RAMMIG (1978), S. 43; BARZILAI (1979), S. 11; YOELI (1980a), S. 73; ROUCAIROL (1982b), S. 15f.

Abweichende Konzepte für das Terminieren von Netzen - und verwandten dynamischen Systemen - vertreten dagegen CERF (1972), S. XVIII u. 143f.; GOSTELOW (1971), S. 20, 22 u. 113; GOSTELOW (1972), S. 20; BYRN (1974), S. II-86, II-171 u. III-42; LÖHR (1974), S. 64; CONRY (1979a), S. 254.

Der Sachverhalt des Terminierens wird zumeist im Zusammenhang mit temporalen Logiken untersucht. Mitunter wird es auch bei der Erörterung von modalen Logiken angesprochen. Das Terminieren eines beliebigen Systems, das keineswegs ein Netz darstellen muß, bedeutet die Endlichkeit des Systemverhaltens in der Anschauungsform Zeit. Mit präziseren Worten heißt das: Ein System terminiert bezüglich einer vorgegebenen Terminierungsbedingung genau dann, wenn nachgewiesen werden kann, daß jedes zulässige Systemverhalten die Terminierungsbedingung nach beliebig großer, aber endlicher Zeit erreicht. Dabei steht die Zeit als *pars pro toto* für beliebige Ressourcen, die beim Ablauf des Systemverhaltens verbraucht wurden. Die Terminierungsbedingungen stellen Abbruchkriterien dar. Sie legen fest, unter welchen - zumeist erwünschten - Umständen ein Systemverhalten als beendet betrachtet wird. Dabei kann es sich z.B. um die Erfüllung eines Sachziels oder um die extreme Erfüllung eines Formalziels handeln. Für ein terminierendes System gilt daher aus einer verallgemeinerten Perspektive: Jedes zulässige Systemverhalten läßt sich mit endlichem Ressourcenverzehr so abwickeln, daß es vorgegebene Abbruchkriterien erfüllt. Die Systemeigenschaft des Terminierens steht daher in einem engen inhaltlichen Zusammenhang mit der allgemeinen Konzeptualisierungsprämisse der Finitheit. Die Untersuchung des Terminierens bietet sich beispielsweise an, um festzustellen, ob ein System mit Sicherheit in vordefinierte Ruhezustände zurückzukehren vermag. Dies spielt aus kybernetischer Perspektive eine Rolle, wenn untersucht wird, ob eine Prozeßregelung entweder Stabilität zu garantieren vermag oder aber durch Aufschaukelungseffekte zu "explodieren" droht. Vgl. ebenso den nachfolgenden Hinweis auf Heimzustände.

24) Dies entspricht genau dem Terminieren von Input/Output-Systemen, das im Zusammenhang mit dem CHURCH/ROSSER-Konzept dargestellt wurde. Hier zeigt sich ein bemerkenswerter Zusammenhang zwischen der Analyse dynamischer Netzeigenschaften mit Analyseproblemen, die mit der Hilfe von Zeitlogiken ausgedrückt sind; vgl. dazu wiederum die vorgenannte Anmerkung.

25) Da die Nullschaltfolge immer geschaltet werden kann, wird nachfolgende Prädikatsdefinition auch dann erfüllt, wenn für den Fall $M_T = M_{E_c}$ mit $M_{E_c} \in RM(M_0)$ Endmarkierungen M_{E_c} betrachtet werden. Diese Möglichkeit wird dagegen explizit ausgeschlossen von HERZOG, O. (1976c), S. 64, ohne dies jedoch näher zu begründen.

26) Ein Heimzustand ist in dynamischen Systemen ein Zustand, der von jedem anderen Systemzustand aus mit Sicherheit in endlicher Zeit erreicht werden kann. Vgl. zu Heimzuständen im Kontext von Petrinetz-Konzept und Transitionssystemen KELLER, R. (1976), S. 379f.; GIRAULT (1980), S. 11; ROUCAIROL (1982b), S. 17ff.; BERTHELOT (1982c), S. 7; BERTHELOT (1982d), S. 25 u. 31; VOSS, K. (1983a), S. 3f.; PAGNONI (1990), S. 143f.

27) Vgl. ROUCAIROL (1982b), S. 17.

28) Es handelt sich dann notwendig um erwünschte Deadlocks. Falls ein terminierendes Netz einen unerwünschten Deadlock enthielte, der per definitionem keine Endmarkierung ist, so könnte von dort aus keine Endmarkierung mehr erreicht werden. Dies widerspräche jedoch der Definition terminierender Netze. Vgl. dazu auch YOELI (1980a), S. 73.

29) Diese Netzkonstruktion verallgemeinert die Anregungen, die sich finden bei BYRN (1974), S. III-42; MERLIN, P. (1976a), S. 618; VALETTE (1977b), S. 105; AGERWALA (1978a), S. 152; VALETTE (1978b), S. 189; AGERWALA (1979), S. 93; YOELI (1980a), S. 73; ROUCAIROL (1982b), S. 16; MONTEL (1983a), S. 295.

30) Darüber hinaus verhält es sich dann auch M_0 -reversibel. Diese Netzeigenschaft wird aber erst im nächsten Kapitel definiert.

31) Vgl. BAER, J. (1973b), S. 19f.; ARAKI (1976b), S. 7 (mit genau einer Endmarkierung); ROUCAIROL (1982b), S. 15f.

32) Andernfalls wäre es möglich, daß die Endmarkierungen zwar von allen erreichbaren Markierungen aus erreicht werden können, aber die dazu gehörenden Schaltprozesse in den Endmarkierungen nicht beendet, sondern fortgesetzt werden. Sie würden dann über die intendierten Endmarkierungen "hinausschießen". Dies läßt sich mit dem Begriff der Endmarkierung durchaus vereinbaren. Er bedeutet nur, daß eine solche Markierung erreicht werden soll, weil sie erwünscht ist. Er sagt jedoch nichts darüber aus, ob diese erwünschte Endmarkierung auch für immer beibehalten werden soll, wenn sie einmal erreicht worden ist. Erst der Deadlockcharakter aller Endmarkierungen stellt dagegen sicher, daß jeder Schaltprozeß in einer Endmarkierung tatsächlich endet, falls er eine Endmarkierung erreicht.

33) Daher kann es keine unerwünschten Deadlocks geben: Jeder Schaltprozeß, der unter einer erreichbaren Markierung gestartet wird, kann beliebig lange fortgesetzt werden, sofern er noch nicht eine der Endmarkierungen erreicht hat.

34) Dadurch wird gewährleistet, daß von jeder erreichbaren Markierung aus eine Endmarkierung tatsächlich erreicht wird, falls der zugrundeliegende Schaltprozeß nur lange genug fortgesetzt wird. Denn es existiert qua Voraussetzung im Erreichbarkeitsgraphen mindestens ein Schaltweg, der einen solchen Schaltprozeß repräsentiert. Ein "Hinausschießen" des Schaltprozesses über eine einmal erreichte Endmarkierung wurde bereits ausgeschlossen. Darüber hinaus wurde sichergestellt, daß sich jeder Schaltprozeß beliebig fortsetzen läßt, solange er noch keine Endmarkierung erreicht hat. Es wäre aber immer noch möglich, daß sich mindestens ein Schaltprozeß, der unter einer erreichbaren Markierung startet, potentiell unendlich fortsetzen läßt und hierbei keine der Endmarkierungen erreicht. Dabei handelt es sich entweder um einen Livelockprozeß, dem im Erreichbarkeitsgraphen des Netzes ein zyklischer Weg entspricht. Oder es liegt ein unbeschränktes Netz mit einem unendlichen Erreichbarkeitsgraphen vor. Dann können auch potentiell unendliche Prozesse ausgeführt werden, die azyklisch verlaufen und dennoch niemals eine der Endmarkierungen mit Deadlockcharakter erreichen. In beiden vorgenannten Fällen muß mindestens eine Transition unendlich oft geschaltet werden, da die Transitionenanzahl eines jeden Netzes als endlich vorausgesetzt wurde. Dies wird durch die o.a. dritte Bedingung verhindert, welche die Existenz potentiell unendlicher Schaltprozesse ausschließt.

35) Daher kann in einem echt terminierenden Netz durch jeden Schaltprozeß jede Transition nur endlich oft geschaltet werden; vgl. CONRY (1979a), S. 254; MEMMI (1979), S. 95f.

5.2.2.2.5 Robustheit, Promptheit und Reversibilität

Abschließend werden drei Eigenschaften von Netzmodellen behandelt, die sich in besonderer Weise auf den Aspekt der Koordinierungsflexibilität konzentrieren. Damit knüpfen sie an den Argumentationshintergrund dieser Arbeit - die Prozeßkoordinierung in Flexiblen Fertigungssystemen - unmittelbar an. Ausgangspunkt ist die zunächst vage Ausdeutung des Flexibilitätsbegriffs als Anpassungsfähigkeit an veränderliche Produktionssituationen. Die Netzeigenschaft der Robustheit stellt den betriebswirtschaftlich vertrautesten Ansatz dar¹⁾, um das Ausmaß dieser Anpassungsfähigkeit konkret zu bestimmen. Er wird aus technischer Hinsicht um die Eigenschaft der Promptheit ergänzt. Sie spielt für Anpassungen unter Realzeitbedingungen eine besondere Rolle, weil sie den Aspekt der Anpassungsgeschwindigkeit in den Vordergrund rückt. Schließlich wird die Reversibilität von Netzmodellen angesprochen. Diese Netzeigenschaft berücksichtigt in besonderer Weise das Anpassungsvermögen gegenüber Produktionsstörungen²⁾.

Bei der Robustheit eines Netzmodells handelt es sich insofern um eine besonders interessante dynamische Modelleigenschaft, als sie im Rahmen des Petrinetz-Konzepts bislang noch nicht thematisiert wurde. Statt dessen entstammt sie der Diskussion über das betriebswirtschaftliche Konzept Robuster Planung³⁾. Auf diese Weise lassen sich Petrinetze um einen originär betriebswirtschaftlichen Beitrag bereichern⁴⁾. Zugleich findet aber auch eine konzeptionelle Befruchtung in der umgekehrten Richtung statt. Denn der betriebswirtschaftliche Robustheitsbegriff stellt bei der Untersuchung eines Koordinierungsproblems nur auf die *Chancen* ab, die darin bestehen, trotz veränderter Produktionssituationen doch noch eine der intendierten Problemlösungen⁵⁾ zu erreichen⁶⁾. Er suggeriert die scheinbar naheliegende, aber tatsächlich fehlerhafte Annahme, zwischen den Chancen des Erreichens und den *Risiken* des Verfehlens von intendierten Problemlösungen bestehe ein einfacher komplementärer Zusammenhang folgender Art: Je größer die Chancen ausfielen, desto geringere Risiken bestünden.

Nur unter dieser Voraussetzung ist es gerechtfertigt, grundsätzlich solche Koordinierungsentscheidungen als "robuste erste Schritte" zu bevorzugen, bei deren Auswahl die Chance besonders hoch ausfällt, schließlich eine der intendierten Problemlösungen zu erreichen. Tatsächlich braucht die prima facie "plausible" Annahme gegenläufiger Lösungschancen und -risiken jedoch nicht zuzutreffen. Denn anhand von Netzmodellen läßt sich aufzeigen, daß Koordinierungsentscheidungen existieren können, bei denen sowohl die Chancen des Erreichens als auch die Risiken des Verfehlens von intendierten Problemlösungen besonders hoch ausfallen⁷⁾. Diese potentielle Gleichläufigkeit von Chancen und Risiken wird durch die nachfolgend entfalteten Robustheitsmaße für Netzmodelle berücksichtigt⁸⁾.

Ausgangspunkt der Robustheitsanalyse eines Netzmodells ist die vollständige Explizierung seiner dynamischen Struktur durch die Prozeßmenge PROM. Sie umfaßt alle Schaltprozesse $PRO_{r,f}(SF_L, MF_L)$ mit $L \in \mathcal{N}_0$, die im betrachteten Netzmodell zulässig sind. Des weiteren wird ein Koordinierungsproblem vorausgesetzt, das aus entscheidungstheoretischer Perspektive durch die Angabe eines Sach- und Formalzielsystems⁹⁾ wohlbestimmt ist¹⁰⁾. Daher wird die Menge aller intendierten Problemlösungen durch jene Schaltprozesse aus der Prozeßmenge PROM konstituiert, die zwei Anforderungen gerecht werden: Erstens müssen sie als finale Schaltprozesse das vorausgesetzte Sachzielsystem erfüllen. Zweitens sollen sie die Satisfizierungs-, Meliorierungs- oder Optimierungsziele des Formalzielsystems in der jeweils zielspezifisch geforderten Weise erfüllen. Alle Schaltprozesse, die diese beiden Postulate einlösen, werden fortan als intendierte Schaltprozesse bezeichnet und in der intendierten Prozeßmenge IPROM zusammengefaßt¹¹⁾. Ihre Definition wird hier für den Fall von Extremierungszielen exemplarisch angeführt¹²⁾. Dazu wird auf die schon früher definierte Menge ZFPROM aller formal zulässigen Schaltprozesse, die eine Teilmenge der Prozeßmenge PROM darstellt, und auf das Prädikat

NDOM($\text{PRO}_{0,E/e}$) der Nichtdominiertheit durch andere formal zulässige Schaltprozesse zurückgegriffen:

$$\text{IPROM} = \{ \text{PRO}_{0,E/e} : \exists (M_{E/e} \in \text{ME}) \exists (L \in \mathcal{N}_0) : \dots \\ \text{PRO}_{0,E/e}(\text{SF}_L, \text{MF}_L) \in \text{ZFPROM} \wedge \text{NDOM}(\text{PRO}_{0,E/e}) \}$$

Auf keinen Fall intendiert werden dagegen alle Schaltprozesse, die in der Ausgangsmarkierung starten und in einer unerwünschten Deadlockmarkierung¹³⁾ enden. Gleiches gilt für alle Schaltprozesse, die ebenso in der Ausgangsmarkierung starten und eine solche Livelockmarkierung erreichen, von der aus kein Schaltprozeß zu einer der erwünschten Endmarkierungen fortführt¹⁴⁾. In beiden vorgenannten Fällen ist es unmöglich, einen der intendierten Schaltprozesse aus der Menge IPROM zu realisieren¹⁵⁾. Die Schaltprozesse, die eine unerwünschte Dead- oder Livelockmarkierung erreichen, werden in der Menge NIPROM aller nicht-intendierten Schaltprozesse¹⁶⁾ zusammengefaßt:

$$\text{NIPROM} = \{ \text{PRO}_{0,r} : \exists (L \in \mathcal{N}_0) : \text{PRO}_{0,r}(\text{SF}_L, \text{MF}_L) \in \text{PROM} \wedge \text{MF}_L = (M_0, \dots, M_r) \wedge \dots \\ ((\text{DEAD}(M_r) \wedge M_r \notin \text{ME}) \vee (\text{LIVE}(M_r) \wedge (\forall (M_{E/e} \in \text{ME}) : M_{E/e} \text{ RM}(M_r)))) \}$$

Es wird eine beliebige Markierung M_{nf} betrachtet, die von der Ausgangsmarkierung M_0 aus erreichbar ist, aber keine der erwünschten Endmarkierungen $M_{E/e}$ darstellt. Sie wird fortan als nonfinale Markierung M_{nf} bezeichnet. Es wird die Auswahl eines Schaltschritts SS_a erwogen, der unter dieser nonfinalen Markierung aktiviert ist. Die Menge aller dort aktivierten Schaltschritte bildet den lokalen Spielraum, der für die Auswahlentscheidung unter der Markierung M_{nf} offensteht. Jeder Schaltschritt aus dieser Menge stellt eine lokale Entscheidungsalternative dar. Es interessiert hier, welche Chancen und Risiken bestehen, durch die Auswahl eines aktivierten Schaltschritts den Entscheidungsspielraum der Markierung M_{nf} so zu schließen, daß - unter Berücksichtigung aller früheren lokalen Auswahlentscheidungen, welche die Markierung M_{nf} aus der Ausgangsmarkierung M_0 hervorgebracht haben, - für die späteren lokalen Auswahlentscheidungen in allen danach erreichbaren Folgemarkierungen gilt: Entweder wird die Möglichkeit offengehalten, durch jene nachfolgenden Auswahlentscheidungen insgesamt noch einen der intendierten Schaltprozesse aus der Menge IPROM zu realisieren (Chancenfall). Oder aber diese Möglichkeit wird ausgeschlossen, weil die nachfolgenden Auswahlentscheidungen insgesamt nur noch einen der nicht-intendierten Schaltprozesse aus der Menge NIPROM verwirklichen können (Risikofall). Im Chancenfall besteht weiterhin Aussicht, bei Fortsetzung des Schaltprozesses im Netz eine der intendierten Lösungen des modellierten Koordinierungsproblems zu verwirklichen. Im Risikofall geht dagegen diese Option, eine intendierte Problemlösung zukünftig zu realisieren, endgültig unter.

Die vorgenannten Chancen und Risiken lassen sich für jeden Schaltschritt SS_a , der unter einer nonfinalen Markierung M_{nf} aktiviert ist, präzise bestimmen. Die Chancen¹⁷⁾ gründen in den Schaltprozessen aus der intendierten Prozeßmenge IPROM. Es bestehen genau so viele Chancen, wie Schaltprozesse $\text{PRO}_{\text{nf},E/e}(\text{SF}_{L/e}, \text{MF}_{L/e})$ existieren, für die jeweils gilt:

- Der Schaltprozeß $\text{PRO}_{\text{nf},E/e}(\text{SF}_{L/e}, \text{MF}_{L/e})$ startet in der nonfinalen Markierung M_{nf} durch das Ausführen des Schaltschritts SS_a .
- Der Schaltprozeß $\text{PRO}_{\text{nf},E/e}(\text{SF}_{L/e}, \text{MF}_{L/e})$ endet in einer der erwünschten Endmarkierungen $M_{E/e}$ aus der Endmarkierungsmenge ME.

- Wenn der Schaltprozeß $PRO_{0,nf}(SF_{L/0},MF_{L/0})$, der von der Ausgangsmarkierung M_0 des Netzmodells zur nonfinalen Markierung M_{nf} geführt hat, und der Schaltprozeß $PRO_{nf,E/e}(SF_{L/e},MF_{L/e})$ hintereinander ausgeführt werden, dann stellt der resultierende Gesamtprozeß $PRO_{0,E/e}(SF_{L/G},MF_{L/G})$ ¹⁸⁾ einen intendierten Schaltprozeß aus der Menge IPROM dar¹⁹⁾.

Entsprechend wird das Risikoausmaß auf Schaltprozesse aus der nicht-intendierten Prozeßmenge NIPROM zurückgeführt. Das Risiko ist genau so groß, wie Schaltprozesse $PRO_{nf,r}(SF_{L/r},MF_{L/r})$ existieren, für die jeweils gilt:

- Der Schaltprozeß $PRO_{nf,r}(SF_{L/r},MF_{L/r})$ startet in der nonfinalen Markierung M_{nf} durch das Ausführen des Schaltschritts SS_a .
- Der Schaltprozeß $PRO_{nf,r}(SF_{L/r},MF_{L/r})$ endet entweder in einer unerwünschten Deadlockmarkierung M_r oder erreicht eine ebenso unerwünschte Livelockmarkierung M_r .
- Wenn der Schaltprozeß $PRO_{0,nf}(SF_{L/0},MF_{L/0})$, der von der Ausgangsmarkierung M_0 des Netzmodells zur nonfinalen Markierung M_{nf} geführt hat, und der Schaltprozeß $PRO_{nf,r}(SF_{L/r},MF_{L/r})$ hintereinander ausgeführt werden, dann stellt der resultierende Gesamtprozeß $PRO_{0,r}(SF_L,MF_L)$ ²⁰⁾ einen nicht-intendierten Schaltprozeß aus der Menge NIPROM dar²¹⁾.

Es ist keineswegs ausgeschlossen, daß durch dasselbe Paar (M_{nf},SS_a) aus der nonfinalen Markierung M_{nf} und dem dort aktivierten Schaltschritt SS_a sowohl mindestens ein intendierter als auch mindestens ein nicht-intendierter Schaltprozeß verlaufen. Daher kann dieselbe Auswahlentscheidung, die im lokalen Spielraum einer Markierung M_{nf} zugunsten eines Schaltschritts SS_a erfolgt, sowohl die Chance bedeuten, eine der intendierten Problemlösungen noch zu verwirklichen, als auch das Risiko, dazu nicht mehr in der Lage zu sein²²⁾.

Chancen und Risiken jeder lokalen Auswahlentscheidung werden durch die beiden partiellen²³⁾ Funktionen "chance" bzw. "risiko" quantifiziert. Dabei wird eine normierte Darstellungsweise benutzt. Sie bezieht die Anzahl aller intendierten oder nicht-intendierten Schaltprozesse, die das Paar (M_{nf},SS_a) umfassen, auf die Anzahl aller intendierten bzw. nicht-intendierten Schaltprozesse, die im untersuchten Netz überhaupt definiert sind²⁴⁾. Aufgrund der voranstehenden Erläuterungen gilt dafür mit $RM(M_0)$ als Menge aller erreichbaren Markierungen und SSM als Menge aller Schaltschritte, die für ein Netz definiert sind:

$$\text{chance: } RM(M_0) \times SSM \rightarrow [0;1]$$

$$(M_{nf},SS_a) \rightarrow \text{chance}(M_{nf},SS_a); \text{ falls } M_{nf} \in (RM(M_0) - ME) \wedge \text{AKT}(SS_a, M_{nf})$$

mit:

$$\text{chance}(M_{nf},SS_a) = \#(\{PRO_{nf,E/e}(SF_{L/e},MF_{L/e})\}) : \#(IPROM)$$

und:

$$\{PRO_{nf,E/e}(SF_{L/e},MF_{L/e})\} = \dots$$

$$\{PRO_{nf,E/e}(SF_{L/e},MF_{L/e}) : PRO_{nf,E/e}(SF_{L/e},MF_{L/e}) \in PROM \wedge \dots$$

$$((SF_{L/e} = (SS_{a(1/e)}, \dots, SS_{a(L/e)}) \wedge MF_{L/e} = (M_{r(0/e)}, M_{r(1/e)}, \dots, M_{r(L/e)}))$$

$$\rightarrow (SS_{a(1/e)} = SS_a \wedge M_{r(0/e)} = M_{nf}$$

$$\wedge (\exists (PRO_{0,E/e}(SF_{L/G},MF_{L/G}) \in IPROM) : \dots$$

$$SF_{L/G} = (SS_{a(1/G)}, \dots, SS_{a(1/e)}, \dots, SS_{a(L/e)}) \wedge \dots$$

$$MF_{L/G} = (M_0, \dots, M_{r(0/e)}, M_{r(1/e)}, \dots, M_{r(L/e)}))\}$$

$$\begin{aligned}
\text{risiko: } & \text{RM}(M_0) \times \text{SSM} \rightarrow [0;1] \\
& (M_{\text{nf}}, \text{SS}_a) \rightarrow \text{risiko}(M_{\text{nf}}, \text{SS}_a); \text{ falls } M_{\text{nf}} \in (\text{RM}(M_0) - \text{ME}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}}) \\
\text{mit:} & \\
\text{und:} & \text{risiko}(M_{\text{nf}}, \text{SS}_a) = \#(\{\text{PRO}_{\text{nf.r}}(\text{SF}_{L/r}, \text{MF}_{L/r})\}) : \#(\text{NIPROM}) \\
& \{\text{PRO}_{\text{nf.r}}(\text{SF}_{L/r}, \text{MF}_{L/r})\} = \dots \\
& \quad \{\text{PRO}_{\text{nf.r}}(\text{SF}_{L/r}, \text{MF}_{L/r}) : \text{PRO}_{\text{nf.r}}(\text{SF}_{L/r}, \text{MF}_{L/r}) \in \text{PROM} \wedge \dots \\
& \quad ((\text{SF}_{L/r} = (\text{SS}_{a(1/r)}, \dots, \text{SS}_{a(L/r)}) \wedge \text{MF}_{L/r} = (\text{M}_{r(0/r)}, \text{M}_{r(1/r)}, \dots, \text{M}_{r(L/r)})) \\
& \quad \rightarrow (\text{SS}_{a(1/r)} = \text{SS}_a \wedge \text{M}_{r(0/r)} = M_{\text{nf}} \\
& \quad \wedge (\exists (\text{PRO}_{0,r}(\text{SF}_{L/G}, \text{MF}_{L/G}) \in \text{NIPROM}) : \dots \\
& \quad \text{SF}_{L/G} = (\text{SS}_{a(1/G)}, \dots, \text{SS}_{a(1/r)}, \dots, \text{SS}_{a(L/r)}) \wedge \dots \\
& \quad \text{MF}_{L/G} = (M_0, \dots, \text{M}_{r(0/r)}, \text{M}_{r(1/r)}, \dots, \text{M}_{r(L/r)}))\}
\end{aligned}$$

Die Robustheit einer lokalen Entscheidung, die aus dem Spielraum einer nonfinalen Markierung M_{nf} einen dort aktivierten Schaltschritt SS_a auswählt, läßt sich nunmehr innerhalb eines Netzmodells präzisieren. Sie fällt zunächst aus der Perspektive des vertrauten betriebswirtschaftlichen Robustheitsbegriffs um so größer aus, je höher die Chancen stehen, aufgrund dieser Entscheidung eine der intendierten Problemlösungen für das modellierte Koordinierungsproblem zu verwirklichen. Genau dies mißt die Funktion "chance". Robustheitsmindernd wirkt sich jedoch zugleich der komplementäre Risikoaspekt aus, daß die Auswahlentscheidung auch zu einer unerwünschten Dead- oder Livelockmarkierung führen und somit die Realisierung einer intendierten Problemlösung verhindern kann. Daher muß der Entscheidungsträger zwischen den lösungsbezogenen Chancen und Risiken aller Schaltschritte abwägen, die im lokalen Spielraum einer nonfinalen Netzmarkierung zur Auswahl stehen.

Die diffizile Problematik, die Einstellung des Entscheidungsträgers gegenüber - zunächst unsicheren²⁵⁾ - Chancen und Risiken formal adäquat wiederzugeben, wird keineswegs verkannt²⁶⁾. Aber es besteht kein Interesse daran, sie in dieser Arbeit weiter zu vertiefen. Statt dessen beschränkt sich der Verf. auf eine einfache Formalisierung der Unsicherheitspräferenz²⁷⁾. Sie besteht aus zwei Parametern α_c und α_r für die Chancen- bzw. die Risikoneigung des Entscheidungsträgers²⁸⁾. Für jeden Parameter wird der normierte Ausprägungsbereich $[0;1]$ vereinbart. Je größer einer dieser Parameter ausfällt, in desto größerem Ausmaß nimmt der Entscheidungsträger die objektiv²⁹⁾ definierten Chancen oder Risiken seiner Auswahlentscheidung subjektiv wahr. Die Parametersumme wird auf den Einheitswert "1" normiert³⁰⁾.

Die Robustheit der Entscheidung, einen Schaltschritt SS_a unter einer nonfinalen Netzmarkierung M_{nf} auszuwählen, wird durch die partielle Funktion "rob" gemessen³¹⁾. Sie beurteilt die normierten Entscheidungschancen und -risiken auf der Basis der Unsicherheitspräferenz des Entscheidungsträgers³²⁾:

$$\begin{aligned}
\text{rob: } & \text{RM}(M_0) \times \text{SSM} \rightarrow [-1;+1] \\
& (M_{\text{nf}}, \text{SS}_a) \rightarrow \text{rob}(M_{\text{nf}}, \text{SS}_a); \text{ falls } M_{\text{nf}} \in (\text{RM}(M_0) - \text{ME}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}}) \\
\text{mit:} & \\
& \text{rob}(M_{\text{nf}}, \text{SS}_a) = \alpha_c \cdot \text{chance}(M_{\text{nf}}, \text{SS}_a) - \alpha_r \cdot \text{risiko}(M_{\text{nf}}, \text{SS}_a)
\end{aligned}$$

Die Auswahl eines Schaltschritts SS_a , der unter einer nonfinalen Markierung M_{nf} aktiviert ist, wird als ein robuster Schritt bezeichnet³³⁾, falls im selben lokalen Entscheidungsspielraum keine Auswahlentscheidung mit größerer Robustheit getroffen werden kann. Es handelt sich daher um eine Auswahlentscheidung mit maximaler Robustheit³⁴⁾. Sie wird durch das Prädikat

$ROB(M_{nf}, SS_a)$ notiert. Diese Auswahlentscheidung kann nur unter einer lebendigen Markierung erfolgen. Denn nur dann ist mindestens eine Transition aktiviert. Andernfalls wäre überhaupt kein aktivierter Schaltschritt definiert, so daß auch kein Spielraum für eine Auswahlentscheidung offenstünde. Daher gilt für das Robustheitsprädikat³⁵⁾:

$$ROB(M_{nf}, SS_a) \\ \Leftrightarrow M_{nf} \in (RM(M_0) - ME) \wedge LEB(M_{nf}) \wedge AKT(SS_a, M_{nf}) \wedge \dots \\ rob(M_{nf}, SS_a) = \max \{ rob(M_{nf}, SS_k) : SS_k \in SSM \wedge AKT(SS_k, M_{nf}) \}$$

Falls es sich bei einer betrachteten nonfinalen und lebendigen Markierung M_{nf} um die Ausgangsmarkierung M_0 des untersuchten Netzes handelt, kann ein robuster "erster Schritt"³⁶⁾ ausgewählt werden. Ein solcher Schritt erfolgt genau dann, wenn unter der Ausgangsmarkierung M_0 ein Schaltschritt SS_a selektiert wird, der das Prädikat $ROB(M_0, SS_a)$ erfüllt.

Jeder erreichbaren Markierung M_r eines Netzes wird ein markierungsspezifischer Robustheitswert $mrob(M_r)$ zugeordnet. Für jede nonfinale und lebendige Markierung wird er mit der Robustheit $rob(M_r, SS_a)$ der dort möglichen robusten Auswahlentscheidungen identifiziert³⁷⁾. Unter einer nonfinalen und toten Markierung ist überhaupt kein Schaltschritt aktiviert. A fortiori ist dort die Robustheit einer Auswahlentscheidung ebenso wenig definiert. Da es sich notwendig um eine unerwünschte Deadlockmarkierung handelt, wird ihr Robustheitswert zu $mrob(M_r) = -1$ festgelegt³⁸⁾. Entsprechend wird für alle finalen Netzmarkierungen der Robustheitswert zu $mrob(M_r) = +1$ vereinbart. Daher gilt für alle erreichbaren Markierungen eines Netzes die - nunmehr vollständig definierte³⁹⁾ - Robustheitsfunktion "mrob" mit⁴⁰⁾:

$$mrob: RM(M_0) \rightarrow [-1; +1]$$

mit:

$$M_r \rightarrow mrob(M_r) = \begin{cases} +1; & \text{falls } M_r \in ME \\ rob(M_r, SS_a); & \text{falls } M_r \in (RM(M_0) - ME) \\ & \wedge LEB(M_r) \wedge ROB(M_r, SS_a) \\ -1; & \text{falls } M_r \in (RM(M_0) - ME) \wedge DEAD(M_r) \end{cases}$$

Die Robustheit eines gesamten Netzes kann zunächst durch den Vektor $\underline{grob}(SN)$ der Robustheitswerte $mrob(M_r)$ seiner erreichbaren Markierungen definiert werden. Hierdurch wird das Potential robuster Entscheidungen, die in den lokalen Entscheidungsspielräumen des Netzmodells getroffen werden können, vollständig abgebildet:

$$\underline{grob}(SN) = (mrob(M_r) : M_r \in RM(M_0))$$

Statt dessen können auch unterschiedliche skalare Kennzahlen für die Robustheit eines Netzes gewonnen werden. Sie verdichten jeweils die Robustheitsinformation des Vektors $\underline{grob}(SN)$, indem die markierungsspezifischen Robustheitswerte aggregiert werden. Eine naheliegende Aggregationsmöglichkeit⁴¹⁾ besteht darin, die skalare Robustheit eines Netzmodells mit der Robustheit $mrob(M_0)$ seiner Ausgangsmarkierung gleichzusetzen⁴²⁾:

$$grob_0(SN) = mrob(M_0)$$

In diesem Fall wird von der initialen Netzrobustheit $grob(SN)$ gesprochen. Sie gibt die Robustheit aller Auswahlentscheidungen an, die als robuste erste Schritte im betrachteten Netz beim

Start eines Schaltprozesses - aus der Perspektive des Robustheitskriteriums - bestmöglich getroffen werden können.

Unterschiedliche Netze können hinsichtlich ihrer Robustheit miteinander verglichen werden. Mit einem skalar verdichteten Robustheitsmaß wie der initialen Netzrobustheit $\text{grob}_0(\text{SN})$ ist dies für beliebige Netze möglich. Der Robustheitsvektor $\text{grob}(\text{SN})$ läßt dagegen nur einen Vergleich zwischen Netzen mit identischen Erreichbarkeitsmengen zu. Solche netzbezogenen Robustheitskennzahlen erlauben es, die relative Robustheit alternativer Netzmodelle zu beurteilen, die jeweils unterschiedliche Konzeptualisierungen des gleichen, gemeinsam zugrundeliegenden realen Koordinierungsproblems darstellen.

Die zuvor erläuterte Robustheit von Netzen beleuchtete nur die Fähigkeit zur Anpassung an verschiedene Produktionssituationen. Die Anpassungsgeschwindigkeit spielte dabei keine Rolle. Dieser Aspekt tritt dagegen bei der Netzeigenschaft der Promptheit⁴³⁾ in den Vordergrund. Sie ist von besonderem Interesse, wenn ein Netzmodell die Koordinierung eines realen Systems im Realzeitbetrieb unterstützen soll⁴⁴⁾. Es wird dann von einem Koordinierungsnetz gesprochen⁴⁵⁾. Das reale System und das Koordinierungsnetz werden durch den wechselseitigen Austausch von Nachrichten miteinander gekoppelt. Die Nachrichten informieren in der einen Richtung das Koordinierungsnetz über Ereignisse im realen System (Systemmeldungen). In der entgegengesetzten Richtung leiten sie Anweisungen aus dem Koordinierungsnetz an das reale System weiter, die dort auszuführende Aktionen anstoßen (Steuerungsanweisungen). Im Netz wird das Eintreffen einer Systemmeldung SK_i ⁴⁶⁾ mit $i \in \{1, \dots, I\}$ und $I \in \mathcal{N}_+$ durch das Schalten einer ausgezeichneten Meldetransition t_{SK_i} dargestellt. Dem Absenden einer Steuerungsanweisung, mit der auf die Systemmeldung SK_i geantwortet wird, entspricht das Schalten einer korrespondierenden Anweisungstransition t_{KS_i} . Es wird vorausgesetzt, daß diese beiden Transitionen niemals zusammenfallen: $t_{\text{SK}_i} \neq t_{\text{KS}_i}$ für alle $i \in \{1, \dots, I\}$.

Ein Schaltprozeß $\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L)$, der im Koordinierungsnetz ausgeführt wird, um auf eine eintreffende Systemmeldung mit der entsprechenden Steuerungsanweisung zu reagieren, wird als ein Reaktionsprozeß bezeichnet. Er beginnt mit einem Schaltschritt, der die Meldetransition t_{SK_i} umfaßt ($\text{SS}_{a(1)} \supseteq \{t_{\text{SK}_i}\}$), und endet mit einem Schaltschritt, der die Anweisungstransition t_{KS_i} enthält ($\text{SS}_{a(L)} \supseteq \{t_{\text{KS}_i}\}$). Die Schaltschrittzahl L stellt die Reaktionsspanne des Reaktionsprozesses $\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L)$ dar. Da Melde- und Anweisungstransitionen immer verschieden sind, muß jeder Reaktionsprozeß aus einer nicht-degenerierten Schaltfolge SF_L aus mindestens zwei Schaltschritten bestehen. Daher wird stets $L \geq 2$ erfüllt.

Die Reaktionsspanne R_i des Koordinierungsnetzes ist die maximale Schaltschrittzahl L aller Reaktionsprozesse, mit denen das Koordinierungsnetz auf Systemmeldungen SK_i im ungünstigsten Fall zu reagieren vermag. Dabei werden alle erreichbaren Markierungen M_r des Koordinierungsnetzes einbezogen, unter denen die Meldetransition t_{SK_i} aktiviert ist, und alle von dort aus ausführbaren Reaktionsprozesse, die mit dem Schalten der korrespondierenden Anweisungstransition t_{KS_i} enden. Die Reaktionsspanne kann auch unendlich groß sein⁴⁷⁾. Daher gilt im allgemeinen $R_i \in (\mathcal{N}_+ \setminus \{\omega\})$ und $R_i \geq 2$. Ein Koordinierungsnetz heißt R_i -prompt⁴⁸⁾, wenn es für die Systemmeldung SK_i eine endliche Reaktionsspanne R_i mit $R_i \in \mathcal{N}_+$ besitzt. Sie gestattet es, von jeder erreichbaren Markierung aus, unter der durch das Schalten der Transition t_{SK_i} eine Systemmeldung SK_i eintrifft, nach höchstens R_i Schaltschritten mit der zugehörigen Steuerungsanweisung zu antworten. Ein Koordinierungsnetz wird als R -prompt bezeichnet, wenn die größte Reaktionsspanne R_i für alle definierten Systemmeldungen SK_i den endlichen Wert R annimmt: $R = \max \{R_i; i = 1, \dots, I\}$ und $R \in \mathcal{N}_+$ ⁴⁹⁾.

Im Regelfall ist ein Netz, das die Koordinierung eines realen Systems im Realzeitbetrieb leisten soll, auf einem Automatischen Informationssystem implementiert. Dann läßt sich durch Koordinierungsexperimente die Größenordnung der Verarbeitungszeit bestimmen, die für das Ausführen eines Schaltschritts im Koordinierungsnetz erforderlich ist. Aus dieser Kenntnis folgt

unmittelbar, daß die Größenordnung der Reaktionszeit eines R-prompten Koordinierungsnetzes im ungünstigsten Fall das R-fache der schaltschrittbezogenen Verarbeitungszeit beträgt. Diese Information gestattet eine erste grobe Einschätzung der Realzeittauglichkeit von Koordinierungsnetzen.

Von einem stark prompten⁵⁰⁾ Koordinierungsnetz wird gesprochen, wenn bezüglich aller Systemmeldungen SK_i mit $i \in \{1, \dots, I\}$ mindestens eine natürliche Zahl R existiert, für die sich das Netz R-prompt verhält. Die starke Promptheit unterscheidet sich also von der R-Promptheit nur dadurch, daß die Reaktionsspanne R nicht explizit angegeben wird. Die starke Promptheit eines Koordinierungsnetzes kann durchaus eine interessante Eigenschaft darstellen. Denn oftmals ist die genaue Kenntnis der Reaktionsspanne R nicht sehr aussagekräftig. Sie stellt nur einen groben Maßstab für die Größenordnung der tatsächlichen Reaktionszeit eines Koordinierungsnetzes dar. Dagegen kann es eine erhebliche Rolle spielen, ob für die Reaktionszeit überhaupt eine obere Schranke ihrer Größenordnung existiert⁵¹⁾.

Eine deutlich abweichende Variante der Netzpromptheit stellt dagegen die Eigenschaft der schwachen Promptheit⁵²⁾ dar. Ihr zufolge muß lediglich garantiert sein, daß ein Koordinierungsnetz nach *endlich* vielen Schaltschritten auf das Eintreffen einer beliebigen Systemmeldung mit dem Absenden einer Steuerungsanweisung zu reagieren vermag⁵³⁾. Dabei wird aber nicht gefordert, daß für alle erreichbaren Markierungen des Netzes und für alle definierten Systemmeldungen eine gemeinsame Reaktionsspanne R existiert. Statt dessen darf für jede Netzmarkierung M_r und jede Systemmeldung SK_i die obere Schranke für die Schaltschrittzahl, die zur Erzeugung einer Netzreaktion erforderlich ist, einen anderen Wert $R_{i,r}$ annehmen. Sofern es sich um ein unbeschränktes Koordinierungsnetz handelt, kann sein Erreichbarkeitsgraph unendlich viele Markierungsknoten M_r besitzen. Dann läßt sich nicht mehr ausschließen, daß mindestens eine der endlichen, aber systemmeldungs- und markierungsspezifischen Reaktionsspannen $R_{i,r}$ über jede obere Schranke hinauswächst⁵⁴⁾. Deshalb kann keine obere Schranke R_i für die Anzahl der Schaltschritte garantiert werden, die zwischen dem Eintreffen einer Systemmeldung SK_i und dem Absenden der zugehörigen Steuerungsanweisung im Netz ausgeführt werden. Folglich droht die reale Reaktionszeit des Koordinierungsnetzes für mindestens eine Systemmeldung SK_i jede endliche Schranke zu überschreiten. Daher sind schwach prompte Netze für die Koordination von realen Systemen im Realzeitbetrieb grundsätzlich nicht geeignet. Gleiches gilt a fortiori für alle Koordinierungsnetze, die noch nicht einmal schwach prompt sind, also keine der drei vorgenannten Promptheitsvarianten erfüllen.

Die Netzeigenschaft der Promptheit interessiert in dieser Arbeit aus der Perspektive der Realzeitkoordination realer Systeme. Daher werden nur R-prompte und stark prompte Netze unter der Klasse der prompten Netze subsumiert. Alle anderen Netze - also auch die schwach prompten - werden dagegen als unprompt bezeichnet⁵⁵⁾. Eine hinreichende⁵⁶⁾ Bedingung für die Unpromptheit eines Netzes ist vor allem die Existenz einer Livelockmarkierung, die zwei Bedingungen erfüllt: Sie gehört erstens zur Markierungsfolge, die vom Reaktionsprozeß zwischen den Schaltakten einer Melde- und einer korrespondierenden Anweisungstransition hervorgebracht wird. Zweitens muß die Livelockmarkierung mindestens einen Livelockprozeß definieren, der nicht mit dem Reaktionsprozeß zusammenfällt. Dann kann sich das Netzverhalten nach dem Eintreffen einer Systemmeldung in einem zyklischen Schaltprozeß verfangen⁵⁷⁾. Dabei wird der Schaltzyklus des Livelockprozesses beliebig oft wiederholt, ohne daß die erforderliche Steuerungsanweisung erzeugt und abgesandt würde.

Die voranstehenden Erläuterungen zur Anpassungsgeschwindigkeit von Netzen werden durch die Prädikate $PMT_{R,i}(SN)$, $PMT_R(SN)$, $PMT_S(SN)$ und $PMT_s(SN)$ formal präzisiert. Sie drücken die R_i -Promptheit, die R-Promptheit, die starke bzw. die schwache Promptheit eines Koordinierungsnetzes SN aus. Für sie gilt:

$$\begin{aligned}
& \text{PMT}_{R_i}(\text{SN}) \\
:\Leftrightarrow & \quad (\exists(R_i \in \mathcal{N}_+) \forall(M_r \in \text{RM}(M_0)) \forall(\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L) \in \text{PROM}): \dots \\
& \quad (\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L) = (M_r, \text{SS}_{a(1)}, \dots, \text{SS}_{a(L)}, M_f) \\
& \quad \wedge (\text{SS}_{a(1)} \supseteq \{t_{\text{SK},i}\}) \wedge (\text{SS}_{a(L)} \supseteq \{t_{\text{KS},i}\})) \\
\rightarrow & \quad R_i = \max \{L: \text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L)\}
\end{aligned}$$

$$\begin{aligned}
& \text{PMT}_R(\text{SN}) \\
:\Leftrightarrow & \quad (\forall(i \in \{1, \dots, I\}): \text{PMT}_{R_i}(\text{SN}) \wedge R = \max \{R_i: i = 1, \dots, I\})
\end{aligned}$$

$$\begin{aligned}
& \text{PMT}_S(\text{SN}) \\
:\Leftrightarrow & \quad (\exists(R \in \mathcal{N}_+): \text{PMT}_R(\text{SN}))
\end{aligned}$$

$$\begin{aligned}
& \text{PMT}_{S_i}(\text{SN}) \\
:\Leftrightarrow & \quad (\forall(M_r \in \text{RM}(M_0)) \exists(R_{i,r} \in \mathcal{N}_+) \forall(\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L) \in \text{PROM}): \dots \\
& \quad (\text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L) = (M_r, \text{SS}_{a(1)}, \dots, \text{SS}_{a(L)}, M_f) \\
& \quad \wedge (\text{SS}_{a(1)} \supseteq \{t_{\text{SK},i}\}) \wedge (\text{SS}_{a(L)} \supseteq \{t_{\text{KS},i}\})) \\
\rightarrow & \quad R_{i,r} = \max \{L: \text{PRO}_{r,f}(\text{SF}_L, \text{MF}_L)\}
\end{aligned}$$

Robustheit und Promptheit eines Netzmodells bezogen sich auf die Anpassungsfähigkeit bzw. -geschwindigkeit eines Koordinierungskonzepts gegenüber veränderlichen Produktionssituationen. Der Charakter der betroffenen Produktionssituationen spielte dabei keine Rolle. Im Gegensatz dazu wird nunmehr der Standpunkt eingenommen, daß eine flexible Prozeßkoordination insbesondere auch Anpassungen an Produktionsstörungen erlauben soll⁵⁸⁾. Bei Untersuchungen dieser Störfallflexibilität werden Produktionsstörungen entweder als unerwünschte Ereignisse in oder als unerwünschte Zustände von einzelnen Komponenten⁵⁹⁾ des jeweils modellierten Produktionssystems erfaßt. Unter diesen Voraussetzungen läßt sich die Modellierung von Anpassungsmaßnahmen⁶⁰⁾ und die Feststellung ihres Anpassungserfolgs wesentlich vereinfachen, indem für jede störungsanfällige Komponente des Produktionssystems ein wohldefinierter, störungsfreier Ausgangszustand⁶¹⁾ festgelegt wird. Dann gilt für die Flexibilität der Prozeßkoordination gegenüber Störungen der jeweils betrachteten Systemkomponente: Eine "vollflexible"⁶²⁾ Koordinierung, die sich an alle Störungen der Komponente anzupassen vermag, ist immer dann möglich, wenn aus jedem Zustand der Systemkomponente in deren störungsfreien Ausgangszustand zurückgekehrt werden kann⁶³⁾.

Die gleiche gesicherte Rückkehrfähigkeit kann auch auf ein modelliertes Produktionssystem als Ganzes bezogen werden: Ein Produktionssystem gestattet eine vollflexible Prozeßkoordination, falls sich aus allen Systemzuständen durch entsprechende Anpassungsmaßnahmen zum störungsfreien Ausgangszustand des Produktionssystems zurückkehren läßt. Ein Produktionssystem, das eine solche vollflexible Koordinierung seiner Produktionsprozesse ermöglicht, erweist sich in einem besonderen, störfallbezogenen Sinne als robust⁶⁴⁾: Es kann durch koordinierende Eingriffe an alle Störungen so angepaßt werden, daß es über die Rückkehr zum störungsfreien Ausgangszustand immer wieder möglich ist, die Ausführung der - vorübergehend gestörten - Produktionsprozesse fortzusetzen. Die Prozeßausführung läßt sich im Prinzip unbegrenzt fortsetzen, solange die intendierten Sachziele noch nicht erfüllt sind und sofern alle

tatsächlich möglichen Produktionsstörungen im Modell des Produktionssystems erfaßt wurden⁶⁵).

Die garantierte Rückkehrmöglichkeit zu einem ungestörten Ausgangszustand wird in Netzmodellen durch die Eigenschaft der M_f -Reversibilität⁶⁶ präzisiert. Dabei spielt es keine Rolle, ob das untersuchte Netz entweder das Gesamtmodell eines ganzen Produktionssystems oder aber das Teilmodell einer Systemkomponente darstellt. Darüber hinaus wird die betrachtete Netzmarkierung M_f zunächst keiner besonderen Qualifizierung unterzogen. Unter diesen Voraussetzungen gilt: Ein Netz SN verhält sich genau dann M_f -reversibel, wenn es möglich ist, von jeder erreichbaren Markierung M_r aus die ausgezeichnete Markierung M_f durch einen endlichen Schaltprozeß zu erreichen. Diesen Sachverhalt drückt das Prädikat $REV(SN, M_f)$ aus, das definiert ist durch:

$$\begin{aligned} & REV(SN, M_f) \\ :\Leftrightarrow & \quad \forall (M_r \in RM(M_0)): M_r \in RM(M_f) \end{aligned}$$

Aus der Perspektive der Störfallflexibilität interessiert jedoch nur, von jeder erreichbaren Netzmarkierung M_r aus zur Ausgangsmarkierung M_0 zurückkehren zu können⁶⁷). Dabei wird vorausgesetzt, daß die Ausgangsmarkierung den störungsfreien Ausgangszustand des jeweils modellierten Produktionssystems oder einer daraus stammenden Systemkomponente darstellt. Dann gilt für die M_0 -Reversibilität⁶⁸ eines Netzmodells mit $REV_0(SN)$ als vereinfachter Notation von $REV(SN, M_0)$:

$$\begin{aligned} & REV_0(SN) \\ :\Leftrightarrow & \quad REV(SN, M_f) \wedge M_f = M_0 \\ \Leftrightarrow & \quad \forall (M_r \in RM(M_0)): M_0 \in RM(M_r) \end{aligned}$$

Die Netzeigenschaft der M_0 -Reversibilität läßt sich von der voranstehend behandelten Störfallflexibilität auf alle Netzmodelle übertragen, mit deren Hilfe untersucht werden soll, ob Prozesse unbegrenzt oft wiederholt werden können⁶⁹). Dies ist z.B. für solche Netzmodelle von Interesse, mit deren Hilfe eine zyklische Prozeßplanung beabsichtigt ist⁷⁰). Dabei wird in jedem Planungszyklus für das modellierte Produktionssystem ein zyklusspezifisches Auftragspaket vorgegeben. Seine Abwicklung konstituiert das Sachzielsystem der Prozeßkoordinierung für jeden Planungszyklus. Die Planungszyklen knüpfen auf jeden Fall dann in wohldefinierter Weise aneinander an, wenn das Netzmodell nach jeder vollständigen Auftragspaketabwicklung in seinen Ausgangszustand zurückkehrt⁷¹). Genau das leistet die M_0 -Reversibilität des Netzmodells. Wenn sie erfüllt ist, können im Prinzip beliebig viele Planungszyklen nacheinander durchlaufen werden. Die M_0 -Reversibilität erhält dann die Qualität einer unbegrenzten Wiederverwendbarkeit von Netzmodellen für gleichartig iterierte Planungsaufgaben. Diese Charakteristik der unbegrenzten Wiederverwendbarkeit wird auch später eine Rolle spielen, wenn Netzmodule für die Spezifizierung des Schaltverhaltens von Makrotransitionen erörtert werden. Dann dient die M_0 -Reversibilität als ein Kriterium für die Korrektheit der Netzmodule.

Anmerkungen zum Kapitel:

- 1) Der betriebswirtschaftliche Robustheitsbegriff wurde bereits eingeführt. Er liegt auch den anschließenden Erläuterungen der Robustheit von Netzmodellen zugrunde. Vgl. darüber hinaus die Würdigung Robuster Planung für die Bewältigung dynamischer Koordinierungsprobleme. Sie erfolgte anlässlich der Diskussion des entscheidungstheoretischen Strukturierungsparadigmas.
 - 2) Produktionsstörungen stellen eine besondere - unerwünschte - Art von Produktionssituationen dar.
 - 3) Konzepte der Robusten Planung wurden im Bezugsrahmen dieser Arbeit erwähnt. Der Einfachheit halber werden hier die wesentlichen Arbeiten zu diesem Planungskonzept noch einmal genannt: GUPTA, S. (1968), S. B-20ff.; ROSENHEAD (1972), S. 418ff.; HANSSMANN (1978a), S. 174ff.; ROSENHEAD (1980a), S. 214f.; ROSENHEAD (1980b), S. 331ff.; HANSSMANN (1987), S. 74f., 129 u. 229ff.; DELFMANN (1989a), S. 222ff., insbesondere S. 224ff.; HANSSMANN (1989), Sp. 1758ff.; HANSSMANN (1990), S. 107f., 322f., 327, 335ff. u. 446.
 - 4) Einen anderen betriebswirtschaftlich motivierten Beitrag stellen die Optimierungsnetze dar, die an späterer Stelle ausführlich behandelt werden.
 - 5) Der Begriff der intendierten Problemlösung wurde bereits in einer früheren Anmerkung eingeführt. Er wird in Kürze für die hier untersuchten Netzmodelle präzisiert.
 - 6) Vgl. dazu die Erläuterung des Robustheitsbegriffs in einer früheren Anmerkung. Dort wurde die Robustheit eines Produktionsmodells durch die Anzahl derjenigen Veränderungen von Produktionssituationen gemessen, in denen die ursprünglich ermittelten, intendierten Modellösungen unverändert gültig bleiben. Vgl. aber auch die Anmerkung, in der auf Nuancen des hier entwickelten, netzbasierten Robustheitsbegriffs hingewiesen wird.
 - 7) Solche Koordinierungsentscheidungen setzen die Existenz von Dead- oder Livelocks voraus. In betriebswirtschaftlichen Modellierungen von Koordinierungsproblemen werden diese dynamischen Systemeigenschaften im allgemeinen nicht berücksichtigt. Dies erklärt die oben erwähnte einseitige Ausrichtung des betriebswirtschaftlichen Robustheitsbegriffs an den Chancen, die hinsichtlich des Erreichens intendierter Problemlösungen bestehen.
 - 8) In der Explizierung und sachlich gerechtfertigten Berücksichtigung des Risikoaspekts liegt die oben erwähnte Befruchtung, die der betriebswirtschaftliche Robustheitsbegriff seitens des Petrinetz-Konzepts erfährt.
 - 9) Das Entscheidungsfeld liegt bereits durch das modellierte Produktionssystem fest.
 - 10) Die Robustheit eines Netzmodells wird daher nicht nur auf das modellierte Produktionssystem, sondern auch auf das ebenso modellierte Koordinierungsproblem bezogen. Folglich können für dasselbe Produktionssystem verschiedene Robustheitsgrade ausgewiesen werden je nachdem, welches Koordinierungsproblem mit welchen Sach- und Formalzielen aktuell zugrundeliegt. Auf dieser Perspektive beruht auch das betriebswirtschaftliche Konzept Robuster Planung. Denn ein Plan kann nur zur Lösung eines konkret bestimmten Koordinierungsproblems aufgestellt und dabei (oder danach) hinsichtlich seiner Robustheit beurteilt werden. Dies erfordert allerdings, die hier betrachtete Anpassungsfähigkeit an veränderliche Produktionssituationen zu präzisieren. Es wird daran erinnert, daß die Produktionssituation als Gesamtheit aus aktuellem Zustand des Produktionssystems sowie aktuell gültigem Sach- und Formalzielsystem festgelegt wurde. Situationsveränderungen können daher auch Variationen von Sach- oder Formalzielen bedeuten. Solche Veränderungen führen jedoch dazu, daß jeweils ein neues Koordinierungsproblem konstituiert wird. Dies ist jedoch für die Beurteilung der Modellrobustheit unbeachtlich, da sie sich - wie oben dargelegt - auf die Einheit aus modelliertem Produktionssystem *und* ebenso modelliertem Koordinierungsproblem erstreckt. Deshalb werden hier nur noch solche Variationen der Produktionssituation betrachtet, bei denen Ereignisse im Produktionssystem geschehen und dadurch den aktuellen Systemzustand verändern. Eine derart veränderte Produktionssituation tangiert nicht die Gültigkeit des invariant vorausgesetzten, problemspezifischen Sach- und Formalzielsystems.
- Eine Ausweitung der zuvor präzisierten Anwendungsbedingungen des Robustheitsbegriffs ist allerdings dadurch möglich, daß unvollständig spezifizierte Koordinierungsprobleme zugrundegelegt werden. Sie besitzen zwar jeweils ein wohlbestimmtes Sachzielsystem, aber für sie ist kein Formalzielsystem festgelegt. Dann fallen die intendierten Problemlösungen mit den finalen Schaltprozessen zusammen, die jeweils per definitionem die vorausgesetzten Sachziele erfüllen. Die daraus abgeleitete Robustheit eines Netzmodells bezieht sich dann nur noch auf das abgebildete Produktionssystem und das vorgegebene Sachzielsystem. Eine noch weiter reichende Vernachlässigung auch der Sachziele läßt sich dagegen nicht mehr vorstellen. Denn in diesem Fall wären überhaupt keine intendierten Problemlösungen definiert. Ohne sie läßt sich der Robustheitsbegriff aber nicht mehr in der betriebswirtschaftlich vertrauten Weise anwenden. (Auf eine abweichende, störfallbezogene Verwendung des Robustheitsbegriffs wird weiter unten eingegangen. Sie hängt nicht von Sach- oder Formalzielvorgaben ab.) Die vorgenannte Ausweitungsmöglichkeit auf unvollständig spezifizierte Koordinierungsprobleme wird nicht weiter ausgeführt. Denn sie läßt sich in die nachfolgend definierten Robustheitsmaße ohne Schwierigkeiten aufnehmen. Dazu braucht lediglich die relevante Menge intendierter Schaltprozesse entsprechend weiter definiert zu werden.

11) Jeder Schaltprozeß aus der intendierten Prozeßmenge IPROM legt eine globale Entscheidungsalternative im Sinne der früher entfalteten entscheidungstheoretischen Struktur von dynamischen Koordinierungsproblemen fest. Die Schaltschritte aus einem solchen Schaltprozeß, die durch einen Entscheidungsträger zur Ausführung ausgewählt wurden, stellen jeweils eine lokale Entscheidungsalternative dar. Ihre Aufeinanderfolge im Schaltprozeß bildet eine Sequenz lokaler Koordinierungsentscheidungen. Jede solche Entscheidungssequenz liefert eine Lösung desjenigen Koordinierungsproblems, das mit Hilfe des betrachteten Netzes modelliert wurde. Neben den vorgenannten Schritten, die vom Entscheidungsträger ausgewählt werden, enthält ein Schaltprozeß aus dem Netzmodell im allgemeinen weitere Schaltschritte. Sie repräsentieren autonome Ereignisgeschehnisse im koordinierten Produktionssystem. Die jeweils aktuelle Produktionssituation wird im Netzmodell durch dessen aktuelle, situationsspezifische Markierung wiedergegeben. Veränderungen der Produktionssituation werden daher im Netzmodell durch das Ausführen von Schritten dargestellt, die entweder auf Auswahlakten des Entscheidungsträgers oder aber auf Ereignisgeschehnissen im Produktionssystem beruhen. Ein Schaltprozeß umfaßt deshalb sowohl Situationsveränderungen im Produktionssystem als auch die Reaktionen des Entscheidungsträgers darauf. Aus diesem Grund entspricht jeder Schaltprozeß des Netzmodells einer möglichen Reaktion(ssequenz) des Entscheidungsträgers auf eine mögliche Entwicklung der Produktionssituation, die durch autonome Ereigniseintritte im Produktionssystem vorangetrieben wird.

12) Die nachfolgend benutzte Menge ZFPROM umfaßt nur solche Schaltprozesse, die auch alle Satisfizierungsziele aus dem zugrundeliegenden Formalzielsystem erfüllen. Daher brauchen Satisfizierungsziele nicht gesondert betrachtet zu werden. Falls ein Formalzielsystem ausschließlich aus Satisfizierungszielen besteht, reicht es aus, in der anschließenden Definition der intendierten Prozeßmenge das Prädikat der Nichtdominiertheit ersatzlos zu streichen. Die Berücksichtigung von Meliorierungszielen bereitet dagegen mehr Aufwand. Erstens muß das einstellige Prädikat der Nichtdominiertheit durch ein zweistelliges Prädikat ersetzt werden, das die echte Besserstellung gegenüber der Referenzalternative ausdrückt. Diesen Zweck erfüllt das Prädikat BES_{\leq} , das bereits im Kontext des Dominanzkonzepts definiert wurde. Darüber hinaus muß als Referenzalternative ein zulässiger finaler Schaltprozeß aus der Prozeßmenge PROM des untersuchten Netzes vorgegeben werden. Da keine Optimierung angestrebt wird, gilt es, diejenige endliche echte Teilmenge der zulässigen Finalprozeßmenge ZFPROM festzulegen, deren Schaltprozesse als potentiell besserstellende (globale) Entscheidungsalternativen untersucht werden sollen. Schließlich ist aus allen zulässigen finalen Schaltprozessen, die erstens aus dieser Teilmenge stammen und zweitens gegenüber der Referenzalternative echt besser stellen, mindestens einer derjenigen auszuwählen, die von keinem anderen Schaltprozeß gleicher Qualifikation dominiert werden. Es kann auch der Fall eintreten, daß überhaupt kein zulässiger finaler Schaltprozeß existiert, der aus der vorgenannten Teilmenge stammt und zugleich gegenüber der Referenzalternative echt besser stellt.

Bei der erweiterten Robustheitsanalyse, die in einer früheren Anmerkung angesprochen wurde, umfaßt die ausgezeichnete Prozeßmenge IPROM alle finalen Schaltprozesse des Netzmodells. Sie fällt daher mit der Finalprozeßmenge FPROM zusammen: $IPROM = FPROM$. Dies entspricht dem Außerachtlassen der zweiten, formalzielbezogenen Anforderung an die intendierten Schaltprozesse.

13) Erwünschte Markierungen aus der Endmarkierungsmenge ME können dagegen Deadlockmarkierungen darstellen. Um das Erreichen derart erwünschter Deadlockmarkierungen nicht versehentlich auszuschließen, kommt es darauf an, keineswegs alle Schaltprozesse als nicht-intendiert zu klassifizieren, die in einer toten Zielmarkierung enden. Dies wird unten in der Definition der Prozeßmenge NIPROM durch das Konjugat " $Dead(M_i) M_i ME$ " berücksichtigt.

14) Die letztgenannte Einschränkung ist erforderlich, weil Livelockmarkierungen durchaus zur Markierungsfolge MF_i eines intendierten Schaltprozesses $PRO_{0,E/e}(MF_i, SF_i)$ gehören können. Dann wird der intendierte Schaltprozeß so ausgeführt, daß er sich nicht in dem Livelockprozeß verfängt, der durch die Livelockmarkierung verläuft. In diesem Sonderfall wird von einer unschädlichen Livelockmarkierung gesprochen. Alle Livelockmarkierungen, welche die o.a. Einschränkung erfüllen, werden dagegen als unerwünschte Livelockmarkierungen bezeichnet.

15) Die Prozeßmenge PROM enthält im allgemeinen weit mehr Schaltprozesse als die voranstehend definierten intendierten und nicht-intendierten Schaltprozesse. Denn diese beiden Prozeßklassen erstrecken sich ausschließlich auf Schaltprozesse, die in der Ausgangsmarkierung des Netzes starten. Alle anderen Schaltprozesse blieben daher unberücksichtigt. Darüber hinaus werden auch jene Schaltprozesse ausgeklammert, die zwar in der Ausgangsmarkierung begonnen haben, aber weder in einer der erwünschten Endmarkierungen noch in einer der unerwünschten Dead- oder Livelockmarkierungen angelangt sind.

16) Sie wird auch kurz als nicht-intendierte Prozeßmenge bezeichnet.

17) Die hier eingeführten Chancen quantifizieren die Anzahl derjenigen Veränderungen von Produktionssituationen, die eintreten können, ohne daß auf das Erreichen einer intendierten Lösung des zugrundeliegenden Produktionsmodells verzichtet werden muß. Dies entspricht dem oben angeführten betriebswirtschaftlichen Robustheitsbegriff. Dabei wird das Produktionsmodell hier als Netzmodell gestaltet. Die intendierten Modellösungen sind die erwünschten Endmarkierungen des Netzmodells.

Allerdings ist auf drei Besonderheiten hinzuweisen. Erstens wird der Chancen- durch einen komplementären Risikobegriff ergänzt. Darauf wurde bereits hingewiesen. Zweitens zählt der anschließend präzisierter Chancenbegriff strenggenommen keine *Situationsveränderungen*. Statt dessen entspricht jede Chance genau einem *Prozeß* im Netzmodell, durch den sich auf Situationsveränderungen so reagieren läßt, daß dennoch eine erwünschte Endmarkierung erreicht wird. Daher kann durchaus der Fall eintreten, daß bei der Ausführung eines solchen Prozesses auf *mehrere* Veränderungen der Produktionssituation hintereinander reagiert wird. Folglich wird durch den hier konkretisierten Robustheitsbegriff die Invarianz von intendierten Modellösungen nicht mehr direkt auf die Anzahl von Situationsveränderungen bezogen. Vielmehr erstreckt er sich auf die Anzahl derjenigen Prozesse, mit denen auf Situationsveränderungen so reagiert werden kann, daß die intendierte Modellösungen erhalten bleiben. Drittens fordert der betriebswirtschaftliche Robustheitsbegriff nur die Invarianz von intendierten Modellösungen gegenüber Veränderungen der Produktionssituation. Damit ist aber noch nicht präzise festgelegt, ob jeweils genau eine intendierte Modellösung *als solche* erhalten bleiben soll (starke Robustheit) oder ob es ausreicht, trotz variierender Produktionssituationen *irgendeine* intendierte Modellösung zu erreichen (schwache Robustheit). Diese Unterbestimmtheit wird hier durch die Robustheitsmaße für Netzmodelle zugunsten der schwachen Robustheitsauffassung beseitigt. Denn bei der Messung der Chancen, weiterhin eine intendierte Modellösung zu erreichen, werden jeweils Prozesse im Netzmodell betrachtet, die zu *irgendeiner* Markierung aus der Menge der erwünschten Endmarkierungen führen.

Das Chancenkonzept, das hier zur Operationalisierung der Robustheit von Produktionsmodellen präsentiert wird, liegt auch dem Ansatz zugrunde, den FOX, B. (1987a), S. 234, für die Konkretisierung des Prinzips kleinstmöglicher Entscheidungsbindung vorgeschlagen hat: "... the principle of least commitment dictates ... [to] ... pursue a course of action which preserves and guarantees future flexibility. This can be accomplished, in part, by selecting the action which guarantees the greatest *number of ways* of completing the ... task." (Ergänzung [...] und kursive Hervorhebung durch den Verf.). Damit zeigt sich die enge Verknüpfung der hier vorgetragenen Präzisierung des Robustheitsbegriffs für Netzmodelle mit der früheren Konzeptualisierung dynamischer Koordinierungsprobleme. Vgl. dazu insbesondere die Ausführungen zum Prinzip kleinstmöglicher Entscheidungsbindung und zur opportunistischen Prozeßkoordinierung.

18) Der Gesamtprozeß $PRO_{0.E/e}(SF_{L/G}, MF_{L/G})$ läßt sich ohne Schwierigkeiten aus den beiden Teilprozessen $PRO_{0.nf}(SF_{L/0}, MF_{L/0})$ und $PRO_{nf.E/e}(SF_{L/e}, MF_{L/e})$ zusammensetzen. Hierfür gilt mit $SS_a = SS_{a(1/e)}$ und $L_G = L_0 + L_e$:

$$\begin{aligned} (SF_{L/0} = (SS_{a(1/0)}, \dots, SS_{a(L/0)}) \wedge MF_{L/0} = (M_0, \dots, M_{nf}) \\ \wedge SF_{L/e} = (SS_{a(1/e)}, \dots, SS_{a(L/e)}) \wedge MF_{L/e} = (M_{nf}, \dots, M_{E/e})) \\ \Rightarrow (SF_{L/G} = (SS_{a(1/0)}, \dots, SS_{a(L/0)}, SS_{a(1/e)}, \dots, SS_{a(L/e)}) \\ \wedge MF_{L/G} = (M_0, \dots, M_{nf}, \dots, M_{E/e})) \end{aligned}$$

Die nonfinale Markierung M_{nf} kann auch die Ausgangsmarkierung M_0 sein. In diesem Grenzfall handelt es sich beim Schaltprozeß $PRO_{0.nf}(SF_{L/0}, MF_{L/0})$ um den entarteten Prozeß $PRO_{0.0}(SF_0, MF_0)$. Seine Verknüpfung mit dem Schaltprozeß $PRO_{nf.E/e}(SF_{L/e}, MF_{L/e})$ gehorcht im Prinzip den oben dargelegten Festlegungen. Es sind lediglich die Besonderheiten $SF_{L/0} = SF_0 = ()$ und $MF_{L/0} = MF_0 = (M_0)$ zu beachten.

19) Diese dritte Anforderung besitzt nur verdeutlichenden Charakter. Sie wird von jedem Schaltprozeß $PRO_{nf.E/e}(SF_{L/e}, MF_{L/e})$, der den beiden ersten Anforderungen gerecht wird, notwendig erfüllt.

20) Der Gesamtprozeß $PRO_{0.r}(SF_{L/G}, MF_{L/G})$ wird aus den beiden Teilprozessen $PRO_{0.nf}(SF_{L/0}, MF_{L/0})$ und $PRO_{nf.r}(SF_{L/r}, MF_{L/r})$ in derselben Weise zusammengesetzt, wie es in einer früheren Anmerkung für den dort behandelten "Chancenfall" vereinbart wurde.

21) Diese dritte Anforderung besitzt abermals nur verdeutlichenden Charakter. Sie wird von jedem Schaltprozeß $PRO_{nf.r}(SF_{L/r}, MF_{L/r})$, der den beiden ersten Anforderungen gerecht wird, notwendig erfüllt.

22) Das betriebswirtschaftlich vertraute Robustheitskonzept berücksichtigt nur den Aspekt der Chancen. Der Risikofall bleibt dagegen unbeachtet. Darauf wurde schon oben aufmerksam gemacht.

23) Es handelt sich um partielle Funktionen, weil sie nicht für jedes Paar aus einer Markierung und einem Schaltschritt definiert sind. Statt dessen erstrecken sie sich nur auf Schaltschritte SS_a , die unter finalen Markierungen M_{nf} aktiviert sind.

24) Dies unterstützt die Vergleichbarkeit der Robustheitsmaße für Netzmodelle, die später definiert werden. Darüber hinaus entspricht es der üblichen Normierung des betriebswirtschaftlichen Robustheitsbegriffs. Ebenso wird die Verwendung derart normierter Anzahlen von Chancen- und Risikofällen dem Konzept der "Pseudowahrscheinlichkeiten" gerecht, das KRELLE (1968), S. 178ff., für die rationale Bewältigung von Unsicherheit eingeführt hat. Es wird auch von BAMBERG (1989), S. 112, positiv gewürdigt. Am Rande wird darauf hingewiesen, daß diese Normie-

ung später erlaubt, zwei Unsicherheitsparameter einzuführen, die sich in KRELLE's Konzept der Entscheidung unter Unsicherheit kohärent einfügen.

25) Chancen und Risiken einer Auswahlentscheidung wurden definiert, ohne jemals Bezug auf Eintrittswahrscheinlichkeiten für Ereignisse zu nehmen, deren autonome Geschehnisse im Produktionssystem die jeweils aktuelle Produktionssituation (Netzmarkierung) verändern würden. Dies entspricht dem vollständigen Prognoseverzicht, der eingangs im Rahmen der entscheidungstheoretischen Konzeptualisierung von Koordinierungsproblemen motiviert wurde. Da keine stochastischen Aspekte in das Netzmodell eingeflossen sind, liegt eine typische Entscheidung unter Unsicherheit vor.

26) Vgl. zu den Schwierigkeiten der Entscheidung unter Unsicherheit und den dafür diskutierten formalen Lösungskonzepten, die oftmals auch als Entscheidungen bei Ungewißheit thematisiert werden, KRELLE (1968), S. 176ff. u. 185ff.; BAMBERG (1989), S. 104ff.; SIEBEN (1990), S. 51ff.

27) Aufgrund ihrer Additivität und Normierung besitzt sie den Vorteil großer Transparenz. Da es hier nur um eine Explikation des Robustheitsbegriffs, nicht aber um eine Erörterung von Entscheidungen unter Unsicherheit geht, bevorzugt der Verf. grundsätzlich eine transparente Formalisierung der Unsicherheitspräferenz.

28) Die beiden Parameter nähern sich grob an das Konzept der "Unsicherheitspräferenzfunktion" von KRELLE (1968), S. 179ff., insbesondere S. 184, an. Dort werden Zielbeiträge, deren Eintrittswahrscheinlichkeiten unbekannt sind, durch eine Präferenzfunktion überlagert, welche die Einstellung des Entscheidungsträgers gegenüber Unsicherheit wiedergibt. Allerdings wird KRELLE's Präferenzkonzept hier in einer wesentlichen Hinsicht vereinfacht. Denn durch die oben diskutierten Chancen und Risiken, die intendierten Problemlösungen zu verwirklichen bzw. zu verfehlen, werden nur genau zwei Niveaus möglicher Zielbeiträge auf ordinaler Skala identifiziert. KRELLE erörtert dagegen kardinal gemessene Zielbeiträge. Vgl. zu KRELLE's Unsicherheitspräferenzfunktionen auch BAMBERG (1989), S. 110ff.

29) "Objektiv" meint hier nur die Unabhängigkeit von allen Präferenzen des Entscheidungsträgers, welche die Schaltschrittauswahl determinieren. Andere subjektbezogene Einflüsse, die bei der Konstruktion des untersuchten Netzmodells seitens des Modellierungsträgers eingeflossen sein können, bleiben davon unberührt.

30) Aufgrund der Normierung der Parametersumme gilt: Wenn der Entscheidungsträger Chancen und Risiken subjektiv gleich stark wahrnimmt, muß $\alpha_c = \alpha_r = 0,5$ gelten. Für einen risikoblinden Entscheidungsträger gelten $\alpha_c = 1,0$ und $\alpha_r = 0,0$. Ein chancenblinder Entscheidungsträger verhält sich invers mit $\alpha_c = 0,0$ und $\alpha_r = 1,0$.

31) Die Robustheit stellt also per definitionem zunächst eine Entscheidungseigenschaft dar. Daraus abgeleitet kann jedoch auch von der Robustheit eines Schaltschritts gesprochen werden: Wenn der Auswahl des Schaltschritts SS_a der unter der Markierung M_{nr} aktiviert ist, die Robustheit $rob(M_{nr}, SS_a)$ zukommt, dann läßt sich ebenso davon reden, der Schaltschritt SS_a besitze unter der Markierung M_{nr} die Robustheit $rob(M_{nr}, SS_a)$.

32) Die Funktion "rob" könnte in alternativer Weise auch als Quotient der präferenzgewichteten Chancen und Risiken definiert werden. Dann müßten aber für den risikofreien Fall $risiko(M_{nr}, SS_a) = 0$ besondere Vorkehrungen getroffen werden, um einen unzulässigen Quotienten mit dem Nenner "0" auszuschließen. Ein solcher risikofreier Fall tritt z.B. in dead- und livelockfreien Netzen immer ein. Durch die nachfolgende subtraktive Funktionsdefinition werden diese Schwierigkeiten eliminiert.

33) In synonyme Weise wird auch von einer robusten Auswahlentscheidung gesprochen.

34) Dies entspricht - sofern vom Risikoaspekt abstrahiert wird - der Ausdeutung des Prinzips kleinstmöglicher Entscheidungsbindung durch FOX, B. (1987a), S. 234. Sie wurde bereits in einer früheren Anmerkung erwähnt, hier aber der Übersichtlichkeit halber noch einmal wiederholt: "... the principle of least commitment dictates ... [to] ... pursue a course of action which preserves and guarantees future flexibility. This can be accomplished, in part, by selecting the action which guarantees the greatest number of ways of completing the ... task."; Ergänzung [...] und kursive Hervorhebung durch den Verf.).

35) In den beiden Grenzfällen der Chancen- oder der Risikoblindheit mit $\alpha_c = 0$ bzw. $\alpha_r = 0$ kann die additive Berechnung des Robustheitswerts $rob(M_{nr}, SS_a)$ zu einem Informationsverlust führen. Dies ist bei Chancenblindheit genau dann der Fall, wenn unter einer nonfinalen Markierung M_{nr} mehrere alternative Schaltschritte SS_a mit gleichem Risiko $risiko(M_{nr}, SS_a)$, aber verschiedenen Chancen $chance(M_{nr}, SS_a)$ aktiviert sind. Denn dann führt die Gewichtung der Chancen mit dem Parameter $\alpha_c = 0$ dazu, diese unterschiedlichen Chancen bei der Auswahl eines robusten Schrittes nicht mehr berücksichtigen zu können. Ein solches Übersehen von Chancen aufgrund der - letztlich willkürlich gewählten - Formalisierung der Unsicherheitspräferenz läßt sich jedoch vermeiden. Dazu reicht es aus, bei Chancenblindheit zu einem modifizierten Robustheitsprädikat $ROB_c(M_{nr}, SS_a)$ überzugehen, das sich vom oben eingeführten Robustheitsprädikat $ROB(M_{nr}, SS_a)$ lediglich dadurch unterscheidet, daß die additive Verrechnung von Chancen und Risiken in der Abbildungsvorschrift der Robustheitsfunktion "rob" durch ein Max/Min-Kriterium ersetzt wird:

$$\text{ROB}_c(M_{np}, SS_a)$$

$$\begin{aligned} :\Leftrightarrow & M_{np} \in (\text{RM}(M_0) - \text{ME}) \wedge \text{LEB}(M_{np}) \wedge \text{AKT}(SS_a, M_{np}) \\ & \wedge \text{chance}(M_{np}, SS_a) = \max \{ \text{chance}(M_{np}, SS_k^*): SS_k^* \in \text{SSM} \wedge \text{AKT}(SS_k^*, M_{np}) \wedge \dots \\ & \text{risiko}(M_{np}, SS_k^*) = \min \{ \text{risiko}(M_{np}, SS_k): SS_k \in \text{SSM} \wedge \text{AKT}(SS_k, M_{np}) \} \} \end{aligned}$$

Auf das Max/Min-Kriterium des modifizierten Robustheitsprädikats $\text{ROB}_c(M_{np}, SS_a)$ wurde bereits eingangs im Zusammenhang mit der Flexiblen Planung angespielt, ohne es dort explizit darzulegen.

In analoger Weise läßt sich der Informationsverlust, der bei einem risikoblinden Entscheidungsträger mit $\alpha_r=0$ droht, dadurch vermeiden, daß zu einem entsprechend modifizierten Robustheitsprädikat $\text{ROB}_r(M_{np}, SS_a)$ übergegangen wird, für das gilt:

$$\text{ROB}_r(M_{np}, SS_a)$$

$$\begin{aligned} :\Leftrightarrow & M_{np} \in (\text{RM}(M_0) - \text{ME}) \wedge \text{LEB}(M_{np}) \wedge \text{AKT}(SS_a, M_{np}) \\ & \wedge \text{risiko}(M_{np}, SS_a) = \min \{ \text{risiko}(M_{np}, SS_k^*): SS_k^* \in \text{SSM} \wedge \text{AKT}(SS_k^*, M_{np}) \wedge \dots \\ & \text{chance}(M_{np}, SS_k^*) = \max \{ \text{chance}(M_{np}, SS_k): SS_k \in \text{SSM} \wedge \text{AKT}(SS_k, M_{np}) \} \} \end{aligned}$$

Dieses Min/Max-Entscheidungskriterium hält zwar entsprechend der Risikoblindheit des Entscheidungsträgers an der Dominanz des Chancenaspekts fest. Dennoch ist es in der Lage, ein "Restrisiko" zu berücksichtigen, das allen chancenmaximalen Entscheidungsalternativen zukommt. Es kann der Fall eintreten, daß mehrere solcher nicht-dominierten Entscheidungsalternativen existieren und sich zusätzlich mindestens zwei von ihnen bezüglich ihres Risikos des Verfehlens einer erwünschten Endmarkierung unterscheiden. Unter diesen Voraussetzungen gestattet das o.a. Min/Max-Kriterium, eine derjenigen chancenmaximalen Entscheidungsalternativen auszuwählen, die sich durch ihre minimalen "Restrisiken" auszeichnen.

36) Dieser robuste erste Schritt entspricht konzeptionell genau den robusten "ersten Schritten", die aus der Perspektive des betriebswirtschaftlichen Robustheitsbegriffs angeführt wurden. Hier ist allerdings eine zweifache Erweiterung erfolgt. Erstens wird der robuste erste Schritt nicht nur auf die Chancen, sondern auch auf die Risiken bezogen, die hinsichtlich der Verwirklichung einer intendierten Problemlösung bestehen. Zweitens werden nicht nur robuste erste Schritte definiert, sondern ebenso robuste Auswahlsschritte für alle anderen erreichbaren und nonfinalen Netzmarkierungen.

37) Für dieselbe nonfinale und lebendige Markierung M_r können mehrere robuste Auswahlentscheidungen offenstehen. Dies ist genau dann der Fall, wenn unter der Markierung M_r mehrere verschiedene Schaltschritte SS_a aktiviert sind, deren Auswahl jeweils *dieselbe* maximale Robustheit $\text{rob}(M_r, SS_a)$ zukommt. Dann erfüllen alle Entscheidungen zugunsten eines dieser Schaltschritte das Prädikat $\text{ROB}(M_{np}, SS_a)$.

38) Diese Konvention lehnt sich an den Wertebereich $[-1; +1]$ der oben definierten Robustheitsfunktion "rob" an. Analoges gilt für die anschließende Vereinbarung hinsichtlich aller finalen Markierungen.

39) Dies gilt allerdings nur unter der Voraussetzung, daß unter jeder nonfinalen und lebendigen Netzmarkierung mindestens ein Schaltschritt für den Entscheidungsträger zur Auswahl steht. Dies muß nicht immer der Fall sein. Statt dessen kann ein Koordinierungsproblem auch so modelliert werden, daß in seinem Netzmodell mindestens eine nonfinale Markierung existiert, unter der ausschließlich Ereignisse im koordinierten Produktionssystem geschehen können. Da diese Ereignisse im Netzmodell als Transitionen abgebildet werden, müssen diese Transitionen unter der betrachteten nonfinalen Markierung aktiviert sein, damit ihre Geschehnismöglichkeit korrekt erfaßt wird. Folglich handelt es sich um eine lebendige Markierung, obwohl unter ihr kein Entscheidungsspielraum für den Entscheidungsträger besteht. Weil unter dieser Markierung keine Auswahlentscheidungen definiert sind, kann ihr auch kein Robustheitswert zugeordnet werden. Folglich müßten alle Markierungen, welche die vorgenannten Bedingungen erfüllen, aus dem Anwendungsbereich der Robustheitsfunktion "rob_M" ausgeschlossen werden. Entsprechend resultierte eine nur noch partiell definierte Funktion. Von dieser Komplikation wird hier jedoch abgesehen.

40) Die Robustheitsfunktion wird so allgemein definiert, daß sie auch zwei Extremfälle abdeckt: Erstens kann die Ausgangsmarkierung M_0 eine unerwünschte Deadlockmarkierung darstellen. Dann handelt es sich um ein totes Netz, dessen einzige erreichbare Markierung den Robustheitswert $\text{rob}_M(M_0) = -1$ besitzt. Zweitens ist es ebenso möglich, daß es sich bei der Ausgangsmarkierung M_0 bereits um eine der erwünschten Endmarkierungen aus der Menge ME handelt. In diesem Fall kommt der Ausgangsmarkierung der Robustheitswert $\text{rob}_M(M_0) = +1$ zu.

41) Eine andere Aggregationsmöglichkeit könnte darin gesehen werden, für alle erreichbaren Markierungen eines Netzes deren subjektive Eintrittswahrscheinlichkeiten zu schätzen. Dann ließe sich eine skalare Netzrobustheit z.B.

auch als Erwartungswert über allen markierungsspezifischen Robustheitswerten errechnen. Wegen des früher gerechtfertigten Verzichts auf stochastische Modellierungen von Koordinierungsproblemen wird diese Alternative jedoch nicht weiter verfolgt.

42) Diese Aggregation bedeutet zwar einen Informationsverlust bezüglich der Robustheitswerte aller übrigen erreichbaren Markierungen. Doch sie entspricht dem betriebswirtschaftlichen Konzept robuster "erster Schritte". Auch dort wird ein Plan zur Lösung eines Koordinierungsproblems nur für den aktuell vorliegenden Ausgangszustand des modellierten Produktionssystems beurteilt. Vgl. dazu die Quellen, die bereits zur Planung robuster erster Schritte angeführt wurden.

43) Vgl. zur Promptheit von Netzen HACK,M. (1975b), S. 50; VALK (1983b), S. 40ff.; VALK (1983c), S. 359f.

44) Solche Netzmodelle können unter anderem für die Realzeitkoordinierung flexibler Fertigungssysteme herangezogen werden. Dies bildet den sachlichen Hintergrund der anschließenden Ausführungen. Vgl. dazu auch die Beiträge, die später zur Realzeitkopplung zwischen Netzmodellen und modellierten Realitätsausschnitten angeführt werden.

45) Auf diese Realzeitkopplung zwischen Netzen und realen Systemen wird später aus dem Blickwinkel der interpretierten Netze ausführlicher zurückgekommen.

46) Die Buchstaben "S" und "K" verweisen auf das System und das Koordinierungsnetz, zwischen denen die Nachrichten ausgetauscht werden.

47) Darauf wird in Kürze näher eingegangen.

48) Vgl. HACK,M. (1975b), S. 50 (allerdings ohne Differenzierung zwischen unterschiedlichen Systemmeldungen).

49) Es ist trivial, daß ein R-promptes Netz auch jede größere Reaktionsspanne R' mit $R' \in \mathcal{N}_i$ und $R' \geq R$ einzuhalten vermag. Denn beim Wert R handelt es sich nur um die *kleinstmögliche* Reaktionsspanne, die für *alle* Systemmeldungen SK_i mit $i \in \{1, \dots, I\}$ selbst für den *ungünstigsten* Fall garantiert werden kann. Daher ließe sich die R-Promptheit eines Netzes auch in schwächerer Form durch $R_i \leq R'$ für alle $i \in \{1, \dots, I\}$ definieren. Der Verf. bevorzugt aber die oben präsentierte, stärkere Variante, weil ihr Informationsgehalt größer ist: Sie gibt nicht nur *eine* obere Schranke der meldungsspezifischen Reaktionsspannen R_i an, sondern sogar *die* kleinstmögliche obere Schranke.

50) Vgl. VALK (1983b), S. 41; VALK (1983c), S. 359.

51) Auf den Fall, daß eine solche Beschränkung nicht existiert, wird später unter dem Stichwort der Unpromptheit zurückgekommen.

52) Vgl. zur schwachen Promptheit, die allerdings zumeist ohne präzisierendes Attribut nur als Promptheit angesprochen wird, VALK (1983b), S. 41; VALK (1983c), S. 359.

Die schwache Promptheit wird im Kontext dynamischer Systeme in vielfältigen Varianten thematisiert, die sich jedoch inhaltlich kaum unterscheiden. Dabei wird die Bezeichnung "Promptheit" nicht verwendet. Hierzu gehören vor allem:

- die Fähigkeit eines Systems, auf Umweltreize mit endlicher Verzögerung zu reagieren (finite delay property);
- die Eigenschaft eines Systems, ihm gestellte Anfragen in endlicher Zeit korrekt zu beantworten (responsiveness), die vor allem im Bereich dynamischer oder temporaler Logikkalküle behandelt wird; vgl. PNUELI (1977), S. 49; PNUELI (1979), S. 11f.; MANNA (1979), S. 397f.;
- die Eigenschaft der Impuls-Beantwortung, die von GABBAY (1980), S. 167, diskutiert wird.

Daneben klingt die schwache Promptheit auch bei HOLT,A. (1975d), S. 158, an. Er behandelt im Kontext automatischer Informationsverarbeitungssysteme das Vermeiden von Outputverzögerungen, die unbeschränkt lange andauern.

53) Die schwache Promptheit eines Netzes kann sowohl durch Dead- als auch durch Livelocks verhindert werden. Dies ist immer dann der Fall, wenn ein Dead- oder Livelock dazu führt, daß die Anweisungstransition, deren Schaltakt zum Aussenden einer Steuerungsanweisung erforderlich wäre, tatsächlich niemals geschaltet wird. Der Umkehrschluß trifft jedoch nicht zu. Denn die Dead- und Livelockfreiheit eines Netzes reichen im allgemeinen noch nicht aus, um die schwache Promptheit eines Netzes zu garantieren. Vielmehr kann es in Netzen mit unendlichen Erreichbarkeitsgraphen zu Schaltprozessen kommen, in denen die Anweisungstransition niemals geschaltet wird, ohne daß es sich um zyklische Livelockprozesse handelt. Vgl. dazu die Skizze eines solchen unendlichen, aber nicht-zyklischen Schaltprozesses in einer früheren Anmerkung. Er läßt die Markenanzahl auf einer Stelle unbeschränkt anwachsen. Daher liegt ein Netz mit einem unendlichen Erreichbarkeitsgraphen vor. Solche Prozesse werden aber in dieser Arbeit durch die Voraussetzung endlicher Erreichbarkeitsgraphen ausgeschlossen.

54) Ein einfaches Beispiel dafür findet sich bei HACK,M. (1975b), S. 51. Dort hängt die Reaktionsspanne R von der Markenanzahl auf einer Stelle ab, die über jede endliche obere Schranke hinaus anwachsen kann. Da unter jeder Netzmarkierung diese Markenanzahl endlich ist, bleibt zwar auch die Anzahl der Schaltschritte für die Netzreaktion

endlich. Doch läßt sich wegen der Unbeschränktheit der Markenanzahl für die Schaltschrittzahl keine feste obere Schranke R angeben, die für alle erreichbaren Markierungen gilt.

55) Aufgrund dieser Festlegung kann sich ein unpromptes Netz also durchaus noch schwach prompt verhalten. Dies ist etwa für das Beispiel von HACK, M. (1975b), S. 51, der Fall, das bereits in einer früheren Anmerkung angesprochen wurde.

56) Es handelt sich jedoch um keine notwendige Bedingung. Denn ein Netz kann sich z.B. auch dadurch unprompt verhalten, daß in ihm ein nicht-zyklischer, aber potentiell unendlicher Schaltprozeß abläuft, der die Markenanzahl mindestens einer Stelle über jede endliche obere Schranke hinaus wachsen läßt. Sie wurden schon kurz zuvor erwähnt.

57) Vgl. PRIESE (1979), S. 22 u. 48.

58) Es wird vorausgesetzt, daß alle Produktionsstörungen, die aus der Sicht des Modellierungsträgers für die Prozeßkoordinierung von Interesse sind, im Netzmodell des koordinierten Produktionssystems abgebildet wurden.

59) Als solche Systemkomponenten kommen z.B. Bearbeitungsstationen in Betracht. Störungen ihrer Betriebsbereitschaft erfahren in der späteren Fallstudie zur Modellierung Flexibler Fertigungssysteme besondere Aufmerksamkeit.

60) Die Anpassungsmaßnahmen, die zur Störungsbeseitigung in Betracht kommen, werden im Netzmodell des Produktionssystems durch entsprechende Schaltprozesse abgebildet.

61) Es wird hier der Einfachheit halber nur der störungsfreie Ausgangszustand betrachtet. Dies ist aber keineswegs notwendig. Statt dessen könnten auch mehrere störungsfreie "Wiederaufsetzpunkte" betrachtet werden, zu denen die gestörte Systemkomponente nach Ausführung von Anpassungsmaßnahmen zurückkehren soll. Solche Wiederaufsetzpunkte werden zumeist beim Realzeitbetrieb von Systemen eingerichtet, um angesichts einer Betriebsstörung baldmöglichst in einen wohldefinierten Betriebszustand übergehen zu können; vgl. AYACHE (1979a), S. 1052; VOSS, K. (1983a), S. 3.

Darüber hinaus existiert ein bemerkenswerter Ansatz innerhalb des Petrinetz-Konzepts, der es erlaubt, Wiederaufsetzpunkte zu definieren und im Störfall dorthin zurückkehren. Zu diesem Zweck wird ein störungsanfälliges System zunächst auf der Objektebene durch ein beliebiges Netz modelliert. Verhaltensweisen des Systems werden als Schaltprozesse in diesem Netzmodell repräsentiert. Hinzu kommt auf der Metaebene ein besonderes Dokumentationsnetz. Es handelt sich um ein zyklen- und konfliktfreies Synchronisationsnetz. Dieses Dokumentationsnetz wird schritthaltend zur Ausführung eines Schaltprozesses im Netzmodell konstruiert. Dabei wird es jeweils um diejenigen Transitionen erweitert, die bei der Schaltprozeßausführung im Netzmodell tatsächlich geschaltet worden sind. Hinzu kommen jeweils diejenigen Stellen, die im Netzmodell durch das letzte Schalten von Transitionen neu markiert worden sind. Auf der Metaebene wird daher dokumentiert, wie im Netzmodell auf der Objektebene ein Schaltprozeß ausgeführt wird. Das Dokumentationsnetz stellt eine schaltprozeßspezifische "Abwicklung" des zugrundeliegenden Netzmodells dar. Wenn bei der Schaltprozeßausführung im Netzmodell ein zulässiger Wiederaufsetzpunkt des zugrundeliegenden Systems ohne Schwierigkeiten durchlaufen wird, so wird dies im Dokumentationsnetz lediglich als eine potentielle Wiederaufsetzmarkierung festgehalten. Falls jedoch im zugrundeliegenden System eine Verhaltensstörung eintritt, kann im Netzmodell der zugehörige Schaltprozeß nicht fortgesetzt werden. Aber das Dokumentationsnetz erlaubt aufgrund seiner Zyklen- und Konfliktfreiheit, durch Umkehrung der Kantenrichtung im Dokumentationsnetz seine zuletzt neu erzeugten Transitionen "rückwärts" zu schalten. Dies geschieht so lange, bis im Dokumentationsnetz eine seiner Wiederaufsetzmarkierungen erreicht wird. Die Wiederaufsetzmarkierung ist wegen der Konfliktfreiheit des Dokumentationsnetzes eindeutig bestimmt. Sie repräsentiert einen wohldefinierten Wiederaufsetzpunkt des zugrundeliegenden Systems, von dem aus die zuvor gestörte Systemverhaltensweise von Neuem in Gang gesetzt werden kann. Auf diese Weise erlaubt es das Petrinetz-Konzept einerseits, ein System und seine Verhaltensweisen auf der Objektebene durch ein Netzmodell und dessen Schaltprozesse zu repräsentieren. Andererseits wird auf der Metaebene ein Dokumentationsnetz schritthaltend konstruiert, um nach Systemstörungen zu wohldefinierten Wiederaufsetzpunkten zurückkehren zu können. Vgl. zu diesem Ansatz MERLIN, P. (1978), S. 129ff.; BEST, E. (1980a), S. 242ff.

62) Die Koordinierungsweise verhält sich strenggenommen nur flexibel gegenüber jenen Produktionsstörungen, die im Modell der betrachteten Komponente des Produktionssystems erfaßt wurden. Nur unter der - wenig realistischen Prämisse - daß alle überhaupt möglichen Störungen der Komponente modelliert wurden, liegt auch eine vollständige Koordinierungsflexibilität im hier thematisierten Sinne vor. Unter Anerkennung dieses Vorbehalts wird vereinbart, daß sich das Attribut "vollflexibel" stets nur auf die modellierten Produktionsstörungen bezieht.

63) Diese Feststellung der vollflexiblen Koordinierung ist nur als eine Implikation ("immer dann"), nicht aber als eine Äquivalenz ("genau dann") formuliert. Denn sie bezieht sich auf die gesicherte Rückkehrmöglichkeit aus *jedem* Systemzustand. Damit stellt sie eine strengere Forderung auf, als es zur Anpassung an Störungen der Systemkomponente erforderlich wäre. Für solche Anpassungsmaßnahmen würde die Garantie ausreichen, nach jedem störungsrepräsentierenden Ereignis oder aus jedem störungsdarstellenden Komponentenzustand in den störungs-

freien Ausgangszustand der Komponente zurückkehren zu können. Dann wäre es jedoch erforderlich, die vorgeannten Ereignisse bzw. Zustände als solche zu identifizieren. Darüber hinaus müßten alle Komponentenzustände, die entweder nach den Geschehnissen von störungsrepräsentierenden Ereignissen vorliegen oder aber selbst Störungszustände darstellen, von allen anderen ungestörten Komponentenzuständen unterschieden werden. Dieser erhebliche Modellierungsaufwand wird vermieden, wenn die o.a. strengere Forderung untersucht wird, aus jedem Komponentenzustand in den ungestörten Ausgangszustand zurückkehren zu können.

64) Vgl. zu diesem störfallbezogenen Robustheitsverständnis, das vom eingangs thematisierten betriebswirtschaftlichen Robustheitsbegriff deutlich unterschieden werden muß, RUSCA (1988), S. 117; LIGGESMEYER (1990), S. 26.

65) Diese Prämisse ist im allgemeinen nicht erfüllt. Daher handelt es sich hier nur um eine hypothetische Erwägung.

66) Vgl. zur M_f -Reversibilität ARAKI (1976c), S. 7ff.; ARAKI (1977b), S. 112.

Strengere Definitionen der Reversibilität als die oben vorgestellte finden sich bei CARDOZA (1976), S. 50; JAFFE (1977), S. 9; GRABOWSKI, J. (1980c), S. 18; PETERSON, J. (1981), S. 150; SEICHE (1990), S. 19; ABEL, D. (1990), S. 12. Sie laufen darauf hinaus, nicht nur die Erreichbarkeit einer einzelnen ausgezeichneten Markierung M_f von allen anderen erreichbaren Markierungen aus sicherzustellen. Statt dessen wird die markierungsspezifische Eigenschaft der M_f -Reversibilität zur netzspezifischen Eigenschaft der Reversibilität bezüglich aller Markierungen verallgemeinert, die sich von der Ausgangsmarkierung aus durch Schaltprozesse erreichen lassen. Dann ist es möglich, von jeder erreichbaren Markierung eines Netzes aus jede andere erreichbare Markierung zu erreichen. Für diese universelle Netzreversibilität $REV(SN)$ muß also gelten:

$REV(SN)$

$$:\Leftrightarrow \forall(M_f \in RM(M_0)) \forall(M_f \in RM(M_0)): M_f \in RM(M_f)$$

Die universelle Netzreversibilität wird zumeist nur kurz als Reversibilität - oder auch als Wiederholbarkeit - von Netzen angesprochen. Der Deutlichkeit halber wird sie aber hier M-Reversibilität genannt, um sie von den oben thematisierten M_f - und M_0 -Reversibilitäten abzuheben. Die M-Reversibilität interessiert in dieser Arbeit nicht weiter, weil für die später behandelten Koordinierungsaufgaben in Flexiblen Fertigungssystemen nur die garantierte Rückkehrmöglichkeit zur Ausgangsmarkierung M_0 von Bedeutung ist. Diese Fokussierung auf den Ausgangszustand eines Netzmodells findet sich auch bei ABEL, D. (1990), S. 44, obwohl er zunächst im Sinne der M-Reversibilität argumentiert hat (S. 12).

Darüber hinaus läßt sich zeigen, daß M- und M_0 -Reversibilität äquivalente Netzeigenschaften darstellen. Einerseits ist es trivial, daß jedes M-reversible auch ein M_0 -reversibles Netz sein muß. Denn in einem M-reversiblen Netz ist jede erreichbare Markierung von allen anderen erreichbaren Markierungen aus erreichbar. Also ist auch die Ausgangsmarkierung M_0 , die wegen $M_0 \in RM(M_0)$ selbst eine erreichbare Markierung darstellt, von allen anderen erreichbaren Markierungen aus erreichbar. Folglich ist das betrachtete Netz auch M_0 -reversibel. Andererseits trifft auch die Umkehrung zu, daß jedes M_0 -reversible Netz zugleich M-reversibel sein muß. Dies läßt sich auf indirekte Weise durch die Annahme des kontradiktorischen Gegenteils beweisen: Für ein beliebiges M_0 -reversibles Netz wird hypothetisch unterstellt, es sei nicht M-reversibel. Dann muß wegen fehlender M-Reversibilität mindestens eine erreichbare Markierung M_f mit $M_f \in RM(M_0)$ existieren, die sich von mindestens einer anderen erreichbaren Markierung M_i mit $M_i \in RM(M_0)$ nicht erreichen läßt. Da das Netz jedoch als M_0 -reversibel vorausgesetzt wurde, muß es möglich sein, von der kritischen Markierung M_f zur Ausgangsmarkierung M_0 zurückzukehren. Von dort aus läßt sich die Markierung M_i wegen $M_i \in RM(M_0)$ erreichen. Abermals infolge M_0 -Reversibilität kann zur Ausgangsmarkierung M_0 zurückgekehrt werden, um von dort aus wegen $M_f \in RM(M_0)$ schließlich zur kritischen Markierung M_f zu gelangen. Auf diese Weise ist es möglich, von der Markierung M_i über die Markierung M_0 die Markierung M_f zu erreichen. Dies widerspricht aber der zunächst angenommenen Hypothese, die Markierung M_f lasse sich von der Markierung M_i aus nicht erreichen. Folglich muß in einem M_0 -reversiblen Netz jede erreichbare Markierung von allen anderen erreichbaren Markierungen aus erreichbar sein. Daher ist jedes M_0 -reversible Netz zugleich auch M-reversibel; q.e.d.

Aufgrund der bewiesenen Äquivalenz zwischen M_0 - und M-Reversibilität ist es zulässig, zwischen beiden Reversibilitätsvarianten nicht weiter zu differenzieren. Auf diesen Sachverhalt wird später zurückgegriffen; vgl. z.B. die Erläuterung zu kondensierten Erreichbarkeitsgraphen.

67) Darauf wurde schon kurz zuvor hingewiesen. Die Ausgangsmarkierung M_0 stellt einen Heimzustand dar. Darüber hinaus ist die Ausgangsmarkierung M_0 per constructionem eine Livelockmarkierung. Sie konstituiert eine nicht-leere, im Regelfall vielelementige Menge von Livelockprozessen, die als zyklische Netzverhaltensweisen in der Ausgangsmarkierung starten und auch dorthin wieder zurückkehren. Diese Livelockprozesse können per definitionem beliebig oft wiederholt werden. Dies begründet die bereits oben festgestellte technische Robustheit von Produktionssystem(komponenten), die sich an Produktionsstörungen so anzupassen vermögen, daß sich Produktionsprozesse unbegrenzt fortsetzen lassen.

68) M_0 -reversible Netze werden fortan auch kurz als reversible Netze angesprochen. Sie werden oftmals als ordentliche ("proper") Netze, mitunter auch als zyklische Netze bezeichnet.

Vgl. zur Reversibilität (im Sinne der M_0 -Reversibilität), Ordentlichkeit oder Zyklizität von Netzen VALETTE (1977a), S. 76f.; VALETTE (1978b), S. 188f.; BOUSSIN (1978), S. 1530; TREMOLIERES (1979), S. 61; TOULOTTE (1979), S. 224; WINDISCH (1979), S. 16; AYACHE (1979a), S. 1052; ROUCAIROL (1982b), S. 16 (dort wird die M_0 -Reversibilität nicht direkt angesprochen, aber eine besonders übersichtliche Konstruktionsweise für M_0 -reversible Netze präsentiert); MONTEL (1983a), S. 295; ABEL, D. (1990), S. 44 u. 48.

Abweichende Definitionen der Netzordentlichkeit vertreten dagegen JANICKI (1978), S. 301; JANICKI (1981a), S. 460; DE CINDIO (1982), S. 274; DE CINDIO (1983a), S. 30.

69) Vgl. AYACHE (1979a), S. 1052; GRABOWSKI, J. (1980c), S. 18.

70) Dies trifft z.B. für zyklisch wiederholte Neuplanungen zu, die in konventionellen Produktionsplanungs- und -steuerungssystemen des öfteren praktiziert werden. Sie wurden bereits unter dem Aspekt angesprochen, daß ihre Produktionspläne "eingefroren" werden, solange die Abwicklung eines Auftragspakets noch andauert.

71) Die Ausgangsmarkierung eines M_0 -reversiblen Netzes stellt dann nicht nur einen Heimzustand, sondern zugleich auch die genau eine erwünschte Endmarkierung des Netzes dar. Bei allen Livelockprozessen, die als zyklische Netzverhaltensweisen in der Ausgangsmarkierung starten und dorthin wieder zurückkehren, handelt es sich um finale, aber nicht terminierende Prozesse.

5.2.3 Untersuchung von Netzeigenschaften

5.2.3.1 Überblick über Auswertungstechniken

Auswertungstechniken¹⁾ werden in dieser Arbeit als Komplexe aufgefaßt, die sich aus einem Analysekonzept, aus einer endlichen und nicht-leeren Menge von Analysemethoden und aus einer endlichen Menge von Analyseinstrumenten zusammensetzen. Analysekonzepte legen dar, wie die Aufgabe, eine Netzeigenschaft zu untersuchen, grundsätzlich bewältigt werden kann. Analysemethoden konkretisieren ein Analysekonzept so weit, daß sie die Vorgehensweise beschreiben, in der sich die betrachtete Untersuchungsaufgabe erfüllen läßt. Sie gestatten es, die Realisierungsmöglichkeit des betroffenen Konzeptaspekts intersubjektiv nachzuvollziehen²⁾. Analyseinstrumente unterstützen die Anwendung der Analysemethoden. Dabei handelt es sich in der Regel um Methodenimplementierungen auf Automatischen Informationsverarbeitungssystemen. Die Analyseinstrumente nehmen dann die Gestalt von Programmen an, die das methodenspezifische Analysewissen enthalten.

Auswertungstechniken für statische Netzeigenschaften werden grundsätzlich nicht betrachtet. Denn die Untersuchung, ob ein Netz eine statische Eigenschaft aufweist, stellt sich als trivial heraus. Aufgrund der topologiebezogenen Definition aller statischen Netzeigenschaften läßt sich aus der jeweils vorliegenden Netztopologie unmittelbar erkennen, welche statischen Eigenschaften von ihr erfüllt oder verletzt werden. Beispielsweise kann die statische Eigenschaft, daß ein Netz keine 1-Schleifen besitzt, direkt anhand der graphischen Repräsentation der Netztopologie überprüft werden.

Für die Untersuchung von dynamischen Netzeigenschaften bietet das Petrinetz-Konzept dagegen eine reichhaltige Sammlung unterschiedlicher Auswertungstechniken an. Sie lassen sich zu fünf Klassen zusammenfassen³⁾:

- ❑ die Deduktionsanalyse, bei der dynamische aus statischen Netzeigenschaften abgeleitet werden;
- ❑ die gleichungsbasierte Invariantenanalyse;
- ❑ die Reduktionsanalyse auf der Basis von Transformationsregeln;
- ❑ die graphenbasierte Erreichbarkeitsanalyse;
- ❑ die Simulationsanalyse, die sich darauf beschränkt, die Schaltregel des jeweils untersuchten Netzes anzuwenden.

Die fünf vorgenannten Analyseklassen verhalten sich keineswegs disjunkt. Sie typisieren lediglich die verfügbaren Auswertungstechniken nach dem Kriterium, jeweils gleichartige Untersuchungsansätze zu einer Gruppe zusammenzufassen. Der sachlogische Zusammenhang der Analyseklassen untereinander wird keinen weiteren Restriktionen unterworfen. Daher können die Analyseklassen in vielfältigen Beziehungen zueinander stehen. Beispielsweise bildet das Verhältnis der Invarianten- zur Deduktionsanalyse einen Spezialisierungszusammenhang. Einem Kombinationszusammenhang unterliegt dagegen die Reduktionsanalyse in bezug auf die vier anderen Analyseklassen. Erreichbarkeits- und Simulationsanalyse fallen sogar in einer gemeinsamen Schnittmenge zusammen, die aus der Konstruktion von Erreichbarkeitsgraphen einerseits und der erweiterten omnivalenten Simulationsanalyse andererseits besteht.

Anmerkungen zum Kapitel:

1) Auswertungstechniken werden in synonyme Weise auch als Untersuchungs- oder Analysetechniken bezeichnet. Der Begriff "Auswertungstechnik" knüpft unmittelbar an die modelltheoretische Unterscheidung zwischen den beiden Hauptphasen der Konstruktion und der Auswertung von Modellen an.

2) Die Beschreibung der aufgabenerfüllenden Vorgehensweise durch eine Methode wird im allgemeinen nur so weit getrieben, daß sich die Konzeptrealisierung überprüfen läßt. Daher bleibt die Vorgehensbeschreibung zumeist immer noch recht abstrakt. Es steht jedoch nichts im Wege, eine Methode auf verschiedenen Abstraktionsniveaus zu formulieren. Bei zunehmender Konkretisierung werden die Methoden oftmals auch als Algorithmen bezeichnet. Eine scharfe Unterscheidung zwischen Methoden und Algorithmen wird in dieser Arbeit nicht vorgenommen. Statt dessen werden beide Bezeichnungen synonym verwendet. Auf mögliche Präzisierungen des Algorithmusbegriffs, die zu einer Abgrenzung vom weniger präzise festgelegten Methodenbegriff herangezogen werden könnten, wird zwar in einer anderen Anmerkung hingewiesen. Sie werden jedoch nicht weiter verfolgt, da die hier behandelten Aspekte durch eine solche präzisierende Differenzierung in keiner Weise befruchtet werden. Wenn eine Methode so weit konkretisiert ist, daß sie sich durch ein Verrichtungssystem - vor allem: ein Automatisches Informationsverarbeitungssystem - ohne weitere Erläuterungen unmittelbar ausführen läßt, wird von einer Prozedur gesprochen.

3) Die Analysebezeichnungen werden als generische Begriffe verwendet. So werden beispielsweise unter dem Begriff der Deduktionsanalyse alle Analysetechniken zusammenfaßt, die zum Typ "der" Deduktionsanalyse gehören.

5.2.3.2 Die Deduktionsanalyse

Die Deduktionsanalyse¹⁾ beruht auf dem Konzept, dynamische Netzeigenschaften ausschließlich aus Eigenschaften der statischen Netzstruktur abzuleiten. Dahinter steht die praktische Erfahrung, daß die analytische Berücksichtigung dynamischer Netzaspekte außerordentlich hohen Aufwand bereitet, sobald die Schaltregel von Netzen beachtet werden muß. Ihr prozeduraler, bedingter, permissiver Charakter wurde an früherer Stelle ausführlich dargelegt. Er führt dazu, daß sich die Schaltregel in analytische Kalküle im allgemeinen nur mit großen Anstrengungen integrieren läßt. Daher kann die Analyse von Netzen erheblich vereinfacht werden, falls bei der Untersuchung dynamischer Netzeigenschaften die Schaltregel nicht berücksichtigt zu werden braucht. Dies ist zumindest dann möglich, wenn es gelingt, Erkenntnisse über die Netzdynamik aus Eigenschaften der statischen Netzstruktur abzuleiten.

Doch leidet der Ansatz der Deduktionsanalyse unter einer fundamentalen konzeptionellen Inadäquanz. Sie beruht auf der mangelnden Berücksichtigung der Ausgangsmarkierung eines Netzes²⁾. Diese Ausgangsmarkierung stellt eine wesentliche Determinante der dynamischen Netzstruktur dar. Es kann nicht ausgeschlossen werden, daß sich eine dynamische Netzeigenschaft verändert, wenn die Ausgangsmarkierung eines Netzes modifiziert wird³⁾. Da die Deduktionsanalyse ausschließlich die statische Netzstruktur erfaßt, ist es unmöglich, solche markierungsinduzierten Variationen von dynamischen Netzeigenschaften zu berücksichtigen.

Allerdings eignet sich die Deduktionsanalyse für die "strukturelle" Untersuchung von dynamischen Netzeigenschaften⁴⁾. Dabei interessieren jedoch nur solche dynamischen Netzeigenschaften, deren Vorliegen oder deren Fehlen unabhängig davon ist, welche zulässige Ausgangsmarkierung einem - im übrigen unveränderten - Netz zugeordnet wird. Es werden nur *notwendige* Bedingungen aufgestellt, die von der statischen Netzstruktur erfüllt werden müssen, damit die intendierte dynamische Netzeigenschaft vorliegen *kann*⁵⁾. Daraus läßt sich nur die Erkenntnis ziehen, daß ein Netz die dynamische Netzeigenschaft mit Sicherheit nicht besitzt, falls es die notwendige Anforderung an die statische Netzstruktur nicht erfüllt. Wenn hingegen ein Netz der notwendigen Bedingung gerecht wird, dann folgt daraus noch kein Wissen darüber, ob das Netz die dynamische Eigenschaft unter einer bestimmten Ausgangsmarkierung tatsächlich besitzt. Dies könnte erst dann zuverlässig überprüft werden, wenn eine *hinreichende* Bedingung bekannt wäre, deren Erfüllung durch die statische Netzstruktur das Vorliegen der interessierenden dynamischen Netzeigenschaft garantierte.

Solche *hinreichenden* Bedingungen für dynamische Netzeigenschaften vermag die strukturelle Analyse dynamischer Netzeigenschaften im allgemeinen nicht zu liefern. Denn ihre Untersuchung von Netzen für *jede* zulässige Ausgangsmarkierung schließt es in der Regel aus, das Vorliegen einer dynamischen Netzeigenschaft in Abhängigkeit von einer *bestimmten* Ausgangsmarkierung zu beurteilen. Nur in Sonderfällen kann nachgewiesen werden, daß eine dynamische Eigenschaft von Variationen der Ausgangsmarkierung überhaupt nicht abhängt⁶⁾. Dann ist es möglich, auch hinreichende Bedingungen für das Vorliegen einer dynamischen Netzeigenschaft so zu formulieren, daß ausschließlich auf Eigenschaften der statischen Netzstruktur Bezug genommen wird⁷⁾. Diese *Sonderfälle* eignen sich jedoch nicht als Fundament für eine *allgemein* einsetzbare Analysetechnik. Daher bleibt die strukturelle Untersuchung dynamischer Netzeigenschaften unbefriedigend.

Jedoch läßt sich die Deduktionsanalyse mit den Ausgangsmarkierungen der jeweils untersuchten Netze kombinieren. Dadurch wird zwar der Anspruch aufgegeben, dynamische Netzeigenschaften *ausschließlich* aus Eigenschaften der statischen Netzstruktur abzuleiten. Doch wird zugleich die oben angesprochene Schwierigkeit überwunden, daß dynamische Netzeigenschaften sensitiv auf die Variation von Ausgangsmarkierungen reagieren können. Darüber hinaus befindet sich die Ausgangsmarkierung eines Netzes ohnehin in einer Übergangszone, in der sich

der statische oder dynamische Charakter von Strukturkomponenten nicht eindeutig entscheiden läßt⁸⁾. Für die praktische Anwendung einer Analysetechnik ist die Frage, ob die Analyse mit rein statischen Netzeigenschaften auskommt oder auch dynamische Aspekte einbeziehen muß, ohnehin unerheblich, solange die Unterscheidung zwischen statischer und dynamischer Netzstruktur rein definitorischen Charakter trägt. Dies ist hinsichtlich der Ausgangsmarkierung tatsächlich der Fall. Denn die praktischen Schwierigkeiten bei der Analyse dynamischer Netzeigenschaften setzen - wie bereits dargelegt wurde - erst dann ein, wenn die Schaltregel von Netzen berücksichtigt werden muß. Dies ist jedoch noch nicht der Fall, wenn die statische Netzstruktur *nur* um den Aspekt der Ausgangsmarkierungen bereichert wird. Daher wird fortan eine derart erweiterte Deduktionsanalyse betrachtet.

Es ist möglich, sowohl notwendige als auch hinreichende Bedingungen für die Erfüllung einer dynamischen Netzeigenschaft anzugeben, die sich nur auf Eigenschaften der statischen Netzstruktur und die zusätzlich berücksichtigte Ausgangsmarkierung beziehen. Die Untersuchung von statischer Netzstruktur und Ausgangsmarkierung erstreckt sich jedoch bisher nur auf Netzklassen, deren zulässige Topologien besonders rigiden Einschränkungen unterliegen⁹⁾. Dazu gehören vor allem Erkenntnisse über notwendige und hinreichende Bedingungen für die Lebendigkeit von Synchronisationsnetzen¹⁰⁾. Auf die starken Restriktionen, denen die Topologien von Synchronisationsnetzen unterworfen sind, wurde bereits hingewiesen. Bisher ließen sich diese Lebendigkeitserkenntnisse, die schon zu Beginn der siebziger Jahre gewonnen wurden, noch nicht einmal auf Stelle/Transition-Netze mit unbeschränkter Netztopologie übertragen¹¹⁾. Eine Ausweitung ihres Geltungsbereichs auf die wesentlich komplexeren Höheren Netze erscheint derzeit ausgeschlossen. Daher sieht der Verf. in dieser Richtung keinen fruchtbaren Ansatz, um eine zufriedenstellende Analysetechnik für Synthetische Netze bereitzustellen¹²⁾.

Anmerkungen zum Kapitel:

1) Vgl. dazu die Quellen, die in den nachfolgenden Anmerkungen angeführt werden. Darüber hinaus lag das Konzept der Deduktionsanalyse schon der stellenbezogenen Definition von Deadlocks zugrunde. Auch dort bestand die Absicht, einen dynamischen Netzaspekt - die Deadlocks - so festzulegen, daß nur auf statische Netzkonstituenten Bezug genommen wird.

2) Dieser Erfassungsmangel betrifft nur diejenige "reine" Form der Deduktionsanalyse, die von *allen* Aspekten der dynamischen Netzstruktur absieht *und* zugleich die Ausgangsmarkierung als eine Komponente der Netzdynamik betrachtet. Diese beiden Voraussetzungen brauchen nicht geteilt zu werden. An anderer Stelle werden Ausweitungen der Deduktionsanalyse diskutiert, die möglich werden, wenn ihre "reine" Form mit der Berücksichtigung von Ausgangsmarkierungen kombiniert wird.

3) Ein anschauliches Beispiel dafür präsentiert BAUER,F. (1981), S. 407, Abb. 6.13: Ein einfaches Netz erweist sich als schwach lebendig, wenn eine bestimmte Ausgangsmarkierung vorliegt. Wird diese Ausgangsmarkierung lediglich dadurch verändert, daß eine Marke von einer Stelle auf eine andere Stelle verschoben wird, so verhält sich das modifizierte Netz nicht mehr schwach lebendig. Durch eine simple Variation der Ausgangsmarkierung wird - bei Konstanz der gesamten statischen Netzstruktur - aus einem deadlockfreien Netz ein deadlockbehaftetes Netz. Auf diese Weise wird eine wichtige dynamische Netzeigenschaft in ihr Gegenteil verkehrt, obwohl die statische Netzstruktur invariant erhalten bleibt.

Es erfolgte sogar ein Aufruf, Netze zu kreieren, die bei Veränderung ihrer Ausgangsmarkierungen ihre Lebendigkeit verlieren; vgl. CHRZASTOWSKI-WACHTEL (1983b), S. 22. Entsprechende Resultate finden sich bei THIAGARAJAN (1983a), S. 22f.; GHOSH,S. (1983), S. 22.

Weitere Hinweise auf die Abhängigkeit dynamischer Netzeigenschaften von der jeweils vorausgesetzten Ausgangsmarkierung finden sich bei LAUTENBACH (1975a), S. 10ff.; WINDISCH (1979), S. 76; REISIG (1979d), S. 37; REISIG (1983c), S. 4.

4) Vgl. zu solchen strukturellen Untersuchungen von dynamischen Netzeigenschaften - vor allem der Netzlebendigkeit -, die nur auf die Aspekte der statischen Netzstruktur Bezug nehmen, HERZOG,O. (1977), S. 213ff.; SIFAKIS (1978), S. 474ff.; MEMMI (1978a), S. 126 u. 134f.; MEMMI (1979), S. 102f.; MEMMI (1982), S. 172ff.; REISIG (1982a), S. 131.

5) Der Charakter von strukturellen Untersuchungen dynamischer Netzeigenschaften, lediglich notwendige Bedingungen zu liefern, läßt sich z.B. klar erkennen bei SIFAKIS (1978), S. 476 u. 479; MEMMI (1978a), S. 126 u. 134; MEMMI (1979), S. 102f.; MEMMI (1982), S. 175f.

6) Dies ist z.B. der Fall, wenn ein Netz für jede zulässige Ausgangsmarkierung ein totes Netz darstellt. Eine *hinreichende* Bedingung dafür, die sich nur auf statische Netzeigenschaften bezieht, lautet beispielsweise für den leicht zu überblickenden Fall von Stelle/Transition-Netzen: Alle Ausgangskanten (s_m, t_n) aller Stellen s_m eines Netzes besitzen ein Gewicht $G(s_m, t_n)$, das größer ist als die Markenkapazität $K(s_m)$ ihrer adjazenten Stellen. Dann folgt aus der Schaltregel, daß unter jeder zulässigen Ausgangsmarkierung, die auf keiner Stelle mehr Marken ablegt, als von deren Markenkapazität erlaubt ist, keine Transition aktiviert ist. Folglich ist ein solches Netz für *jede* zulässige Ausgangsmarkierung tot. Der praktische Erkenntniswert dieses Sonderfalls konvergiert jedoch gegen Null. Denn es ist nicht zu ersehen, welche Realprobleme derart gewichtete Netze repräsentieren könnten. Diese toten Netze besitzen die Qualität theoretischer Konstrukte ohne jede realproblembezogene Relevanz.

7) Dafür wurde in der voranstehenden Anmerkung ein konkretes Beispiel vorgelegt.

8) Vgl. dazu die Diskussion, ob die Ausgangsmarkierung eines Netzes entweder zu dessen statischer oder aber zu dessen dynamischer Struktur gerechnet werden müsse.

9) Vgl. zu Bedingungen, die nur auf die statischen Strukturen und die Ausgangsmarkierungen von Netzen mit eingeschränkter Topologie (Zustandsmaschinen-Netze, Free-Choice-Netze usw.) Bezug nehmen, aber für dynamische Eigenschaften - vor allem die Netzlebendigkeit - zumindest hinreichend, seltener auch notwendig sind, HACK,M. (1972), S. 19, 43ff., 57ff. u. 62f.; HACK,M. (1974a), S. 3; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 12ff. u. 173ff.; BEST,E. (1974), S. 72; LAUTENBACH (1975a), S. 9f., 12, 14 u. 18; MARIN (1975), S. 2/7 u. 2/10ff.; MAYR,E. (1975), S. 59ff., insbesondere S. 71ff.; SHAPIRO,R. (1975b), S. 144; COOPRIDER (1976), S. 23; HERZOG,O. (1976a), S. 310ff., insbesondere S. 312; JONES,N. (1977), S. 281; SCHMID,H. (1978), S. 6ff.; BAUMGARTEN (1978), S. 45ff.; MEMMI (1978a), S. 139ff.; LAUER (1978a), S. 320f. u. 323; PRINOTH (1978a), S. 70; PRINOTH (1978b), S. 25; LAUTENBACH (1979a), S. 99ff.; WINDISCH (1979), S. 61ff.; MEMMI (1979), S. 104; GRIESE,W. (1979), S. 1, 6, 10 u. 13ff.; PAKAS-SKEWES (1979), S. 34; REISIG (1980a), S. 250; JANTZEN (1980a), S. 198ff.; LAUER (1980), S. 503; STARKE (1980), S. 121ff. u. 132; GRIESE,W. (1980), S. 257, 261 u. 263; HRUSCHKA (1980a), S. 275; REISIG (1982a), S. 126 u. 133; SANDRING (1982), S. 219ff.; REISIG (1983c), S. 4; REISIG (1986a), S. 121ff. u. 127f. Vgl. darüber hinaus die Quellen in der nachfolgenden Anmerkung, die den gleichen Aspekt für die spezielle Klasse der Synchronisationsnetze beleuchten.

10) Vgl. HOLT,A. (1970a), S. 26f.; HOLT,A. (1970b), S. 100; COMMONER (1971a), S. 513; GENRICH (1971b), S. 6ff.; GENRICH (1973a), passim, insbesondere S. 148f. u. 157f.; JUMP (1973), S. 70; BERNSTEIN (1973), S. 10; MASSACHUSETTS COMPUTER ASSOCIATES (1974c), S. 29; JUMP (1975), S. 598; COOPRIDER (1976), S. 23; AMIN (1976), S. 296; SCHESCHONK (1977), S. 51ff.; MURATA,TA. (1977a), S. 413; MURATA,TA. (1977b), S. 401; MURATA,TA. (1977c), S. 7; BAUMGARTEN (1978), S. 40ff.; LAUER (1978a), S. 313; LAUTENBACH (1979a), S. 95; STARKE (1980), S. 122; LAUER (1980), S. 503; JANTZEN (1980a), S. 204; WINAND (1980), S. 1249; MÜLLER,B. (1981), S. 204; REISIG (1986a), S. 127f.

11) BEST,E. (1975a), S. 192, bedauert, daß hinreichende Bedingungen für die Lebendigkeit eines Netzes nur für bestimmte Netzklassen, nicht aber für Netze im allgemeinen bekannt sind, wenn diese Bedingungen nur an statische Netzeigenschaften anknüpfen dürfen. LAUER (1978a), S. 311, weist darauf hin, daß es unmöglich ist, die dynamische Eigenschaft "Deadlock" für Systeme (Netze) ausschließlich anhand statischer Systemeigenschaften zu überprüfen, solange diese Systeme keinen besonderen Einschränkungen unterworfen wurden. Die Gültigkeit beider Feststellungen besteht bis heute fort.

12) Gleicher Ansicht ist BEST,E. (1980c), S. 304; BEST,E. (1982d), S. 165. Er glaubt nicht, daß sich dynamische Netzeigenschaften wie die Lebendigkeit jemals durch die Analyse der rein statischen Netzstruktur (und der Ausgangsmarkierungen) erkennen oder widerlegen lassen. Im selben Sinne lassen sich die Ausführungen von BEST,E. (1975a), S. 192, und LAUER (1978a), S. 311, auslegen, sofern sie jeweils auf Netze bezogen werden, die keiner topologischen Einschränkung unterliegen.

5.2.3.3 Die Invariantenanalyse

Die Invariantenanalyse¹⁾ spielt für die Analyse von Synthetischen Netzen keine Rolle. Dies bedarf einer ausführlicheren Rechtfertigung, weil diese Analysetechnik in der Netzliteratur breite Beachtung findet²⁾.

Die Invariantenanalyse stimmt mit dem Konzept der Deduktionsanalyse, aus Untersuchungen der statischen Netzstruktur Erkenntnisse über dynamische Netzeigenschaften abzuleiten, weitgehend überein. Denn seitens der Invariantenanalyse wird ausschließlich auf die Inzidenzmatrix eines Netzes Bezug genommen. Diese Matrix faßt nahezu alle Informationen über die topologische Netzstruktur in komprimierter Form zusammen. Zusätzlich enthält sie die Angaben über alle Kantengewichte. Damit spiegelt sie weitgehend die statische Netzstruktur wieder³⁾. Dennoch fällt die Invarianten- mit der Deduktionsanalyse nicht zusammen. Denn die Invariantenanalyse beruht ausschließlich auf dem methodischen Ansatz, Erkenntnisse über dynamische Netzeigenschaften durch die Auswertung linearer homogener diophantischer Gleichungssysteme zu gewinnen. Die Deduktionsanalyse kennt dagegen eine solche methodische Fokussierung nicht. In dieser Hinsicht kann die Invarianten- als ein Sonderfall der Deduktionsanalyse aufgefaßt werden.

Da die Invariantenanalyse dem Untersuchungskonzept der Deduktionsanalyse im Prinzip folgt, gelten die Schwierigkeiten, die bei der Deduktionsanalyse aus ihrer Bezugnahme auf die statische Netzstruktur resultieren, für die Invariantenanalyse in gleicher Weise:

- Invariantenanalysen eignen sich zwar für die strukturelle Untersuchung von dynamischen Netzeigenschaften⁴⁾. Doch aus der statischen Netzstruktur lassen sich im allgemeinen⁵⁾ nur notwendige Bedingungen für das Vorliegen einer dynamischen Netzeigenschaft ableiten⁶⁾. Dies bleibt jedoch unbefriedigend, weil die Erfüllung einer notwendigen Bedingung noch nicht ausreicht, um zu wissen, ob die untersuchte dynamische Netzeigenschaft auch tatsächlich vorliegt⁷⁾. Lediglich das Nichtvorliegen einer dynamischen Netzeigenschaft steht definitiv fest, falls die zugehörige notwendige Bedingung verletzt ist.
- Invariantenanalysen lassen sich zwar mit dem Wissen über die Ausgangsmarkierungen der untersuchten Netze kombinieren⁸⁾. Dann können sowohl notwendige als auch hinreichende Bedingungen für das Vorliegen einer dynamischen Netzeigenschaft angegeben werden. Diese Bedingungen beziehen sich nur auf Aspekte der statischen Netzstruktur und der Ausgangsmarkierung⁹⁾. Doch die Netzklassen, für die solche notwendigen und hinreichenden Bedingungen bekannt sind, unterliegen rigiden Einschränkungen ihrer zulässigen Topologien¹⁰⁾.

Diese Defizite gelten für alle Analysetechniken, die dynamische auf statische Netzeigenschaften und Ausgangsmarkierungen zurückführen. Daneben treten noch weitere, jedoch invariantenspezifische Analyseprobleme. Um sie präzise identifizieren zu können, wird der konzeptionelle Ansatz der Invariantenanalyse näher ausgeführt.

Mit Hilfe der Invariantenanalyse wird untersucht, welche Größen unter allen erreichbaren Markierungen eines Netzes unverändert - also "invariant" - gelten. Falls sich solche invarianten Größen identifizieren lassen, verhalten sie sich per definitionem invariant gegenüber allen Markierungsveränderungen, die in einem Netz durch Ausführen von Schaltprozessen hervorgerufen werden können. Die derart ausgezeichneten Größen werden kurz als Invarianten des jeweils untersuchten Netzes bezeichnet.

Die formale Definition von Invarianten knüpft grundsätzlich an der Inzidenzmatrix \underline{C} eines Netzes an. Die Inzidenzmatrix wurde für Stelle/Transition-Netze schon an früherer Stelle eingeführt. Auf ihrer Grundlage lassen sich für Stelle/Transition-Netze die Invarianten in besonders einfacher Weise definieren. Bei Höheren Netzen - insbesondere auch bei Synthetischen Netzen -

bereitet die Definition von Invarianten dagegen größeren formalen Aufwand. Die nachfolgenden Erörterungen werden jedoch schon anhand der Stelle/Transition-Netze erhebliche konzeptionelle Defizite¹¹⁾ der Invariantenanalyse aufzeigen. Daher wird hier darauf verzichtet, die kompliziertere Invariantendefinition für Höhere Netze im einzelnen auszubreiten¹²⁾.

In Stelle/Transition-Netzen stellen Invarianten ganzzahlige Spaltenvektoren \underline{i}_h dar, die sich durch eine spezielle formale Eigenschaft auszeichnen: Ihr Vektorprodukt mit der Inzidenzmatrix \underline{C} eines Netzes ergibt jeweils den Nullvektor $\underline{0}$. Invarianten sind also durch Vektorgleichungen definiert. Dabei lassen sich T- und S-Invarianten \underline{i}_h bzw. \underline{i}_h unterscheiden je nachdem, ob es sich um einen transitionenbezogenen N-stelligen Spaltenvektor oder aber um einen stellenbezogenen M-stelligen Spaltenvektor handelt. Beide Fälle sind mit Ga als Menge aller ganzen Zahlen in analoger Weise festgelegt durch¹³⁾:

a) für T-Invarianten¹⁴⁾:

$$\underline{C} \cdot \underline{i}_h = \underline{0}_M$$

$$\text{mit: } \underline{i}_h^{\text{tr}} = (i_{h,n} : n = 1, \dots, N \wedge i_{h,n} \in Ga)$$

b) für S-Invarianten¹⁵⁾:

$$\underline{C}^{\text{tr}} \cdot \underline{i}_h = \underline{0}_N$$

$$\text{mit: } \underline{i}_h^{\text{tr}} = (i_{h,m} : m = 1, \dots, M \wedge i_{h,m} \in Ga)$$

Die voranstehenden Vektorgleichungen besitzen stets die trivialen Lösungen $\underline{i}_h = \underline{0}_M$ und $\underline{i}_h = \underline{0}_N$. Daher wird im Regelfall nur die Existenz von nicht-trivialen Invarianten untersucht.

Darüber hinaus werden T-Invarianten nur dann näher betrachtet, wenn es sich um semi-positive Invarianten handelt¹⁶⁾. Sie erfüllen die strengere Anforderung, daß alle Komponenten $i_{h,n}$ ihres Spaltenvektors \underline{i}_h nicht-negative Ganzzahlen darstellen und mindestens eine dieser Komponenten sogar eine positive Ganzzahl ist. Dann können T-Invarianten als Schaltvektoren interpretiert werden. Jede ihrer Komponenten $i_{h,n}$ gibt für genau eine Transition t_n aus der Transitionenmenge T des jeweils untersuchten Netzes an, wie oft diese Transition in einer denkmöglichen Schaltfolge geschaltet wird¹⁷⁾. Dabei repräsentiert dieselbe T-Invariante den gemeinsamen Schaltvektor einer Klasse von Schaltfolgen¹⁸⁾. Zu dieser Klasse gehören alle Schaltfolgen, deren Schaltvektoren mit der semi-positiven T-Invariante zusammenfallen. Die klassenzugehörigen Schaltfolgen unterscheiden sich daher nicht in den *Anzahlen* der Schaltakte ihrer Transitionen. Dagegen können die - nebenläufigen oder sequentiellen - *Anordnungen* der Schaltakte von Schaltfolge zu Schaltfolge variieren. Der Schaltvektor einer T-Invariante ist also "blind" gegenüber den folgeninternen Schaltaktanordnungen.

Es ist nicht sichergestellt, daß die denkmöglichen Schaltfolgen, deren Schaltaktanzahlen eine semi-positive T-Invariante wiedergibt, im zugrundeliegenden Netz auch tatsächlich ausgeführt werden können¹⁹⁾. Denn in die Definition der T-Invarianten fließen weder Netzmarkierungen noch Aktivierungsbedingungen ein. Daher kann eine semi-positive T-Invariante den gemeinsamen Schaltvektor sowohl von zulässigen als auch von unzulässigen Schaltfolgen²⁰⁾ darstellen²¹⁾. Falls bekannt ist, daß mindestens eine Schaltfolge aus der Schaltfolgenklasse einer semi-positiven T-Invariante im zugrundeliegenden Netz zulässig ist, so wird kurz von einer zulässigen T-Invariante gesprochen²²⁾. Wenn jedoch feststeht, daß keine Schaltfolge aus der Schaltfolgenklasse einer semi-positiven T-Invariante im Netz ausgeführt werden kann, dann liegt eine unzulässige T-Invariante vor²³⁾. Schließlich wird ohne präzisierendes Attribut so lange nur von einer T-Invariante gesprochen, wie unbekannt ist, ob es sich um eine entweder zulässige oder aber unzulässige T-Invariante handelt.

Für jede zulässige Schaltfolge, die unter einer erreichbaren Markierung M_r aktiviert ist und deren Schaltvektor durch eine semi-positive T-Invariante wiedergegeben wird, gilt: Das Ausführen der aktivierten Schaltfolge reproduziert die Markierung M_r in nicht-trivialer Weise²⁴⁾. Die Netzmarkierung M_r bleibt also unverändert ("invariant") erhalten, obwohl die Schaltfolge durch jeden ihrer Schaltschritte zu vorübergehenden Veränderungen der aktuellen Netzmarkierungen führt. Aber die Markierungsveränderungen wirken insgesamt so zusammen, daß am Ende der Schaltfolgenausführung wieder die ursprünglich vorgegebene Markierung M_r vorliegt. Es wird auch kurz davon gesprochen, daß die T-Invariante gestattet, die Markierung M_r zu reproduzieren.

Solange die Zulässigkeit einer semi-positiven T-Invariante noch nicht bekannt ist, trifft nur ein schwächerer Sachverhalt zu: Wenn die Ausführung einer Schaltfolge eine erreichbare Markierung in nicht-trivialer Weise reproduzieren soll, dann muß eine semi-positive T-Invariante existieren, die den Schaltvektor der reproduzierenden Schaltfolge darstellt. Falls keine solche T-Invariante existiert, steht zugleich auch fest, daß es keine reproduzierende Schaltfolge für eine beliebige erreichbare Markierung geben kann. Die Existenz einer semi-positiven T-Invarianten ist also nur die *notwendige* - aber keineswegs hinreichende - Voraussetzung für die nicht-triviale Reproduktion einer erreichbaren Markierung²⁵⁾. *Hinreichend* für die nicht-triviale Reproduktion einer erreichbaren Markierung M_r ist dagegen erst die Erfüllung von zwei Voraussetzungen: Erstens muß mindestens eine semi-positive T-Invariante existieren. Zweitens muß mindestens diese eine T-Invariante der Schaltvektor einer zulässigen Schaltfolge sein, die unter der erreichbaren Markierung M_r aktiviert ist.

Für nicht-triviale S-Invarianten \underline{is}_h läßt sich dagegen zeigen²⁶⁾, daß das Vektorprodukt $\underline{is}_h^{tr} \bullet \underline{M}_r$ für alle erreichbaren Markierungen M_r eines Netzes denselben konstanten ("invarianten") skalaren Wert "mi" annimmt: $\underline{is}_h^{tr} \bullet \underline{M}_r = mi$ für alle $M_r \in RM(M_0)$. Dabei wird der konstante Skalar "mi" durch die S-Invariante \underline{is}_h selbst und die Ausgangsmarkierung \underline{M}_0 des Netzes festgelegt als das Vektorprodukt: $mi = \underline{is}_h^{tr} \bullet \underline{M}_0$. Folglich muß für jede nicht-triviale S-Invariante \underline{is}_h mit $\underline{is}_h \neq \underline{0}_M$ gelten:

$$\exists(mi \in \mathcal{N}_0) \forall(M_r \in RM(M_0)): \underline{is}_h^{tr} \bullet \underline{M}_r = mi = \underline{is}_h^{tr} \bullet \underline{M}_0$$

Für alle nicht-trivialen S-Invarianten, die für ein Netz ermittelt werden können, folgt daraus: Alle denkmöglichen Markierungen M_r mit $\underline{is}_h^{tr} \bullet \underline{M}_r \neq \underline{is}_h^{tr} \bullet \underline{M}_0$ stellen keine Markierungen dar, die sich von der Ausgangsmarkierung \underline{M}_0 aus erreichen lassen²⁷⁾. Wesentlich ist, daß diese Erkenntnisse über die Nichterreichbarkeit von Markierungen *ohne* das Anwenden der Schaltregel unmittelbar aus der statischen Struktur eines Netzes und seiner Ausgangsmarkierung gewonnen werden.

Trotz ihrer formal übersichtlichen und kompakten Definition von Netzinvarianten leidet die Invariantenanalyse jedoch unter erheblichen Schwierigkeiten²⁸⁾. Sie erstrecken sich im wesentlichen²⁹⁾ auf folgende Defizite:

- ❑ Der grundsätzliche Ansatz der Invariantenanalyse ist konzeptionell unvollständig und inadäquat (konzeptionelles Defizit).
- ❑ Des öfteren bereitet es Schwierigkeiten, den zunächst rein formal definierten Netzinvarianten eine materielle Bedeutung im Sinne einer interessanten dynamischen Netzeigenschaft zuzuordnen (interpretatives Defizit).
- ❑ Netze mit beschränkten Markkapazitäten können nicht unmittelbar untersucht werden (kapazitives Defizit).
- ❑ Unreine Netze, die 1-Schleifen enthalten, werden ebenso wenig erfaßt wie Netze mit Informationsstellen und -kanten³⁰⁾ (topologisches Defizit).
- ❑ Die Reduzierung aller Schaltfolgen auf Schaltvektoren kann zu einem Informationsverlust führen (vektorielles Defizit).

- Für Höhere Netze sind keine universellen Methoden bekannt, um alle Invarianten aufzuspüren (Effektivitätsdefizit).

Das konzeptionelle Defizit der Invariantenanalyse beruht zunächst darauf, daß in den Definitionsgleichungen der T- und S-Invarianten ausschließlich auf diejenigen Netzkomponenten Bezug genommen wird, die durch die Inzidenzmatrix \underline{C} repräsentiert werden³¹⁾. Die Inzidenzmatrix ist jedoch grundsätzlich nicht in der Lage, die dynamische Struktur eines Netzes vollständig wiederzugeben. Diese prinzipielle dynamische Unvollständigkeit der Inzidenzmatrix läßt sich als Repräsentationslücke identifizieren. Sie gründet im eingeschränkten Formulierungspotential von Matrizenkalkülen.

Die Repräsentationslücke der Inzidenzmatrix betrifft die Aktivierungsbedingung von Transitionen. Diese Aktivierungsbedingung stellt eine herausragende Determinante der Schaltregel aller Netze dar. Sie beeinflußt daher das potentielle Schaltverhalten eines Netzes erheblich. Dennoch berücksichtigt jeder Koeffizient $c_{m,n}$ der Inzidenzmatrix \underline{C} ausschließlich die Schaltwirkung, die eine Transition t_n auf eine Stelle s_m ausübt³²⁾, falls die Transition tatsächlich geschaltet wird. Ob die Transition überhaupt geschaltet werden darf, hängt jedoch von der Erfüllung ihrer Aktivierungsbedingung ab. Diese Bedingungserfüllung fließt aber nicht in die Inzidenzmatrix eines Netzes - und somit auch nicht in die Definitionsgleichungen der Netzinvarianten - ein. Daher kann die Invariantenanalyse zu fehlerhaften Ergebnissen führen, die tatsächlich nicht erfüllte Netzeigenschaften vorspiegeln. Diese Gefahr droht immer dann, wenn eine dynamische Netzeigenschaft aufgrund der Schaltwirkungen von Transitionen vorliegen *könnte*, aber dem untersuchten Netz wegen der Verletzung mindestens einer Aktivierungsbedingung in Wirklichkeit nicht zukommt.

Ein Beispiel für die mangelhafte Berücksichtigung von Aktivierungsbedingungen wurde bereits oben angeführt: Eine semi-positive T-Invariante wird als Schaltvektor einer Klasse von Schaltfolgen interpretiert. Die Existenz dieser Schaltfolgenklasse *könnte* eine dynamische Netzeigenschaft darstellen oder zumindest das Fundament bilden, aus dem sich eine interessante dynamische Netzeigenschaft ableiten läßt. Es ist aber durchaus möglich, daß einzelne oder alle Schaltfolgen aus dieser Schaltfolgenklasse tatsächlich unzulässig sind, weil ihre Aktivierungsbedingungen nicht erfüllt werden. Dies läßt sich jedoch allein anhand der ermittelten T-Invariante nicht feststellen. Daher muß für jede Schaltfolge, deren Schaltvektor die T-Invariante darstellt, zusätzlich überprüft werden, ob die Aktivierungsbedingungen ihrer Transitionen wirklich erfüllt sind.

Das Unvermögen, mit Hilfe einer Inzidenzmatrix die Aktivierungsbedingungen von Transitionen zu berücksichtigen, läßt sich grundsätzlich nicht beseitigen. Denn das Formulierungspotential von Matrizenkalkülen läßt es nicht zu, die Koeffizienten von Matrizen in konditionaler Weise auszudrücken. Solche bedingten Matrixkoeffizienten wären aber erforderlich, um die Schaltregel von Netzen vollständig wiederzugeben: Der Koeffizient $c_{m,n}$ darf die Schaltwirkung einer Transition t_n bezüglich einer Stelle s_m nur dann abbilden, falls die Transition auch tatsächlich aktiviert ist. Dies wurde früher durch die Verwendung von *partiell definierten* Schaltregel-funktionen geleistet. Ihre Abbildungsvorschriften können nur dann angewendet werden, wenn die einschränkende Aktivierungsbedingung erfüllt ist. Matrizenkalküle sind jedoch nicht in der Lage, solche partiell definierten Funktionen oder gleichwertige konditionale Konstrukte in die Formulierung ihrer Matrixkoeffizienten aufzunehmen³³⁾. Daher läßt sich die Repräsentationslücke von Inzidenzmatrizen, die hinsichtlich der Berücksichtigung von Aktivierungsbedingungen besteht, niemals schließen.

Das konzeptionelle Defizit der Invariantenanalyse erstreckt sich aber nicht nur auf die unvollständige Erfassung der dynamischen Netzstruktur durch Inzidenzmatrizen. Hinzu kommt auch eine grundsätzliche Inadäquanz des zugrundeliegenden Gleichungskalküls. Die Vektorgleichungen, die Netzinvarianten definieren, lassen sich in skalarer Weise als lineare³⁴⁾ homogene³⁵⁾ diophantische³⁶⁾ Gleichungssysteme reformulieren. Solche Gleichungssysteme sind für

die Repräsentation statischer Wissensinhalte ausgelegt, die trotz aller verhaltensbedingten Netzveränderungen einem untersuchten Netz immer in der *gleichen* Weise - also "invariant" - zu kommen. Dies widerspricht aber dem Ziel, aus der Untersuchung des Verhaltenspotentials von Netzen deren dynamischen Eigenschaften zu erkennen. Das gilt zumindest in dem Ausmaß, wie sich diese Eigenschaften nicht von der Untersuchung konkreter Netzverhaltensweisen abkoppeln lassen.

Beispielsweise erscheint es unmöglich, mit Hilfe der Invariantenanalyse die R-Promptheit eines Netzes zu untersuchen, das unter Realzeitbedingungen auf alle Meldungen über Veränderungen in der realen Netzumwelt mit einer maximalen Reaktionsspanne antworten soll. Es liegt außerhalb des Erkenntnisinteresses dieser Arbeit, einen entsprechenden Unmöglichkeitsnachweis zu führen. Aber bereits der Umstand, daß bis heute noch keine Promptheitsuntersuchung für Netze auf der Basis einer Invariantenanalyse bekannt wurde, rechtfertigt den Verdacht, ein solches Unterfangen sei von vornherein zum Scheitern verurteilt. Ebenso läßt sich kein Ansatz erkennen, der es gestatten würde, mit Hilfe der Invariantenanalyse die Existenz von Schaltkonflikten zu untersuchen, die auf der Konkurrenz von Schaltschritten um knappe Ressourcen beruhen. Ressourcenbedingte Schaltkonflikte wurden jedoch als ein wesentlicher Aspekt der Modellierungsfähigkeit des Petrinetz-Konzepts herausgestellt.

Selbst wenn sich zeigen ließe, daß durch Hilfskonstruktionen alle interessierenden dynamischen Netzeigenschaften auf Gleichungssysteme zurückgeführt werden können³⁷⁾, so änderte dies nichts an der konzeptionellen Inadäquanz statischer Gleichungssysteme. Denn der Verf. betrachtet ein Konzept für die Untersuchung dynamischer Netzeigenschaften erst dann als adäquat, wenn es auf einer Wissensrepräsentation beruht, in der sich die dynamische Charakteristik der Eigenschaften explizit wiederfindet. In Netzen, die das Erkenntnisobjekt der Invariantenanalyse bilden, wird diese dynamische Qualität durch die Veränderungen von Netzmarkierungen und zugehörigen Faktenmengen konkretisiert. Solche Markierungsänderungen spiegeln sich jedoch in statischen *Gleichungssystemen* grundsätzlich nicht explizit wieder³⁸⁾.

Das interpretative Defizit³⁹⁾ der Invariantenanalyse ist eine Folge ihrer konzeptionellen Inadäquanz. Denn die dynamischen Netzeigenschaften auf der einen Seite und die statischen Gleichungssysteme auf der anderen Seite klaffen so weit auseinander, daß es oftmals erhebliche Schwierigkeiten bereitet, aus der Erkenntnis von Netzinvarianten auf interessante Netzeigenschaften zu schließen. T- und S-Invarianten, die eine Invariantenanalyse hervorbringt, stellen zunächst abstrakte formale Konstrukte ohne materiellen Gehalt dar. Es läßt sich ihnen nicht unmittelbar ansehen, welche Erkenntnisse über interessante dynamische Netzeigenschaften in den Netzinvarianten verborgen sein mögen. Daher erhebt sich die Frage, in welchem Ausmaß die Formalkonstrukte der Invarianten in der Lage sind, einen interessanten materiellen Beitrag zur Bewältigung von Modellierungsproblemen zu leisten.

Für semi-positive T-Invarianten wurde bereits oben eine grundsätzliche Interpretationsmöglichkeit aufgezeigt: Sie lassen sich als Schaltvektoren von Schaltvektoren verstehen, die eine beliebige Referenzmarkierung reproduzieren können. Aber diese Invarianteninterpretation vermag aus zwei Gründen noch nicht zu befriedigen. Erstens handelt es sich um keine zuverlässige Interpretation. Es wurde dargelegt, daß die T-Invarianten zwar Schaltvektoren von zulässigen Schaltfolgen darstellen können, aber keineswegs müssen. Denn die T-Invarianten stellen keine hinreichenden, sondern nur notwendige Voraussetzungen für die Reproduktion von Netzmarkierungen dar. Welche Erkenntnis über das Verhaltenspotential eines Netzes aus T-Invarianten gezogen werden soll, die zu unzulässigen Schaltfolgen führen, bleibt vollkommen offen. Zweitens liegt aber selbst dann, wenn der Nachweis einer zulässigen Schaltfolge gelingt, zunächst eine bedeutungslose formale Netzeigenschaft vor. Denn die Reproduktionsmöglichkeit beliebiger Netzmarkierungen stellt keinen Selbstzweck dar. Auf den ersten Blick scheint sie keiner derjenigen dynamischen Eigenschaften zu entsprechen, die an früherer Stelle als interessante Erkenntnisobjekte für die Untersuchung von Netzmodellen ausgezeichnet wurden.

Bei genauerem Hinsehen zeigt sich jedoch, daß tatsächlich mindestens⁴⁰⁾ eine materiell interessante Interpretation für semi-positive T-Invarianten existiert. Sie erstreckt sich auf die Netzeigenschaft der M_0 -Reversibilität. Mit Hilfe der M_0 -Reversibilität läßt sich z.B. die Garantie ausdrücken, daß ein modelliertes Produktionssystem immer wieder in einen ungestörten Ausgangszustand zurückkehren kann. Die M_0 -Reversibilität kann ebenso benutzt werden, um eine Korrektheitsanforderung an die Module aus einem modular gestalteten Gesamtmodell zu spezifizieren. Aus der Definition der M_0 -Reversibilität und aus der Charakteristik von semi-positiven T-Invarianten folgt unmittelbar: Die Existenz einer semi-positiven T-Invariante ist *notwendig* für die M_0 -Reversibilität eines Netzmodells oder -moduls⁴¹⁾.

Daher kann zumindest ein bescheidener, aber materiell gehaltvoller Schluß gezogen werden: Ein Netz, das keine semi-positive T-Invariante besitzt, ist mit Sicherheit nicht M_0 -reversibel. Leider schlägt jedoch auch hier durch, daß Invarianten in der Regel keine hinreichenden, sondern nur notwendige Bedingungen zu formulieren gestatten. Denn aus der Existenz einer semi-positiven T-Invarianten läßt sich keineswegs folgern, daß das betrachtete Netz die erwünschte Eigenschaft der M_0 -Reversibilität tatsächlich besitzt⁴²⁾. In der Modellierungspraxis bleibt es aber zumeist unbefriedigend, nur feststellen zu können, daß angestrebte Modelleigenschaften nicht verwirklicht wurden. Statt dessen wäre es willkommener, den Nachweis einer solchen Eigenschaft führen zu können. Aber gerade das gestattet die Invariantenanalyse zumeist nicht.

Weitere materiell gehaltvolle Anwendungen von semi-positiven T-Invarianten sind möglich, wenn es gelingt, die Invarianten mit interessanten Netzeigenschaften zu verknüpfen. Zu diesem Zweck kommen grundsätzlich Theoreme in Betracht, die seitens der Netztheorie diskutiert werden. Ein Beispiel lieferte bereits die zuvor erläuterte M_0 -Reversibilität⁴³⁾. Ein ähnliches Netztheorem läßt sich für Livelocks aufstellen: Jede zulässige semi-positive T-Invariante kennzeichnet mindestens eine Livelockmarkierung⁴⁴⁾. Zugleich weist sie alle potentiellen Livelock-Transitionen aus⁴⁵⁾. Sie drohen zu "verhungern", weil sie bei der Ausführung von Livelockprozessen, welche die Livelockmarkierung reproduzieren, niemals geschaltet werden. Weitere Netztheoreme befassen sich mit den Abhängigkeiten zwischen T-Invarianten und Netzlebendigkeit⁴⁶⁾. Schließlich werden später zwei Netztheoreme diskutiert⁴⁷⁾, die gestatten, die Inkonsistenz eines Netzmodells aufzuzeigen. Dabei werden zulässige semi-positive T-Invarianten betrachtet. Sie stellen die Schaltvektoren von Schaltfolgen dar, die in der Lage sind, die markenfreien Ausgangsmarkierungen der Netzmodelle zu reproduzieren.

Zwar konnten zuvor einige Beispiele für gehaltvolle Ausdeutungen von Invarianten aufgelistet werden. Dennoch bleibt das materielle Interpretationsangebot der Invariantenanalyse recht dürftig. Dabei sind vier Punkte festzuhalten:

- ❑ Bislang sind nur wenige Theoreme bekannt, in denen formal berechenbare Netzinvarianten mit materiell gehaltvollen Netzeigenschaften verknüpft werden. Diese Theoreme decken nur einige wenige derjenigen Eigenschaften von Netzmodellen ab, die in dieser Arbeit als interessant ausgezeichnet wurden. Für zahlreiche andere Netzeigenschaften - wie z.B. für die Finalität oder die Promptheit von Netzmodellen - fehlen dagegen entsprechende Netztheoreme.
- ❑ Zumeist werden nur notwendige Bedingungen für das Vorliegen interessanter Netzeigenschaften vorgelegt. Die praktisch bedeutsameren hinreichenden Bedingungen bleiben im allgemeinen unbekannt.
- ❑ In den wenigen Fällen, in denen hinreichende Bedingungen für interessante Netzeigenschaften formuliert werden, knüpfen sie an die Zulässigkeit der Invarianten an⁴⁸⁾. Die Zulässigkeit einer Invariante läßt sich aber mit Hilfe der Auswertungstechniken für Gleichungssysteme, von denen die Invariantenanalyse beherrscht wird, nicht überprüfen⁴⁹⁾.

- T-Invarianten spielen die weitaus dominierende Rolle. Auf sie beziehen sich alle oben vorgelegten Beispiele. Für S-Invarianten existieren nur einige wenige überzeugende, dann aber auch auf spezielle Fragestellungen beschränkte Interpretationsansätze⁵⁰⁾.

Daher besteht derzeit eine deutliche interpretative Kluft zwischen der Ermittlung von Netzinvarianten einerseits und der Erkenntnis von interessanten Netzeigenschaften andererseits. Besonders offensichtlich ist diese Interpretationslücke für S-Invarianten. Diese Interpretationsarmut von S-Invarianten führt dazu, daß in dieser Arbeit überwiegend auf Aspekte von T-Invarianten eingegangen wird⁵¹⁾.

Angesichts des interpretativen Defizits der Invariantenanalyse überrascht es, daß in der Netzliteratur die Ermittlung von Invarianten so große Beachtung findet⁵²⁾. Es drängt sich der Eindruck auf, die Technik der Invariantenanalyse werde angewendet, weil ihr formaler Apparat zur Verfügung steht. Dabei wird jedoch der Frage, welchen Erkenntnisgewinn diese Analysetechnik zu leisten vermag, kaum größere Beachtung gewidmet. Aus diesem Blickwinkel droht die Beschäftigung mit der Invariantenanalyse zu einem Selbstzweck zu entarten. Die ursprüngliche Motivation von Analysetechniken, mit ihrer Hilfe nicht-triviale Einsichten in Netzmodelle zu erlangen, tritt in den Hintergrund. Entsprechend selten finden sich in der Netzliteratur Beispiele dafür, wie durch die Ermittlung und Interpretation von Netzinvarianten konkrete Erkenntnisse über *interessante* Modelleigenschaften erzielt werden können⁵³⁾.

Das voranstehend erörterte interpretative Defizit der Invariantenanalyse resultierte aus ihrer konzeptionellen Inadäquanz. Aus der konzeptionellen Unvollständigkeit der Invariantenanalyse folgen dagegen ihr kapazitives und ihr topologisches Defizit. Gemeinsamer Ausgangspunkt dieser beiden Defizite ist der methodische Ansatz, die Anwendung der Invariantenanalyse von vornherein dort auszuschließen, wo die dynamische Unvollständigkeit von Inzidenzmatrizen die Erkenntnis dynamischer Netzeigenschaften mit Sicherheit verfälscht. Durch entsprechende Anwendungsprämissen werden notwendige, aber keineswegs hinreichende⁵⁴⁾ Bedingungen für die Korrektheit der Invariantenanalyse aufgestellt.

Die dynamische Unvollständigkeit von Inzidenzmatrizen kann zunächst immer dort zu Analysefehlern führen, wo das Schalten von Transitionen auf mindestens einer Stelle mehr Marken ablegt, als sich dort gemäß ihrer Markenzapazität höchstens befinden dürfen. Ein solches unzulässiges Schalten wird zwar durch die Aktivierungsbedingung in der Schaltregel von Netzen verhindert. Aber die Invariantenanalyse ist nicht in der Lage, diese einschränkende Wirkung der Aktivierungsbedingung zu berücksichtigen⁵⁵⁾. Statt dessen muß die Invariantenanalyse zu der "konzeptionellen Krücke" greifen, diese Möglichkeit unzulässiger Schaltwirkungen dadurch auszuschließen, daß für alle Stellen eines Netzes *unbeschränkte* Markenzapazitäten vorausgesetzt werden⁵⁶⁾. Durch diese Prämisse wird der Anwendungsbereich der Invariantenanalyse von vornherein auf unbeschränkte Netze reduziert. Sie läßt sich auf solche Netze, in denen mindestens eine Stelle eine beschränkte Markenzapazität besitzt, nicht unmittelbar anwenden⁵⁷⁾. Darin besteht das kapazitive Defizit der Invariantenanalyse.

Bei der Voraussetzung unbeschränkter Markenzapazitäten handelt es sich jedoch um eine "heroische" Prämisse. Sie erweist sich für die Modellierung realer Probleme in der Regel als inakzeptabel. Denn oftmals spielen reale Objekte eine Rolle, die - wie z.B. Lager oder Bearbeitungsstationen - jeweils nur eine endliche Kapazität besitzen. Aufgrund der früher gerechtfertigten Modellierungsprämisse, die Ausdrucksmächtigkeit eines Modells gegenüber Aspekten der Modellauswertung zu bevorzugen, kann auf die Möglichkeit, begrenzte Objektkapazitäten abzubilden, grundsätzlich nicht verzichtet werden. Die unmittelbare Modellierung solcher realen Kapazitätsbeschränkungen durch Stellen mit endlichen Markenzapazitäten wird jedoch seitens der Invariantenanalyse ausgeschlossen.

Zwar besteht die Möglichkeit, Stellen mit beschränkten Markenzapazitäten durch die Einführung von Komplementärstellen in Stellen mit unbeschränkten Markenzapazitäten zu transformieren. Doch leidet die Verwendung von Komplementärstellen unter drei erheblichen Nach-

teilen. Erstens werden die Netzmodelle durch die zusätzlichen Stellen und weitere Hilfskanten, die diese Komplementärstellen mit Knoten des ursprünglichen Netzes verknüpfen, aufgebläht. Dadurch werden die resultierenden Netzmodelle oftmals unübersichtlich und schwer zu handhaben. Zweitens widersprechen die Komplementärstellen der Anforderung einer möglichst natürlichen Modellgestaltung. Denn bei diesen Stellen und ihren Hilfskanten handelt es sich um artifizielle Konstrukte, die keinen "natürlichen" Bezug zum modellierten Realitätsausschnitt besitzen. Drittens lassen sich Komplementärstellen nur auf Stelle/Transition-Netze ohne Schwierigkeiten anwenden. Aus allen vorgenannten Gründen eignen sich Komplementärstellen nicht, um für Synthetische Netze das kapazitive Defizit der Invariantenanalyse zu überwinden.

Das topologische Defizit der Invariantenanalyse beruht wiederum auf dem Motiv, aus ihrem Anwendungsbereich solche Netze auszuschließen, bei denen die dynamische Unvollständigkeit von Inzidenzmatrizen zu fehlerhaften Untersuchungsergebnissen verleiten könnte. Topologisch definierte Netzkomponenten, die von Inzidenzmatrizen grundsätzlich nicht berücksichtigt werden können, aber dennoch die Zulässigkeit des Schaltens von Transitionen beeinflussen, liegen vor allem⁵⁸⁾ in zwei Fällen vor. Erstens handelt es sich um 1-Schleifen. Dort hängt die Aktivierung der schleifenzugehörigen Transition von der Markierung der schleifenzugehörigen Stelle ab. Zweitens sind Informationsstellen betroffen. Sie können vermittels ihrer Informationskanten und durch ihre aktuellen Markierungen die Aktivierung jener Transitionen beeinflussen, die sich in ihren Nachbereichen befinden.

Eine Transition t_n , die in einer 1-Schleife enthalten ist, erfordert für ihre Aktivierung, daß die schleifenzugehörige Stelle s_m hinreichend markiert sind. Beim Schalten der Transition aus einer 1-Schleife wird die aktivierende Markenanzahl nicht nur von der schleifenzugehörigen Stelle abgezogen, sondern dort *uno actu* auch wieder abgelegt. Nur diese Schaltwirkung wird in der Inzidenzmatrix erfaßt. Da die Markenanzahl auf der schleifenzugehörigen Stelle durch das Schalten der Transition insgesamt nicht verändert wird, nimmt der schleifenrepräsentierende Matrixkoeffizient den Wert $c_{m,n}=0$ an. Anhand dieses Nullkoeffizienten wird besonders deutlich, daß der Aktivierungsaspekt keine Berücksichtigung findet. Denn die *positive* Markenanzahl, die zur Aktivierung der Transition t_n aus der 1-Schleife auf jeden Fall⁵⁹⁾ erforderlich ist, spiegelt sich im *Nullkoeffizienten* überhaupt nicht wieder. Statt dessen repräsentiert die Inzidenzmatrix aufgrund ihres Nullkoeffizienten eine 1-Schleife zwischen einer Transition t_n und einer Stelle s_m in exakt derselben Weise wie den Fall, daß dieses Knotenpaar überhaupt nicht durch Kanten verknüpft wird⁶⁰⁾.

Das potentielle Schaltverhalten eines Netzes kann jedoch wesentlich davon abhängen, ob eines seiner Knotenpaare entweder eine 1-Schleife bildet oder aber unverknüpft ist⁶¹⁾. Diesen Unterschied der dynamischen Netzstruktur vermag die Inzidenzmatrix des Netzes nicht abzubilden. Das Ausdrucksvermögen von Inzidenzmatrizen reicht grundsätzlich nicht aus, um die unterschiedliche dynamische Qualität von unverknüpften Knotenpaaren und 1-Schleifen zu erfassen. Daher ist die Repräsentation der dynamischen Netzstruktur durch eine Inzidenzmatrix notwendig unvollständig⁶²⁾. Falls ein Netz tatsächlich mindestens eine 1-Schleife besitzt, kann sich diese Unvollständigkeit als fehlerhafte Repräsentation der Netzdynamik auswirken⁶³⁾. Folglich erweist sich die Invariantenanalyse, die sich ausschließlich auf die Netzrepräsentation durch Inzidenzmatrizen stützt, als potentiell fehlerhaft. Sie droht zu falschen Erkenntnissen über das Verhaltenspotential von Netzen zu gelangen, sobald in der Topologie der untersuchten Netze 1-Schleifen auftauchen⁶⁴⁾. Um diese Fehlermöglichkeit auszuschließen, darf die Invariantenanalyse nur auf Netze angewandt werden, die keine 1-Schleifen enthalten⁶⁵⁾. Aus dem gleichen Grund müssen alle unreinen Netze ausgegrenzt werden, die per definitionem mindestens eine 1-Schleife enthalten. Daher leidet die Invariantenanalyse unter dem topologischen Defizit, von vornherein auf die Untersuchung reiner Netze beschränkt zu sein⁶⁶⁾.

Das topologische Defizit der Invariantenanalyse wird noch dadurch verstärkt, daß ebenso Netze ausgeschlossen werden müssen, die über Informationsstellen und -kanten verfügen. Diese Konstrukte wurden aber in dieser Arbeit eingeführt, um die Ausdruckskraft Synthetischer Netze

zu erhöhen. Ihre Ausgrenzung aus der Invariantenanalyse wird erforderlich, weil sie sich genau so wie 1-Schleifen verhalten: Eine Informationsstelle beeinflusst zusammen mit ihrer Informationskante die Aktivierung ihrer Ausgangstransition, ohne daß die Markierung der Informationsstelle durch die Schaltwirkung jener Transition verändert wird. Deshalb wird ein solches Paar aus einer Informationsstelle s_m und ihrer Ausgangstransition t_n in einer Inzidenzmatrix wiederum durch einen Nullkoeffizienten $c_{m,n}=0$ repräsentiert⁶⁷⁾. Die Schwierigkeiten, die der Invariantenanalyse von 1-Schleifen bereitet werden, gelten daher in der gleichen Weise für Informationsstellen von Synthetischen Netzen. Folglich müssen auch Netze mit Informationsstellen aus dem Anwendungsbereich der Invariantenanalyse ausgesondert werden, um die oben erläuterten Fehlergefahren zu vermeiden⁶⁸⁾.

Die Unfähigkeit, Netze zu berücksichtigen, die 1-Schleifen oder Informationsstellen enthalten, kann für die netzbasierte Modellierung realer Probleme im allgemeinen nicht hingegenommen werden. Denn beide Konstrukte dienen zur Repräsentation von Nebenbedingungen, die bei der Konzeptualisierung von Realproblemen oftmals eine größere Rolle spielen. Daher schränkt die Ausgrenzung von 1-Schleifen und von Informationsstellen den Anwendungsbereich der Invariantenanalyse so weit ein, daß ihr Einsatz als Untersuchungstechnik für Netzmodelle schon aus diesem Grund im Regelfall nicht in Betracht kommt.

Die Mängelliste der Invariantenanalyse ist aber noch keineswegs erschöpft. Hinzu kommt als eine weitere Schwäche das vektorielle Defizit der Analyse von semi-positiven T-Invarianten. Es beruht auf der Verengung des Schaltverhaltens von Netzen auf die Betrachtung von Schaltvektoren. Denn die Schaltvektoren enthalten nur noch die absoluten Anzahlen der Schaltakte, mit denen alle Transitionen eines Netzes an einem Schaltprozeß teilhaben. Diese absoluten Schaltaktanzahlen enthalten aber keine Informationen mehr über den kausalen oder zeitlichen Zusammenhang der Schaltakte. So können sich zwei Schaltprozesse trotz identischer Schaltvektoren immer noch dadurch unterscheiden, daß in ihnen die Schaltakte der Transitionen in unterschiedlichen Reihenfolgen geschehen⁶⁹⁾. Ebenso ist es möglich, daß manche Schaltakte kausal unabhängig und zeitlich parallel erfolgen, weil sie zu einem Schaltschritt nebenläufig aktivierter Transitionen gehören. Beide Fälle lassen sich durch die Schaltvektoren der Invariantenanalyse aber nicht erfassen. Daher kann die Reduzierung von Schaltfolgen auf Schaltvektoren zu einem Informationsverlust führen⁷⁰⁾.

Dieser potentielle Informationsverlust ist für die Untersuchung dynamischer Netzeigenschaften besonders gravierend. Denn das Petrinetz-Konzept zeichnet sich dadurch aus, bei der nebenläufigen Aktivierung von Transitionen sowohl alternative sequentielle Schaltprozesse mit jeweils verschiedenen Reihenfolgen der involvierten Schaltakte als auch nebenläufige Schaltprozesse zuzulassen. Dieser Ausdrucksreichtum für die Formulierung von Schaltprozessen in Netzmodellen wird durch das vektorielle Defizit der Invariantenanalyse grundsätzlich unterdrückt.

Das Effektivitätsdefizit der Invariantenanalyse für Höhere Netze beruht auf der hohen formalen Komplexität, die den invariantendefinierenden Vektorgleichungen für diese Netze zukommt. Die skalare Reformulierung der Vektorgleichungen liefert zwar für alle Netzklassen lineare homogene Gleichungssysteme. Diese Gleichungssysteme nehmen aber nur für Stelle/Transition-Netze eine relativ einfache formale Struktur an, die sich mit konventionellen Techniken der numerischen Informationsverarbeitung ohne grundsätzliche Schwierigkeiten bewältigen läßt⁷¹⁾. Die Einfachheit der Gleichungssystemstruktur beruht auf zwei Gründen. Erstens sind die invariantendefinierenden Gleichungssysteme bei Stelle/Transition-Netzen auf rein arithmetische Weise definiert. Denn nur dort bestehen sowohl der Spaltenvektor der Invarianten i_h mit $i_h \in \{i_{t_h}, i_{s_h}\}$ als auch die Inzidenzmatrix C immer aus *ganzzahligen* Komponenten bzw. Koeffizienten (diophantische Gleichungssysteme). Zweitens handelt es sich bei allen Koeffizienten $c_{m,n}$ der Inzidenzmatrix um Konstanten. Daraus folgt, daß in den Gleichungssystemen alle Variablen lediglich additiv verknüpft sind.

Für Höhere Netze besitzen die invariantendefinierenden Gleichungssysteme dagegen im allgemeinen eine komplexere formale Struktur. Denn die Koeffizienten $c_{m,n}$ der Inzidenzmatrix können beliebig komplizierte algebraisch-prädikatenlogische Ausdrücke darstellen. Bei Synthetischen Netzen handelt es sich beispielsweise um Multimengen aus teilevaluierten atomaren Formelvorkommnissen. Die Koeffizienten der Inzidenzmatrix dürfen auch Variablen darstellen oder als Werte von Funktionen definiert sein, die in ihren Argumenten Variablen enthalten. Der Ausdrucksreichtum zulässiger Matrixkoeffizienten kann dazu führen, daß sich die Gleichungssysteme, die Invarianten für Höhere Netze definieren, oftmals nur unter großen Anstrengungen oder im ungünstigen Fall sogar überhaupt nicht mehr lösen lassen⁷²⁾. Daher steht derzeit kein "effektives"⁷³⁾ mathematisches Verfahren zur Verfügung, das imstande wäre, auch dann noch alle nicht-trivialen Lösungen der o.a. linearen homogenen Gleichungssysteme zu bestimmen, wenn das algebraische Ausdrucksvermögen vollständig ausgeschöpft wird.

Das Effektivitätsdefizit der Invariantenanalyse beruht letztlich auf der Schwierigkeit, das algebraisch-prädikatenlogische Ausdruckspotential Höherer Netze mit der Hilfe von Gleichungssystemen zu beherrschen. Dies gilt insbesondere auch für die hier entwickelten Synthetischen Netze, die dieses Ausdrucksvermögen bei der Definition sortierter Marken intensiv nutzen. Es existieren lediglich einige wenige Versuche, die Invarianten von Höheren Netzen zu berechnen⁷⁴⁾. Dabei treten erhebliche Ermittlungsschwierigkeiten zu Tage. Darüber hinaus erweisen sich die eingesetzten Rechenverfahren als hochgradig intransparent⁷⁵⁾. Beide Gründe lassen es nicht erwarten, daß in der näheren Zukunft effektive Lösungsmethoden für die invariantendefinierenden Gleichungssysteme entwickelt werden, die auch die wesentlich komplexer strukturierten Attribut- und Kompositmarken aus dieser Arbeit bewältigen könnten.

Die voranstehend erläuterten vielfältigen Defizite, unter denen die Untersuchungen von Netzinvarianten leiden, lassen sich als "Dilemma der Invariantenanalyse" zusammenfassen: Entweder kann sie dynamische Eigenschaften von Netzen zufriedenstellend analysieren. Dafür muß ihr Anwendungsbereich aber so weit eingeschränkt werden, daß die Ausdrucksfähigkeit von Netzen gravierend beeinträchtigt wird⁷⁶⁾. Dies widerspricht der früher aufgestellten Zielsetzung, ein möglichst ausdrucksstarkes Modellierungskonzept zu entwickeln. Oder es wird am ungeschmälerten Ausdrucksvermögen des Petrinetz-Konzepts festgehalten. Dann ist die Invariantenanalyse mit derart folgenschweren Mängeln behaftet, daß sie als allgemeine Untersuchungstechnik für Netzmodelle ausscheidet.

Angesichts dieses Dilemmas vermag sich der Verf. der breiten Resonanz, die der Invariantenanalyse in der Netzliteratur zuteil wird, nicht anzuschließen. Statt dessen wird er die Invariantenanalyse nicht weiter berücksichtigen. Er sieht in ihrer Vernachlässigung keinen beklagenswerten Nachteil. Denn es wird später aufgezeigt, daß sich alle interessanten dynamischen Netzeigenschaften durch die Analyse von Erreichbarkeitsgraphen gewinnen lassen⁷⁷⁾. Dabei erlaubt die Erreichbarkeitsanalyse, für die untersuchten Netzeigenschaften nicht nur notwendige⁷⁸⁾, sondern sogar hinreichende Bedingungen anzugeben⁷⁹⁾. Dazu braucht lediglich die Finitheit der betrachteten Netzmodelle vorausgesetzt zu werden. Diese Prämisse wird im Rahmen dieser Arbeit stets erfüllt. Anderen Einschränkungen unterliegt die Erreichbarkeitsanalyse eines Netzmodells nicht. Daher ist die Invariantenanalyse nicht nur mit gravierenden Schwächen behaftet. Vielmehr erweist sie sich im hier gesetzten Rahmen sogar als überflüssig, weil eine leistungsfähigere Alternative bereitsteht.

Anmerkungen zum Kapitel:

1) Vgl. zu Invariantenanalysen im Rahmen des Petrinetz-Konzepts LAUTENBACH (1973), S. 65ff. u. 111ff.; LAUTENBACH (1975a), S. 5ff.; COTRONIS (1977), S. 200f.; SIFAKIS (1978), S. 474ff.; MEMMI (1978c), S. 507ff.; LAUTENBACH (1979a), S. 93ff.; AZEMA (1980), S. 602; MEMMI (1981a), S. 10f. u. 15ff.; REISIG (1983c), S. 4; PASCOLETTI (1985), S. 119ff.; REISIG (1985b), S. 77ff. u. 133ff.; REISIG (1986a), S. 88ff. u. 158ff.; LAUTENBACH (1987a), S. 148ff., insbesondere S. 152ff.; MURATA,TA. (1988b), S. 488ff.; BATTISTON (1988), S. 32ff.; FREEDMAN (1988b), S. 334f. u. 337f.; REISIG (1989a), S. 15ff. u. 43; REISIG (1989b), S. 6ff.; ZELEWSKI (1989c), S. 16 u. 29ff.; ZELEWSKI (1989e), S. 74ff. u. 85ff.; PAGNONI (1990), S. 25, 139ff. u. 145ff., insbesondere S. 153ff.; ABEL,D. (1990), S. 85ff. u. 111ff.; ROSENSTENGEL (1991), S. 63ff. u. 89f.

2) Dies belegt schon die Anzahl der Quellenangaben in der voranstehenden Anmerkung. Darüber hinaus zeichnen sich diese Beiträge zur Invariantenanalyse zumeist auch durch eine hohe Erörterungsintensität aus. Schließlich stützen sich die meisten Softwarepakete, die eine Untersuchung dynamischer Netzeigenschaften gestatten, auf die Technik der Invariantenanalyse. Dies liegt insofern nahe, als diese Analysetechnik - wie nachstehend erläutert wird - auf arithmetisch oder algebraisch formulierten Gleichungssystemen beruht. Zumindest die arithmetischen Gleichungssysteme lassen sich mit konventionellen Algorithmen der numerischen Informationsverarbeitung relativ leicht bearbeiten. (Bezugspunkte sind die Verarbeitungsaufgaben, die für die Techniken der Deduktions-, Reduktions- und Erreichbarkeitsanalyse erfüllt werden müssen. Die Simulationsanalyse bereitet dagegen einen ähnlich geringen Informationsverarbeitungsaufwand.)

3) Auf dennoch bestehende Defizite bei der Erfassung der statischen Netzstruktur wird noch zurückgekommen.

4) Vgl. zum Einfluß der Invariantenanalyse auf strukturelle Untersuchungen von dynamischen Netzeigenschaften - insbesondere der Netzlebendigkeit - z.B. LAUTENBACH (1975a), S. 5; SIFAKIS (1978), S. 474ff.; MEMMI (1978c), S. 508; MEMMI (1979), S. 102f.; MEMMI (1980), S. 218; REISIG (1986a), S. 94ff., insbesondere S. 96f.; REISIG (1989b), S. 6ff.; ABEL,D. (1990), S. 89 u. 111ff.

5) Eine Ausnahme findet sich bei REISIG (1986a), S. 94 i.V.m. S. 93, und ABEL,D. (1990), S. 88: Wenn ein Netz eine *positive* S-Invariante i_{S_h} mit $i_{S_{h_m}} \in \mathcal{N}_+$ für alle Stellen $s_m \in S$ besitzt, dann ist das Netz mit Sicherheit beschränkt. Wenn die Ausgangsmarkierung M_0 des Netzes bekannt ist, kann sogar eine obere Schranke K_{\max} für die K-Beschränktheit des Netzes angegeben werden (vgl. ABEL,D. (1990), S. 88).

6) Vgl. ABEL,D. (1990), S. 28, 89, 114 u. 118. Er stellt u.a. fest, "daß die eleganter wirkenden Verfahren der linearen Algebra für *allgemeine* Petri-Netze lediglich notwendige Bedingungen liefern können." (S. 28; kursive Hervorhebung durch den Verf.).

7) Dies wird später anhand der nicht-trivialen Reproduktion von Netzmarkierungen durch semi-positive T-Invarianten näher ausgeführt.

8) Besonders deutlich wird diese Kombination von Invarianten und Ausgangsmarkierungen bei LAUTENBACH (1973), S. 66; COTRONIS (1977), S. 200f.; MEMMI (1978c), S. 507; LAUTENBACH (1979a), S. 99 u. 101; REISIG (1986a), S. 95ff.

9) Vgl. - vor allem in bezug auf die Netzlebendigkeit - LAUTENBACH (1973), S. 66; LAUTENBACH (1975a), S. 10ff., insbesondere S. 12; REISIG (1983c), S. 4; REISIG (1986a), S. 96f.

10) Diese Einschränkung bestätigt mittelbar das Zitat von ABEL, das schon in einer Anmerkung kurz zuvor angeführt wurde. Aus der Umkehrung der zitierten Feststellung folgt nämlich, daß sich hinreichende Bedingungen nur für nicht-allgemeine, also irgendwie eingeschränkte Petrinetze angeben lassen.

11) Diese Mängel liegen *nicht* in den Charakteristika von Stelle/Transition-Netzen begründet, sondern betreffen den konzeptionellen Ansatz der Invariantenanalyse grundsätzlich. Daher erübrigt sich der potentielle Einwand, durch den Übergang zu Höheren Netzen könnten die anschließend erörterten Schwächen der Invariantenanalyse nachträglich behoben werden.

12) Die Komplikationen folgen daraus, daß der ganzzahlige Charakter von Inzidenzmatrix und Invarianten verloren geht, wenn von Stelle/Transition-Netzen zu Höheren Netzen übergewechselt wird. Beispielsweise sind bei Synthetischen Netzen die Koeffizienten $c_{m,n}$ der Inzidenzmatrix \underline{C} nicht mehr als Differenzen $c_{m,n} = W(t_p, s_m) - W(s_m, t_n)$ der ganzzahligen Kantengewichte von Stelle/Transition-Netzen festgelegt. An ihre Stelle treten die Differenzen $c_{m,n} = \text{bfm}(t_p, s_m) - \text{bfm}(s_m, t_n)$ der entsprechenden Beschriftungen der Kanten von Synthetischen Netzen durch Multimengen aus teilevaluierten atomaren Formelvorkommnissen. Entsprechend nehmen die Invarianten in Synthetischen Netzen die formale Gestalt von Multimengen aus teilevaluierten atomaren konstanten Formelvorkommnissen an. Auf die daraus resultierenden formalen Schwierigkeiten wird an späterer Stelle unter dem Aspekt des Effektivitätsdefizits der Invariantenanalyse zurückgekommen. Die komplexere Definition der Invarianten von Synthetischen Netzen bedeutet z.B. im Hinblick auf semi-positive T-Invarianten, daß sie nicht mehr nur die Schaltanzahlen von Transitionen, sondern auch die zugehörigen Schaltfarben ausdrücken. Die Schaltfarben stellen aber als die Bilder

von Variablenbindungsfunktionen, die innerhalb eines sortierten prädikatenlogischen Ansatzes definiert wurden, recht aufwendige formale Konstrukte dar.

13) Mit $\underline{0}_M$ und $\underline{0}_N$ werden der M- bzw. N-stellige Spaltenvektor notiert, der in allen seinen Komponenten jeweils den Wert "0" annimmt (Nullvektor). \underline{C}^t ist die transponierte Form der Inzidenzmatrix \underline{C} . Erste geht aus zweiter durch zeilen-/spaltenweise Notation aller Matrixspalten/-zeilen hervor.

14) Vgl. REISIG (1986a), S. 88 u. 108; ABEL,D. (1990), S. 88f. (jedoch nur in bezug auf nicht-negative Ganzzahlen); PAGNONI (1990), S. 151f. (dort allerdings nicht als Definition, sondern als Konsequenz) u. 155 (dort als Definition).

15) Vgl. REISIG (1986a), S. 88 u. 91; ABEL,D. (1990), S. 87; Mitunter wird für die Definition von S-Invarianten auch die äquivalente Formulierung $\underline{is}_h^t \cdot \underline{C} = \underline{0}_N^t$ gewählt. Vgl. z.B. PAGNONI (1990), S. 150f. (dort allerdings nicht als Definition, sondern als Konsequenz) u. 155 (dort als Definition).

16) Beispielsweise schränkt ABEL,D. (1990), S. 88, seinen Begriff der T-Invarianten von vornherein auf den Fall semi-positiver Vektoren ein.

Allerdings werden bei LAUTENBACH (1987a), S. 159, auch negative Werte für die Komponenten $it_{h,n}$ einer T-Invariante zugelassen. Sie würden ein "Rückwärtsschalten" von Transitionen erfordern, das in der vorliegenden Arbeit jedoch ausgegrenzt wurde. Denn es handelt sich dabei um eine Schaltregelvariante, die sich formal definieren, aber in bezug auf modellierte reale Sachverhalte nicht gehaltvoll interpretieren läßt. Daher wird die Berücksichtigung von Invarianten mit negativen Komponenten nicht weiter verfolgt.

17) T-Invarianten verallgemeinern daher die Schaltvektoren, die für das einmalige Schalten einer einzelnen Transition eingeführt wurden, auf das beliebig oft wiederholte Schalten beliebiger Transitionen eines Netzes.

18) Es wird zugelassen, anstelle der Klasse aller Schaltfolgen, deren gemeinsamen Schaltvektor eine T-Invariante darstellt, nur von einer Schaltfolge zu sprechen. Diese Schaltfolge besitzt dann generischen Charakter. Sie ist eine "typische" Schaltfolge in dem Sinne, daß sie - wie alle anderen Schaltfolgen aus der vorgenannten Klasse - den gleichen Schaltvektor besitzt.

19) Vgl. REISIG (1986a), S. 109f.; ZELEWSKI (1989c), S. 16 u. 31f.

20) Eine Schaltfolge heißt genau dann (un)zulässig bezüglich einer Markierung, wenn die Schaltfolge unter dieser Markierung (nicht) aktiviert ist. Die Aktivierung einer Schaltfolge unter einer Markierung wurde schon im Kontext von Stelle/Transition-Netzen definiert.

21) Welche Schaltfolgen tatsächlich zulässig sind, kann mit Rechenverfahren für die Auswertung von Gleichungssystemen nicht festgestellt werden. Statt dessen muß für jeden Schaltschritt aus einer Schaltfolge überprüft werden, ob unter der jeweils erreichten Markierung seine Schaltvoraussetzung erfüllt ist. Dies kann auf zwei Weisen geschehen. Entweder wird auf andere Analysetechniken zurückgegriffen, die nicht zur Invariantenanalyse gehören. Dafür kommt z.B. die Erreichbarkeitsanalyse in Betracht. Denn die Überprüfung von Schaltvoraussetzungen (Aktivierungsbedingungen) bildet einen zentralen Aspekt der Konstruktion von Erreichbarkeitsgraphen. Oder die Invariantenanalyse wird um entsprechende Prüfungen von Schaltvoraussetzungen erweitert. Dies kann z.B. derart geschehen, daß die Rechenverfahren für die Lösung von Gleichungssystemen erweitert werden um Verfahren, mit denen sich die Erfüllung von Ungleichungssystemen überprüfen läßt. Denn die Schaltvoraussetzungen von Schaltschritten und die zugrundeliegenden Aktivierungsbedingungen ihrer zugehörigen Transitionen beruhen immer wieder auf zwei charakteristischen Tests: Auf den Einflußstellen müssen hinreichend viele von den erforderlichen Markenkopien vorhanden sein, und auf den Ausgangsstellen muß genügend freie Markenzapazität zur Verfügung stehen. Beide Testvarianten lassen sich auf die Erfüllung von Ungleichungssystemen zurückführen. Sofern im folgenden keine ausdrücklich anderen Festlegungen erfolgen, wird die Invariantenanalyse so eng ausgelegt, daß sie im wesentlichen nur die Rechenverfahren für die Auswertung linearer homogener diophantischer Gleichungssysteme umfaßt.

22) Eine zulässige semi-positive T-Invariante liegt vor, wenn für mindestens eine erreichbare Netzmarkierung nachgewiesen werden kann, daß die T-Invariante der Schaltvektor mindestens einer Schaltfolge ist, die unter dieser erreichbaren Markierung aktiviert ist. Die semi-positive T-Invariante muß nicht unbedingt der Schaltvektor einer Schaltfolge sein, die bereits unter der aktuellen Referenzmarkierung M_r aktiviert ist. Vielmehr reicht es aus, wenn von der aktuellen Referenzmarkierung M_r aus mindestens eine Folgemarkierung M_f erreicht werden kann, unter der dann mindestens eine Schaltfolge aktiviert ist, deren Schaltvektor die T-Invariante bildet. Allerdings ist zu beachten, daß die nachfolgend erörterte Reproduktion von erreichbaren Markierungen immer an denjenigen Markierungen anknüpft, unter denen die reproduzierenden Schaltfolgen aktiviert sind. Im zuvor erwähnten Fall würde die semi-positive T-Invariante also nicht die Referenzmarkierung M_r , sondern die Folgemarkierung M_f zu reproduzieren gestatten.

23) Dieser Fall tritt ein, wenn für alle Schaltfolgen aus der Schaltfolgenklasse einer semi-positiven T-Invariante gilt: Die Schaltfolgen sind weder unter der Ausgangsmarkierung des untersuchten Netzes noch unter einer von dort aus erreichbaren Markierung aktiviert. Eine solche T-Invariante repräsentiert ausschließlich unzulässige Schaltfolgen.

Nur für ein Netz, für das keine bestimmte Ausgangsmarkierung M_0 vorgegeben wird, läßt sich zeigen: Wenn das Netz mindestens eine semi-positive T-Invariante besitzt, dann kann das Netz mit einer Ausgangsmarkierung so versehen werden, daß die T-Invariante den Schaltvektor mindestens einer Schaltfolge darstellt, die unter dieser Ausgangsmarkierung oder unter einer von dort aus erreichbaren Markierung aktiviert ist. Dies läßt sich beispielsweise durch folgende Konstruktion erreichen: Zunächst werden alle Stellen s_m des Netzes mit der Nullmarkierung $M_0(s_m)=0$ versehen. Dann werden alle Transitionen betrachtet, die nach Maßgabe der betrachteten T-Invariante geschaltet werden sollen. Die Eingangsstellen jeder dieser Transitionen werden mit so vielen Marken ausgestattet, daß sie ausreichen, um die Transitionen so oft zu schalten, wie es durch den Schaltvektor der T-Invariante angezeigt wird. Unter der derart modifizierten Ausgangsmarkierung ist per constructionem mindestens eine Schaltfolge aktiviert, deren Schaltfolge die betrachtete T-Invariante darstellt.

24) Vgl. REISIG (1986a), S. 108; LAUTENBACH (1987a), S. 159; REISIG (1989b), S. 9; PAGNONI (1990), S. 147ff., insbesondere S. 149, u. S. 156; ABEL,D. (1990), S. 89 u. 92.

Die Reproduktion der Referenzmarkierung M_r durch jede Schaltfolge, die unter dieser Markierung aktiviert ist und eine semi-positive T-Invariante \underline{it}_h als Schaltvektor besitzt, folgt aus der Definitionsgleichung von T-Invarianten und aus der inzidenzmatrixbezogenen Schaltregelvariante $SR_{FS,3}$. Dabei wird der schaltfolgenbezogene Schaltvektor \underline{sy} für das Ausführen einer Schaltfolge SF_L der Länge L durch die semi-positive T-Invariante \underline{it}_h ersetzt. Diese T-Invariante wird als Schaltvektor \underline{it}_h aller Schaltfolgen SF interpretiert, für die jeweils gilt: Jede Transition t_n mit $n \in \{1, \dots, N\}$ wird in der Schaltfolge SF genau so oft geschaltet, wie die Komponente $it_{h,n}$ der T-Invariante \underline{it}_h angibt. Die Länge L der Schaltfolgen ist jetzt nicht mehr fixiert. Denn durch unterschiedliches Zusammenfassen der geschalteten Transitionen zu verschiedenen Schaltschritten ist es möglich, sowohl kürzere Schaltfolgen mit relativ umfangreichen Schaltschritten als auch längere Schaltfolgen mit relativ kleinen Schaltschritten mit der gleichen T-Invariante \underline{it}_h als Schaltvektor zu vereinbaren. Vgl. dazu den Hinweis zur "Anordnungsblindheit" der T-Invarianten \underline{it}_r . Daher lautet die schaltschritt- und schaltfolgenbezogene Schaltregelvariante $SR_{FS,3}$, wenn die Definitionen von Vor- und Nachbereich der Schaltregel-Funktion außer acht gelassen werden, für jede Schaltfolge SF mit dem Schaltvektor \underline{it}_r :

$$SR_{FS,3}: (\underline{M}_r, SF) \rightarrow \underline{M}_f = SR_{FS,3}(\underline{M}_r, SF) = \underline{M}_r + \underline{C} \cdot \underline{it}_h; \text{ sofern } AKT(SF, \underline{M}_r)$$

Daraus folgt mit Hilfe der Definition $\underline{C} \cdot \underline{it}_h = \underline{Q}_M$ von T-Invarianten durch elementare Äquivalenzumformungen:

$$\begin{aligned} \underline{M}_f &= \underline{M}_r + \underline{C} \cdot \underline{it}_h & // \quad \underline{C} \cdot \underline{it}_h &= \underline{Q}_M \\ \Leftrightarrow \underline{M}_f &= \underline{M}_r + \underline{Q}_M \\ \Leftrightarrow \underline{M}_f &= \underline{M}_r \end{aligned}$$

Folglich wird die Referenzmarkierung M_r durch die Schaltfolge SF_r mit dem Schaltvektor \underline{it}_h reproduziert; q.e.d.

Da nur semi-positive T-Invarianten betrachtet werden, stellen sie immer die Schaltvektoren von Schaltfolgen SF dar, in denen mindestens eine Transition mindestens einmal geschaltet wird. Dadurch wird die Nullschaltfolge, die jede Referenzmarkierung in trivialer Weise reproduziert, von vornherein ausgeschlossen. Dies entspricht der o.a. Nichtbeachtung der trivialen Lösung $\underline{it}_h = \underline{Q}_M$ für die Definitionsgleichung von T-Invarianten.

25) Dies wird besonders deutlich bei ABEL,D. (1990), S. 10 u. 12. Zwar bezieht sich ABEL nicht unmittelbar auf T-Invarianten. Vielmehr befaßt er sich mit der Analyse erreichbarer Markierungen auf der Basis von Inzidenzmatrizen. Aber die hier interessierenden T-Invarianten lassen sich als ein Spezialfall der Ausführungen ABEL's rekonstruieren. Denn ABEL betrachtet den Übergang zwischen der Ausgangsmarkierung M_0 und einer erreichbaren Markierung M_r (S. 10; dort als Markierungen "m₀" bzw. "m"). Für den speziellen Fall $M_r = M_0$ resultiert daraus die Reproduktion einer beliebigen Markierung M_r . Aus den Betrachtungen ABEL's sind lediglich die trivialen Reproduktionen auszuschließen, da er den Nullvektor als Schaltvektor nicht gesondert beachtet. Später spricht ABEL,D. (1990), S. 89, sogar explizit davon, daß T-Invarianten die Schaltanzahlen angeben, die zur Reproduktion einer Referenzmarkierung notwendig sind.

26) Vgl. zum folgenden (unabhängig davon, ob die skalare Konstante "mi" oder das konstante Vektorprodukt " $\underline{is}_h \cdot \underline{M}_0$ " bevorzugt wird) REISIG (1986a), S. 92f.; LAUTENBACH (1987a), S. 154f.; ABEL,D. (1990), S. 87; PAGNONI (1990), S. 155.

Die anschließend benutzte Beziehung $\underline{is}_h \cdot \underline{M}_r = mi = \underline{is}_h \cdot \underline{M}_0$ kann mit Hilfe der Definition von S-Invarianten gewonnen werden. Dabei wird auf die schaltschritt- und schaltfolgenbezogene Schaltregelvariante $SR_{FS,3}$ zurückgegriffen. Wenn abermals von den Definitionen des Vor- und Nachbereichs der Schaltregel-Funktion $SR_{FS,3}$ abstrahiert wird, liefert sie für die Ausführung einer Schaltfolge SF mit dem Schaltvektor \underline{sy} , die unter der Referenzmarkierung M_r aktiviert ist, eine Folgemarkierung M_f , die bestimmt ist durch:

$$SR_{FS,3}: (\underline{M}_r, SF) \rightarrow \underline{M}_f = SR_{FS,3}(\underline{M}_r, SF) = \underline{M}_r + \underline{C} \cdot \underline{sy}; \text{ sofern } AKT(SF, \underline{M}_r)$$

Diese Beziehung wird nun auf das Ausführen einer Schaltfolge SF angewendet, die unter der Ausgangsmarkierung M_0 aktiviert ist und von dort aus zu einer beliebigen erreichbaren Markierung M_r führt. Dann muß gelten:

$$\begin{aligned} \underline{M}_r &= \underline{M}_0 + \underline{C} \cdot \underline{sv} \\ \Leftrightarrow \underline{M}_r - \underline{M}_0 &= \underline{C} \cdot \underline{sv} \\ \Leftrightarrow \underline{M}_r^{\text{tr}} - \underline{M}_0^{\text{tr}} &= (\underline{C} \cdot \underline{sv})^{\text{tr}} && // \text{ vgl. zur Transponierten eines Vektorprodukts} \\ &&& \text{ABEL, D. (1990), S. 87.} \\ \Leftrightarrow \underline{M}_r^{\text{tr}} - \underline{M}_0^{\text{tr}} &= \underline{sv}^{\text{tr}} \cdot \underline{C}^{\text{tr}} && // \cdot \underline{is}_h \\ \Leftrightarrow (\underline{M}_r^{\text{tr}} - \underline{M}_0^{\text{tr}}) \cdot \underline{is}_h &= \underline{sv}^{\text{tr}} \cdot \underline{C}^{\text{tr}} \cdot \underline{is}_h && // \underline{C}^{\text{tr}} \cdot \underline{is}_h = \underline{0}_N \\ \Leftrightarrow \underline{M}_r^{\text{tr}} \cdot \underline{is}_h - \underline{M}_0^{\text{tr}} \cdot \underline{is}_h &= \underline{sv}^{\text{tr}} \cdot \underline{0}_N \\ \Leftrightarrow \underline{M}_r^{\text{tr}} \cdot \underline{is}_h - \underline{M}_0^{\text{tr}} \cdot \underline{is}_h &= 0 \\ \Leftrightarrow \underline{M}_r^{\text{tr}} \cdot \underline{is}_h &= 0 + \underline{M}_0^{\text{tr}} \cdot \underline{is}_h \\ \Leftrightarrow \underline{M}_r^{\text{tr}} \cdot \underline{is}_h &= \underline{M}_0^{\text{tr}} \cdot \underline{is}_h \\ \Leftrightarrow (\underline{M}_r^{\text{tr}} \cdot \underline{is}_h)^{\text{tr}} &= (\underline{M}_0^{\text{tr}} \cdot \underline{is}_h)^{\text{tr}} \\ \Leftrightarrow \underline{is}_h^{\text{tr}} \cdot (\underline{M}_r^{\text{tr}})^{\text{tr}} &= \underline{is}_h^{\text{tr}} \cdot (\underline{M}_0^{\text{tr}})^{\text{tr}} \\ \Leftrightarrow \underline{is}_h^{\text{tr}} \cdot \underline{M}_r &= \underline{is}_h^{\text{tr}} \cdot \underline{M}_0 = \underline{mi} \end{aligned}$$

27) Vgl. LAUTENBACH (1987a), S. 155; PAGNONI (1990), S. 151.

28) Vgl. zu den Problemen von Invariantenanalysen auch VALETTE (1979b), S. 156ff.

29) Auf ein weiteres konzeptionelles Defizit der Invariantenanalyse von Gleichungssystemen weist BYRN (1974), S. II-129, hin. Der Verf. stuft die dort thematisierte Problematik aber nicht als wesentlich ein. Darüber hinaus kann der gleichungsbasierte Ansatz der Invariantenanalyse grundsätzlich nicht auf manche Erweiterungen des Kernkonzepts Synthetischer Netze angewandt werden. Dies gilt z.B. für Inhibitorkanten und für die erweiterten Schaltvorschriften der Realtransitionen aus Interpretierten Netzen. Beide Netzkonstrukte werden aber erst später eingeführt. Daher wird hier nicht näher darauf eingegangen, auf welche Weise sie die Unanwendbarkeit der Invariantenanalyse verursachen. Eine Andeutung findet sich aber für den Fall Interpretierter Netze bei VALETTE (1979b), S. 158.

30) Informationsstellen und -kanten gehören zwar nicht unmittelbar zur topologischen Struktur eines Netzes. Sie wurden aber eingeführt, um Nebenbedingungen zu repräsentieren, die sonst zumeist durch 1-Schleifen ausgedrückt werden. Die 1-Schleifen stellen ihrerseits einen typischen Aspekt der Netztopologie dar. Daher werden hier Informationsstellen und -kanten noch als topologischer Gesichtspunkt im weitesten Sinne behandelt.

31) Denn außer der Inzidenzmatrix enthalten die invariantendefinierenden Vektorgleichungen nur noch die Invarianten selbst und den Nullvektor. Die Invarianten tragen als Definienda noch keine Information über das jeweils zugrundeliegende Netz. Der Nullvektor kann als Konstante, die für *alle* Netze gilt, grundsätzlich keine Information über die dynamische Struktur eines speziellen Netzes enthalten.

32) An früherer Stelle wurden die Koeffizienten der Inzidenzmatrix durch den Nettoeffekt definiert, den das Schalten einer Transition bezüglich der Markierung einer Stelle hervorruft.

33) Dies läßt sich auch so umschreiben: Matrizenkalküle setzen stets vollständig definierte Funktionen voraus, deren Funktionswerte die Matrixkoeffizienten bilden. Den größeren Ausdrucksreichtum von partiell definierten Funktionen können Matrizenkalküle in dem Ausmaß nicht erfassen, in dem er über das Ausdrucksvermögen vollständig definierter Funktionen hinausgeht. Gerade dieses zusätzliche Formulierungspotential partieller Funktionen wird aber benötigt, um die Abhängigkeit der Schaltwirkung einer Transition von der Erfüllung ihrer Aktivierungsbedingung auszudrücken.

34) Die Linearität folgt daraus, daß in den Gleichungssystemen alle Variablen nur in ihrer ersten Potenz vorkommen. Das schließt jedoch nicht aus, daß die Gleichungssysteme auch Produkte solcher Variablen enthalten dürfen. Dies ist immer dann der Fall, wenn die Koeffizienten $c_{m,n}$ der Inzidenzmatrix keine Konstanten, sondern Variablen darstellen. Die variablen Koeffizienten werden dann mit den Variablen aus dem invariantendefinierenden Spalten-

vektor multiplikativ verknüpft. Solche Variablen-Produkte können zu erheblichen Schwierigkeiten führen, wenn es gilt, die Lösungen der Gleichungssysteme zu ermitteln.

35) Die Homogenität der Gleichungssysteme drückt lediglich aus, daß die involvierten Gleichungen keine absoluten Glieder besitzen (oder mit anderen Worten: das absolute Glied Null besitzen). Dies brachte in den o.a. invarianten-definierenden Vektorgleichungen bereits der Nullvektor zum Ausdruck.

36) Ein Gleichungssystem heißt diophantisch, wenn seine Koeffizienten ganzzahlig sind und wenn für seine Variablen ausschließlich ganzzahlige Lösungen interessieren. Dies trifft auf die Invariantenanalyse zu. Einerseits werden die Koeffizienten der Gleichungssysteme, die es bei der Invariantenanalyse zu lösen gilt, durch die Inzidenzmatrix des jeweils zu untersuchenden Netzmodells bestimmt. Die Matrixkoeffizienten sind entweder ganzzahlige Konstanten oder aber Variablen oder aber Funktionswerte von variablen Argumenten. In den beiden letztgenannten Fällen besitzen die Definitionsbereiche der Variablen und die Wertebereiche der Funktionsargumente abermals ganzzahligen Charakter. Andererseits wurden die T- und S-Invarianten als *ganzzahlige* Spaltenvektoren definiert. Diese Spaltenvektoren sind zu Analysebeginn unbekannt. Daher werden ihre Komponenten zunächst als Variablen formuliert. Für diese Variablen gilt es solche Substitutionen durch ganzzahlige Konstanten zu bestimmen, welche die invarianten-definierenden Gleichungssysteme erfüllen. Als Komplikation bei Höheren Netzen - insbesondere Prädikat/Transition-Netzen - ist lediglich zu beachten, daß die "ganzzahligen" Konstanten, Variablen und Funktionswerte oftmals auch formale Produkte aus Ganzzahlen und symbolischen Ausdrücken darstellen können. Darauf wird später aus der Perspektive des Effektivitätsdefizits zurückgekommen.

Vgl. zur Definition (linearer homogener) diophantischer Gleichungssysteme z.B. HUET (1978), S. 144 (dort allerdings ohne negative Ganzzahlen als Lösungen); CLAUSEN, M. (1989), S. 202; ABEL, D. (1990), S. 85 (ansatzweise).

37) Allgemein läßt sich nicht ausschließen, daß ein inadäquates Untersuchungskonzept durch "konzeptionelle Krücken" so ergänzt werden kann, daß es schließlich die erwünschten Analyseleistungen doch noch zu erbringen vermag. Dies trifft auch im besonderen auf die Invariantenanalyse zu.

38) Zur Verdeutlichung wird auf die Erreichbarkeitsanalyse verwiesen: Dort repräsentieren die Erreichbarkeitsgraphen dynamische Netzstrukturen adäquat, weil die Markierungsveränderungen in den Graphen als Schaltkanten zwischen jeweils zwei aufeinander folgenden Markierungsknoten explizit enthalten sind.

39) Vgl. VALETTE (1979b), S. 156; VALK (1981b), S. 148 (mit der Nuancierung, daß sich die Invarianten von Stelle/Transition-Netzen noch recht gut, die Invarianten von Prädikat/Transition-Netzen aber oftmals nicht mehr interpretieren ließen); PAGNONI (1990), S. 161 (nur in bezug auf Prädikat/Transition-Netze).

40) Auf eine zweite Interpretationsmöglichkeit wird in Kürze aus der Perspektive der Modellierungskonsistenz näher eingegangen.

Einen dritten Deutungsvorschlag unterbreitet ABEL, D. (1990), S. 92 u. 118. Er interpretiert eine semi-positive T-Invariante als die Charakterisierung eines in sich abgeschlossenen, von seiner Umgebung weitgehend unabhängigen Teilmodells. Dabei argumentiert er analog zur Interpretation von S-Invarianten, die in einer anderen Anmerkung näher erläutert wird. Daher wird hier auf seine Deutung von T-Invarianten nur knapp eingegangen. Jeder semi-positiven T-Invariante \underline{it}_h läßt sich eine Unterstützungsmenge US_h aus denjenigen Transitionen t_n zuordnen, die mit positiven Komponenten $it_{h,n}$ aus der T-Invariante \underline{it}_h korrespondieren: $t_n \in US_h \Leftrightarrow it_{h,n} > 0$. Zwar erweisen sich die Beispiele, die ABEL präsentiert, als intuitiv einleuchtend. Aber es läßt sich schwer überprüfen, ob dies entweder an der Invarianteninterpretation oder aber an der Einfachheit und "glücklichen" Konstruktion der Beispiele liegt. Der Verf. hegt in dieser Hinsicht erhebliche Skepsis, ob es für die semi-positiven T-Invarianten aus *beliebigen* Netzmodellen ebenso möglich ist, ähnlich überzeugende Bestimmungen von eigenständigen Teilmodellen vorzunehmen. Denn die Reproduktion von Referenzmarkierungen, die für semi-positive T-Invarianten typisch ist, stellt kein inhaltlich überzeugendes Argument zugunsten der inneren Abgeschlossenheit von Teilmodellen dar. Im Gegensatz zu S-Invarianten kann auch nicht auf Objektflüsse als Abgrenzungskriterium verwiesen werden, die das jeweils identifizierte Teilmodell in konstantem Gesamtvolumen durchsetzen.

41) Vgl. ABEL, D. (1990), S. 89.

Einerseits verhält sich ein Netz genau dann M_0 -reversibel, wenn sich seine Ausgangsmarkierung M_0 von jeder erreichbaren Markierung M_r aus erreichen läßt. Daher existiert für jede erreichbare Markierung M_r (mit $M_r \neq M_0$) mindestens eine semi-positive T-Invariante als Schaltvektor derjenigen Schaltfolge, die von der Ausgangsmarkierung M_0 aus startet, über die Markierung M_r führt und schließlich die Ausgangsmarkierung M_0 reproduziert. Diese Schaltfolgen sind mit Sicherheit zulässig, weil die Erreichbarkeit der Markierungen M_r und M_0 vorausgesetzt wurden. Also gilt für jedes Netz, das neben seiner Ausgangsmarkierung M_0 noch mindestens eine andere erreichbare Markierung M_r besitzt: Wenn das Netz M_0 -reversibel ist, dann besitzt es mindestens eine semi-positive T-Invariante. Die Umkehrung braucht jedoch nicht zuzutreffen. Denn eine semi-positive T-Invariante ist der gemeinsame Schaltvektor einer Klasse von Schaltfolgen, die jede Referenzmarkierung nur dann zu reproduzieren vermögen, falls es sich um zulässige Schaltfolgen handelt. Es kann doch der Fall eintreten, daß in einem Netz alle Schaltfolgen, die zu den semi-positiven T-Invarianten des Netzes gehören, unzulässig sind.

Dagegen behauptet REISIG (1986a), S. 110, daß sich ein Netz *genau* dann M-reversibel verhält, wenn es eine semi-positive T-Invariante besitzt. Bereits in einer früheren Anmerkung wurde aufgezeigt, daß die M-Reversibilität eines Netzes mit dessen M_0 -Reversibilität äquivalent ist. Daher schein REISIG's Diktum dem Voranstehenden zu widersprechen. Dies trifft jedoch aus zwei Gründen nicht zu. Erstens nimmt REISIG in seine Äquivalenzbehauptung die Prämisse auf, daß alle Stellen des betroffenen Netzes unbeschränkte Markenkapazitäten besitzen. Damit grenzt er schon eine größere Anzahl von Fällen aus, in denen die Schaltfolgen von semi-positiven T-Invarianten wegen fehlender freier Markenkapazitäten unzulässig werden können. Zweitens ist sein Beweis auf S. 110 nicht stringent, weil er nur die Schaltwirkungen der Transitionen aus Schaltfolgen berücksichtigt. Der Beweis enthält jedoch keine Prüfung, ob in den Schaltfolgen auch stets die Schaltvoraussetzungen (Aktivierungsbedingungen) der Transitionen erfüllt sind. Damit übersieht er alle jene Fälle, in denen Transitionen aus einer Schaltfolge nicht aktiviert sind, weil sich auf ihren Einflußstellen zu wenig Markenkopien befinden. Dies trifft insbesondere auf Einflußstellen zu, die mit den Transitionen jeweils 1-Schleifen bilden und *unmarkiert* sind. Die Schaltverhinderung solcher unmarkierten 1-Schleifen wird von REISIG's "Beweis" grundsätzlich übersehen, weil er sich auf die Inzidenzmatrix des untersuchten Netzes stützt. Der Informationsverlust, den Inzidenzmatrizen bei 1-Schleifen verursachen können, wurde schon an früherer Stelle behandelt.

42) Vgl. dazu die Argumentation in der voranstehenden Anmerkung. Selbst wenn aufgezeigt werden könnte, daß eine semi-positive T-Invariante den Schaltvektor von mindestens einer *zulässigen* Schaltfolge wiedergibt, so hilft dies im allgemeinen nicht viel weiter. Denn dann steht nur fest: Von der Ausgangsmarkierung M_0 kann über alle erreichbaren Markierungen M_r , die von der zulässigen Schaltfolge hervorgebracht werden, zur Ausgangsmarkierung M_0 zurückgekehrt werden. Diese involvierten Markierungen M_0 und M_r werden jedoch in der Regel die Erreichbarkeitsmenge des betrachteten Netzes nicht vollständig ausschöpfen. Daher werden im allgemeinen erreichbare Markierungen existieren, die von der zulässigen Schaltfolge nicht hervorgebracht werden. Für diese erreichbaren Markierungen steht nicht fest, ob von ihnen aus zur Ausgangsmarkierung zurückgekehrt werden kann. Folglich kann selbst eine semi-positive T-Invariante mit zulässiger Schaltfolge nicht sicherstellen, daß sich das zugrundeliegende Netz M_0 -reversibel verhält.

43) Es liegt das Netztheorem zugrunde: Wenn sich ein Netz M_0 -reversibel verhält, dann besitzt es mindestens eine semi-positive T-Invariante. Vgl. dazu die konzise Formulierung von ABEL,D. (1990), S. 89.

44) Eine *zulässige* semi-positive T-Invariante ist per definitionem der Schaltvektor von mindestens einer Schaltfolge, die unter mindestens einer erreichbaren Markierung M_r aktiviert ist. Des weiteren reproduziert zumindest diese eine Schaltfolge die Referenzmarkierung M_r , wenn die Schaltfolge tatsächlich ausgeführt wird. Sobald die Schaltfolge zur Markierung M_r zurückgeführt hat, ist sie dort per constructionem abermals aktiviert, kann erneut ausgeführt werden usw. ad infinitum. Also stellt die Referenzmarkierung M_r einen Livelock dar (vgl. dazu die Livelock-Definition).

Falls die Zulässigkeit der T-Invariante unbekannt ist, gilt wieder nur eine schwächere notwendige Bedingung: Wenn ein Netz eine Livelockmarkierung M_r besitzt, dann muß mindestens eine semi-positive T-Invariante existieren. Andernfalls - wenn das Netz keine semi-positive T-Invariante aufweist - wäre keine Schaltfolge möglich, die jemals eine Markierung M_r als Livelockmarkierung zu reproduzieren vermag.

45) Eine Livelock-Transition ist jede Transition t_n , deren zugehörige Komponente $it_{h,n}$ in einer zulässigen semi-positiven T-Invariante \underline{it}_h den Wert $it_{h,n}=0$ annimmt. Denn das Schalten der Transition t_n kommt in keiner derjenigen Schaltfolgen vor, deren Schaltvektor durch die T-Invariante \underline{it}_h dargestellt wird. Wenn diese Schaltfolgen unter einer erreichbaren Markierung M_r aktiviert sind und - wie in der voranstehenden Anmerkung erläutert - in einem Schaltprozeß des Netzes potentiell unendlich oft wiederholt ausgeführt werden, dann wird die Transition t_n mit $it_{h,n}=0$ in diesem Schaltprozeß niemals geschaltet. Falls die Transition t_n dabei potentiell unendlich oft aktiviert ist, handelt es sich um eine Livelock-Transition. Ob die Transition t_n tatsächlich immer aktiviert ist, läßt sich aus der o.a. T-Invariante \underline{it}_h allerdings nicht erkennen. Daher stellen die Transitionen t_n mit $it_{h,n}=0$ lediglich potentielle Livelock-Transitionen dar.

46) Vgl. dazu die Quellenangaben zur strukturellen Untersuchung dynamischer Netzeigenschaften mit Hilfe der Invariantenanalyse. Daraus bietet vor allem ABEL,D. (1990), S. 89, eine prägnante Formulierung als Netztheorem: Wenn ein Netz (markierungsschwach) lebendig und beschränkt ist, dann besitzt es mindestens eine *positive* T-Invariante. Aus der Kontraposition folgt: Wenn ein beschränktes Netz keine positive T-Invariante besitzt, dann kann es nicht (markierungsschwach) lebendig sein. Ein praktisches Anwendungsbeispiel für diese Umkehrung findet sich bei ABEL,D. (1990), S. 115f.

47) Vgl. dazu die Ausführungen zu den Netztheoremen von MURATA und ZHANG einerseits sowie von LAUTENBACH andererseits.

48) Dies trifft auf das o.a. Netztheorem für Livelockmarkierungen zu.

49) Zu diesem Zweck muß auf andere Analysetechniken - z.B. die Erreichbarkeitsanalyse - zurückgegriffen werden.

50) Zwar existiert für S-Invarianten eine "Standardinterpretation": Jede S-Invariante charakterisiert eine Menge von Stellen, auf denen sich unter jeder erreichbaren Markierung die gleiche *gewichtete* Gesamtanzahl von Markenkopien befindet. Vgl. z.B. REISIG (1986a), S. 92; ABEL,D. (1990), S. 87; PAGNONI (1990), S. 147 u. 149. Diese "Interpretation" umschreibt jedoch lediglich die formale Definition von S-Invarianten in natürlicher Sprache. Einen materiellen Gehalt kann der Verf. darin jedoch nicht entdecken. Zumindest gibt die gewichtete Gesamtanzahl von Markenkopien keinen Hinweis auf die dynamischen Netzeigenschaften, die in dieser Arbeit als interessant ausgezeichnet wurden. Ebenso bleibt vollkommen offen, welche Bedeutung der speziellen Gewichtung zukommen soll, der die stellenspezifischen Anzahlen von Markenkopien in jeder S-Invariante unterworfen werden. Die Gewichte stellen in der Regel Artefakte dar, die aus Gründen der formalen "Stimmigkeit" notwendig sind. Einer "natürlichen", problembezogenen Interpretation entziehen sie sich jedoch im Regelfall. S-Invarianten geben daher oftmals nur dann zufriedenstellende Antworten, wenn die Fragestellungen zuvor entsprechend "zurechtgebogen" worden sind; vgl. dazu beispielsweise das Vorgehen von PAGNONI (1990), S. 153ff., insbesondere S. 155.

S-Invarianten erlauben nur in einigen Sonderfällen eine intuitive überzeugende Ausdeutung, die sich auf den Hintergrund des jeweils bearbeiteten Realproblems bezieht. Dazu gehört zunächst der Spezialfall der Einheitsgewichtung. Er zeichnet sich dadurch aus, daß alle Komponenten $is_{h,m}$ einer semi-positiven S-Invariante is_h duale Werte annehmen: $is_{h,m} \in \{0,1\}$ für alle $s_m \in S$. Dann bietet es sich an, auf das Konstrukt der Unterstützungsmengen (supports) zurückzugreifen. Die Unterstützungsmenge einer S-Invariante ist eine Teilmenge der Stellenmenge S des untersuchten Netzes. Die Unterstützungsmenge US_h einer semi-positiven S-Invariante is_h umfaßt im allgemeinen eine Stelle s_m genau dann, wenn die stellenspezifische Komponente $is_{h,m}$ positiv ist: $s_m \in US_h \Leftrightarrow is_{h,m} > 0$. Für den speziellen Fall der hier vorausgesetzten Einheitsgewichtung gilt noch anschaulicher: $s_m \in US_h \Leftrightarrow is_{h,m} = 1$ und $s_m \notin US_h \Leftrightarrow is_{h,m} = 0$. Die Unterstützungsmenge US_h einer semi-positiven *und* einheitsgewichteten S-Invariante is_h erlaubt nun eine intuitiv eingängige Interpretation. Ihr zufolge charakterisiert die Unterstützungsmenge durch ihre Stellen einen in sich abgeschlossenen Ausschnitt aus dem zugrundeliegenden Netzmodell. Es handelt sich um ein invariantenspezifisches Teilmodell. Innerhalb dieses geschlossenen Teilmodells zirkuliert ein konstanter Markenfluß. Er umfaßt unter jeder erreichbaren Markierung eine feste, unveränderliche Anzahl "mi" von Markenkopien. Dies folgt unmittelbar aus der allgemeinen Gleichung " $is_h^t \cdot M_t = is_h^u \cdot M_0$ " für S-Invarianten und aus der speziell vorausgesetzten Einheitsgewichtung. Denn für alle erreichbaren Markierungen M_t gilt:

$$\begin{aligned}
 & is_h^t \cdot M_t = is_h^u \cdot M_0 = mi \\
 \Leftrightarrow & \sum_{(s_m \in US_h)} is_{h,m} \cdot M_t(s_m) + \sum_{(s_m \in (S-US_h))} is_{h,m} \cdot M_t(s_m) \\
 = & \sum_{(s_m \in US_h)} is_{h,m} \cdot M_0(s_m) + \sum_{(s_m \in (S-US_h))} is_{h,m} \cdot M_0(s_m) = mi \\
 & // s_m \in US_h \Leftrightarrow is_{h,m} = 1 \text{ und } s_m \in (S-US_h) \Leftrightarrow is_{h,m} = 0 \\
 \Leftrightarrow & \sum_{(s_m \in US_h)} 1 \cdot M_t(s_m) + \sum_{(s_m \in (S-US_h))} 0 \cdot M_t(s_m) \\
 = & \sum_{(s_m \in US_h)} 1 \cdot M_0(s_m) + \sum_{(s_m \in (S-US_h))} 0 \cdot M_0(s_m) = mi \\
 \Leftrightarrow & \sum_{(s_m \in US_h)} M_t(s_m) + 0 = \sum_{(s_m \in US_h)} M_0(s_m) + 0 = mi \\
 \Leftrightarrow & \sum_{(s_m \in US_h)} M_t(s_m) = \sum_{(s_m \in US_h)} M_0(s_m) = mi
 \end{aligned}$$

Folglich wird durch die Ausgangsmarkierung M_0 aller Stellen s_m , die zur Unterstützungsmenge US_h gehören, die konstante Anzahl "mi" von Markenkopien fixiert. Die "mi" Markenkopien können zwar unter allen erreichbaren Markierungen M_t zwischen den Stellen aus der Unterstützungsmenge US_h hin- und hergeschoben werden. Aber die Anzahl der Markenkopien wird weder vermehrt noch vermindert. Gleiches gilt für die Objekte des Netzmodells, die von den Markenkopien repräsentiert werden. Aus dieser Perspektive läßt sich das Teilmodell, das von den Stellen der Unterstützungsmenge US_h aufgespannt wird, als eine zusammenhängende Einheit mit einem in sich geschlossenen Objektfluß ansehen. Diese Interpretation von semi-positiven und einheitsgewichteten S-Invarianten findet sich z.B. bei ABEL,D. (1990), S. 92 u. 111ff.

Eine leichte intuitive Zugänglichkeit kann ABEL's Ausdeutung von S-Invarianten nicht abgesprochen werden. Dennoch leidet sie unter drei wesentlichen Einschränkungen. Erstens ist sie nur so lange zulässig, wie der Sonderfall der Einheitsgewichtung vorliegt. Zwar spricht ABEL,D. (1990), S. 87, davon, daß dieser Sonderfall "häufig" auftritt. Doch belegt er dies an keiner Stelle. Ein Vergleich mit den Invarianten, die in den Quellen aus der eingangs angeführten Anmerkung diskutiert werden, vermag ABEL's Häufigkeitsbehauptung ebenso nicht zu stützen. Zweitens trifft die Feststellung von ABEL im allgemeinen nicht zu, daß die repräsentierten Objekte in den invariantenspezifischen Teilmodellen nur "verschoben werden" (S. 92). Denn dieses Urteil läßt sich allenfalls auf die strukturlosen Markenkopien von Niederen Netzen - wie z.B. Stelle/Transition-Netzen - beziehen. Dort erscheint es aber widersinnig, vom "Verschieben" von Objekten sprechen zu wollen, obwohl diese Objekte nicht als mit sich identisch wiedererkannt werden können. Denn es ist unmöglich, mit den unterschiedslosen Kopien der Basismarke in Nieder-

ren Netzen eine Objektidentität zu modellieren. In Höheren Netzen kann dagegen das behauptete Verschieben von Objekten eklatant verletzt werden. Denn die einheitsgewichteten S-Invarianten fixieren nur die *Anzahl* der Objekte im invariantenspezifischen Teilmodell. Damit ist keineswegs ausgeschlossen, daß beim Schalten von Transitionen einzelne objektrepräsentierende Markenkopien vernichtet werden, während eine gleich große Anzahl von Markenkopien, die andere Objekte repräsentieren, uno actu erschaffen wird. Drittens trägt die Auszeichnung von invariantenspezifischen Teilmodellen - so sehr sie auch im Einzelfall intuitiv überzeugen mag - nicht zur Untersuchung derjenigen dynamischen Netzeigenschaften bei, die in dieser Arbeit als interessant ausgezeichnet wurden.

Dem letzten Vorbehalt begegnen die Ausführungen von FREEDMAN (1988b), S. 337f. i.V.m. S. 334. Dort dienen S-Invarianten dazu, die Lebendigkeit und Beschränktheit (Sicherheit) von Netzen zu untersuchen. Diese beiden Netzeigenschaften zählen zu den hier interessierenden dynamischen Aspekten von Netzmodellen. Allerdings trifft der Ansatz von FREEDMAN (1988b) nur für die Klasse der Bedingung/Ereignis-Netze zu. Diese Netzklasse ist ohnehin schon stark eingeschränkt. Sie wird aber ihrerseits noch weiter auf die Teilklasse der stark zusammenhängenden Bedingung/Ereignis-Netze verengt (vgl. FREEDMAN (1988b), S. 334). Wegen dieser überaus restriktiven Anwendungsvoraussetzungen wird der Beitrag von FREEDMAN (1988b) hier nicht weiter beachtet.

Eine weitere Anwendungsoption für S-Invarianten besteht darin zu untersuchen, ob Transitionen unter keiner erreichbaren Markierung aktiviert sind. Dabei können zwei Fälle unterschieden werden. Im ersten Fall wird überprüft, ob es sich bei *einzelnen* ausgezeichneten Transitionen um netztheoretische Fakten (faktische Transitionen) handelt. Mit ihnen läßt sich die Allgemeingültigkeit von Sachverhalten ausdrücken, die in einem Netzmodell immer zutreffen sollen; vgl. z.B. REISIG (1986a), S. 140ff., insbesondere S. 142f. Der zweite Fall erstreckt sich dagegen auf *alle* Transitionen eines Netzes. Es wird dann getestet, ob ein vorliegendes Netz zur Klasse der Faktetze gehört. Ein Faktetz liegt vor, wenn alle Transitionen tot sind, d.h. wenn keine von ihnen unter keiner erreichbaren Markierung aktiviert ist. Faktetze können u.a. benutzt werden, um aussagen- oder prädikatenlogische Formelsysteme zu repräsentieren. In dieser Arbeit werden Faktetze aber nicht weiter berücksichtigt, weil für die Repräsentation prädikatenlogischer Formelsysteme zwei andere Ansätze ausgewählt wurden: die operationalen und die deklarativen Netzmodelle. Daher spielt hier der Einsatz von S-Invarianten, der zur Überprüfung des Vorliegens von Faktetzen dienen soll, keine weitere Rolle. Vgl. statt dessen z.B. den Beitrag von THIELER-MEVISSSEN (1987a), S. 533, 535, 537 u. 541ff.

Dagegen ist die zuvor erwähnte Untersuchungsvariante, mit S-Invarianten den faktischen Charakter einzelner ausgezeichneter Transitionen zu testen, nicht ganz von der Hand zu weisen. Denn auch in der hier vorgelegten Ausarbeitung werden faktische Transitionen zugelassen, um Sachverhalte zu modellieren, die allgemeingültig sein sollen. Es handelt sich um die Repräsentation von Integritätsbedingungen durch faktische Transitionen. Solche Integritätsbedingungen werden in einem Netzmodell mit Sicherheit eingehalten, wenn gezeigt werden kann, daß sie allgemeingültige Sachverhalte ausdrücken: Dies ist genau dann der Fall, wenn die bedingungsrepräsentierenden Transitionen unter keiner erreichbaren Markierung aktiviert sind - und deshalb faktische Transitionen darstellen. Für diese spezielle Aufgabe, die universelle Nichtaktivierung und somit den faktischen Charakter einzelner Transitionen nachzuweisen, lassen sich S-Invarianten durchaus nutzen. Damit hat sich vor allem REISIG in seinen Werken REISIG (1985b), S. 93f. u. 133ff., und REISIG (1986a), S. u. 107f. u. 158ff., intensiv befaßt. Allerdings sieht der Verf. in dieser begrüßenswerten Interpretationsmöglichkeit für S-Invarianten keine besondere Bereicherung des Petrinetz-Konzepts. Denn die gleiche Frage, ob eine Transition unter allen erreichbaren Markierungen nicht aktiviert ist, läßt sich ebenso anhand eines Erreichbarkeitsgraphen beantworten: Dort braucht nur überprüft zu werden, ob die betrachtete Transition tot ist. Später wird sich zeigen, daß die Totheit einer Transition im Rahmen einer Erreichbarkeitsanalyse in leicht verständlicher Weise festgestellt werden kann. Die o.a. Beiträge von REISIG fallen jedoch wegen ihrer umfangreichen Beweisführungen mit aufwendigen Fallunterscheidungen vergleichsweise intransparent aus. Daher lassen S-Invarianten zwar in bezug auf faktische Transitionen eine interessante Interpretation zu. Doch sie erweisen sich dabei anderen Analysetechniken als deutlich unterlegen.

Eine letzte erwähnenswerte Einsatzmöglichkeit von S-Invarianten fand schon an früherer Stelle Beachtung. Dort wurde demonstriert, wie sich mit ihrer Hilfe feststellen läßt, daß bestimmte Markierungen in einem Netz unerreichbar sind. Aus dieser Erkenntnis folgt aber noch keine direkte Auskunft über eine jener dynamischen Netzeigenschaften, die in dieser Arbeit interessieren. Daher bleibt der Erkenntniswert der Unerreichbarkeit von Markierungen für die hier vorgelegte Ausarbeitung unbefriedigend. Interpretationen von S-Invarianten, die über die vorgenannten hinaus eingesetzt werden könnten, um interessante dynamische Eigenschaften von Netzmodellen zu untersuchen, sind dem Verf. unbekannt.

51) Es erstaunt, daß S- und T-Invarianten zwar in formaler Hinsicht vollkommen symmetrisch definiert sind, aber ihre Interpretationsmöglichkeiten drastisch auseinanderfallen. Dieses Phänomen eines interpretativen Symmetriebruchs hat bis heute noch keine befriedigende Erklärung gefunden.

52) Darauf wurde schon eingangs hingewiesen.

53) In den Invariantenanalysen von Netzen werden zwar durchaus Versuche unternommen, die formal identifizierten Invarianten materiell auszudeuten. Vgl. z.B. REISIG (1986a), S. 95ff. u. 160ff.; REISIG (1989b), S. 8f.; ABEL, D. (1990), S. 92 u. 111ff.

Doch diese Interpretationsversuche hinterlassen in den meisten Fällen einen äußerst dürftigen Eindruck: Entweder zeichnet die "Bedeutung" einer Invariante einen artifiziellen Aspekt aus, dessen praktische Relevanz nicht nachvollzogen werden kann. Dazu gehört z.B. die "Standardinterpretation" von S-Invarianten, die in einer früheren Anmerkung erwähnt wurde. Interpretationen dieser Art besitzen aus der problemorientierten Perspektive, vorgegebene produktionswirtschaftliche Modellierungsprobleme zu bearbeiten, keinen Wert. Oder es handelt sich bei der Bedeutung einer Invariante tatsächlich um eine interessante Netzeigenschaft im Sinne jener Festlegungen, die an früherer Stelle erfolgten. Dazu gehören z.B. Erkenntnisse über die Lebendigkeit oder Inkonsistenz eines Netzes. Diese begrüßenswerten Interpretationen unterliegen aber anderen Einschränkungen der Invariantenanalyse, die in diesem Kapitel erörtert werden. Dazu gehören beispielsweise im Hinblick auf die zuvor erwähnte Netzlebendigkeit und -inkonsistenz das Problem unzulässiger Schaltfolgen, die Voraussetzung unbeschränkter Markenzapazitäten und die Ermittlungsschwierigkeiten für Höhere Netze. Vgl. zu den beiden voranstehend skizzierten Fällen auch das "Dilemma der Invariantenanalyse".

54) Daß die Anwendungsprämissen nicht ausreichen, um die Korrektheit der Invariantenanalyse zu garantieren, läßt sich durch zwei Hinweise verdeutlichen.

Erstens wurde schon betont, daß die Interpretation einer semi-positiven T-Invariante als Schaltvektor einer Schaltfolge keineswegs korrekt sein muß. Statt dessen kann die betroffene Schaltfolge auch unzulässig sein, weil sie im untersuchten Netz überhaupt nicht aktiviert ist. Aufgrund ihrer konzeptionellen Unvollständigkeit ist die Invariantenanalyse aber nicht in der Lage, die Erfüllung der Aktivierungsbedingungen für alle Transitionen aus einer Schaltfolge zu gewährleisten. Dies wurde bereits begründet.

Zweitens werden bei der methodischen Ausgestaltung der Invariantenanalyse mitunter zusätzliche Vorkehrungen getroffen, um den Ausweis von Invarianten zu verhindern, deren Interpretation als dynamische Netzeigenschaften inkorrekt wäre. So wird mitunter darauf geachtet, daß bei der Invariantenanalyse nur solche Operationen ausgeführt werden, die den Orthanten nicht-negativer Ganzzahlen für Markenzahlen nicht verlassen. Auf diese Weise wird das Schalten von Transitionen unterbunden, das von mindestens einer ihrer Eingangsstellen so viele Marken abzüge, daß eine unzulässige negative Markenzahl entstünde. Andernfalls könnte eine solche negativ markierte Stelle zu einer - unzulässigen - Schaltfolge gehören, deren Schaltvektor eine semi-positive T-Invariante darstellt. Denn es läßt sich nicht ausschließen, daß diese T-Invariante den Schaltvektor einer Schaltfolge angibt, für die gilt: Falls ihre Transitionen in unzulässiger Weise geschaltet würden, so riefen sie auf einer Stelle zunächst eine negative Markenzahl hervor. Später würden andere Transitionen dieses Markendefizit wieder auffüllen, so daß am Ende der Schaltfolge wieder eine zulässige Netzmarkierung vorläge. Solche Kompensationen unzulässiger Schaltakte können grundsätzlich nicht verhindert werden, solange die Invariantenanalyse ohne zusätzliche Vorkehrungen nach Lösungen der o.a. Vektorgleichungen sucht.

55) Die Markenzapazitäten der Stellen konstituieren zwar auch einen Teilaspekt der statischen Netzstruktur. Aber *dieser* Teilaspekt wird in der Inzidenzmatrix eines Netzes per constructionem nicht erfaßt.

56) Vgl. zur Voraussetzung unbeschränkter Markenzapazitäten LAUTENBACH (1987a), S. 148; PAGNONI (1990), S. 139.

57) Zwar werden in Verfeinerungen der Invariantenanalyse auch Netze mit beschränkten Markenzapazitäten zugelassen; vgl. LAUTENBACH (1987a), S. 161f.; REISIG (1989a), S. 37.

Doch handelt es sich hierbei nur um eine *mittelbare* Anwendung der Invariantenanalyse. Denn die Netze mit beschränkten Markenzapazitäten werden durch die Einführung von Komplementärstellen zunächst in Ersatznetze mit unbeschränkten Markenzapazitäten transformiert. Erst auf diese Ersatznetze wird die Invariantenanalyse angewendet. Daher bleibt die Invariantenanalyse selbst auch weiterhin auf die Prämisse unbeschränkter Markenzapazitäten angewiesen.

58) Daneben sind auch Netze mit Inhibitorkanten betroffen. Sie werden hier aber nicht gesondert behandelt. Erstens sind sie noch nicht eingeführt worden. Zweitens wurde bereits angedeutet, daß sie sich als ein Sonderfall der hier thematisierten Informationsstellen und -kanten auffassen lassen.

59) Darüber hinaus können die Kantengewichte der 1-Schleife in Synthetischen Netzen dazu führen, daß nicht nur bestimmte Markenzahlen, sondern auch Marken mit bestimmten Attributausprägungen erforderlich sind, um die schleifenzugehörige Transition zu aktivieren.

60) Wenn zwischen ein zunächst unverknüpftes Paar aus einer Stelle s_m und einer Transition t_n zwei Kanten (s_m, t_n) und (t_n, s_m) eingefügt werden, so liegt eine 1-Schleife im früher festgelegten, engeren Sinn nur dann vor, falls beide Kanten gleich große Kantengewichte erhalten. Dann verhält sich aber der Koeffizient $c_{m,n}$ der Inzidenzmatrix \underline{C} bezüglich der Ergänzung des o.a. Kantenpaares invariant. Dies gilt unabhängig davon, ob Stelle/Transition-Netze mit Kantengewichten $W(s_m, t_n)$ und $W(t_n, s_m)$ oder ob Synthetische Netze mit Kantengewichten $\text{bfm}(s_m, t_n)$ und $\text{bfm}(t_n, s_m)$ betrachtet werden. Denn in beiden Fällen kann sich der Koeffizient $c_{m,n}$ wegen der Gleichgewichtung der Kanten aus einer 1-Schleife nicht verändern. Mit $c_{m,n,\text{prac}}$ und $c_{m,n,\text{post}}$ als Werten des Koeffizienten $c_{m,n}$ vor bzw. nach Hinzufügen des schleifenkonstituierenden Kantenpaares gilt z.B. für Synthetische Netze:

$$c_{m,n,\text{post}} = c_{m,n,\text{prae}} + \text{bfm}(t_n, s_m) - \text{bfm}(s_m, t_n) \quad // \quad \text{bfm}(t_n, s_m) = \text{bfm}(s_m, t_n)$$

$$\Leftrightarrow c_{m,n,\text{post}} = c_{m,n,\text{prae}} + 0$$

$$\Leftrightarrow c_{m,n,\text{post}} = c_{m,n,\text{prae}}$$

Weil in der Inzidenzmatrix zwischen einem unverknüpften Knotenpaar und einer 1-Schleife nicht differenziert wird, läßt sich mit ihrer Hilfe der topologisch bedingte Schaltzusammenhang zwischen den Stellen und Transitionen eines Netzes grundsätzlich nicht eindeutig repräsentieren, falls 1-Schleifen zugelassen werden. Denn der Koeffizient $c_{m,n}=0$ kann in einer Inzidenzmatrix einerseits ausdrücken, daß zwischen der Stelle s_m und der Transition t_n kein schaltrelevanter Zusammenhang besteht, weil sie durch überhaupt keine Kante miteinander verknüpft sind. Andererseits läßt sich derselbe Koeffizient $c_{m,n}=0$ auch in der Weise auslegen, daß zwischen der Stelle s_m und der Transition t_n eine 1-Schleife existiert, in der die Stelle s_m als Nebenbedingung das Schaltverhalten der Transition t_n beeinflusst. Trotz ihrer so unterschiedlichen Bedeutung für das potentielle Schaltverhalten eines Netzes lassen sich beide Interpretationen mit demselben Koeffizientenwert $c_{m,n}=0$ vereinbaren. Denn für den Fall Synthetischer Netze gilt:

a) kein schaltrelevanter Zusammenhang:

$$(t_n, s_m) \notin F \wedge (s_m, t_n) \notin F$$

$$\Rightarrow \text{bfm}(t_n, s_m) = \text{bfm}(s_m, t_n) = 0$$

$$\Rightarrow c_{m,n} = \text{bfm}(t_n, s_m) - \text{bfm}(s_m, t_n) = 0 - 0$$

$$\Rightarrow c_{m,n} = 0$$

b) Schaltzusammenhang im Sinne einer Nebenbedingung:

$$(t_n, s_m) \in F \wedge (s_m, t_n) \in F \wedge \text{bfm}(t_n, s_m) = \text{MTAN}_{u,v} \wedge \text{bfm}(s_m, t_n) = \text{MTAV}_{u,v} \wedge \text{MTAN}_{u,v} = \text{MTAV}_{u,v} = \text{MTA}_{u,v}$$

$$\Rightarrow c_{m,n} = \text{bfm}(t_n, s_m) - \text{bfm}(s_m, t_n) = \text{MTA}_{u,v} - \text{MTA}_{u,v}$$

$$\Rightarrow c_{m,n} = 0$$

61) Beispielsweise läßt sich das Verhaltenspotential eines zunächst reinen Netzes verändern, indem das Netz lediglich um zwei zusätzliche Kanten so erweitert wird, daß sie ein zuvor unverknüpftes Paar aus einer Stelle s_m und einer Transition t_n zu einer 1-Schleife zusammenführen. Dadurch können Schaltprozesse, die im originären, reinen Netz *zulässig* waren, im erweiterten, aber nunmehr unreinen Netz *unzulässig* werden. Dies ist immer dann möglich, wenn die neu eingeführte 1-Schleife eine Nebenbedingung ausdrückt, die von mindestens einem zulässigen Schaltprozeß des originären Netzes nicht erfüllt wird. Das wurde schon im Kontext von Stelle/Transition-Netzen ausführlicher dargelegt.

Ebenso können durch 1-Schleifen Zugriffe mehrerer Teilprozesse auf eine gemeinsam benötigte, aber knappe Ressource dargestellt werden. Die Ausführung der Teilprozesse hängt davon ab, ob die Ressource aktuell verfügbar oder aber gerade von einem anderen Teilprozeß in Anspruch genommen ist. Vgl. zur Bedeutung von 1-Schleifen für die Modellierung von gemeinsam genutzten knappen Ressourcen FREEDMAN (1988b), S. 337.

62) In einer früheren Anmerkung wurde aufgezeigt, daß sich die Koeffizienten $c_{m,n}$ der Inzidenzmatrix \underline{C} eines Netzes nicht verändern, obwohl ein gleichgewichtetes Kantenpaar ergänzt wird, das eine 1-Schleife konstituiert. Daher wird die hinzugefügte 1-Schleife so behandelt, als ob sie nicht existieren würde, obgleich sie das Schaltverhalten eines Netzes zu beeinflussen vermag. Deshalb ist die Repräsentation des Verhaltenspotentials von unreinen Netzen notwendig unvollständig, wenn sie sich nur auf die Verwendung von Inzidenzmatrizen gründet.

63) Wenn die Ergänzung einer 1-Schleife die Menge zulässiger Schaltfolgen eines Netzes verringert, diese Veränderung seines Verhaltenspotentials aber in der unvollständigen Netzrepräsentation durch die Inzidenzmatrix nicht angezeigt wird, dann ist diese Netzrepräsentation fehlerhaft. Denn die Netzrepräsentation täuscht weiterhin ein unverändertes Verhaltenspotential vor. Vgl. auch FREEDMAN (1988b), S. 337 ("Incidence Matrix, which is inherently unsuitable for modelling self-loops").

64) Diese Fehlerhaftigkeit droht z.B. für den Fall, der in einer früheren Anmerkung skizziert wurde. Dort zeigte sich, daß die Menge zulässiger Schaltfolgen in einem Netz verringert werden kann, wenn das - ansonsten unverän-

derte - Netz um Kanten erweitert wird, die mindestens eine 1-Schleife bilden. Diese ergänzte 1-Schleife findet aber keinen Niederschlag in der Inzidenzmatrix des Netzes. Folglich können sich auch die Netzinvarianten, die mit Hilfe dieser Inzidenzmatrix durch die o.a. Vektorgleichungen definiert werden, nicht verändern. Daher ist es *grundsätzlich unmöglich*, die beschriebene Änderung der dynamischen Netzstruktur, die sich als Reduzierung zulässiger Schaltfolgen äußert, im Rahmen der Invariantenanalyse zu erkennen. Falls das Vorliegen einer interessierenden dynamischen Netzeigenschaft u.a. von jenen Schaltfolgen abhängt, die aufgrund der hinzugefügten 1-Schleife unzulässig geworden sind, dann läßt sich diese Netzeigenschaft mittels der Invariantenanalyse nicht mehr korrekt untersuchen.

65) Vgl. VALETTE (1979b), S. 156.

66) Vgl. zur Einschränkung der Invariantenanalyse auf reine Netze SIFAKIS (1978), S. 474ff.; LAUTENBACH (1987a), S. 148; PAGNONI (1990), S. 25, 139 u. 149.

Zwar läßt LAUTENBACH (1987a), S. 160f., auch unreine Netze zu. Doch für diese Netzvariante werden nur Schleifen i.w.S. erlaubt. Hierfür setzt LAUTENBACH - in der Notation von Stelle/Transition-Netzen - *unterschiedliche* Kantengewichte voraus, welche die Beziehung $W(s_m, t_n) - W(t_n, s_m) = 1$ erfüllen. Davon werden aber die Schleifen i.e.S., die oben als 1-Schleifen angesprochen wurden, nicht erfaßt. Denn diese Schleifen zeichnen sich dadurch aus, daß die Markierung der schleifenzugehörigen Stelle s_m vom Schalten der Transition t_n nicht beeinflusst wird. Daher müssen 1-Schleifen notwendig *gleiche* Kantengewichte besitzen, für die $W(s_m, t_n) - W(t_n, s_m) = 0$ gilt.

67) Die Inzidenzmatrix vermag nur zwischen den Ein- oder Ausgangsstellen von Transitionen zu differenzieren. Dies liegt daran, daß in den Differenzgleichungen $c_{m,n} = W(t_n, s_m) - W(s_m, t_n)$ und $c_{n,m} = \text{bfm}(t_n, s_m) - \text{bfm}(s_m, t_n)$ für die Definition der Matrixkoeffizienten zwischen den Ein- und Ausgangsstellen derselben Transition durch ein negatives bzw. (implizites) positives Vorzeichen unterschieden wird. Da ein Drittes jenseits von positiven oder aber negativen Vorzeichen nicht definiert ist, besteht keine Möglichkeit, in solchen Differenzgleichungen die eigenständige Bedeutung, die Informationsstellen für das Schaltverhalten ihrer Ausgangstransitionen besitzen, auszudrücken.

68) Gleiches gilt für ein weiteres Netzkonstrukt, das erst später im Detail dargestellt wird: die Nulltestkanten. Diese Nulltestkanten können eine Stelle s_m und eine Transition t_n wiederum so miteinander verknüpfen, daß die Markierung der Stelle die Aktivierungsbedingung der Transition beeinflusst, ohne jedoch vom Schalten der Transition verändert zu werden. Daher gelten die voranstehenden Argumente für Netze, die solche Nulltestkanten enthalten, analog. Folglich müssen auch diese Netze aus dem Anwendungsbereich der Invariantenanalyse ausgeschlossen werden. Dies erweist sich als höchst problematisch, weil später aufgezeigt wird, daß die Darstellungsmöglichkeit von Nulltestkanten das Ausdrucksvermögen von Netzen erheblich ausweitet. Vgl. dazu die Erläuterungen zur TURING-Mächtigkeit von Netzen mit Nulltestkanten. Daher bedeutet der Einsatz der Invariantenanalyse, von vornherein auf eine Netzklasse mit besonders großem Ausdruckspotential zu verzichten.

69) Vgl. ZELEWSKI (1989c), S. 33.

70) Vgl. MEMMI (1981a), S. 10f.

71) Für Stelle/Transition-Netze existieren Lösungsmethoden, mit deren Hilfe sich grundsätzlich alle T- und S-Invarianten ermitteln lassen. Dazu gehören vor allem die beiden Lösungsmethoden, die von PASCOLETTI (1985), S. 119ff. i.V.m. S. 105ff., vorgelegt wurden. Mit ihrer Hilfe läßt sich die Menge aller "einfachen" T- und S-Invarianten eines Stelle/Transition-Netzes *vollständig* bestimmen. Solche einfachen Varianten sind aus keinen anderen Varianten zusammengesetzt. Alle weiteren Invarianten eines Netzes lassen sich aus diesen einfachen Varianten als deren Linearkombinationen ableiten. Die Menge aller zusammengesetzten Netzinvarianten kann zwar unendlich groß sein. Dennoch ist auch eine solche unendliche Invariantenmenge durch die vollständig bestimmbaren einfachen Netzinvarianten und durch die Zusammensetzungsoperation der Linearkombination wohldefiniert. ABEL, D. (1990), S. 93ff., befaßt sich ebenso ausführlich mit den Lösungsmöglichkeiten für lineare diophantische Gleichungssysteme. Allerdings bezieht er sich dabei weniger deutlich auf den Anlaß, Netzinvarianten zu bestimmen (dieses Ermittlungsziel wird erst auf S. 100 relevant). Vgl. am Rande auch ZELEWSKI (1989c), S. 33f., zu den Möglichkeiten, die Invarianten von Stelle/Transition-Netzen ohne grundsätzliche Schwierigkeiten zu ermitteln. Weitere Beiträge zur Lösung von linearen homogenen diophantischen Gleichungssystemen, die sich jedoch nicht auf die Invariantenanalyse von Netzen beziehen, präsentieren z.B. HUET (1978), S. 145ff., und CLAUSEN, M. (1989), S. 204ff. Dabei stellt CLAUSEN drei Methoden für die vollständige Generierung aller minimalen Lösungen vor, aus denen sich alle anderen Lösungen linear kombinieren lassen. Die minimalen Lösungen von CLAUSEN entsprechen daher den "einfachen" Invarianten von PASCOLETTI.

Allerdings kann das "grundsätzlich problemlose" Bewältigen linearer homogener diophantischer Gleichungssysteme immer noch erheblichen Lösungsaufwand bereiten. Vgl. dazu die zurückhaltenden Formulierungen von ABEL, D. (1990), S. 85f. u. 104 (er redet von "Schwierigkeiten" und "Problemen" der Invariantenermittlung). Vgl. ebenso PAGNONI (1990), S. 152. Die Autorin berichtet über Invariantenanalysen von Stelle/Transition-Netzen mit 20 bis 500 Knoten, die insgesamt 2 Monate Rechenzeit für den Zentralprozessor eines VAX 11/750-Systems erfordern hätten. Eine detaillierte Dokumentation dieser Invariantenanalysen findet sich bei COLOM (1989b), S. 81ff. Die längste Einzelanalyse brauchte für ein mittleres Netz, das nur 93 Knoten umfaßte, mehr als 3,5 Stunden Rechenzeit

auf dem zuvor genannten VAX-System; vgl. COLOM (1989b), S. 84, Table 3, Example BD_04, Alg. A2. Von solchen exorbitant hohen Rechenzeiten wird hier "wohlwollend" abgesehen. Statt dessen interessieren im folgenden nur die noch weitaus größeren, *prinzipiellen* Schwierigkeiten der Invariantenanalyse von Höheren Netzen.

72) Vgl. zu den rechentechnischen Problemen, denen die Invariantenermittlung von Höheren Netzen unterliegt, MURATA,TA. (1988b), S. 490; PAGNONI (1990), S. 161; ABEL,D. (1990), S. 41f.

Die Schwierigkeiten der Invariantenanalyse Höherer Netze ergeben sich aus der Aufgabe, die invariantendefinierenden Gleichungssysteme zu bewältigen. Diese Gleichungssysteme haben bei Höheren Netzen eine komplexe innere Struktur. Sie resultiert unmittelbar aus der hohen Komplexität der Inzidenzmatrizen von Höheren Netzen. Vgl. dazu beispielsweise die Inzidenzmatrizen und die daraus abgeleiteten Gleichungssysteme bei REISIG (1986a), S. 157, 161 u. 163; REISIG (1989b), S. 7; ABEL,D. (1990), S. 41.

Die Komplexität von Inzidenzmatrizen und Gleichungssystemen wird besonders deutlich, wenn es sich um Höhere Netze in der Gestalt von Prädikat/Transition-Netzen handelt. Ihre Inzidenzmatrizen enthalten im allgemeinen Koeffizienten, die entweder Variablen oder aber die Werte von Funktionen darstellen, deren Argumente mindestens eine Variable umfassen. Die skalare Auflösung der invariantendefinierenden Vektorgleichungen führt dazu, daß die Variable(n) aus dem Matrixkoeffizienten mit den Variablen aus dem Spaltenvektor der gesuchten Invarianten multipliziert werden muß (müssen). Solche Variablenprodukte bereiten bei der Lösung von Gleichungssystemen jedoch in der Regel erhebliche Schwierigkeiten. Darüber hinaus besitzen die invariantendefinierenden Gleichungssysteme bei Höheren Netzen in der Regel nicht mehr rein ganzzahligen Charakter, sondern stellen Produkte aus Ganzzahlen und symbolischen Ausdrücken dar. Dies trifft auch auf Gefärbte Netze zu. Im Falle von Synthetischen Netzen handelt es sich bei den symbolischen Ausdrücken um die formalen Objekte aus den sortenspezifischen Objektmengen OB_i (sofern diese formalen Objekte nicht zufällig mit Ganzzahlen zusammenfallen). Die formalen Objekte stellen Ausprägungen von Markenattributen dar.

Bislang ist dem Verf. noch kein Verfahren bekannt geworden, das gestatten würde, die Invarianten eines Höheren Netzes *in jedem denkmöglichen Fall* und ohne intuitive Eingriffe des Verfahrenanwenders zu ermitteln.

Allerdings wird mitunter ein "Trick" empfohlen, um die rechentechnischen Schwierigkeiten der Invariantenermittlung für Höhere Netze elegant zu umgehen: Es wird auf die Möglichkeit zurückgegriffen, Höhere Netze auf Stelle/Transition-Netze abzuwickeln. Diese Netzabwicklung wurde schon in einer früheren Anmerkung erwähnt. Die Stelle/Transition-Netze setzen einer Invariantenanalyse keine rechentechnischen Schwierigkeiten mehr entgegen. Denn sie besitzen eine einfache arithmetische Qualität, die weder Variablen und Funktionen noch symbolische Ausdrücke umfaßt.

Der Verf. steht der Abwicklung von Höheren Netzen zu Stelle/Transition-Netzen jedoch skeptisch gegenüber. Dazu bewegen ihn im wesentlichen zwei Vorbehalte. Erstens führt die Abwicklung eines Höheren Netzes zu einem überaus umfangreichen Stelle/Transition-Netz. Dies mag bei kleinen Demonstrationsbeispielen noch nicht ins Gewicht fallen. Aber bei anspruchsvolleren Höheren Netzen kommen doch Zweifel auf, ob die derzeit angebotenen automaten-gestützten Analysewerkzeuge die abgewickelten Stelle/Transition-Netze überhaupt noch speichern und dann innerhalb "akzeptabler" Rechenzeiten auswerten können. Beispielsweise ist dem Verf. bis heute noch keine Invariantenanalyse für ein abgewickelteres Höheres Netz bekannt geworden, das die Größenordnung der später präsentierten Fallstudie erreicht. Aber der Verf. räumt ein, daß dieser Kapazitäts- und Effizienzvorbehalt keine stringente Argumentationskraft besitzt. Der zweite Einwand besitzt dagegen prinzipielle Qualität. Denn die Abwicklung eines Höheren Netzes auf ein Stelle/Transition-Netz ist grundsätzlich nur so lange möglich, wie das Abwicklungsergebnis ein *endliches* Stelle/Transition-Netz liefert. Diese Anforderung wird jedoch keineswegs von allen Höheren Netzen erfüllt. Ihre Einhaltung kann lediglich beim Gebrauch von Gefärbten Netzen garantiert werden. Ihre spezielle Finitheitsvoraussetzung, ausschließlich endliche Mengen verschiedenartiger "gefärbter" Marken zuzulassen, sorgt für die Endlichkeit der abgewickelten Stelle/Transition-Netze. Die "individuellen" Marken aus Prädikat/Transition-Netzen brauchen eine analoge Finitheitsvoraussetzung jedoch nicht zu erfüllen. Daher sind Prädikat/Transition-Netze möglich, die eine unbegrenzte Vielfalt verschiedenartiger "individueller" Marken gestatten. Derart infinite Prädikat/Transition-Netze lassen sich grundsätzlich nicht zu endlichen Stelle/Transition-Netzen abwickeln. Daher besitzt die o.a. abwickelnde Invariantenanalyse keinen universellen Anwendungsbereich: Sie kann nicht auf jedes beliebige Höhere Netz angewendet werden. Nun wäre es möglich einzuwenden, Prädikat/Transition-Netze mit unbegrenzter Markenvielfalt stellten einen irrelevanten Sonderfall dar. Dies trifft jedoch im Rahmen der hier vorgelegten Ausarbeitung nicht zu. Denn es wurde schon dargelegt, daß eine solche unbegrenzte Vielfalt bei Synthetischen Netzen schon dann auftritt, wenn mindestens ein Attribut von mindestens einer Attributmarke einen unendlichen Definitionsbereich zulässiger Attributausprägungen besitzen darf. Dies wurde grundsätzlich zugelassen, um die Ausdrucksmächtigkeit Synthetischer Netze nicht unnötig einzuschränken. Daher kann zumindest auf die hier entfalteten Synthetischen Netze - oder ihre Rekonstruktion als Prädikat/Transition-Netze - die abwickelnde Invariantenanalyse nicht universell angewendet werden. Daher wird diese "trickreiche" Variante der Invariantenanalyse in dieser Arbeit nicht weiter berücksichtigt. Vgl. darüber hinaus zur Kritik an der Abwicklung Höherer Netze zu Stelle/Transition-Netzen ABEL,D. (1990), S. 42.

73) Der Begriff der Verfahrenseffektivität wird hier benutzt, um auszudrücken, ob ein Verfahren grundsätzlich in der Lage ist, ein vorgegebenes Sachziel zu erfüllen. Im Gegensatz dazu betrifft die Verfahrenseffizienz den formal-

zielbezogenen Aspekt, in welcher Relation Nutzen und Kosten der Verfahrensanwendung zueinander stehen. Vgl. zu dieser Unterscheidung zwischen Effektivitäts- und Effizienzbegriff die Erläuterungen in einer früheren Anmerkung.

74) Vgl. REISIG (1985b), S. 129ff.; REISIG (1986a), S. 153ff.; BATTISTON (1988), S. 32ff.; REISIG (1989a), S. 15ff. u. 43; REISIG (1989b), S. 6ff.

75) Vgl. dazu die Quellen, die kurz zuvor erwähnt wurden. Beispielsweise ist kaum nachzuvollziehen, wie die Produktregeln für S- und für T-Invarianten bei REISIG (1989b), S. 7 bzw. 9, zustandekommen.

76) Diese Verengung der Ausdrucksfähigkeit von Netzmodellen stellt eine Variante der Prokrustesbett-Modellierung dar, die an früherer Stelle kritisiert wurde: Um Invariantenanalysen auf Netzmodelle anwenden zu können, müssen die analysierbaren Netze zunächst so weit vereinfacht werden, daß sie zu einer realitätsnahen Modellierung nur noch wenig beizutragen vermögen. Es erfolgt eine lösungsorientierte Modellierung. Denn aus dem Interesse heraus, Analyseprobleme mit der Hilfe von Netzinvarianten lösen zu können, werden nur solche Netze zugelassen, die eine Invariantenanalyse gestatten. Ob die formalen Probleme, die sich dann noch analysieren lassen, für die Bewältigung von realen Koordinierungsproblemen irgendeine substantielle Bedeutung tragen, wird dagegen kaum noch beachtet. Daher besteht die Gefahr, einen typischen Fehler 3. Art zu begehen: Es werden "falsche" - weil zu ausdrücksschwach modellierte - Formalprobleme mit der Hilfe von Invariantenanalysen "richtig" gelöst.

77) Dies klang bereits an, als es darum ging zu untersuchen, ob ausgezeichnete Transitionen unter keiner erreichbaren Markierung aktiviert sind.

78) Die Invariantenanalyse gestattet hingegen nur die Überprüfung von notwendigen Bedingungen, sofern nicht stark eingeschränkte Netze vorausgesetzt werden.

79) Vgl. ABEL, D. (1990), S. 28.

5.2.3.4 Die Reduktionsanalyse

Die Reduktionsanalyse¹⁾ von Netzen stellt prima facie ein faszinierendes Konzept dar, um die Komplexität potentieller Netzverhaltensweisen mit dem Instrument der kontrollierten Komplexitätsreduktion zu beherrschen. Die Komplexitätsreduktion wird durch eine nicht-leere, endliche Menge von Transformationsregeln geleistet. Sie gestatten es, vorliegende komplexe Netze sukzessiv in jeweils weniger komplexe zu transformieren. Verschiedene Varianten der Reduktionsanalyse unterscheiden sich durch ihre spezifischen Transformationsregelmengen.

Die Netzreduktion erfolgt kontrolliert in zweierlei Hinsicht. Erstens sind die zulässigen Reduktionsschritte durch die Transformationsregeln eindeutig und vollständig definiert. Zweitens liegt jeder regelgestützten Netzreduktion mindestens ein Invarianztheorem zugrunde. Ihm zufolge erweisen sich bestimmte, durch das Theorem jeweils ausgezeichnete dynamische Netzeigenschaften invariant gegenüber beliebigen Anwendungen von Transformationsregeln aus der zugehörigen Regelmenge. Daher verhalten sich das jeweils untersuchte originäre Netz und das daraus abgeleitete reduzierte Netz äquivalent in bezug auf diejenigen Eigenschaften, deren Erhaltung die Invarianztheoreme garantieren.

Bei jeder Reduktionsanalyse werden die originären, komplexen²⁾ Netze durch Anwenden von Transformationsregeln so lange eigenschaftsäquivalent reduziert, bis das zuletzt gewonnene derivative Netz gestattet, die interessierenden Netzeigenschaften unmittelbar zu erkennen. Falls diese Eigenschaften nicht sofort intuitiv erfaßt werden können, kann die Reduktionsanalyse auf der Ebene der reduzierten Netze mit jeder anderen Technik der Netzanalyse kombiniert werden³⁾. Da bereits stark vereinfachte Netze vorliegen, lassen sich jene Untersuchungstechniken auf die reduzierten Netze tendenziell wesentlich leichter anwenden als auf die ursprünglich gegebenen, komplexen Netze.

Trotz ihres interessanten theoretischen Konzepts leidet die Reduktionsanalyse jedoch unter zwei erheblichen Schwierigkeiten, die ihre praktische Analyseeignung wesentlich einschränken. Erstens läßt sie sich bisher nur auf die Klasse der Stelle/Transition-Netze ohne größere Probleme anwenden. Reduktionsanalysen für Höhere Netze sind noch nicht bekannt geworden. Dies läßt vermuten, daß die Reduktionsmethoden entweder nicht hinreichend ausgereift oder sogar grundsätzlich ungeeignet sind, um die Komplexität Höherer Netze auf reduktive Weise zu beherrschen. Dies gilt insbesondere hinsichtlich der internen Strukturierung von Marken aus Höheren Netzen. Zweitens beziehen sich die unterschiedlichen Varianten der Reduktionsanalyse auf verschiedene dynamische Netzeigenschaften. Es können jeweils nur jene Eigenschaften untersucht werden, auf welche die Analysevarianten mit ihren Transformationsregelmengen und Invarianztheoremen speziell zugeschnitten sind. Derzeit besteht keine erfolgversprechende Perspektive, die unterschiedlichen Transformationsregelmengen und Invarianztheoreme aller Varianten der Reduktionsanalyse zu einer einzigen integrierten Analysetechnik zusammenzuführen. Daher bereitet es erheblichen Aufwand, für die Überprüfung mehrerer interessierender Netzeigenschaften alle hierfür erforderlichen Analysevarianten separat vorhalten und anwenden zu müssen.

Aufgrund der voranstehend skizzierten Schwierigkeiten stellt die Reduktionsanalyse keine Analysetechnik dar, die sich für die Untersuchung der dynamischen Eigenschaften von Synthetischen Netzen eignet. Dies betrifft zumindest die Reduktionsanalyse in ihrer heute vorliegenden Entwicklungsstufe⁴⁾.

Allerdings könnte der Einwand erhoben werden, daß sich alle Höheren Netze auf verhaltensäquivalente Stelle/Transition-Netze reduzieren lassen. Auf die resultierenden Stelle/Transition-Netze könnte dann die Reduktionsanalyse angewendet werden. Infolge der gesicherten Verhaltensäquivalenz wäre es schließlich möglich, die Erkenntnisse aus der Reduktionsanalyse der

Stelle/Transition-Netze auf die zugrundeliegenden Höheren Netze zurückzuübertragen. Der Verf. schließt sich dieser Argumentation aus drei Gründen nicht an.

Erstens berührt sie in keiner Weise die zweite der beiden o.a. Schwierigkeiten: Die Einschränkung aller Varianten der Reduktionsanalyse auf einzelne, jeweils variantenspezifische Netzeigenschaften, die sich untersuchen lassen, wird nicht überwunden.

Zweitens besteht die Möglichkeit, Höhere Netze auf verhaltensäquivalente Stelle/Transition-Netze zurückzuführen, nicht universell. Denn sie läßt sich nur so lange garantieren, wie der Spielraum aller Schaltmöglichkeiten für jede Transition eines Netzes endlich ist⁵⁾. Dies muß jedoch keineswegs der Fall sein. Synthetische Netze z.B. bieten ihren Transitionen unendlich große Schaltspielräume, falls die Mengen zulässiger Schaltfarben durch keine obere Schranke begrenzt werden.

Drittens ist die Reduzierung von Höheren Netzen auf Stelle/Transition-Netze selbst dann, wenn sie theoretisch möglich ist, oftmals praktisch unbefriedigend. Denn Höhere Netze werden insbesondere deswegen benutzt, weil sich mit ihrem größeren Ausdrucksreichtum komplexe Realprobleme kompakter und transparenter modellieren lassen als mit Stelle/Transition-Netzen. Falls nun solche Höheren Netze, die komplexe Realprobleme abbilden, auf Stelle/Transition-Netze reduziert werden, so müssen die letztgenannten notwendig sehr umfangreich und undurchsichtig ausfallen. Die mangelhafte Netztransparenz betrifft zwar nicht den Einsatz automaten-gestützter Analyseinstrumente. Aber der Umfang von Stelle/Transition-Netzen, die aus der Reduzierung Höherer Netze hervorgehen, kann schnell so groß werden, daß sie sich selbst mit den Mitteln der Automatischen Informationsverarbeitung nicht mehr praktisch handhaben lassen⁶⁾.

Ohne die Grenzen der praktischen Handhabbarkeit hier präzisieren zu wollen, kann doch darauf verwiesen werden, daß alle Softwarepakete für die Erreichbarkeitsanalyse von Stelle/Transition-Netzen Restriktionen unterliegen. Hierdurch wird entweder die Größe der zu untersuchenden Netze oder aber die Größe der zugehörigen Erreichbarkeitsgraphen begrenzt. In beiden Fällen ist es unmöglich, *beliebig* große Stelle/Transition-Netze untersuchen zu können⁷⁾. Die Restriktionen der Softwarepakete ließen sich zwar dadurch umgehen, daß auf den Einsatz der Automatischen Informationsverarbeitung verzichtet wird. Doch scheidet diese Denkmöglichkeit für die praktische Netzanalyse in der Regel ebenso aus. Denn die Erreichbarkeitsgraphen für Netze, die interessante Realprobleme modellieren, fallen ohnehin so umfangreich aus, daß ihre Untersuchung ohne Automatenunterstützung praktisch indiskutabel ist. Erschwerend kommt noch die o.a. Intransparenz jener Stelle/Transition-Netze hinzu, die aus der Reduzierung Höherer Netze im allgemeinen hervorgehen.

Anmerkungen zum Kapitel:

- 1) Vgl. zu Reduktionsanalysen von Netzen HOLT,R. (1971a), S. 51ff., 127ff. u. 162; MAYR,E. (1975), S. 18ff. u. 57f.; ULLRICH (1976), S. 5/17 (als Vorstufe zu einer anschließenden Erreichbarkeitsanalyse); GOSH (1977), S. 14ff., insbesondere S. 17; BERTHELOT (1986), S. 20ff.; BEKHI (1989), S. 245.
- 2) Die Reduktionsanalyse eignet sich aufgrund ihrer Vorgehensweise nur für die Untersuchung komplexer Netze. Netze mit einer einfachen Struktur bleiben daher hier von vornherein außer Acht. Der Verf. verzichtet darauf, den Begriff der Netzkomplexität zu präzisieren, da die Reduktionsanalyse von Netzen in dieser Arbeit keine Rolle spielt. Dies wird anschließend gerechtfertigt.
- 3) Kombinationen der Reduktions- mit der Erreichbarkeitsanalyse präsentieren z.B. ULLRICH (1976), S. 5/17; VALMARI (1988a), S. 96ff., insbesondere S. 96, 99 u. 106.
- 4) Damit unterscheidet sich das negative Eignungsurteil bezüglich der Reduktionsanalyse vom ebenso negativen Eignungsurteil hinsichtlich der Invariantenanalyse. Denn die Invariantenanalyse leidet unter einer strukturellen Inadäquanz, die sie für die Untersuchung dynamischer Netzeigenschaften *grundsätzlich* untauglich erscheinen läßt. Die Reduktionsanalyse erweist sich dagegen aus der Perspektive ihres Analysekonzepts als durchaus vielversprechend. Sie könnte *grundsätzlich* eine willkommene Analysetechnik darstellen. Aber ihre bisher erfolgte Ausreifung läßt es noch nicht zu, sie für die Eigenschaftsanalyse Höherer Netze praktisch einzusetzen.
- 5) Denn bei der Reduzierung Höherer Netze wird jede Transition t_n , die mit insgesamt C_n unterschiedlichen Schaltfarben c_n geschaltet werden kann, in einem verhaltensäquivalenten Stelle/Transition-Netz durch C_n verschiedene Transitionen ersetzt. Da in jedem Netz die Transitionenmenge endlich sein muß, ist eine solche Substitution nicht mehr möglich, falls die Schaltfarbenanzahl C_n unendlich groß sein darf.
- 6) Vgl. dazu die Erreichbarkeitsnetze, die an früherer Stelle eingeführt wurden. Mit ihrer Hilfe läßt sich zwar die dynamische Struktur beliebiger Netze, die zunächst durch Erreichbarkeitsgraphen ausgedrückt wurde, als ein Stelle/Transition-Netz darstellen. Aber bereits das einfache Beispiel, das zur Illustrierung der Optimierung von Netzmodellen präsentiert wurde, führt plastisch vor Augen, wie drastisch das Volumen des Erreichbarkeitsnetzes gegenüber dem zugrundeliegenden Netz angestiegen ist.
- 7) Bei der Beschränkung der zulässigen Netzgröße ist dies unmittelbar offensichtlich. Falls hingegen die Größe der zugehörigen Erreichbarkeitsgraphen begrenzt wird, so gilt nur ein mittelbarer Zusammenhang: Größere Netze müssen nicht notwendig größere Erreichbarkeitsgraphen besitzen. Aber es läßt sich leicht zeigen, daß für jede vorgegebene obere Schranke der Größe eines Erreichbarkeitsgraphen stets mindestens ein Stelle/Transition-Netz konstruiert werden kann, dessen zugehöriger Erreichbarkeitsgraph diese Schranke übersteigt. Dies kann z.B. durch Netze bewerkstelligt werden, die mindestens eine unsterbliche Transition besitzen, welche die Markenanzahl auf mindestens einer ihrer Ausgangsstellen durch jeden Schalttakt streng monoton erhöht. Dann kann die Anzahl der Markierungsknoten des zugehörigen Erreichbarkeitsgraphen über jede obere Schranke hinaus vermehrt werden, indem die Markenkapazität dieser einen Ausgangsstelle beliebig erhöht wird.

5.2.3.5 Die Erreichbarkeitsanalyse

5.2.3.5.1 Grundsätzliches

Die Erreichbarkeitsanalyse¹⁾ besteht aus der Untersuchung des Erreichbarkeitsgraphen eines Netzmodells. Diese Analysetechnik steht im Zentrum dieser Arbeit²⁾. Denn nur sie ist in der Lage, zwei wichtige Universalitätsanforderungen zugleich zu erfüllen. Dabei handelt es sich um die beiden Postulate:

- eines universellen Erkenntnispotentials und
- eines universellen Anwendungsbereichs.

Das universelle Erkenntnispotential der Erreichbarkeitsanalyse resultiert aus dem früher dargelegten Sachverhalt, daß für jedes Netz gilt: Sein Erreichbarkeitsgraph repräsentiert die dynamische Netzstruktur vollständig und explizit. Daher lassen sich grundsätzlich alle dynamischen Eigenschaften eines Netzmodells, die für ein Netz definiert werden können³⁾, durch die Auswertung seines Erreichbarkeitsgraphen untersuchen⁴⁾. Folglich ist es grundsätzlich möglich, jede dynamische Netzeigenschaft anhand eines Erreichbarkeitsgraphen festzustellen⁵⁾. Dieses universelle Erkenntnispotential vermögen die übrigen Analysetechniken dagegen nicht zu garantieren. Denn sie leiden unter einer unvollständigen Repräsentation der dynamischen Netzstruktur. So wurde aufgezeigt, daß die Deduktions- und die Invariantenanalyse wesentliche Aspekte der dynamischen Netzstruktur, die mit schaltbedingten Veränderungen von Netzmarkierungen zusammenhängen, nicht zu erfassen vermögen. Die Reduktionsanalyse ist nur in der Lage, *einzelne* - von der jeweils zugrundeliegenden Analysevariante abhängige - dynamische Netzeigenschaften korrekt zu erkennen. Sie kann aber nicht die Invarianz *aller* dynamischen Netzeigenschaften bezüglich der jeweils angewandten Netztransformationen garantieren.

Ihren universellen Anwendungsbereich⁶⁾ verdankt die Erreichbarkeitsanalyse ihrem konzeptionellen Ansatz, nicht Netze unmittelbar, sondern nur vermittels ihrer Erreichbarkeitsgraphen zu untersuchen. Damit verhält sich die Erreichbarkeitsanalyse invariant gegenüber beliebigen Modifikationen der zugrundeliegenden Netze, solange diese nur wohldefinierte Erreichbarkeitsgraphen besitzen. Die letztgenannte einschränkende Bedingung wird aber von allen Netzen erfüllt, die überhaupt eine dynamische Struktur besitzen. Denn eine dynamische Netzstruktur wird durch die Definition einer Ausgangsmarkierung und einer Schaltregel konstituiert. Sobald ein Netz über diese beiden Komponenten verfügt⁷⁾, läßt sich für sie aber auch immer der zugehörige Erreichbarkeitsgraph konstruieren. Dazu braucht lediglich die Schaltregel auf die Ausgangsmarkierung und alle von dort aus erreichbaren Markierungen iterativ angewendet zu werden⁸⁾. Daher kann die Erreichbarkeitsanalyse auf alle Netze mit wohldefinierter Netzstruktur angewendet werden. Bestimmte Netztopologien brauchen dabei weder vorausgesetzt noch ausgeschlossen zu werden.

Dagegen besitzen Deduktions-, Invarianz- und Reduktionsanalysen kein derart universelles Anwendungspotential. Sie lassen sich ohne größere Schwierigkeiten nur auf die relativ ausdrucksarme Klasse der Stelle/Transition-Netze anwenden. Darüber hinaus unterliegen sie weiterführenden, analysespezifischen Einschränkungen. Als *pars pro toto* wird auf die Voraussetzung unbeschränkter Markierungskapazitäten und auf den Ausschluß von unreinen Netzen seitens der Invariantenanalyse verwiesen.

Die Erreichbarkeitsanalyse stellt daher zur Zeit die einzige Analysetechnik dar, die sich für die Untersuchung *aller* dynamischen Eigenschaften von *beliebigen* Netzen eignet⁹⁾, sofern diese Netze überhaupt eine dynamische Struktur besitzen¹⁰⁾. Da diese doppelte Universalität bislang ausschließlich der Erreichbarkeitsanalyse zukommt, bevorzugt der Verf. diese Analysetechnik

für die Untersuchung dynamischer Netzeigenschaften grundsätzlich gegenüber allen anderen Analysetechniken. Dies gilt insbesondere auch im Hinblick auf die Invariantenanalyse¹¹⁾.

Erreichbarkeitsgraphen besitzen bei der Analyse dynamischer Netzeigenschaften gegenüber Netzinvarianten einen weiteren entscheidenden Vorteil: Die Erreichbarkeitsgraphen verfügen über eine analyseadäquate Struktur. Auf die Invariantenanalyse trifft dies jedoch nicht zu. Denn es wurde schon früher darauf hingewiesen, daß die Untersuchung von Netzinvarianten auf der Lösung von Gleichungssystemen beruht, die grundsätzlich für die Repräsentation *statischer* Wissensinhalte ausgelegt sind. Dies widerspricht dem Ziel, aus der Untersuchung des Verhaltenspotentials von Netzen deren dynamischen Eigenschaften zu erkennen. Erreichbarkeitsgraphen sind dagegen von vornherein so strukturiert, daß sie das gesamte Wissen über die Netzdynamik durch eine adäquate *dynamische* Ausdrucksform repräsentieren. Die zentrale Rolle spielen dabei die Schaltkanten eines Erreichbarkeitsgraphen. Ihre Beschriftung durch Schaltschritte, deren Ausführungen Referenz- in Folgemarkierungen transformieren, und ihre Ausrichtung entsprechend der Abfolge zwischen Referenz- und Folgemarkierungen bringen die dynamische Qualität von schaltbedingten Markierungsveränderungen unmittelbar zum Ausdruck.

Die Erreichbarkeitsanalyse von Netzen zeichnet sich auch dadurch aus, daß sie aus logischer Perspektive ein sehr mächtiges Instrument für die Untersuchung dynamischer Wissensinhalte darstellt. Denn das Ausdrucksvermögen von Erreichbarkeitsgraphen übersteigt das Formulierungspotential der Prädikatenlogik (1. Stufe)¹²⁾. Dies folgt bereits aus dem Umstand, daß durch die Markierungen von Netzen zeitlich variables Wissen repräsentiert wird, das die Monotonieprämisse der konventionellen Prädikatenlogik verletzt. Die nonmonotonen Veränderungen solcher Wissensinhalte werden durch Markierungsübergänge ausgedrückt, die in Erreichbarkeitsgraphen als Schaltkanten repräsentiert werden. Diese Markierungsübergänge werden in den zugrundeliegenden Netzen durch das Schalten von Transitionen bewirkt. Die Schaltakte realisieren jeweils das allgemeine Übergangsschema ÜS, das mit seinen Zuweisungsoperatoren ":= " und seiner sequentiellen Ausführungsreihenfolge der involvierten Teiloperationen Ausdrucksmittel umfaßt, die der Prädikatenlogik fremd sind.

Statt dessen bedeutet die Formulierung von schaltbedingten Markierungsübergängen, daß die Prädikatenlogik um Aspekte von Modal-, Zeit-, Aktions- oder Handlungslogiken oder von dynamischen Logiken bereichert wird. Allerdings tritt diese logische Bereicherung nicht unmittelbar zu Tage. Denn für die Spezifizierung der einzelnen Netzmarkierungen reicht die prädikatenlogische Formulierungskraft vollkommen aus: Jede Netzmarkierung M_i ist eine Interpretation I_i desjenigen prädikatenlogischen Formelsystems, das durch ein Netzmodell äquivalent repräsentiert wird. Die Übergänge von Referenz- zu ihren Folgemarkierungen werden in Erreichbarkeitsgraphen in rein graphentheoretischer Weise ausgedrückt, die vollkommen ohne die o.a. logischen Erweiterungen auskommt. Denn die Definition eines mathematischen Graphen reicht in keiner Weise über die konventionelle Prädikatenlogik hinaus. Allerdings beschreiben in einem Erreichbarkeitsgraphen die Schaltkanten zwischen den Netzmarkierungen auf graphentheoretische Weise, wie alternative Interpretationen des jeweils repräsentierten prädikatenlogischen Formelsystems ineinander transformiert werden können. Dieses Transformationswissen besitzt trotz seiner vordergründigen graphentheoretischen Formulierung hintergründig die Qualität der o.a. angeführten Erweiterungen der Prädikatenlogik.

Beispielsweise¹³⁾ läßt sich zeigen, daß eine modallogische Wissensrepräsentation vorliegt, wenn prädikatenlogische Beschreibungen der Zustände eines Systems mit einer Erreichbarkeitsrelation über der Zustandsmenge dieses Systems kombiniert werden¹⁴⁾. Dann stellt jede Zustandsbeschreibung für sich allein ein gewöhnliches prädikatenlogisches Formelsystem dar, das z.B. die Monotonieprämisse der Prädikatenlogik erfüllt. Wenn KRIPKE's modallogische Semantik multipler Welten¹⁵⁾ zu Hilfe genommen wird, läßt sich jedes solche zustandsbeschreibende Formelsystem als ein in sich geschlossenes, rein prädikatenlogisches Weltmodell vorstellen. Die Erreichbarkeitsrelation über der Zustandsmenge beschreibt dann die Möglichkeiten, von einem Weltmodell aus andere Weltmodelle zu erreichen. Bei einem solchen Wechsel der Welt-

modelle wird jeweils eine alte prädikatenlogische Weltmodellierung gegen eine neue ausgetauscht. Dadurch droht immer eine Verletzung der Monotonieprämisse: Es kann grundsätzlich nicht garantiert werden, daß alle Wissensinhalte, die im alten Weltmodell zutrafen, auch noch im neuen Weltmodell unverändert zutreffen.

Nun lassen sich die prädikatenlogisch beschriebenen Systemzustände aus der voranstehend skizzierten modallogischen Wissensrepräsentation unmittelbar mit den Zuständen eines prädikatenlogischen Objektmodells identifizieren. Dieses Objektmodell wird durch ein äquivalentes Netzmodell abgebildet. Dann gibt jede Markierung des Netzmodells durch ihre Faktenmenge das Wissen wieder, das der Modellierungsträger über den aktuellen Zustand des prädikatenlogischen Objektmodells besitzt. Die Erreichbarkeitsrelation über der Zustandsmenge aus der modallogischen Wissensrepräsentation entspricht genau der Erreichbarkeitsrelation des Netzmodells. Der Erreichbarkeitsgraph des Netzmodells bildet in seiner Knotenmenge alle Netzmarkierungen und in seiner Kantenmenge die gesamte Erreichbarkeitsrelation des Netzmodells ab. Folglich repräsentiert der Erreichbarkeitsgraph des Netzmodells auch den gesamten zustands- und erreichbarkeitsbezogenen Gehalt der zugrundegelegten modallogischen Wissensrepräsentation. Also besitzt der Erreichbarkeitsgraph eine inhärent modallogische Qualität.

Die voranstehend dargelegten Vorzüge der Erreichbarkeitsanalyse sind allerdings nur in dem Ausmaß gesichert, wie die untersuchten Netze *endliche* Erreichbarkeitsgraphen besitzen. Solche Erreichbarkeitsgraphen zeichnen sich durch eine endliche Knoten-¹⁶⁾ und eine endliche Kantenmenge¹⁷⁾ aus. Sie garantieren, daß sich die Untersuchung jeder dynamischen Netzeigenschaft mit endlichem Ressourceneinsatz bewältigen läßt. Infolgedessen sind alle dynamischen Eigenschaften von Netzen mit endlichen Erreichbarkeitsgraphen entscheidbar, d.h. sie können in endlicher Zeit entweder bestätigt oder aber widerlegt werden. Daher wird in dieser Arbeit vorausgesetzt, daß alle untersuchten Netze jeweils einen endlichen Erreichbarkeitsgraphen aufweisen (Endlichkeitsprämisse)¹⁸⁾.

Es könnte der Einwand erhoben werden, die Endlichkeit der Erreichbarkeitsgraphen brauche überhaupt nicht unterstellt zu werden. Denn das Konzept der Überdeckbarkeitsgraphen¹⁹⁾ ermöglicht es, auch Netze mit unendlichen Erreichbarkeitsgraphen²⁰⁾ zu berücksichtigen. Für jedes Netz mit einem unendlichen Erreichbarkeitsgraphen könne ein endlicher Überdeckbarkeitsgraph konstruiert werden²¹⁾. Es reiche aus, diesen endlichen Überdeckbarkeitsgraphen zu analysieren. Gegen die Verwendung von Überdeckbarkeitsgraphen für die Untersuchung dynamischer Netzeigenschaften sprechen jedoch zwei gewichtige Gründe.

Erstens leiden Überdeckbarkeitsgraphen unter dem Nachteil, daß sie unbeschränkte Knoten enthalten können²²⁾. Ein Knoten $M^{\bar{U}}$ des Überdeckbarkeitsgraphen heißt unbeschränkt, falls er für mindestens eine Stelle s_m mit $M^{\bar{U}}(s_m) = \omega$ eine unbeschränkte Markenanzahl ausweist. Die tatsächliche Markenanzahl einer solchen Stelle s_m vermag über jede endliche obere Schranke hinauszuwachsen. Es brauchen dazu nur hinreichend ausgedehnte Schaltprozesse betrachtet zu werden, die in erreichbaren Markierungen enden, die vom Knoten $M^{\bar{U}}$ "überdeckt" werden. Jeder solche unbeschränkte Knoten involviert die Gefahr, daß Informationen über die dynamische Netzstruktur verloren gehen²³⁾. Dieser Informationsverlust kann bedeuten, daß sich charakteristische dynamische Netzeigenschaften - wie etwa die Netzlebendigkeit - auf der Basis von Überdeckbarkeitsgraphen nicht mehr sicher erkennen lassen²⁴⁾. Schon aus diesem Grunde stellen Überdeckbarkeitsgraphen keine befriedigende konzeptionelle Basis dar, auf der eine allgemeine Analysetechnik für dynamische Netzeigenschaften fußen könnte²⁵⁾.

Zweitens trifft der oben angeführte Einwand, für Netze mit unendlichen Erreichbarkeitsgraphen ließen sich endliche Überdeckbarkeitsgraphen konstruieren, nur partiell zu. Denn das Konzept der Überdeckbarkeitsgraphen kümmert sich nur darum, das Verhaltenspotential von Netzen mit unbeschränkt anwachsenden Markenanzahlen in endlicher Weise zu behandeln. Dies reicht zwar für Stelle/Transition-Netze aus, aber nicht für Höhere Netze. Denn nur in Stelle/Transition-Netzen sind unendliche Schaltprozesse notwendig damit verbunden, daß die

Markenanzahl mindestens einer Stelle über jede beliebige, aber endliche obere Schranke anzuwachsen vermag. Das größere Ausdruckspotential Höherer Netze - insbesondere auch das von Synthetischen Netzen - ermöglicht jedoch, daß in ihnen unendliche Schaltprozesse ausgeführt werden können, *obwohl* die Markenanzahlen aller ihrer Stellen eine endliche obere Schranke nicht übersteigen²⁶⁾.

Dieser "pathologische" Fall kann z.B. für Synthetische Netze eintreten, wenn der Definitionsbereich mindestens eines Attributs von mindestens einer Attributmarke unendlich viele zulässige Attributausprägungen umfaßt. Dann ist es möglich, daß ein unendlicher Schaltprozeß existiert, in dem eine Kopie der Attributmarke nacheinander unendlich viele verschiedene Attributausprägungen annimmt. Dabei braucht die Anzahl der Markenkopien im Netz überhaupt nicht verändert zu werden. Abb. 65 auf der nächsten Seite zeigt ein triviales Zählernetz²⁷⁾. In ihm kann ein Schaltprozeß beliebig - potentiell unendlich - lange fortgesetzt werden, welcher den Wert der Zählermarke unablässig um den Wert "1" erhöht. Dennoch handelt es sich um ein 1-beschränktes Netz, weil sich auf seiner Stelle, die mit dem Prädikatssymbol "Zählerstand" beschriftet ist, unter jeder erreichbaren Markierung immer nur genau eine Marke befindet. Dieses Netz besitzt nicht nur einen unendlichen Erreichbarkeits-, sondern ebenso einen *unendlichen* Überdeckbarkeitsgraphen²⁸⁾. Denn die Technik von Überdeckbarkeitsgraphen, unbeschränkt anwachsende Markenanzahlen durch unbeschränkte Markierungsknoten abzufangen, läßt sich auf unendlich variierende Attributausprägungen nicht mehr anwenden²⁹⁾. Ein kleiner Ausschnitt aus dem unendlichen Überdeckbarkeitsgraphen des Zählernetzes, der mit dessen Erreichbarkeitsgraphen zusammenfällt, ist in Abb. 65 auf der nächsten Seite dargestellt.

In dieser Arbeit werden für die Attribute von Attributmarken grundsätzlich unendliche Definitionsbereiche zugelassen³⁰⁾. Daher können unendliche Überdeckbarkeitsgraphen nicht ausgeschlossen werden. Um die Analyse der dynamischen Eigenschaften eines Netzes in endlichem Rahmen vollziehen zu können, müßte daher auch für Überdeckbarkeitsgraphen eine Endlichkeitsprämisse aufgestellt werden. Daher entfällt der o.a. Einwand, mit der Hilfe von Überdeckbarkeitsgraphen ließe sich die Endlichkeitsforderung für Erreichbarkeitsgraphen umgehen.

Die Prämisse, ausschließlich endliche Erreichbarkeitsgraphen zu behandeln, erweist sich im Rahmen der hier vorgelegten Untersuchungen als unproblematisch. Zunächst gewährleistet die frühere Vereinbarung, für alle Stellen eines Netzes *beschränkte* Markenkapazitäten festzulegen, daß es sich bei allen analysierten Netzen um beschränkte Netze handelt³¹⁾. Dadurch wird die erste Quelle unendlicher Erreichbarkeitsgraphen, das unbeschränkte Anwachsen von Markenanzahlen, von vornherein ausgeschlossen³²⁾. Wegen der weiterhin zugelassenen unendlichen Definitionsbereiche von Attributen können zwar Netze mit unendlichen Erreichbarkeitsgraphen nicht grundsätzlich verhindert werden. Aber der Einsatz von Netzen für die Modellierung von Realproblemen hat immer wieder die Erfahrung bekräftigt, daß praktisch interessante Netzmodelle endliche Erreichbarkeitsgraphen besitzen³³⁾. Lediglich theoretische Artefakte, wie das oben vorgestellte Zählernetz, führen mitunter trotz beschränkter Markenkapazitäten zu unendlichen Erreichbarkeitsgraphen.

Im Zweifelsfall kann für ein Netzmodell zunächst ein Überdeckbarkeitsgraph konstruiert werden³⁴⁾. Falls er endlich ist und keinen Knoten für eine Markierung M_r mit einer unbeschränkten Markenanzahl $M_r(s_m) = \omega$ für mindestens eine Stelle s_m aufweist, ist nachgewiesen: Das zugrundeliegende Netzmodell ist beschränkt und besitzt einen endlichen Erreichbarkeitsgraphen. Dies ist der Regelfall, der in diesem und den folgenden Kapiteln unterstellt wird. Sobald jedoch während der Konstruktion des Überdeckbarkeitsgraphen ein erster Markierungsknoten M_r mit einer unbeschränkten Markenanzahl $M_r(s_m) = \omega$ für mindestens eine Stelle s_m entdeckt wird, steht mit Sicherheit fest, daß das zugrundeliegende Netzmodell unbeschränkt ist³⁵⁾. Folglich ist sein Erreichbarkeitsgraph unendlich. Es steht dann zur Disposition des Modellierungsträgers, ob er³⁶⁾:

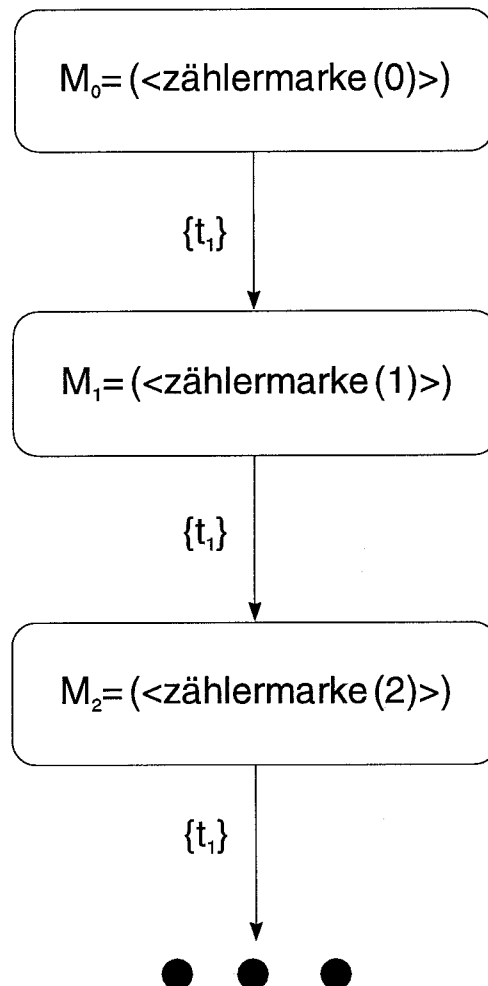
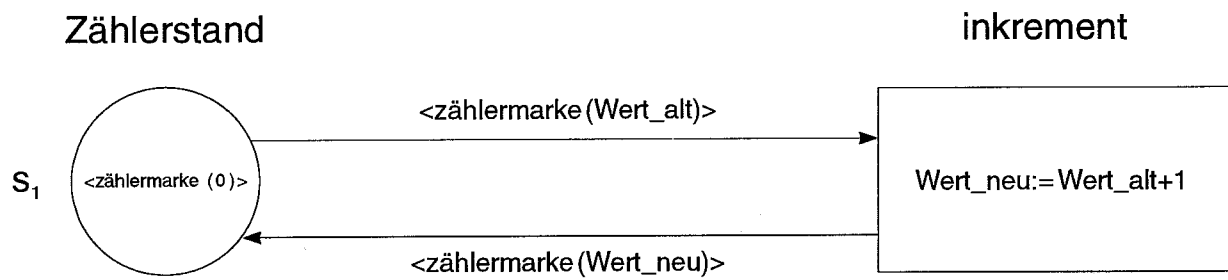


Abb. 65: Zählernetz

- entweder auf die Erreichbarkeitsanalyse seines unveränderten Netzmodells grundsätzlich verzichtet
- oder aber versucht, sein Netzmodell so lange zu modifizieren, bis der Überdeckbarkeitsgraph des Netzmodells endlich wird und keinen Markierungsknoten mehr enthält, der eine unbeschränkte Markenanzahl für mindestens eine Stelle aufweist.

Im ersten Fall muß der Modellierungsträger auf leistungsschwächere Auswertungstechniken zurückgreifen, um sein Netzmodell zu analysieren. Dafür bietet sich z.B. die Invariantenanalyse an, da sie ohnehin leichter angewendet werden kann, wenn mit unbeschränkten Markenskapazitäten für alle Stellen gearbeitet wird. Dann müssen aber auch alle Defekte der Invariantenanalyse in Kauf genommen werden, die bereits dargelegt wurden. Daher bevorzugt der Verf. den zweiten Fall. Er gestattet, die leistungsfähigere Erreichbarkeitsanalyse anzuwenden, sobald die Modifizierungsversuche erfolgreich abgeschlossen sind. Dann liegt qua Voraussetzung ein Netzmodell vor, dessen Erreichbarkeitsgraph die Finitheitsprämisse erfüllt.

Allerdings muß konzediert werden, daß der zweite Fall ein Ergebnisrisiko in sich birgt. Denn es ist kein Konstruktionsverfahren bekannt, das in der Lage wäre, jedes beliebige Netzmodell mit unendlichem Erreichbarkeitsgraphen in ein "ähnliches" Netzmodell mit endlichem Erreichbarkeitsgraphen zu transformieren. Es wäre aber auch verfehlt, die Existenz eines solchen generellen Konstruktionsverfahrens zu erwarten. Denn der Maßstab und das Ausmaß der "Ähnlichkeit", die jeweils präsupponiert werden, können die Suche nach einem modifizierten Netzmodell mit endlichem Erreichbarkeitsgraphen entscheidend beeinflussen - widrigenfalls sogar grundsätzlich vereiteln. Daher kann nicht ausgeschlossen werden, daß es dem Modellierungsträger nicht vergönnt ist, ein modifiziertes, aber noch "hinreichend ähnliches" Netzmodell mit endlichem Erreichbarkeitsgraphen zu erlangen³⁷⁾.

Die voranstehenden Disjunktionen bleiben aber unvollständig. Denn es läßt sich noch der "pathologische" Fall vorstellen, daß die Konstruktion eines Überdeckbarkeitsgraphen nicht terminiert, ohne jemals auf einen Markierungsknoten mit einer unbeschränkten Markenanzahl für mindestens eine Stelle zu stoßen³⁸⁾. Es handelt sich dann um einen *unendlichen* Überdeckbarkeitsgraphen für ein dennoch *beschränktes* Netzmodell. Die Existenzmöglichkeit solcher exotischer Netzmodelle wurde oben anhand des Zählernetzes aufgezeigt. Aufgrund der Unendlichkeit des Überdeckbarkeitsgraphen ist auch der zugehörige Erreichbarkeitsgraph unendlich. Im Prinzip stehen daher abermals die zwei Optionen offen, die schon zuvor erläutert wurden: Entweder verzichtet der Modellierungsträger auf die Erreichbarkeitsanalyse. Oder er versucht, sein Netzmodell so lange zu verändern, bis er ein Netzmodell mit endlichem Erreichbarkeitsgraphen erlangt. Die besondere Schwierigkeit des "pathologischen" Falls liegt darin begründet, daß die Unendlichkeit seiner Überdeckbarkeitsgraphen auf konstruktive Weise überhaupt nicht festgestellt werden kann. Statt dessen ist es mit endlichem Ressourceneinsatz immer nur möglich, einen endlichen Ausschnitt aus dem unendlichen Überdeckbarkeitsgraphen zu konstruieren. Daher muß das Konstruktionsverfahren für einen solchen Überdeckbarkeitsgraphen wegen realer Ressourcenbeschränkungen irgendwann abgebrochen werden. Aus einem Verfahrensabbruch läßt sich aber die Unendlichkeit des Überdeckbarkeitsgraphen nicht stringent folgern. Sie kann nur durch *zusätzliches* Wissen begründet werden, das aus der Analyse der dynamischen Struktur des zugrundeliegenden Netzmodells gewonnen wird³⁹⁾.

Abb. 66 auf der nächsten Seite faßt die zuvor ausgebreiteten Überlegungen zusammen. Sie gibt die vorbereitenden Schritte wieder, die im Zweifelsfall erfolgen können, um die Zulässigkeit der Erreichbarkeitsanalyse eines Netzmodells zu überprüfen. Die anschließenden Ausführungen gehen aber vom oben erwähnten Regelfall aus, daß die Finitheitsprämisse erfüllt ist⁴⁰⁾. Die untersuchten Netzmodelle besitzen daher stets endliche Erreichbarkeitsgraphen.

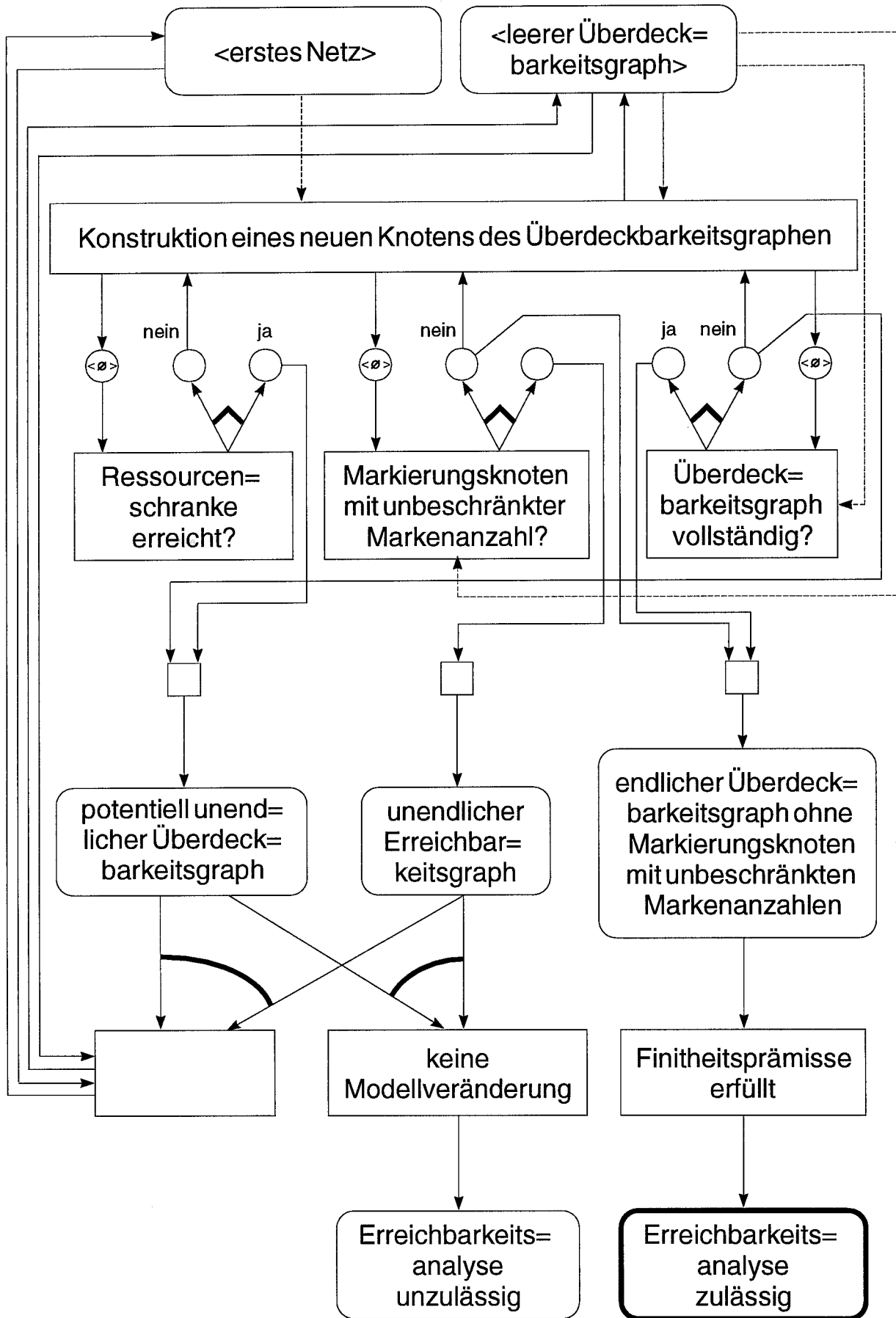


Abb. 66: Überprüfung der Zulässigkeit von Erreichbarkeitsanalysen

Durch die Finitheitsprämisse werden vielfältige Schwierigkeiten vermieden, die bei Netzen mit unendlichen Erreichbarkeitsgraphen drohen. Insbesondere kann die o.a. Entscheidbarkeit aller dynamischen Netzeigenschaften nicht mehr garantiert werden, falls für die Eigenschaftsüberprüfung auch unendliche Erreichbarkeitsgraphen in Betracht gezogen werden müssen. Beispielsweise läßt sich zwar für jedes Netz mit einem endlichen Erreichbarkeitsgraphen durch Untersuchung dieses Graphen entscheiden, ob sich das Netz lebendig verhält⁴¹⁾. Dagegen ist kein Algorithmus bekannt, der es gestatten würde, die Eigenschaft der Netzlebendigkeit selbst dann noch zu entscheiden, wenn auch Netze mit unendlichen Erreichbarkeitsgraphen zugelassen werden⁴²⁾. Es wurde schon oben erwähnt, daß der Rückgriff auf endliche Überdeckbarkeitsgraphen nicht ausreicht, um die Lebendigkeit von Netzen mit unendlichen Erreichbarkeitsgraphen zu entscheiden. Dies schließt zwar nicht aus, daß andere Analysekonzepte, die nicht auf Überdeckbarkeitsgraphen zurückgreifen, in der Lage sein könnten, die Netzlebendigkeit zu entscheiden. Aber derzeit sind solche alternativen Untersuchungskonzepte nicht bekannt. Das Lebendigkeitsproblem für Petrinetze mit unendlichen Erreichbarkeitsgraphen ist daher noch ungelöst⁴³⁾. Es gibt sogar Hinweise, die es plausibel erscheinen lassen, daß das Lebendigkeitsproblem für Netze tatsächlich unentscheidbar wird, wenn die Voraussetzung endlicher Erreichbarkeitsgraphen aufgehoben wird⁴⁴⁾.

Von den voranstehend skizzierten Schwierigkeiten kann aufgrund der oben eingeführten Finitheitsprämisse im folgenden abgesehen werden⁴⁵⁾. Statt dessen rückt eine andere Problematik in den Vordergrund: Die Erreichbarkeitsanalyse leidet unter einem schwerwiegenden Effizienzdefizit⁴⁶⁾. Es verhält sich komplementär zu der oben dargelegten doppelten Universalität dieser Analysetechnik. Damit unterstreicht die Erreichbarkeitsanalyse die allgemein zu beobachtende Tendenz, daß sich Konzepte um so weniger effizient anwenden lassen, desto inhaltsreicher sie formuliert werden. Diese gegenläufige Beziehung zwischen Gehalt und Effizienz wurde schon früher hinsichtlich des Verhältnisses erörtert, in dem Modellierungsgüte und -effizienz zueinander stehen.

Das Effizienzdefizit läßt sich sowohl in theoretischer als auch in praktischer Hinsicht belegen. Die erste Perspektive verfolgen komplexitätstheoretische worst case-Betrachtungen⁴⁷⁾. Bei ihnen wird untersucht, welche Gestalt die Komplexitätsfunktionen von Algorithmen im ungünstigsten Fall (worst case) annehmen, die zur Lösung eines Analyseproblems eingesetzt werden. Solche Komplexitätsfunktionen geben an, wie der Ressourcenverzehr bei der Algorithmusausführung vom Umfang⁴⁸⁾ des jeweils untersuchten Problems ansteigt. Für ein typisches⁴⁹⁾ Problem der Erreichbarkeitsanalyse - das Erreichbarkeitsproblem - konnte bewiesen werden, daß es sich außerordentlich komplex verhält⁵⁰⁾. Denn seine Lösung verbraucht im ungünstigsten Fall noch weitaus mehr Ressourcen, als es für die "NP-vollständigen" Probleme⁵¹⁾ der Fall ist⁵²⁾. Jene NP-vollständigen Probleme werden aber im Bereich des Operations Research bereits zu den "schwierigsten" Problemen gerechnet⁵³⁾. Da schon alle Lösungsalgorithmen für die schwierigen, NP-vollständigen Probleme als ineffizient betrachtet werden⁵⁴⁾, müssen sie für das noch komplexere Erreichbarkeitsproblem erst recht ineffizient ausfallen⁵⁵⁾. Daher läßt sich das Erreichbarkeitsproblem des Petrinetz-Konzepts - bei plastischer Ausdrucksweise - als ein "überschweres" oder "besonders ineffizient lösbares" Untersuchungsproblem charakterisieren⁵⁶⁾. Dieser Effizienzmangel gilt für die Erreichbarkeitsanalyse schlechthin⁵⁷⁾.

Allerdings kann eingewendet werden, daß das komplexitätstheoretische Beurteilungskonzept der worst case-Untersuchungen kein stichhaltiges Argument wider die Erreichbarkeitsanalyse liefere. Denn die Betrachtung des schlechtest denkmöglichen Falles sei für praktische Untersuchungsaufgaben der Erreichbarkeitsanalyse nicht oder allenfalls marginal relevant⁵⁸⁾. Es gelte vielmehr zu untersuchen, mit welcher Komplexitätsfunktion der durchschnittliche Ressourcenverzehr in Abhängigkeit vom Problemumfang ansteige, wenn "repräsentative" Sammlungen⁵⁹⁾ gleichartiger Problemausprägungen gelöst werden⁶⁰⁾. Die Vorziehenswürdigkeit solcher average case-Untersuchungen⁶¹⁾ der Durchschnittskomplexität wird hier nicht bestritten. Doch auch sie leiden unter Unzulänglichkeiten. Beispielsweise⁶²⁾ gilt es diejenigen Bedingungen zu

bestimmen, die Sammlungen gleichartiger Problemausprägungen als repräsentativ für die gesamte untersuchte Problemklasse auszeichnen⁶³).

Selbst wenn von den grundsätzlichen Schwierigkeiten abgesehen wird, denen die Durchschnittsbetrachtung repräsentativer Problemsammlungen unterliegt, so klafft immer noch eine Ausführungslücke. Denn bisher fehlt es für die Erreichbarkeitsanalyse von Netzen an repräsentativen Problemsammlungen und erst recht an entsprechenden Erkenntnissen über die average case-Komplexität des Erreichbarkeitsproblems. Gleiches gilt auch für die anderen zuvor thematisierten Untersuchungsaufgaben der Erreichbarkeitsanalyse, wie z.B. die Feststellung der Lebendigkeit oder Promptheit von Netzen. Lediglich aus dem praktischen Umgang mit einzelnen kleinen Beispielnetzen läßt sich immer wieder die Erfahrung gewinnen, daß ihr Umfang und ihr Untersuchungsaufwand mit zunehmender Netzgröße rasch ansteigen. Dies ist ein schwacher Indikator dafür, daß die Untersuchung dynamischer Netzeigenschaften mit Hilfe der Erreichbarkeitsanalyse auch aus praktischer Perspektive ein überaus anspruchsvolles, ressourcenintensives Unterfangen darstellt.

Sowohl die theoretische Einsicht, das Erreichbarkeitsproblem gehöre zu den "überschweren" Untersuchungsaufgaben, als auch die praktischen Hinweise auf den großen Untersuchungsaufwand, den die Konstruktion und Auswertung von Erreichbarkeitsgraphen bereitet, belegen das o.a. Effizienzdefizit der Erreichbarkeitsanalyse. Allerdings hält der Verf. diese Unzulänglichkeit für weniger bedeutungsvoll als die früher erläuterten Defizite der konkurrierenden Untersuchungstechniken, insbesondere der Invariantenanalyse. Denn bei jenen anderen Defiziten handelte es sich überwiegend um Schwierigkeiten grundsätzlicher Art, für die keine Aussicht besteht, sie in Zukunft überwinden oder zumindest kompensieren zu können⁶⁴). Bei der Erreichbarkeitsanalyse liegen die Verhältnisse hingegen anders.

Erstens kann nicht ausgeschlossen werden, daß noch Algorithmen aufgefunden werden, die Untersuchungsaufgaben der Erreichbarkeitsanalyse mit befriedigender Durchschnittskomplexität zu erfüllen vermögen. Sie würden die theoretischen Ergebnisse von worst case-Betrachtungen dadurch überwinden, daß sie deren praktische Irrelevanz aufzeigten⁶⁵).

Zweitens betreffen worst case-Argumentationen strenggenommen nicht die Komplexität eines Analyseproblems, sondern die Komplexität der Algorithmen, die für die Lösung des Analyseproblems bekannt sind. Daher besteht durchaus die Chance, neuartige Analysealgorithmen zu entdecken, deren worst case-Komplexität kleiner ist als diejenige aller bis dahin bekannten problemlösenden Algorithmen. Dadurch wird auch die Komplexität des Analyseproblems, die früher mit der niedrigsten Komplexität der damals bekannten Algorithmen vorläufig gleichgesetzt worden war, nach unten korrigiert. Diese Möglichkeit, die worst case-Komplexität von Analyseproblemen nachträglich zu reduzieren, stellt keineswegs eine "abstrakte Denkmöglichkeit" dar. Vielmehr wurde sie bereits im Bereich der Erreichbarkeitsanalyse von Netzen verwirklicht. VALMARI gelang es, die worst case-Komplexität eines speziellen Analyseproblems⁶⁶) durch eine geschickte Algorithmuskonstruktion so weit herabzudrücken, daß der Lösungsaufwand vom Problemumfang nicht mehr in exponentiell, sondern nur noch in polynomial - nämlich quadratisch - beschränkter Weise abhängt⁶⁷).

Drittens läßt sich auch hier auf das schon früher benutzte Argument verweisen, daß die praktische Bedeutung von Effizienzdefiziten immer nur für das aktuelle Leistungsniveau der Automatischen Informationsverarbeitung gilt. Das beträchtliche Ansteigen dieses Leistungsvermögens in den vergangenen Jahren und fundierte Prognosen seiner zukünftigen Fortentwicklung lassen erwarten, daß dieses Leistungsniveau noch erheblich zunehmen wird⁶⁸). In diesem Fall würde das Effizienzdefizit der Erreichbarkeitsanalyse durch steigende Verarbeitungsleistungen entsprechender automatengestützter Analyseprogramme zunehmend kompensiert. Einen besonderen Beitrag in dieser Richtung lassen Automatische Informationsverarbeitungssysteme mit parallelen Architekturen und darauf implementierten nondeterministischen Algorithmen erwarten. Sie werden später ausführlicher behandelt.

Die drei voranstehenden Argumentationsstränge ändern zwar nichts an dem gegenwärtigen Effizienzdefizit der Erreichbarkeitsanalyse. Aber sie zeigen doch auf, daß dieser Schwachpunkt keine grundsätzlich unüberwindbare Qualität besitzt. Diese gerechtfertigte Aussicht auf Linderung des Effizienzdefizits zusammen mit der eingangs herausgestellten doppelten Universalität der Erreichbarkeitsanalyse haben den Verf. bewogen, sie gegenüber den anderen vorgestellten Analysetechniken vorzuziehen.

Anmerkungen zum Kapitel:

1) Vgl. zur Erreichbarkeitsanalyse von Netzen BYRN (1974), S. II-112f.; HUEN (1975), S. 59; ULLRICH (1976), S. 5/17f.; VALETTE (1977a), S. 78; STARKE (1980), S. 119; LAUER (1980), S. 503; PETERSON, J. (1981), S. 101; ZELEWSKI (1986c), S. 15ff.; VALMARI (1988a), S. 95ff.; ZELEWSKI (1988b), S. 356ff.; ABEL, D. (1990), S. 60ff. u. 106ff.; SEICHE (1990), S. 16f. u. 19ff.; PAGNONI (1990), S. 23ff. u. 141ff., insbesondere S. 143f.

Es liegen auch Softwarepakete vor, welche die Erreichbarkeitsanalyse von Netzen unterstützen. Dazu gehört z.B. das Softwarepaket PASIPP, das in dieser Arbeit schon mehrfach gewürdigt wurde. Allerdings gestattet PASIPP nur die interaktive Konstruktion von Erreichbarkeitsgraphen. Falls unter einer Markierung mehrere Schaltschritte aktiviert sind, muß der Benutzer jeweils entscheiden, welchen Schaltschritt er ausführen lassen möchte. Eine vollautomatische Konstruktion von Erreichbarkeitsgraphen kann auf diese Weise nicht vorgenommen werden. Eine automatische Analyse von Erreichbarkeitsgraphen ist ohnehin nicht vorgesehen.

Wesentlich weiter reicht dagegen das Leistungsvermögen des Softwarepakets P-NUT. Es ist in der Lage, Erreichbarkeitsgraphen ohne Benutzereingriffe sowohl zu konstruieren als auch zu analysieren. Bei der Analyse von Erreichbarkeitsgraphen ist besonders bemerkenswert, daß sie keineswegs auf vordefinierte Netzeigenschaften eingeschränkt ist. Vielmehr kann der Benutzer beliebige Netzeigenschaften definieren, die sich auf die Stellen, Transitionen, Markierungen oder Markierungsübergänge eines Netzmodells beziehen. Es wird lediglich vorausgesetzt, daß die zu untersuchenden Eigenschaften als prädikatenlogische Formeln 1. Stufe ausgedrückt sind. Dies bestätigt nachträglich die Vorgehensweise in dieser Arbeit, alle interessanten Netzeigenschaften ebenso mit der Hilfe solcher Prädikate formal zu definieren. Darüber hinaus verfügt das Softwarepaket sogar über eine Option, Erreichbarkeitsgraphen in einer baumartigen Weise auszudrucken. Das voranstehend skizzierte Leistungsspektrum von P-NUT wird näher beschrieben bei RAZOUK (1985c), S. 9f. u. 15ff., insbesondere S. 15f. (Graphkonstruktion), S. 17f. (Graphausdruck) u. S. 18ff. (Graphanalyse). Vgl. darüber hinaus zum selben Softwarepaket MORGAN (1986), S. 49ff.; MORGAN (1987), S. 1080ff.

Darüber hinaus wird die Erreichbarkeitsanalyse in prinzipiell gleichartiger Weise auch auf andere dynamische Systeme angewandt. Dabei handelt es sich um Systeme, deren zulässige Verhaltensweisen sich durch das Erreichen von zulässigen Systemzuständen beschreiben lassen. Solche dynamischen Systeme werden des öfteren als Transitionssysteme thematisiert, weil die Übergänge zwischen zwei benachbarten zulässigen Systemzuständen durch "Transitionen" bewirkt werden. Diese "Transitionen" entsprechen weitgehend den Transitionen von Petrinetzen. Nähere Beschreibungen von Transitionssystemen finden sich z.B. bei FINKEL (1987b), S. 499ff. (dort wird die enge Verwandtschaft zwischen Transitionssystemen und Petrinetzen ausführlich gewürdigt).

Vgl. zur Erreichbarkeitsanalyse von dynamischen (Transitions-)Systemen HOLZMANN (1987), S. 339ff.; FINKEL (1987b), S. 500ff.

2) In ähnlicher Weise wird der Erreichbarkeitsanalyse von Netzen eine herausragende Stellung eingeräumt durch VALMARI (1988a), S. 95; PAGNONI (1990), S. 143 u. 161 (allerdings unter Vorbehalt; vgl. S. 23); HOLZMANN (1991), S. 185 (als Standardprozedur für die Analyse von Netzmodellen).

3) Die Erreichbarkeitsanalyse braucht daher nicht auf diejenigen dynamischen Netzeigenschaften beschränkt zu werden, die an früherer Stelle eingeführt und als interessant ausgezeichnet wurden. Vielmehr kann sie auf jede beliebige wohldefinierte Eigenschaft der Netzdynamik angewendet werden.

4) Alle dynamischen Netzeigenschaften sind logische Konsequenzen der dynamischen Netzstruktur. Daher steht das Wissen über *alle* dynamischen Netzeigenschaften - zumindest implizit - zur Verfügung, sobald eine *vollständige* Repräsentation der dynamischen Netzstruktur vorliegt. Dies ist bei Erreichbarkeitsgraphen durch die graphische Repräsentation der Erreichbarkeitsmengen und -relationen von Netzen der Fall. Das Wissen über die dynamischen Netzeigenschaften kann jedoch solange nicht angewandt werden, wie es nur in impliziter Form vorliegt. Die Netzanalyse erfordert daher auch eine Explizierung aller vorerst nur implizit bekannten Netzeigenschaften. Die Explizierung impliziten Wissens kann aber erhebliche Schwierigkeiten bereiten. Dies hat der Verf. an anderer Stelle ausführlicher dargelegt; vgl. ZELEWSKI (1986a), S. 342ff. Die Aufgabe der Wissensexplizierung wird aber wesentlich erleichtert, wenn die Repräsentation der dynamischen Netzstruktur bereits in expliziter Weise vorliegt. Denn dann können zumindest alle Probleme vermieden werden, die sich auf die Explizierung dynamischen Strukturwissens beziehen. Daher erweist es sich als vorteilhaft, daß das Wissen über die dynamische Netzstruktur in Erreichbarkeitsgraphen nicht nur vollständig, sondern auch *explizit* repräsentiert wird. Es verbleibt daher "nur" noch die Aufgabe, dieses explizit verfügbare Strukturwissen so auszuwerten, daß das darin implizit enthaltene Wissen über dynamische Netzeigenschaften ebenfalls expliziert wird. Genau dies leistet die nachfolgend erläuterte Technik der Erreichbarkeitsanalyse.

5) Diesem universellen Erkenntnispotential scheint auf den ersten Blick das Phänomen der Kontaktsituationen zu widersprechen. Kontaktsituationen wurden in dieser Arbeit nicht näher thematisiert, weil sie für die Modellierung realer Produktionssysteme keine Relevanz besitzen. Aber in der Netzliteratur finden Kontaktsituationen des öfteren größere Beachtung. Eine Kontaktsituation wird durch ein Paar aus einer Markierung und einer Transition definiert: Eine Kontaktsituation liegt genau dann vor, wenn die Transition unter der Markierung nur deswegen nicht aktiviert

ist, weil mindestens eine Stelle aus dem Nachbereich der Transition nicht genügend freie Markenkapazität besitzt. Vgl. zur Definition solcher Kontaktsituationen ABEL,D. (1990), S. 18.

Das besondere Interesse an Kontaktsituationen resultiert aus deren Vermeidung. Ein Netz, in dem keine Kontaktsituationen eintreten können wird als kontaktfrei bezeichnet. Aus der Definition von Kontaktsituationen folgt unmittelbar: In einem kontaktfreien Netz hängt die Aktivierung einer jeden Transition unter jeder Markierung *ausschließlich* von den Stellen aus dem Vorbereich der betroffenen Transition ab; vgl. ABEL,D. (1990), S. 18 und - besonders deutlich - S. 29. Daher kann in einem Netz, das als kontaktfrei nachgewiesen ist, auf die Berücksichtigung knapper Markenkapazitäten vollkommen verzichtet werden. Entsprechend läßt sich die Schaltregel für kontaktfreie Netze vereinfachen. Aufgrund dieser Vereinfachungsmöglichkeiten, die sich z.B. für Beweisführungen der Netztheorie nutzen lassen, besteht ein Interesse, die Kontaktfreiheit von Netzen zu überprüfen. Dabei stellt die Kontaktfreiheit eine dynamische Netzeigenschaft dar, weil sie von den erreichbaren Markierungen des untersuchten Netzes abhängt.

Die Kontaktfreiheit eines Netzes läßt sich aber prima facie nicht anhand seines Erreichbarkeitsgraphen überprüfen. Denn der Erreichbarkeitsgraph weist nur alle Markierungen des Netzes aus, die sich durch zulässige Schaltakte von Transitionen tatsächlich erreichen lassen. Dagegen spielen für die fragliche Kontaktfreiheit eines Netzes gerade diejenigen Markierungen eine Rolle, die *nicht* erreicht werden konnten, weil knappe Markenkapazitäten die Zulässigkeit der Schaltakte von Transitionen verhinderten. Solche nicht-erreichbaren Markierungen weist ein Erreichbarkeitsgraph aber per constructionem nicht aus. Daher reicht ein gewöhnlicher Erreichbarkeitsgraph nicht aus, um die Kontaktfreiheit eines Netzes zu überprüfen. Dies wird bei ABEL,D. (1990), S. 66 (zusätzlich Kenntnis von Inzidenzmatrix und Kapazitätsvektor), besonders deutlich. Folglich scheint die oben behauptete universelle Erkenntnismöglichkeit aller dynamischen Netzeigenschaften auf die Erreichbarkeitsanalyse doch nicht zuzutreffen.

Diese Schlußfolgerung wäre jedoch verfehlt, weil sie das inhärente Erkenntnispotential der Erreichbarkeitsanalyse verkennt. Tatsächlich ist es möglich, mit ihrer Hilfe auch die Kontaktfreiheit eines Netzes zu überprüfen. Zu diesem Zweck brauchen lediglich Informationen, die bei der Konstruktion eines Erreichbarkeitsgraphen ohnehin anfallen, sorgsamer ausgewertet zu werden. Denn bei jeder Etablierung eines Markierungsknotens wird überprüft, welche Transitionen des untersuchten Netzes unter der aktuellen Netzmarkierung aktiviert sind. Dabei werden *alle* Transitionen aus der Transitionenmenge des Netzes betrachtet. Falls die Aktivierungsbedingung einer Transition nicht erfüllt ist, wird sie bei gewöhnlich konstruierten Erreichbarkeitsgraphen für den aktuell untersuchten Markierungsknoten nicht weiter beachtet. An diesem Punkt läßt sich aber die Konstruktion des Erreichbarkeitsgraphen erweitern: Für eine jede Transition, deren Aktivierungsbedingung nicht erfüllt ist, wird festgehalten, welche Komponenten der Aktivierungsbedingung verletzt werden. Falls mindestens eine der Einflußstellen der Transition nicht hinreichend markiert ist, spielt die Transition keine weitere Rolle. Wenn jedoch alle ihre Einflußstellen hinreichend markiert sind, muß die Nichterfüllung der Aktivierungsbedingung daran liegen, daß mindestens eine der Ausgangsstellen der Transition keine ausreichende freie Markenkapazität besitzt. Genau dann liegt eine Kontaktsituation vor. Folglich bereitet es an dieser Stelle bei der Konstruktion des Erreichbarkeitsgraphen keine Schwierigkeiten, den untersuchten Markierungsknoten um die Information anzureichern, daß die betroffene Transition zusammen mit der aktuellen Markierung eine Kontaktsituation bildet. Wenn auf diese Weise der gesamte Erreichbarkeitsgraph konstruiert worden ist und keine einzige Kontaktsituation angezeigt wurde, steht die Kontaktfreiheit des zugrundeliegenden Netzes fest. Deshalb läßt sich mit einer geringfügig erweiterten Erreichbarkeitsanalyse auch die dynamische Netzeigenschaft der Kontaktfreiheit überprüfen (q.e.d.).

6) Die universelle Anwendbarkeit der Erreichbarkeitsanalyse klingt auch bei ULLRICH (1976), S. 5/17, und PAGNONI (1990), S. 23 u. 143, an.

7) Dies muß keineswegs der Fall sein, wie die eingangs vorgestellten Allgemeinen Netze und Petrinetze i.e.S. belegen. Diese Netze weisen aber auch keine dynamische Struktur auf.

8) Dies wurde schon für Stelle/Transition-Netze auf informale Weise skizziert. Später wird die Konstruktion von Erreichbarkeitsgraphen für Synthetische Netze detailliert ausgeführt und formal präzisiert.

9) Dies unterstreicht für den speziellen Aspekt der Netzlebensdauer BERTHELOT (1977), S. 217: Er sieht in der Konstruktion von Erreichbarkeitsmenge und zugehörigem Erreichbarkeitsgraphen die einzige Möglichkeit, die Lebensdauer von *beliebigen* Netzen zu untersuchen.

10) Die Erfüllung dieser Prämisse wird fortan für alle angesprochenen Netze vorausgesetzt. Die Prämisse wird daher nicht mehr explizit erwähnt.

11) Vgl. dazu auch die kritische Würdigung der Invariantenanalyse.

12) Daß die Erreichbarkeitsgraphen von Netzen das prädikatenlogische Ausdrucksvermögen übersteigen, wurde schon angedeutet. Auch in dieser Hinsicht überragt die Erreichbarkeits- die Invariantenanalyse. Denn die Invariantenanalyse bleibt durch ihre Untersuchung von arithmetischen oder algebraischen Gleichungssystemen innerhalb der prädikatenlogischen Formulierungskraft.

- 13) Ein alternativer Ansatz, in dem prädikatenlogische Zustandsbeschreibungen und eine Erreichbarkeitsrelation auf das Ausdrucksvermögen einer dynamischen Logik zurückgeführt werden, findet sich bei GRABOWSKI, J. (1980c), S. 59ff.
- 14) Daß die modallogische Wissensrepräsentation auf der einen Seite mit der Kombination aus Prädikatenlogik und Erreichbarkeitsrelation auf der anderen Seite gleichwertig sind, haben MANNA (1979), S. 386ff., 398f. u. 405f.; GRABOWSKI, J. (1980c), S. 53ff., insbesondere S. 56, ausführlicher behandelt.
- 15) Vgl. zur modallogischen Semantik multipler oder möglicher Welten VAN EMDE BOAS (1978), S. 2ff.; MANNA (1979), S. 386ff.; BIBEL (1989), S. 55, 59 u. 61.
- 16) Vgl. SEICHE (1990), S. 17 (endliche Anzahl von Zuständen der Netzmodelle).
- 17) Die Endlichkeit der Kantenmenge folgt nicht notwendig aus der Endlichkeit der Knotenmenge *allein*. Da es sich bei Erreichbarkeitsgraphen um Multigraphen handelt, könnte in extremen Fällen die Kantenmenge trotz endlicher Knotenmenge unendlich sein. Dies wäre der Fall, wenn zwischen mindestens zwei Knoten, die eine Referenz- und eine Folgemarkierung des zugrundeliegenden Netzmodells wiedergeben, ein Kantenbündel aus unendlich vielen Schaltkanten verläuft. Daran könnte in Höheren Netzen gedacht werden, für deren strukturierte Marken ein unendlich großer Bereich verschiedener Markenkopien definiert ist. Daher wäre es möglich, daß der Übergang zwischen zwei benachbarten Markierungsknoten durch eine unendliche Anzahl unterschiedlicher Schaltfarben trotz einer endlichen Anzahl geschalteter Transitionen bewirkt wird. Die unendlich vielen Schaltfarben definierten dann unendlich viele Schaltschritte und somit auch unendlich viele Schaltkanten zwischen den betrachteten zwei Markierungsknoten. Allerdings erzeugen die Schaltakte von Transitionen, die unter derselben Referenzmarkierung aktiviert sind und mit unterschiedlichen Schaltfarben geschaltet werden, auch *verschiedene* Folgemarkierungen. Die Unterschiedlichkeit der Folgemarkierungen widerspricht aber der Voraussetzung, daß sich im Erreichbarkeitsgraphen unendlich viele Schaltkanten zwischen dem Knoten einer Referenzmarkierung und dem Knoten von *einer* Folgemarkierung erstrecken. Lediglich Transitionen, deren Schaltakte *keine* Veränderung der Netzmarkierung bewirken, könnten noch bei der Verwendung von unendlich vielen Schaltfarben zu nur genau einer Folgemarkierung führen. Diese Folgemarkierung würde mit der Referenzmarkierung zusammenfallen. Im Erreichbarkeitsgraphen besäße der Knoten der Referenzmarkierung eine unendliche Anzahl von Schaltkanten, die zum Knoten selbst zurückführen ("0-Schleifen"). Solche wirkungsfreien Transitionen wurden jedoch in dieser Arbeit ausgeklammert. Unter dieser *zusätzlichen* Voraussetzung kann ein Erreichbarkeitsgraph, dessen Knotenmenge endlich ist, immer nur eine endliche Kantenmenge besitzen. Davon wird im folgenden ausgegangen.
- 18) Diese Finitheitsprämisse dominiert auch in der Netzliteratur, soweit sie sich mit Erreichbarkeitsgraphen befaßt; vgl. z.B. ULLRICH (1976), S. 5/17; BERTHELOT (1977), S. 217. Besonders deutlich wird die Endlichkeit von Netzen und ihren Erreichbarkeitsgraphen bei der Modellierung von Produktionssystemen - darunter auch Flexiblen Fertigungssystemen - von SEICHE (1990), S. 17, hervorgehoben. Analoge Finitheitsprämissen werden auch bei anderen Modellierungskonzepten für dynamische Systeme zugrundegelegt; vgl. z.B. FIDELAK (1988b), S. 22 (dort wird eine "endliche Repräsentation aller möglichen Abläufe" gesucht); ABEL, D. (1990), S. 43 (für die "Modellierung technischer Systeme") u. 60.
- 19) Vgl. zu Überdeckbarkeitsgraphen (coverability graphs), die des öfteren auch als Überdeckungsgraphen angesprochen werden, STARKE (1980), S. 97ff.; PETERSON, J. (1981), S. 91ff.; REISIG (1985b), S. 66ff.; REISIG (1986a), S. 75ff.; REISIG (1987a), S. 129ff.; FINKEL (1987b), S. 499 u. 505ff.; ABEL, D. (1990), S. 77ff. Vgl. speziell zu algorithmischen Beschreibungen der Konstruktion von Überdeckbarkeitsgraphen JANTZEN (1980a), S. 171; REISIG (1985b), S. 67; REISIG (1986a), S. 76; REISIG (1987a), S. 131; ABEL, D. (1990), S. 79, mit einem sehr übersichtlichen Struktogramm für einen Konstruktionsalgorithmus.
- 20) Unendliche Erreichbarkeitsgraphen werden z.B. explizit angesprochen bei RAZOUK (1985c), S. 15; ZELEWSKI (1986c), S. 24ff.; FINKEL (1987b), S. 499 u. 502f.; ZELEWSKI (1988b), S. 358f. Anstatt unendlicher Erreichbarkeitsgraphen werden mitunter unendliche Erreichbarkeitsmengen thematisiert; vgl. z.B. ABEL, D. (1990), S. 10f. u. 15 (am Rande auch S. 43). Diese Nuancierung ist hier aber unerheblich. Denn die Erreichbarkeitsmenge eines Netzes ist die Knotenmenge seines Erreichbarkeitsgraphen. Folglich muß ein Netz mit unendlicher Erreichbarkeitsmenge auch einen unendlichen Erreichbarkeitsgraphen besitzen. Die Umkehrung gilt zwar nicht notwendig. Denn ein Erreichbarkeitsgraph kann wegen einer unendlichen Kantenmenge trotz endlicher Knotenmenge unendlich werden. Die wirkungsfreien Transitionen, die dafür erforderlich sind, wurden aber ausgeschlossen. Daher gilt in dieser Arbeit noch strenger: Ein Netz besitzt einen unendlichen Erreichbarkeitsgraphen genau dann, wenn seine Erreichbarkeitsmenge unendlich ist.
- 21) Das folgt aus den Konstruktionsbeschreibungen für Überdeckbarkeitsgraphen, die in den Quellen aus einer der voranstehenden Anmerkungen aufgeführt werden. Die dort beschriebenen Konstruktionen konzentrieren sich auf die Behandlung von zyklischen Schaltfolgen, die beliebig oft wiederholt werden können und dabei die Markenanzahl $M_i(s_m)$ mindestens einer Stelle s_m mit unbeschränkter Markenzahl $K(s_m) = KAP_m = \omega$ über jede obere Schranke hinauswachsen lassen. Dies führt notwendig zu einem unendlichen Erreichbarkeitsgraphen. Denn es müßten im

Erreichbarkeitsgraphen unendlich viele Markierungsknoten erzeugt werden, um die unendlich vielen verschiedenen Markenanzahlen $M_r(s_m)$ auf der Stelle s_m darstellen zu können. In Überdeckbarkeitsgraphen läßt sich dagegen diese Unendlichkeit des Erreichbarkeitsgraphen durch die nachfolgend skizzierte Konstruktion vermeiden.

Für jede Schaltfolge, welche die Markierung $M_r(s_m^*)$ von mindestens einer Stelle s_m^* mit unbeschränkter Markenzapazität $K(s_m^*) = \omega$ über jede vorgegebene Schranke anwachsen ließe, wenn sie hinreichend lange fortgesetzt würde, gilt: Die Schaltfolge ruft im Überdeckbarkeitsgraphen eine Markierung M_f hervor, die einer anderen, im Überdeckbarkeitsgraphen bereits enthaltenen Markierung M_r ähnlich ist. Die Ähnlichkeit erstreckt sich darauf, daß für das Verhältnis zwischen Markierung M_r und Markierung M_f gilt: $M_f(s_m) \geq M_r(s_m)$ für alle Stellen s_m mit $s_m \in S$ und $M_f(s_m^*) > M_r(s_m^*)$ für mindestens eine Stelle s_m^* mit $s_m^* \in S$. Dann kann auf den neuen Markierungsknoten M_f im Überdeckbarkeitsgraphen verzichtet werden. An seine Stelle tritt ein modifizierter Knoten M_f^{\cup} . Er geht aus der Markierung M_f wie folgt hervor: Einerseits werden alle Komponenten $M_f(s_m)$ der Markierung M_f unverändert übernommen, welche die Bedingung $M_f(s_m) = M_r(s_m)$ erfüllen. Andererseits wird jede Komponente $M_f(s_m^*)$, auf welche die Bedingung $M_f(s_m^*) > M_r(s_m^*)$ zutrifft, durch die neuartige Komponente " ω " ersetzt. Folglich gilt für den modifizierten Markierungsknoten M_f^{\cup} :

$$M_f(s_m) = M_r(s_m) \quad \Rightarrow \quad M_f^{\cup}(s_m) = M_f(s_m)$$

$$M_f(s_m^*) > M_r(s_m^*) \quad \Rightarrow \quad M_f^{\cup}(s_m^*) = \omega$$

Der modifizierte Knoten M_f^{\cup} repräsentiert eine Klasse $MK_{f^{\cup}}$ von potentiell unendlich vielen Netzmarkierungen. Die potentiell infinite Markierungsklasse $MK_{f^{\cup}}$ umfaßt alle Markierungen, die von Schaltprozessen hervorgebracht werden, die von der Markierung M_f ausgehen und die Markenanzahl auf der mindestens einen Stelle s_m^* über alle Schranken anwachsen lassen können. Der modifizierte Knoten M_f^{\cup} "überdeckt" jetzt alle erreichbaren Knoten M_r aus der Markierungsklasse $MK_{f^{\cup}}$, d.h. es gilt: $M_f^{\cup} \geq M_r$ für alle $M_r \in MK_{f^{\cup}}$.

Jeder Knoten M_f^{\cup} des Überdeckbarkeitsgraphen, der für mindestens eine Stelle s_m^* die Komponente $M_f^{\cup}(s_m^*) = \omega$ aufweist, stellt keine erreichbare Markierung mehr dar. Denn die Markenanzahl " ω " ist für keine Stelle eines Netzes definiert. Statt dessen repräsentiert der Knoten M_f^{\cup} nur noch eine "überdeckende" Markierung (ABEL, D. (1990), S. 78). Darüber hinaus ist nur bekannt, daß der Knoten M_f^{\cup} alle erreichbaren Knoten M_r aus der Markierungsklasse $MK_{f^{\cup}}$ überdeckt. Aber die Umkehrung braucht keineswegs zuzutreffen: Nicht jede Markierung, die vom Knoten M_f^{\cup} überdeckt wird, muß im zugrundeliegenden Netz auch erreichbar sein; vgl. ABEL, D. (1990), S. 78f. Die Spezialisierung der markierungsüberdeckenden Knoten besteht in einer Finitisierungsoperation: Sie bewirkt, daß im Überdeckbarkeitsgraphen an die Stelle einer jeden *infiniten* Markierungsklasse $MK_{f^{\cup}}$ genau ein Knoten M_f^{\cup} tritt. Dieser Knoten M_f^{\cup} gleicht dem bereits konstruierten Knoten für die Referenzmarkierung M_r bis auf den einen Unterschied, daß der Knoten M_f^{\cup} für mindestens eine Stelle s_m^* die charakteristische Komponente $M_f^{\cup}(s_m^*) = \omega$ aufweist. Es läßt sich beweisen, daß die vorgenannte Finitisierungsoperation die Endlichkeit des resultierenden Überdeckbarkeitsgraphen garantiert. Formal strenge Beweise der Endlichkeit von Überdeckbarkeitsgraphen finden sich bei REISIG (1985b), S. 70f.; REISIG (1986a), S. 81 i.V.m. S. 79ff.; REISIG (1987a), S. 132f.

Allerdings gilt die garantierte Endlichkeit des Überdeckbarkeitsgraphen nur in bezug auf Stellen mit unbeschränkt anwachsenden Markenanzahlen. Sie trifft jedoch nicht auf jene "pathologischen" Fälle zu, in denen ein *beschränktes* Netz dennoch unendlich viele verschiedene Markierungen besitzt. Darauf wird in Kürze anhand eines "Zählernetzes" näher eingegangen. Lediglich bei REISIG (1985b), S. 70, findet sich eine korrekte Einschränkung seines Endlichkeitsbeweises auf Stelle/Transition-Netze.

22) Dies folgt aus der Konstruktion, die in der voranstehenden Anmerkung skizziert wurde. Überdeckbarkeitsgraphen müssen jedoch nicht immer unbeschränkte Markierungsknoten umfassen. Insbesondere besitzen die Überdeckbarkeitsgraphen von allen Netzen mit endlichen Erreichbarkeitsgraphen keine unbeschränkten Markierungsknoten. Denn endliche Erreichbarkeitsgraphen fallen mit Überdeckbarkeitsgraphen zusammen. Daneben werden in Kürze noch Überdeckbarkeitsgraphen vorgestellt, die zwar zu Netzen mit unendlichen Erreichbarkeitsgraphen gehören, aber dennoch keine unbeschränkten Markierungsknoten aufweisen.

23) Vgl. PETERSON, J. (1981), S. 103; ZELEWSKI (1986c), S. 23; ZELEWSKI (1988b), S. 358; ABEL, D. (1990), S. 79, 81 u. 83f.

Ein anschauliches Beispiel für diesen Informationsverlust legt PETERSON, J. (1981), S. 104ff., vor: Er konstruiert zwei Netze, die sich deutlich voneinander unterscheiden. Sie besitzen zwar die gleichen Stellen- und Transitionenanzahlen, aber verschiedenartige Flußrelationen. Dies führt dazu, daß das eine Netz in einen Deadlock gerät, während sich das andere Netz als (transitionsschwach) lebendig erweist. *Dennoch* besitzen die beiden so verschiedenen Netze denselben Überdeckbarkeitsgraphen. Folglich ist in ihrem gemeinsamen Überdeckbarkeitsgraphen diejenige Information über die dynamische Netzstruktur verloren gegangen, die ausgereicht hätte, um die Nichtlebendigkeit des einen von der Lebendigkeit des anderen Netzes abzuheben.

24) Ein Beispiel dafür wurde bereits in der voranstehenden Anmerkung angesprochen. Dort wurde mit Bezug auf PETERSON, J. (1981), S. 104ff., gezeigt, daß es für mindestens ein Netz unmöglich ist, anhand seines Überdeckbarkeitsgraphen die (transitionsschwache) Netzlebendigkeit zuverlässig abzuleiten. Also ist die transitionsschwache

Netzlebendigkeit bei der Verwendung von Überdeckbarkeitsgraphen nicht mehr entscheidbar. Dies bestätigt ein weiteres Beispiel von ABEL, D. (1990), S. 83f. (für den Fall der Abb. 5.14 auf S. 83). Es zeigt, daß eine tatsächlich vorhandene Deadlockmarkierung anhand des Überdeckbarkeitsgraphen für das zugrundeliegende Netz grundsätzlich nicht erkannt werden kann. Folglich ist die Netzeigenschaft der Deadlockfreiheit unentscheidbar. Es wurde bereits dargelegt, daß es sich bei Deadlockfreiheit und transitionsschwacher Lebendigkeit um äquivalente Netzeigenschaften handelt. Folglich hat ABEL ebenso nachgewiesen, daß die transitionsschwache Netzlebendigkeit unentscheidbar ist. Die transitionsschwache Lebendigkeit eines Netzes stellt ihrerseits für die starke Netzlebendigkeit eine notwendige Voraussetzung dar. Daher ist a fortiori auch das Problem der starken Netzlebendigkeit bei der Verwendung von Überdeckbarkeitsgraphen unentscheidbar.

Der Erkenntnis der Unentscheidbarkeit der (transitionsschwachen) Netzlebendigkeit scheint eine Feststellung von STARKE (1980), S. 123, zu widersprechen: Ihr zufolge verhält sich ein Netz genau dann (transitionsschwach) lebendig, wenn sein Überdeckbarkeitsgraph keinen Markierungsknoten ohne Ausgangskanten besitzt. Da der Überdeckbarkeitsgraph per constructionem endlich ist, müßte sich auch für jedes Netz - insbesondere auch für ein Netz mit unendlichem Erreichbarkeitsgraphen - entscheiden lassen, ob es mindestens einen Markierungsknoten ohne Ausgangskanten aufweist. Folglich wäre die (transitionsschwache) Lebendigkeit eines Netzes eine entscheidbare Eigenschaft. Dies läßt sich jedoch prima facie mit dem o.a. Beispiel von PETERSON, J. (1981), S. 104ff., überhaupt nicht vereinbaren.

Der scheinbare Widerspruch zwischen STARKE und PETERSON kann jedoch zugunsten der Unentscheidbarkeit des (transitionsschwachen) Lebendigkeitsproblems aufgelöst werden. Denn eine nähere Kontextanalyse der Entscheidbarkeitsbehauptung von STARKE (1980), S. 123, zeigt, daß er sich ausschließlich auf "gewöhnliche" Netze bezieht (S. 117), in denen alle Kanten (k_n, k_y) das Einheitsgewicht $W(k_n, k_y) = 1$ tragen (S. 56). Für diese einheitsgewichteten Netze trifft STARKE's Entscheidbarkeitsbehauptung für das Lebendigkeitsproblem tatsächlich zu. Dies folgt aus der Entscheidbarkeit des Erreichbarkeitsproblems für einheitsgewichtete Netze. PETERSON, J. (1981), S. 104ff., konstruiert dagegen ein Beispielnetz (Fig. 4.23 auf S. 105), das sich durch zwei Eigenschaften auszeichnet: Erstens besitzt es einen Überdeckbarkeitsgraphen (Fig. 4.25 auf S. 106), das - in der Diktion STARKE's - keinen Markierungsknoten ohne Ausgangskante besitzt. (Zwar weist der Überdeckbarkeitsgraph bei PETERSON solche Markierungsknoten ohne Ausgangskanten auf. Sie müßten jedoch durch jeweils eine Ausgangskante vom Typ des "Schleifenrückstellers" ergänzt werden, damit PETERSON'S Überdeckbarkeitsgraph der leicht abweichenden Konstruktionsvorschrift für Überdeckbarkeitsgraphen bei STARKE (1980), S. 98f., entspricht.) Zweitens handelt es sich bei PETERSON's Beispielnetz um kein einheitsgewichtetes Netz. Denn es enthält zwei Kanten, die jeweils das Gewicht $W(k_n, k_y) = 2$ tragen. Daher läßt sich STARKE's Erkenntnis, das Lebendigkeitsproblem sei für die Teilklasse einheitsgewichteter Netze anhand ihrer Überdeckbarkeitsgraphen entscheidbar, durchaus konsistent vereinbaren mit PETERSON's Gegenbeispiel eines Netzes ohne Einheitsgewichtung, dessen Lebendigkeit sich anhand seines Überdeckbarkeitsgraphen nicht mehr feststellen läßt. Für den allgemeinen Fall beliebiger Netze, deren Kantengewichtung keiner Einschränkung auf das Einheitsgewicht unterliegt, gilt daher: Es ist unmöglich, ihre (transitionsschwache) Lebendigkeit mit der Hilfe ihrer Überdeckbarkeitsgraphen zu entscheiden; q.e.d.

Für einen weiteren Spezialfall hat FINKEL (1987b), S. 508, bewiesen, daß die markierungsschwache Netzlebendigkeit entscheidbar ist: Es handelt sich um Petrinetze, die als wohlstrukturierte Transitionssysteme dargestellt werden können. Vgl. zur vorausgesetzten Terminologie FINKEL (1987b), S. 500 (Transitionssystem), S. 506 (Wohlstrukturiertheit eines Transitionssystems) und S. 500 (Quasi-Lebendigkeit eines Transitionssystems im Sinne der markierungsschwachen Netzlebendigkeit).

25) Vgl. zum eingeschränkten Erkenntnispotential von Überdeckbarkeitsgraphen auch ABEL, D. (1990), S. 84.

26) Diese Möglichkeit übersieht z.B. ABEL, D. (1990), S. 43. Er glaubt, ein Netzmodell könne nur dann eine unendliche Menge von zulässigen Zuständen annehmen, wenn es *unbeschränkt* ist. Tatsächlich zeigt aber das nachfolgende Beispiel, daß sogar ein 1-beschränktes Netz ein unendliches Schaltverhalten besitzen darf. Dabei vermag es unendlich viele zulässige Netzzustände hervorzubringen.

27) Die Netzlegende für dieses Zählernetz nimmt folgende Gestalt an:

Netzlegende:

Marken/Operationssymbole:

wert: INTEGER

zähler = Zählermarke(wert)

Stellen/Prädikatssymbole:

s_1 : Zählerstand(zähler)
markenkapazität $_1 = 1$

Transitionen/Transaktionen:

t_1 : inkrement
 $\langle \text{zählermarke}(\text{Wert_alt}) \rangle \approx \text{zählerstand}(\text{zählermarke}(\text{Wert_alt}))$
 $\text{Wert_neu} := \text{Wert_alt} + 1$
 $\langle \text{zählermarke}(\text{Wert_neu}) \rangle \approx \text{zählerstand}(\text{zählermarke}(\text{Wert_neu}))$

Fakten:

$\langle \text{zählermarke}(0) \rangle \approx \text{fakt}_0(1, \text{zählerstand}(\text{zählermarke}(0)))$

28) Folglich irrt ABEL, D. (1990), S. 81, mit seinem kategorischen Urteil: "Ein Überdeckungsgraph U_N ist für jedes Netz N endlich." (kursive Hervorhebung durch den Verf.). Ebenso trifft der "Satz" von REISIG (1986a), S. 81, nicht zu: "Jeder Überdeckungsgraph eines endlichen Netzes ist endlich." Zwar schränkt REISIG sein Urteil auf endliche Netze ein. Aber alle Netze, die in dieser Arbeit behandelt werden, sind ohnehin von vornherein als endliche Netze definiert. Auch das oben vorgestellte Zählernetz stellt mit nur zwei Knoten und nur zwei Kanten ein endliches Netz dar. Dennoch besitzt dieses kleine, endliche "Spielzeugnetz" einen *unendlichen* Überdeckungsgraphen. Dagegen läßt sich das vorsichtiger Urteil von REISIG (1985b), S. 70, erfolgreich verteidigen: "Every coverability graph of a P/T-net ist finite." Denn die Einschränkung auf Stelle/Transition-Netze ("P/T-nets") bedeutet, daß die hier identifizierte Infinitheitsursache von vornherein ausgeschlossen wird: Unendliche Definitionsbereiche für die Attributausprägungen von Attributmarken aus Höheren Netzen können in Stelle/Transition-Netzen grundsätzlich nicht vorkommen.

Es könnte der Einwand erhoben werden, daß der Überdeckbarkeitsgraph für das Zählernetz endlich sei, weil sich auf seiner einen Stelle s_1 immer nur eine Kopie der Zählermarke befindet. Diese Vorhaltung übersieht aber, daß die Markierung einer Stelle in Höheren Netzen keineswegs von der *Anzahl* der Markenkopien vollständig determiniert wird. Vielmehr hängt die Markierung ebenso von der *Ausprägung* der Markenkopien ab. Gerade darin liegt der Endlichkeitsdefekt von Überdeckbarkeitsgraphen begründet. Denn Konstruktionsalgorithmen für Überdeckbarkeitsgraphen beziehen sich ausschließlich auf *Anzahlen* von Markenkopien. Dies läßt sich anhand des Konstruktionsalgorithmus aus dem Struktogramm von ABEL, D. (1990), S. 79, Abb. 5.11, verdeutlichen:

- Die Knotenmenge $KN_{0,0}$ des Überdeckbarkeitsgraphen umfaßt - in der Kurznotation aus Abb. 65 - zunächst nur die Ausgangsmarkierung $M_0 = \langle \text{zählermarke}(0) \rangle$ des Zählernetzes, unter der sich auf der Stelle s_1 die Markenkopie " $\langle \text{zählermarke}(0) \rangle$ " befindet: $KN_{0,0} = \{M_0\}$.
- Unter der Ausgangsmarkierung M_0 ist die Transition t_1 aktiviert, deren Schalttakt die Folgemarkierung $M_f = M_1 = \langle \text{zählermarke}(1) \rangle$ hervorbringt. Diese Folgemarkierung stimmt - trotz gleicher Anzahl von Markenkopien - mit der Ausgangsmarkierung M_0 nicht überein: $M_1 \neq M_0$. Die Folgemarkierung M_1 ist also noch nicht als Knoten in der Knotenmenge des Überdeckbarkeitsgraphen vorhanden. Daher wird für den Knoten M_1^0 , der in den Überdeckbarkeitsgraphen neu aufzunehmen ist, nach Maßgabe des oben erwähnten Struktogramms folgende Fallunterscheidung getroffen:
 - ➔ $M_1^0(s_1) = \omega$; falls der Markierung M_1 eine frühere erreichbare Markierung M_r vorangeht, für die gilt: $\#(M_1(s_m)) \geq \#(M_r(s_m))$ für alle $s_m \in S$ und $\#(M_1(s_1)) > \#(M_r(s_1))$. Die einzige erreichbare Markierung, die der betrachteten Markierung M_1 vorangeht, ist die Ausgangsmarkierung M_0 . Für die beiden Markierungen M_1 und M_0 trifft zwar die erste Bedingung $\#(M_1(s_m)) \geq \#(M_0(s_m))$ für alle $s_m \in S$ wegen $S = \{s_1\}$ und $\#(M_1(s_1)) = \#(M_0(s_1)) = 1$ zu. Aber die zweite Bedingung $\#(M_1(s_1)) > \#(M_0(s_1))$ wird wegen $\#(M_1(s_1)) = \#(M_0(s_1)) = 1$ verletzt.
 - ➔ $M_1^0(s_1) = M_1(s_1)$; andernfalls. Da bereits gezeigt wurde, daß der erste Fall nicht zutrifft, gilt per definitionem dieser zweite Fall.
- Das Zählernetz besteht nur aus der einen Stelle s_1 . Daher liegt der neu aufzunehmende Knoten M_1^0 für den Überdeckbarkeitsgraphen bereits fest: $M_1^0 = (M_1^0(s_1)) = (M_1(s_1)) = M_1$. Folglich lautet die erstmals erweiterte Knotenmenge des Überdeckbarkeitsgraphen: $KN_{0,1} = \{M_0\} \cup \{M_1\} = \{M_0, M_1\}$.
- Eine Kante, die mit dem Schaltschritt $\{t_1\}$ beschriftet ist, wird im Überdeckbarkeitsgraphen vom Knoten M_0 zum Knoten M_1 gerichtet.

□ Unter der Markierung M_1 ist die Transition t_1 abermals aktiviert. Ihr Schaltakt bringt die Folgemarkierung $M_f = M_2 = (\langle \text{zählermarke}(2) \rangle)$ hervor. Diese Folgemarkierung stimmt - trotz gleicher Anzahl von Markenkopien - mit den beiden bereits vorhandenen Knoten in der Knotenmenge des Überdeckbarkeitsgraphen nicht überein: $M_2 \notin \{M_0, M_1\}$. Daher wird für den Knoten M_2^0 , der in den Überdeckbarkeitsgraphen neu aufzunehmen ist, erneut die oben erläuterte Fallunterscheidung vorgenommen. Daraus folgt für den neuen Knoten M_2^0 des Überdeckbarkeitsgraphen: $M_2^0 = (M_2^0(s_1)) = (M_2(s_1)) = M_2$. Folglich lautet die zum zweiten Mal erweiterte Knotenmenge des Überdeckbarkeitsgraphen: $KN_{0,2} = \{M_0, M_1, M_2\}$.

Offensichtlich läßt sich die Knotenmenge des Überdeckbarkeitsgraphen unendlich oft erweitern. Nach einer beliebigen k -ten Erweiterung mit $k \in \mathcal{N}_+$ nimmt die Knotenmenge des Überdeckbarkeitsgraphen die Gestalt an: $KN_{0,k} = \{M_0, \dots, M_k\}$. Mit $k \rightarrow \infty$ wächst die Knotenmenge - und somit auch der Überdeckbarkeitsgraph selbst - über alle Grenzen an (q.e.d.).

29) Dies folgt aus demjenigen Konstruktionsschritt von Überdeckbarkeitsgraphen, in dem einer Stelle s_m in einem unbeschränkten Markierungsknoten erstmals der Markierungswert $M(s_m) = \omega$ zugewiesen wird. Denn diese Zuweisung geschieht nur dann, wenn die Markenanzahl $M_f(s_m)$ auf dieser Stelle gegenüber einer Referenzmarkierung M_r echt angestiegen ist, d.h. wenn $M_f(s_m) > M_r(s_m)$ gilt. Vgl. dazu beispielsweise STARKE (1980), S. 99 (Fall " $E_r \neq \emptyset$ "). Ein solcher Anstieg der Markenanzahl braucht aber bei variierenden Attributausprägungen für die Kopie einer Attributmarke überhaupt nicht zu erfolgen.

30) Solche unendlichen Definitionsbereiche fallen für Synthetische Netze oftmals an. Beispielsweise wird auf die Objektmengen "INTEGER", "STRING" und "SYMBOL" verwiesen. Sie dienen häufig dazu, um die Bereiche zulässiger Attributausprägungen zu definieren; vgl. etwa das Beispielnetz einer Autovermietung, das an früherer Stelle präsentiert wurde.

31) Dadurch werden auch Schwierigkeiten derjenigen Lebendigkeitskriterien vermieden, die für die Untersuchung von Erreichbarkeitsgraphen vorgestellt werden. Die Anwendung dieser Kriterien kann zu fehlerhaften Analyseergebnissen führen, falls sie auf unbeschränkte Netze bezogen werden. Ein Beispiel dafür findet sich bei BAUMGARTEN (1978), S. 37f.

32) In beschränkten Netzen ist die Markenanzahl auf jeder Stelle nach oben beschränkt. Falls sich in einem Netz zwei Markierungen nur dadurch unterscheiden können, daß ihre Markenanzahlen auf mindestens einer Stelle voneinander abweichen, dann kann sein Erreichbarkeitsgraph nur endlich viele verschiedene Markierungsknoten besitzen. Da in einem solchen Netz für das Schalten von Transitionen keine Schaltfarben definiert sind, können auch nur endlich viele Schaltkanten je zwei Markierungsknoten miteinander verknüpfen. Beides wurde für Stelle/Transition-Netze aufgezeigt. Folglich kann der Erreichbarkeitsgraph eines Netzes, in dem es nur auf Markenanzahlen, nicht aber auf die Ausprägungen von Markenattributen ankommt, immer nur endlich viele Markierungsknoten und Schaltkanten besitzen. Daher sind Erreichbarkeitsgraphen aus dieser Perspektive für beschränkte Netze endlich.

33) Da diese Erfahrung nur auf einer beschränkten Anzahl von Netzmodellen beruht, kann sie nur ein vorläufiges Tendenzurteil abstützen.

34) Diese Vorgehensweise empfiehlt ABEL,D. (1990), S. 60f., insbesondere anhand seines Ablaufdiagramms in Abb. 5.1 auf S. 61.

35) Vgl. ABEL,D. (1990), S. 82.

36) Von den beiden folgenden Alternativen sieht ABEL,D. (1990), in seinem Ablaufdiagramm der Abb. 5.1 auf S. 61, nur die zweite vor.

37) Dieses Ergebnisrisiko ist im Ablaufdiagramm der Abb. 5.1 bei ABEL,D. (1990), S. 61, als Endlosschleife enthalten. Aus den Ausführungen ABEL's ist aber nicht ersichtlich, ob er ein Verfangen in dieser Endlosschleife in Betracht gezogen hat.

38) Da die Konstruktion eines Überdeckbarkeitsgraphen unter Umständen nicht terminiert, irrt sich ABEL,D. (1990), S. 84, mit seiner Feststellung, daß "ein ... Vorteil des Überdeckungsgraphen darin besteht, ... für jedes ... Petri-Netz *effektiv konstruierbar* zu sein." (kursive Hervorhebungen durch den Verf.).

Die Möglichkeit unendlicher Überdeckbarkeitsgraphen wird in der Netzliteratur nach Wissen des Verf. nicht zur Kenntnis genommen. Vgl. dazu den Hinweis auf das fehlerhafte kategorische Urteil, der Überdeckbarkeitsgraph jedes Netzes sei endlich. Die Denkmöglichkeit, daß die Konstruktion eines Überdeckbarkeitsgraphen nicht terminiert, fehlt auch in dem Ablaufdiagramm von ABEL, das schon mehrfach angeführt wurde (ABEL,D. (1990), S. 61, Abb. 5.1).

39) Die dynamische Struktur des oben präsentierten "pathologischen" Zählernetzes war derart simpel, daß sich das erforderliche Zusatzwissen als evident erwies. Die Unendlichkeit seines Überdeckbarkeitsgraphen konnte bereits durch unmittelbare Anschauung aus einem kleinen Ausschnitt seines Erreichbarkeits- und Überdeckbarkeitsgraphen intuitiv erkannt werden.

40) Daher wird auch darauf verzichtet, die Konstruktion von Überdeckbarkeitsgraphen hier im Detail zu erläutern. Entsprechende Anregungen finden sich aber in den Quellen, die bereits in einer früheren Anmerkung zu Überdeckbarkeitsgraphen angeführt wurden.

41) Vgl. zur Entscheidbarkeit der Lebendigkeit von Netzen, die endliche Erreichbarkeitsgraphen besitzen oder beschränkt sind (aus letztem folgt erstes), AZEMA (1976b), S. 111; VALETTE (1977b), S. 105; PRIESE (1979), S. 13.

Die Entscheidbarkeit der Netzlebendigkeit für endliche Erreichbarkeitsgraphen läßt sich intuitiv aus den graphenbezogenen Lebendigkeitskriterien ablesen, die an früherer Stelle angegeben wurden. Es stehen aber auch konkret ausformulierte Algorithmen zur Verfügung, mit deren Hilfe sich die Lebendigkeit aller Netze mit endlichen Erreichbarkeitsgraphen entscheiden läßt. Einen solchen Entscheidungsalgorithmus präsentiert z.B. BERTHELOT (1977), S. 219.

42) Vgl. zur Vakanz dieses Algorithmus' BERTHELOT (1977), S. 217; GRABOWSKI, J. (1980c), S. 19.

43) Die Frage nach der Entscheidbarkeit des Lebendigkeitsproblems für allgemeine, in keiner Weise eingeschränkte Netze wird z.B. für ungelöst gehalten von LÖHR (1974), S. 91; ULLRICH (1976), S. 3/12; BERTHELOT (1977), S. 217. Es fällt allerdings auf, daß sich die Ansicht, das Lebendigkeitsproblem sei ungelöst, seit den achtziger Jahren nicht mehr belegen läßt. Dieses Phänomen könnte darauf zurückgeführt werden, daß das Lebendigkeits- mit dem Erreichbarkeitsproblem äquivalent ist und letztes inzwischen gelöst wurde. Eine solche Argumentation führte jedoch in eine prekäre Argumentationslage, daß das Erreichbarkeitsproblem einerseits als entscheidbar bewiesen wurde, während die Unentscheidbarkeit des Lebendigkeitsproblems naheliegt. Dieser prima facie widersprüchliche Sachverhalt bedarf einer näheren Erläuterung.

Das Erreichbarkeitsproblem des Petrinetz-Konzepts besteht im wesentlichen darin, für jedes Netz und für jede vorgegebene Referenzmarkierung M_r zu entscheiden, ob sich diese Markierung durch eine zulässige Schaltfolge von der Ausgangsmarkierung des Netzes aus erreichen läßt. Dieses Erreichbarkeitsproblem hat der Verf. an anderer Stelle ausführlicher behandelt; vgl. ZELEWSKI (1989a), S. 111 u. 120ff., und die dort genannte weiterführende Literatur.

Es konnte schon früh nachgewiesen werden, daß sich Lebendigkeits- und Erreichbarkeitsproblem rekursiv äquivalent verhalten, weil sie sich wechselseitig aufeinander reduzieren lassen. Vgl. zu diesem Reduktionszusammenhang HACK, M. (1974b), S. 156 u. 159ff.; HACK, M. (1975a), S. 85ff.; PETERSON, J. (1977), S. 240; CRESPI-REGHIZZI (1977), S. 180; LANDWEBER (1978), S. 352; STARKE (1980), S. 117; JANTZEN (1980a), S. 175f. u. 178; PETERSON, J. (1981), S. 129ff.; SANDRING (1982), S. 218f. Die rekursive Äquivalenz der beiden Entscheidungsprobleme bedeutet, daß das Lebendigkeitsproblem genau dann lösbar (entscheidbar/unentscheidbar) sein muß, wenn auch das Erreichbarkeitsproblem lösbar (entscheidbar/unentscheidbar) ist; vgl. ZELEWSKI (1989a), S. 63f.

Mittlerweile konnte das Erreichbarkeitsproblem als hochkomplex, aber immerhin entscheidbar nachgewiesen werden; vgl. SACERDOTE (1977), S. 62ff. (mit einer Beweisführung, die allerdings bis heute keine Akzeptanz fand); MAYR, E. (1980a), S. 51 i.V.m. S. 39ff.; MAYR, E. (1981b), S. 245 i.V.m. S. 243f.; KOSARAJU (1982), S. 281 i.V.m. S. 270ff.; MÜLLER, H. (1982c), S. 5ff.; MÜLLER, H. (1983b), S. 4; MÜLLER, H. (1983c), S. 177ff.; MÜLLER, H. (1983e); MAYR, E. (1984), S. 454 i.V.m. S. 452ff. u. 454ff.; MÜLLER, H. (1985a), S. 378, 382ff. u. 388ff.; LAMBERT, J. (1987), S. 451ff., insbesondere S. 457ff.; BUDINAS (1989), S. 1393ff., insbesondere S. 1410; ZELEWSKI (1989a), S. 123ff., insbesondere S. 124f. (und die dort angeführten Quellen). Wegen des Reduktionszusammenhangs zwischen Lebendigkeits- und Erreichbarkeitsproblem führte die Erkenntnis der Entscheidbarkeit des Erreichbarkeitsproblems zu der Einsicht, daß auch das Lebendigkeitsproblem entscheidbar ist. Vgl. zur derart motivierten Entscheidbarkeit des Lebendigkeitsproblems für Netze VALK (1977a), S. 325ff. (bei unbeschrifteten Transitionen); PETERSON, J. (1977), S. 242 (allerdings mit problematischer Bezugnahme auf einen unvollständigen Entscheidbarkeitsbeweis für das Erreichbarkeitsproblem).

Allerdings zeigen nähere Betrachtungen der Entscheidbarkeitsbeweise für das Erreichbarkeitsproblem von Netzen, daß diese nicht für beliebige Netze geführt worden sind. Statt dessen beziehen sie sich nur auf Netze, die einschneidenden Einschränkungen unterliegen. So wird beispielsweise bei BUDINAS (1989), S. 1393, vorausgesetzt, daß sich die Netze als Vektor-Ersatz-Systeme darstellen lassen. Aus der Art, wie in solchen Vektorersatzsystemen die Zulässigkeit von Zustandsübergängen definiert ist, folgt: Die übergangsbewirkende Schaltregel darf im dargestellten Netz weder auf beschränkte Markenkapazitäten Bezug nehmen noch die Aktivierungsbesonderheiten von 1-Schleifen berücksichtigen. Vgl. zu Definition und Eigenarten von Vektor-Ersatz-Systemen (vector replacement systems), die hier nicht näher behandelt werden, BUDINAS (1989), S. 1393. Mitunter kommt es auch vor, daß für die Kanten der betrachteten Netze das Einheitsgewicht $W(k_n, k_y) = 1$ unterstellt wird. (Die Netze können auch so formuliert sein, daß sie überhaupt keine Gewichtungsfunktion besitzen. Diese ungewichteten Netze entsprechen aber genau Netzen mit Gewichtungsfunktionen, die alle Netzkanten auf das Einheitsgewicht abbilden.) Diese Einschränkung auf einheitsgewichtete Netze wird z.B. bei LAMBERT, J. (1987), S. 452 (ungewichtetes Netz), deutlich. Folglich wurde die Entscheidbarkeit des Erreichbarkeitsproblems *nur für erheblich eingeschränkte Netze* bewiesen. Für Netze, die keinen Restriktionen unterliegen, treffen die Entscheidbarkeitsnachweise dagegen nicht zu. Die inhaltliche Nähe des Erreichbarkeitsproblems für uneingeschränkte Netze zum Halteproblem der TURING-Automaten legt es sogar nahe,

das Erreichbarkeitsproblem im allgemeinen für unentscheidbar anzusehen; vgl. MÜLLER, H. (1985a), S. 376. Denn die Unentscheidbarkeit des Halteproblems ist seit längerem bekannt. Dies wurde in dieser Arbeit bereits dargelegt. Da die Entscheidbarkeit des Erreichbarkeitsproblems bisher nur für eingeschränkte Netze bewiesen werden konnte, lassen sich mit dieser Entscheidbarkeitserkenntnis derzeit sowohl die Ungelöstheit als auch die Unentscheidbarkeit des Lebendigkeitsproblems für uneingeschränkte Netze vereinbaren.

Vgl. zu weiteren Erörterungen der Entscheidbarkeit des Lebendigkeitsproblems für Netze KELLER, R. (1972a), S. 43; HACK, M. (1974b), S. 156ff.; HACK, M. (1975a), S. 25f., 83 u. 91; JONES, N. (1977), S. 279; LANDWEBER (1978), S. 353; HEINEMANN (1980), S. 57ff. (mit einer präzisen formalsprachlichen Problemformulierung); PETERSON, J. (1981), S. 128f.

44) Vgl. zur (vermuteten) Unentscheidbarkeit des Lebendigkeitsproblems für beliebige Netze HACK, M. (1974b), S. 156; BYRN (1974), S. II-62 u. II-85; AZEMA (1976b), S. 111; KERAMIDIS (1979), S. 5.

Für die Vermutung der Unentscheidbarkeit des Lebendigkeitsproblems sprechen im wesentlichen zwei Sachverhalte. Erstens wurde schon angesprochen, daß sich die Eigenschaft der (transitionsschwachen) Lebendigkeit - und a fortiori auch der markierungsschwachen oder starken Lebendigkeit - anhand des Überdeckbarkeitsgraphen eines Netzes nicht sicher erkennen läßt. Überdeckbarkeitsgraphen stellen jedoch zur Zeit das allgemeinste Analysekonzept dar, das auf Netze überhaupt angewendet werden kann. Es lassen sich keine Anzeichen für noch leistungsfähigere Untersuchungskonzepte ausmachen, mit deren Hilfe das Lebendigkeitsproblem doch noch entschieden werden könnte. Zweitens beruhen die Beweise der Entscheidbarkeit des Erreichbarkeitsproblems für einheitsgewichtete Netze auf dem Konzept der Überdeckbarkeitsgraphen. Dies wird vor allem bei der ausführlichen und transparenten Beweisrekonstruktion von LAMBERT, J. (1987), S. 452ff., deutlich. Der Entscheidbarkeitsbeweis für das Erreichbarkeitsproblem von einheitsgewichteten Netzen stützt sich also genau auf jene Überdeckbarkeitsgraphen, mit deren Hilfe sich das Lebendigkeitsproblem für Netze ohne Einschränkung ihrer Gewichtungsfunktion nicht entscheiden läßt. Dies legt den Verdacht nahe, das Erreichbarkeitsproblem könne unentscheidbar werden, sobald keine einheitsgewichteten Netze mehr vorausgesetzt werden. Sollte dies tatsächlich zutreffen, so würde aus dem Reduktionszusammenhang beider Entscheidungsprobleme zwangsläufig folgen, daß auch das Lebendigkeitsproblem für Netze im allgemeinen unentscheidbar ist. Darüber hinaus ist die *Unentscheidbarkeit* des Lebendigkeitsproblems bereits für einige Netzklassen bewiesen, die das Ausdrucksvermögen von Stelle/Transition-Netzen und von Synthetischen Netzen übersteigen; vgl. HEINEMANN (1980), S. 65ff. (Unentscheidbarkeit in Selbstmodifizierenden Netzen); VIDAL-NAQUET (1982b), S. 8 (Unentscheidbarkeit bei Nulltestkanten) u. S. 13 (Unentscheidbarkeit bei unendlichen Mengen zulässiger Schaltfarben, d.h. Variablenbelegungen). Allerdings kann gegen diese Unentscheidbarkeitsresultate für das Lebendigkeitsproblem eingewandt werden, sie würden durch den besonders großen Ausdrucksreichtum jener Netzklassen verursacht. (Vgl. dazu beispielsweise die Ausführungen zur TURING-Mächtigkeit der oben erwähnten Netze mit Nulltestkanten). Daher ließen sich diese Unentscheidbarkeitsresultate nicht auf das Lebendigkeitsproblem für Stelle/Transition-Netze oder Synthetische Netze übertragen. Diese Vorbehalte erkennt der Verf. an. Deshalb stuft er das Lebendigkeitsproblem für die beiden letztgenannten Netzklassen auch nicht als unentscheidbar, sondern nur als ungelöst ein.

45) Allerdings vermag die Endlichkeitsprämisse für Erreichbarkeitsgraphen nicht zu garantieren, daß sich Erreichbarkeitsanalysen in der Praxis immer erfolgreich ausführen lassen. Denn jede praktische Erreichbarkeitsanalyse unterliegt einem Budget maximal einsetzbarer Analyseressourcen. Es kann niemals ausgeschlossen werden, daß ein Erreichbarkeitsgraph zwar endlich ist, aber immer noch so umfangreich ausfällt, daß er sich mit den tatsächlich vorhandenen Analyseressourcen nicht so weit untersuchen läßt, wie es für die Überprüfung einer interessierenden Netzeigenschaft erforderlich wäre. In einem solchen Fall muß die Erreichbarkeitsanalyse erfolglos abgebrochen werden, obwohl die Endlichkeitsprämisse erfüllt wird.

Diese Gefahr droht vor allem dann, wenn Netzeigenschaften untersucht werden, die grundsätzlich erst entschieden werden können, wenn der *gesamte* Erreichbarkeitsgraph vorliegt. Dazu gehört vor allem die Frage, ob sich ein Netzmodell sw^* -optimal verhält. Es existieren aber auch andere Netzeigenschaften, die nur in ungünstigen Fällen die vollständige Untersuchung eines Erreichbarkeitsgraphen erfordern, während sie in günstigen Fällen bereits nach einer Teiluntersuchung des Erreichbarkeitsgraphen entschieden sind. Ein Beispiel hierfür liefert die Eigenschaft der transitionsschwachen Netzlebendigkeit. Sie ist nur dann erfüllt, wenn *alle* Markierungsknoten eines Erreichbarkeitsgraphen keine Deadlock-Markierungen darstellen. Im ungünstigen Fall ist das analysierte Netz tatsächlich transitionsschwach lebendig. Dies läßt sich erst dann feststellen, wenn tatsächlich alle Knoten seines Erreichbarkeitsgraphen untersucht worden sind. Ein günstiger Fall liegt dagegen vor, wenn das Netz nicht transitionsschwach lebendig ist und wenn mindestens eine seiner Deadlock-Markierungen mit den vorhandenen Analyseressourcen tatsächlich aufgefunden wird.

Aufgrund der voranstehend skizzierten Schwierigkeiten kann die praktische Ausführung von Erreichbarkeitsanalysen unbefriedigend ausfallen, obwohl die Prämisse endlicher Erreichbarkeitsgraphen erfüllt wird. Diese Finitheitsprämisse ist daher nur eine notwendige, aber keine hinreichende Bedingung für den praktischen Erfolg von Erreichbarkeitsanalysen.

46) Vgl. zu allgemeinen Klagen über die geringe Effizienz oder mangelhafte Praktikabilität von Netzanalysen, die auf der Basis von Erreichbarkeitsgraphen ausgeführt werden, FREEDMAN (1988b), S. 334; PAGNONI (1990), S. 23

("not useful for large Petri nets") u. S. 143 ("cumbersome"). MEMMI (1983b), S. 221, verweist darauf, daß die Algorithmen zur Analyse von Erreichbarkeitsgraphen lediglich eine exponentiell beschränkte Komplexität besitzen. Auf solche Komplexitätsargumente wird in Kürze ausführlicher eingegangen werden.

Mit analogen Effizienz- und Praktikabilitätsmängeln befaßt sich HOLZMANN (1987), S. 339 u. 342f. Allerdings beziehen sich die Ausführungen auf Erreichbarkeitsanalysen, die für dynamische Systeme außerhalb des Petrinetz-Konzepts angestellt werden. Es wird gezeigt, daß der abstrakte Suchraum erreichbarer Systemzustände für reale Probleme, die zum Bereich der Kommunikationsprotokolle gehören, in der Größenordnung von 8 Millionen Zuständen liegt. Um diese erreichbaren Systemzustände für eine Erreichbarkeitsanalyse zu verwalten, wird ein Speichervolumen von ca. 400 MByte benötigt (S. 342). Um den erreichbaren Zustandsraum auszuwerten, würden auf einer Workstation vom Typ VAX/750 etwa 10 Tage reine Rechenzeit benötigt (S. 324).

Die exorbitante Größe des auszuwertenden Suchraums stellt den tieferen Grund für die geringe Effizienz von Erreichbarkeitsanalysen dar. Dies gilt insbesondere auch für die Effizienzängel der Erreichbarkeitsanalyse von Netzmodellen. Sie beruhen auf der "kombinatorischen Explosion" der zugehörigen Erreichbarkeitsgraphen. Vgl. z.B. PAGNONI (1990), S. 23. Das allgemeine Phänomen der kombinatorischen Explosion wurde schon angesprochen. Bei Erreichbarkeitsgraphen tritt es in zwei besonderen Varianten auf. Die erste gilt für alle Petrinetze mit definierter Ausgangsmarkierung, also z.B. auch für Stelle/Transition-Netze. Die zweite stellt sich dagegen erst ein, wenn sich zur Ausgangsmarkierung auch noch die Zulässigkeit von strukturierten Marken gesellt. Sie trifft daher nur auf Höhere Netze, wie z.B. Synthetische Netze, zu. Hier interessiert lediglich die kombinatorische Explosion von Erreichbarkeitsgraphen als Ursache von Effizienzdefiziten. Daher reicht es aus, sich auf die erste, leichter darzustellende Explosionsvariante zu beschränken. Auf die zweite Explosionsvariante wird später zurückgekommen, wenn die Effizienz des Petrinetz-Konzepts abschließend beurteilt wird.

Die kombinatorische Explosion erster Art läßt sich anhand eines Stelle/Transition-Netzes mit insgesamt M Stellen s_m und $m=1, \dots, M$ verdeutlichen. Dabei wird der Argumentation von BERTHELOT (1982b), S. 251, gefolgt. Vgl. auch die ähnlichen Ausführungen, in denen die Endlichkeit des Erreichbarkeitsgraphen für ein Stelle/Transition-Netz mit beschränkten Markkapazitäten nachgewiesen wurde. Der Einfachheit halber wird hier angenommen, daß es sich um ein K -beschränktes Stelle/Transition-Netz handelt. Daher besitzen alle seine Stellen s_m dieselbe beschränkte Markkapazität KAP_m für jeweils höchstens K Kopien der Basismarke ($KAP_m = K$). Die Anzahl aller denkmöglichen Markierungen einer beliebigen Stelle s_m beträgt $K+1$. Denn neben den 1, 2, ... oder höchstens K Basismarkenkopien, die sich auf der Stelle s_m befinden dürfen, kann die Stelle auch unmarkiert bleiben. Folglich beträgt die Anzahl VA aller kombinatorisch möglichen Variationen von Netzmarkierungen: $VA = (K+1)^M$. Die Netztopologie des zugrundeliegenden Stelle/Transition-Netzes und seine Schaltregel werden zwar dazu führen, daß im allgemeinen nicht alle kombinatorisch möglichen Variationen der Netzmarkierung von der Ausgangsmarkierung des Netzes aus auch tatsächlich erreicht werden können. Folglich wird die Anzahl $\#(RM)$ der tatsächlich erreichbaren Markierungen, die in der Erreichbarkeitsmenge RM des Stelle/Transition-Netzes insgesamt enthalten sind, in der Regel kleiner als die Variationenanzahl VA sein. Welches Ausmaß die Verringerung von kombinatorisch möglichen auf tatsächlich erreichbare Markierungen tatsächlich annimmt, kann jedoch nicht generell bestimmt werden. Daher steht für die Anzahl erreichbarer Netzmarkierungen lediglich fest: $\#(RM) \leq (K+1)^M$. Jede erreichbare Markierung des zugrundeliegenden Stelle/Transition-Netzes bildet im zugehörigen Erreichbarkeitsgraphen genau einen Markierungsknoten (vice versa). Daher gilt für die Anzahl $\#(KN_{RG})$ der Markierungsknoten des Erreichbarkeitsgraphen ebenso: $\#(KN_{RG}) \leq (K+1)^M$. Die Größe des Erreichbarkeitsgraphen wird mit seiner Knotenanzahl gleichgesetzt. Von der Kantenanzahl - und damit der Verknüpfungsdichte - des Erreichbarkeitsgraphen wird also abgesehen. Zugleich wird die Größe des zugrundeliegenden Stelle/Transition-Netzes mit dessen Stellenanzahl M identifiziert. Die Transitionen- und Kantenanzahl des Stelle/Transition-Netzes finden als keine Beachtung als Determinanten der Netzgröße. Aus diesen Vereinbarungen folgt: Die Größe $\#(KN_{RG})$ des Erreichbarkeitsgraphen wächst in Abhängigkeit von der Größe M des zugrundeliegenden Stelle/Transition-Netzes in einem Ausmaß, das sich nach oben nur durch Exponentialfunktionen "fun_K" mit $fun_K(M) = (K+1)^M$ beschränken läßt. Diese Exponentialfunktionen begrenzen das Anwachsen des Erreichbarkeitsgraphen im ungünstigsten aller denkmöglichen Fälle (worst case). Sie präzisieren die eingangs genannte "kombinatorische Explosion" von Erreichbarkeitsgraphen als exponentiell beschränktes Größenwachstum. Wegen des raschen Anwachsens von Exponentialfunktionen kann ebenso - aber plastischer - von einem "exponentiell explodierenden" Größenwachstum geredet werden.

Bisher wurde nur betrachtet, wie sich die Knotenanzahl $\#(KN_{RG})$ eines Erreichbarkeitsgraphen vergrößert, wenn die Stellenanzahl M des zugrundeliegenden Stelle/Transition-Netzes zunimmt. Der Aufwand, der für die Auswertung eines Erreichbarkeitsgraphen betrieben werden muß, kann aber auch von dessen Kantenanzahl abhängen. Jede Schaltkante des Erreichbarkeitsgraphen entspricht genau einem aktivierten Schaltschritt aus dem zugrundeliegenden Stelle/Transition-Netz. Für die Anzahl aller Schaltschritte, deren Aktivierung unter derselben Netzmarkierung denkmöglich ist, läßt sich eine obere Schranke angeben: Wenn N die Anzahl aller Transitionen t_n mit $n=1, \dots, N$ ist, beträgt die Anzahl aller kombinatorisch möglichen Schaltschritte - unabhängig von der jeweils aktuellen Netzmarkierung - insgesamt $2^N - 1$. Denn als Schaltschritt kommt jede nicht-leere Teilmenge der Transitionenmenge des Stelle/Transition-Netzes in Betracht. Da leere Schaltschritte nicht definiert sind und erst recht nicht jemals aktiviert sein können, muß die Anzahl 2^N aller Transitionenteilmengen um die ausgeschlossene leere Teilmenge korrigiert werden. Wie viele von diesen $2^N - 1$ kombinatorisch möglichen Schaltschritten unter einer Netzmarkierung tatsäch-

lich aktiviert sind, bleibt im allgemeinen unbekannt. Aber unter einer beliebigen Netzmarkierung wird die Anzahl der aktivierten Schaltschritte durch die Transitionenanzahl N des Stelle/Transition-Netzes mittels des Terms 2^{N-1} nach oben beschränkt. Diese obere Schranke trifft auf jede erreichbare Netzmarkierung und somit auch für jeden Markierungsknoten des Erreichbarkeitsgraphen zu. Für die Knotenanzahl $\#(KN_{RG})$ des Erreichbarkeitsgraphen steht bereits die obere Schranke $\#(KN_{RG}) \leq (K+1)^M$ fest. Daraus folgt für die obere Schranke seiner Kantenanzahl $\#(KA_{RG})$ unmittelbar: $\#(KA_{RG}) \leq (K+1)^M \cdot (2^{N-1})$. Daher kann nun die Größe eines Erreichbarkeitsgraphen durch die Summe aus seiner maximalen Knotenanzahl $\#(KN_{RG})$ und aus seiner maximalen Kantenanzahl $\#(KA_{RG})$ nach oben beschränkt werden gemäß:

$$\#(KN_{RG}) + \#(KA_{RG}) \leq (K+1)^M + (K+1)^M \cdot (2^{N-1})$$

$$\Leftrightarrow \#(KN_{RG}) + \#(KA_{RG}) \leq 2^N \cdot (K+1)^M$$

Daher wirkt sich auf die obere Schranke der Größe eines Erreichbarkeitsgraphen sowohl die Transitionenanzahl N als auch die Stellenanzahl M des zugrundeliegenden Stelle/Transition-Netzes in exponentieller Weise aus. Das Größenwachstum des Erreichbarkeitsgraphen ist deshalb bei zunehmender Transitionen- oder Stellenanzahl seines Stelle/Transition-Netzes nur exponentiell beschränkt.

Die Auswertung von formalen Konstrukten, die mit der Größe des jeweils zugrundeliegenden Problems so schnell anwachsen, daß sich ihre Größenzunahme nur durch Exponentialfunktionen nach oben einschränken läßt, gilt allgemein als ineffizient. Das wurde schon als "Lösungsdefekt" ausführlicher belegt. Daher wird auch die Auswertung von exponentiell explodierenden Erreichbarkeitsgraphen als ineffizient betrachtet. Gleiches trifft auf alle Auswertungsprobleme für Netzmodelle zu, deren Lösung erfordert, die Erreichbarkeitsgraphen der Netzmodelle auszuwerten. Folglich erweist sich die Erreichbarkeitsanalyse von Netzmodellen als ineffizient. Dies gilt insbesondere auch für die Untersuchung des Erreichbarkeitsproblems als speziellem Anwendungsfall der Erreichbarkeitsanalyse. Von der Ineffizienz der Erreichbarkeitsanalyse ist bereits die einfache Klasse der Stelle/Transition-Netze betroffen. Für Synthetische Netze gilt die gleiche Ineffizienz. Allerdings kommt dort noch die kombinatorische Explosion zweiter Art hinzu, die schon oben erwähnt wurde.

Das drastische Anwachsen der Größe von Erreichbarkeitsgraphen verdeutlicht VALMARI (1988a), S. 96, 101f. u. 111. Er betrachtet Netze, die das Synchronisationsproblem von k speisenden Philosophen modellieren. Für die zugehörigen Erreichbarkeitsgraphen zeigt er, daß sie insgesamt $3^k - 1$ Markierungsknoten und $k \cdot (2 \cdot 3^k - 1)$ Schaltkanten besitzen. Dies bedeutet z.B. bei einem Problemumfang von $k=20$ Philosophen, daß der vollständige Erreichbarkeitsgraph aus fast 3,5 Milliarden Markierungsknoten besteht. RAZOUK (1985c), S. 16, berichtet von einem Erreichbarkeitsgraphen, den er für das gleiche Synchronisationsproblem bei $k=9$ Philosophen konstruierte. Es handelt sich um den größten Erreichbarkeitsgraphen, dessen *tatsächliche* Konstruktion dem Verf. bisher bekannt geworden ist: Er umfaßt nahezu 20.000 Markierungsknoten. Damit wird VALMARI's Berechnungsformel bestätigt ($3^9 - 1 = 19.682$). Die Generierung des Erreichbarkeitsgraphen erfolgte mit Hilfe des Softwarepakets P-NUT, das schon in einer früheren Anmerkung angesprochen wurde, in knapp 7 Minuten CPU-Zeit auf einer Workstation vom Typ VAX/750.

BERTHELOT (1990), S. 6f., betrachtet ein einfaches zyklisches Netz mit S Stellen, auf denen sich insgesamt k Marken befinden. Dafür wird anhand einer übersichtlichen kombinatorischen Überlegung gezeigt, daß sein Erreichbarkeitsgraph insgesamt $((S+k-1)!)/((k! \cdot (S-1)!)$ Markierungsknoten besitzt. Durch Auflösen des Fakultäts-Operators "!" und Kürzen jeweils gleicher Faktoren im Zähler und Nenner des voranstehenden Bruchs läßt sich leicht erkennen, daß auch hier die Knotenanzahl des Erreichbarkeitsgraphen in Abhängigkeit von Parametern " S " oder " k " überaus schnell "explodiert". Numerische Beispiele dafür finden sich auch bei BERTHELOT (1990), S. 7.

Das zuvor aufgezeigte, überaus rasche Anwachsen der Größe von Erreichbarkeitsgraphen wird ebenso angesprochen bei: BERNSTEIN (1973), S. 11; BYRN (1974), S. III-30; PETERSON, J. (1977), S. 243 (allerdings nur mittelbar); AGERWALA (1979), S. 93; BERTHELOT (1982b), S. 251; RAZOUK (1985c), S. 15; FREEDMAN (1988b), S. 334; ABEL, D. (1990), S. 45 ("gewaltige Ausmaße"); PAGNONI (1990), S. 23. Es bleibt hier unbeachtet, daß in den vorgenannten Quellen das exponentielle Explodieren der Erreichbarkeitsgraphen in recht unterschiedlichen Formulierungen thematisiert wird. Beispielsweise werden mitunter nicht die Größe der Erreichbarkeitsgraphen, sondern der äquivalente Umfang der zugrundeliegenden Erreichbarkeitsmengen angesprochen.

Dem Größenwachstum von Erreichbarkeitsgraphen läßt sich zwar durch "Tricks" begegnen, die zu einer Verdichtung der Erreichbarkeitsgraphen führen. Dazu gehören einerseits die Entscheidungsgraphen, die schon im Zusammenhang mit Stelle/Transition-Netzen eingeführt wurden. Andererseits kommen auch kondensierte Erreichbarkeitsgraphen in betracht, die später ausführlicher behandelt werden. Diese Verdichtungsformen sind im allgemeinen aber nur in der Lage, die Effizienzmängel der Erreichbarkeitsanalyse zu lindern, nicht aber zu beseitigen. Dies liegt an zwei Gründen. Erstens lassen sich die verdichteten Erreichbarkeitsgraphen nicht auf alle Analyseprobleme anwenden. Vgl. dazu z.B. die Ausführungen hinsichtlich kondensierter Erreichbarkeitsgraphen. Zweitens unterliegen die verdichteten Erreichbarkeitsgraphen mitunter ebenso einer kombinatorischen Explosion, deren Ausmaß lediglich geringer ausfällt als bei unverdichteten Erreichbarkeitsgraphen. Das trifft vor allem auf Entscheidungsgraphen zu, wenn die zugrundeliegenden Netzmodelle eine Vielzahl von nicht-degenerierten Entscheidungsspielräumen besit-

zen. Schließlich muß auch in Rechnung gestellt werden, daß die Verdichtung der Erreichbarkeitsgraphen in effizienz-mindernder Weise Ressourcen beansprucht.

47) Näheres zu den hier nur kurz angesprochenen komplexitätstheoretischen Untersuchungskonzepten bei COOK, S. (1971), S. 151ff.; KARP (1972), S. 85ff.; KARP (1975b), S. 45ff.; LENSTRA (1977), S. 343ff.; PAUL, W. (1978), S. 181ff.; GAREY (1979), S. 6ff.; BACHEM (1980), S. 813ff.; BRUCKER (1981), S. 26ff.; PARKER, R. (1982a), S. 3ff.; PARKER, R. (1982b), S. 83ff.; COOK, S. (1983), S. 401ff.; KARP (1986), S. 98ff.; ZELEWSKI (1989a), S. 1ff. u. 42ff., und die dort angeführte vertiefende Literatur; vgl. auch die annotierten Bibliographien im Sammelband O'HEIGEARTAIGH (1985) und - als Überblick über neuere Forschungsschwerpunkte - die Beiträge in dem Tagungsband SELMAN (1986).

Allerdings ist darauf zu achten, daß seitens der Komplexitätstheorie nicht die Komplexität von Problemen selbst, sondern die Komplexität derjenigen Algorithmen untersucht wird, die sich zur Problemlösung einsetzen lassen. Im allgemeinen werden in komplexitätstheoretischen Untersuchungen nur die jeweils bekannten Lösungsalgorithmen betrachtet. Dadurch wird strenggenommen nur eine Obergrenze für die algorithmisch definierte Problemkomplexität festgelegt. Denn es kann nicht ausgeschlossen werden, daß später Lösungsalgorithmen mit geringerer Komplexität aufgefunden werden. Dennoch wird der Einfachheit halber zumeist von "der" Problemkomplexität gesprochen. Dies gilt auch für die nachfolgenden Untersuchungen. Es ist zwar möglich, die algorithmenbezogene Problemkomplexität auch für alle überhaupt denkmöglichen Lösungsalgorithmen zu bestimmen, die keineswegs aktuell bekannt sein müssen. Doch diese Komplexitätsanalysen sind wesentlich schwieriger durchzuführen und spielen in der Komplexitätstheorie kaum eine Rolle. Sie liegen auch nicht der anschließenden komplexitätstheoretischen Beurteilung der Erreichbarkeitsanalyse zugrunde. Allerdings wird auf diesen Aspekt an späterer Stelle noch einmal zurückgekommen.

Vgl. zum Obergrenzencharakter der meisten komplexitätstheoretischen Untersuchungen BACHEM (1980), S. 833; TRAUB (1982), S. 60; SCHRADER (1983), S. 1; ZELEWSKI (1989a), S. 48ff., 62, 94 u. 137; vgl. auch die Ausführungen zur Komplexität des Optimierungsproblems der linearen Programmierung.

Vgl. des weiteren zu den besonderen Schwierigkeiten, die Problemkomplexität für alle überhaupt denkmöglichen Lösungsalgorithmen zu ermitteln, HOPCROFT (1974), S. 625; SCHNORR (1974), S. 182; SAVAGE (1976), S. 347; TRAUB (1982), S. 60; COOK, S. (1983), S. 403ff.; ZELEWSKI (1989a), S. 49.

48) Der Problemumfang wird zunächst durch die Anzahl der Determinanten bestimmt, die zur Abbildung und Lösung eines Analyseproblems im jeweils zu untersuchenden Modell explizit repräsentiert werden müssen. Hinzu kommt jeweils der Umfang der Codierungsweise, die für die Repräsentation der Information über problembeeinflussende Determinanten benutzt wird. Vgl. zu Möglichkeiten und Schwierigkeiten der Definition des Problemumfangs bei komplexitätstheoretischen Analysen GAREY (1979), S. 9f. u. 20ff.; ZELEWSKI (1989a), S. 42f., 76f. u. 134ff., sowie die dort angeführte vertiefende Literatur.

49) Das Erreichbarkeitsproblem erweist sich insofern als typisch, als andere interessante Untersuchungsaufgaben auf das Erreichbarkeitsproblem reduziert werden können. Dies wurde früher anhand des Lebendigkeitsproblems exemplarisch dargelegt. Aus den Eigenarten anderer Analyseprobleme - wie etwa der Untersuchung von Promptheit und Terminieren - läßt sich schon intuitiv erkennen, daß sie eng mit der Erreichbarkeit von Markierungen zusammenhängen. Darüber hinaus konnte auch für andere dynamische Netzeigenschaften, die ohne Bezug auf das Erreichbarkeitsproblem formuliert sind, gezeigt werden, daß ihre Untersuchung dennoch zur gleichen Komplexitätsklasse wie das Erreichbarkeitsproblem gehören, sofern es sich um hinreichend anspruchsvolle Untersuchungsaufgaben handelt. Vgl. zu solchen komplexitätsverwandten Analyseproblemen für Netze JANTZEN (1983a), S. 24f.; NIEHUIS (1986), S. 28; ZELEWSKI (1989a), S. 126, Fn. 403.

50) Eine ausführlichere Untersuchung der Komplexität des Erreichbarkeitsproblems hat der Verf. an anderer Stelle vorgelegt; vgl. ZELEWSKI (1989a), S. 120ff.; vgl. auch die dort angeführte vertiefende Literatur.

Die Komplexitätsfunktionen der bisher bekannten Lösungsalgorithmen für das Erreichbarkeitsproblem sind *nach unten* durch eine ACKERMANN-Funktion beschränkt, sofern die Verarbeitungszeit betrachtet wird, die im ungünstigsten Fall zum Gewinnen einer Problemlösung erforderlich ist (Zeitkomplexität); vgl. MÜLLER, H. (1983e); MÜLLER, H. (1985a), S. 387; MÜLLER, H. (1985b), S. 245; BUDINAS (1989), S. 1417; ZELEWSKI (1989a), S. 125f.; PAGNONI (1990), S. 142 (dort wird die ACKERMANN-Funktion aber nur mittelbar als eine nicht primitiv rekursive Funktion angesprochen). Die Gestalt von ACKERMANN-Funktionen wird näher erläutert bei KLEENE (1952), S. 271f.; HERMES (1961), S. 83ff.; STEGMÜLLER (1973a), S. 45; CARDOZA (1976), S. 53; ZELEWSKI (1989a), S. 35. Bei ACKERMANN-Funktionen handelt es sich um nicht primitiv rekursive Funktionen, die mit der Problemgröße wesentlich rascher anwachsen als Exponentialfunktionen; vgl. ZELEWSKI (1989a), S. 125. Daher besitzt die Minimalkomplexität in Größe der ACKERMANN-Funktion eine größere Aussagekraft als die frühere Angabe, die (Zeit-) Komplexität des Erreichbarkeitsproblems sei nach unten durch eine Exponentialfunktion beschränkt. Vgl. zu dieser weniger gehaltvollen Festlegung PETERSON, J. (1977), S. 242; LANDWEBER (1978), S. 352; GRABOWSKI, J. (1978), S. 616.

Wenn hingegen der Speicherplatz untersucht wird, der im schlechtesten Fall zum Gewinnen einer Problemlösung bereitgestellt werden muß (Raumkomplexität), so sind die bekannten Lösungsalgorithmen für das Erreichbar-

keitsproblem *nach unten* durch eine Exponentialfunktion beschränkt; vgl. HACK,M. (1975a), S. 171; CARDOZA (1976), S. 50; JONES,N. (1977), S. 293; LENSTRA (1979), S. 125; JANTZEN (1980a), S. 187; MAYR,E. (1980a), S. 53; PETERSON,J. (1981), S. 146 u. 148; MÜLLER,H. (1983c), S. 182f.; MÜLLER,H. (1985a), S. 377; HOWELL (1987c), S. 47; ZELEWSKI (1989a), S. 127 u. Fn. 401 auf S. 126; PAGNONI (1990), S. 142. In diesem Zusammenhang wird auch immer wieder auf LIPTON (1976) verwiesen. Leider ließ sich diese Quelle jedoch nicht auf dem üblichen Fernleihweg beschaffen.

Weitere Erkenntnisse über die unteren Schranken für die Zeit- und Raumkomplexität des Erreichbarkeitsproblems liegen vor, wenn Stelle/Transition-Netze betrachtet werden, die zusätzliche einschränkende Bedingung erfüllen. Dazu gehören z.B. Stelle/Transition-Netze vom Typ der Zustandsautomaten- oder vom Typ der Synchronisationsnetze. Eine besonders reichhaltige Übersicht über diese Minimalkomplexitäten bietet JONES,N. (1977), S. 286ff. u. 298. Vgl. zu weiteren Beiträgen, die sich mit den unteren Schranken der Zeit- und Raumkomplexitäten von Auswertungsproblemen für Stelle/Transition-Netze und deren Teilklassen befassen:

- Lebendigkeitsproblem: JONES,N. (1977), S. 281ff., 290f. u. 298;
- Deadlockfreiheit: JONES,N. (1977), S. 281;
- K-Beschränktheit: JONES,N. (1977), S. 288, 298 u. 290;
- Beschränktheit: LANDWEBER (1978), S. 353 u. 361;
- Überdeckbarkeit: JONES,N. (1977), S. 290 u. 298; PETERSON,J. (1977), S. 243;
- Persistenz: JONES,N. (1977), S. 289;
- Konservativität: JONES,N. (1977), S. 286 u. 298;
- Kontrollierbarkeit: JONES,N. (1977), S. 291ff.

Dagegen liegen bisher noch keine Erkenntnisse darüber vor, durch welche Komplexitätsfunktionen die bisher bekannten Lösungsalgorithmen für das Erreichbarkeitsproblem *nach oben* beschränkt sind. Zwar gibt BUDINAS (1989), S. 1417, vor, er habe sich mit oberen Schranken für die Zeitkomplexität des Erreichbarkeitsproblems befaßt. Aber eine genauere Analyse seiner Argumentation zeigt, daß er auch er eine ACKERMANN-Funktion nur als untere Schranke der Zeitkomplexität behandelt. Denn BUDINAS konstatiert: "F(n) may be taken as a primitive recursive function with respect to some ... Ackermann function" (S. 1417). Folglich kann die Komplexitätsfunktion selbst schneller anwachsen als die jeweils zugrundeliegende ACKERMANN-Funktion. Das exakte Größenwachstum der Komplexitätsfunktion ist jedoch noch nicht bekannt. Es steht lediglich fest, daß dieses Wachstum bezüglich der ACKERMANN-Funktion primitiv rekursiv beschränkt ist. Diese relative Wachstumsbeschränkung der Komplexitätsfunktion gibt aber keine Auskunft darüber, welche mathematische Gestalt die Komplexitätsfunktion selbst annimmt. Deshalb ist - wie bereits oben festgestellt - bislang nur bekannt, daß ACKERMANN-Funktionen eine *untere* Schranke für die Komplexitätsfunktionen der Zeitkomplexität des Erreichbarkeitsproblems bilden.

51) Ein Problem heißt NP-vollständig, falls es NP-komplex ist und sich alle anderen NP-komplexen Probleme auf dieses - oder ein beliebiges anderes - NP-vollständige Problem reduzieren lassen. Ein Problem ist genau dann NP-komplex, wenn für alle bekannten Lösungsalgorithmen gilt: Ihre Ausführungszeit kann im schlechtest denkmöglichen Fall durch eine polynomiale Komplexitätsfunktion ("P") nach oben beschränkt werden, falls die Algorithmusformulierung auf nondeterministischen TURING-Automaten ("N") beruht.

Grob gesprochen handelt es sich bei polynomialen Funktionen um erweiterte Polynome. Während die Summanden eines Polynoms jeweils nur eine potenzierte Variable enthalten dürfen, können die Summanden eines Polynomials Produkte aus potenzierten Variablen darstellen. Näheres zu polynomialen Funktionen bei HACK,M. (1975a), S. 101; PAUL,W. (1978), S. 91f.; JONES,N. (1977), S. 280; PETERSON,J. (1981), S. 118; ZELEWSKI (1989a), Fn. 29 auf S. 8. Vgl. zur Definition NP-komplexer und NP-vollständiger Probleme KARP (1972), S. 91f.; SCHNORR (1974), S. 183; ULLMAN (1976), S. 145; HARTMANIS (1976), S. 32; GAREY (1979), S. 13, 17, 29, 31f. u. 37; LENSTRA (1979), S. 123; BACHEM (1980), S. 823; BRUCKER (1981), S. 155f.; PARKER,R. (1982a), S. 9; PARKER,R. (1982b), S. 84; CLAUSEN,J. (1986), S. 32; ZELEWSKI (1989a), Fn. 162 auf S. 51 sowie S. 59 u. 67.

Vgl. zu den zugrundeliegenden nondeterministischen TURING-Automaten SCHNORR (1974), S. 183f.; KARP (1975b), S. 45, 47 u. 58 (nondeterministische Algorithmen); HÄUSSLER (1976), S. 20f.; ULLMAN (1976), S. 144f.; SAVAGE (1976), S. 348f.; ZERVOS (1977), S. 297ff.; GAREY (1979), S. 30ff. i.V.m. S. 28f.; BRUCKER (1979), S. 78f.; LENSTRA (1979), S. 123; BRUCKER (1981), S. 42 u. 153ff.; ZELEWSKI (1986a), S. 323ff.; ZELEWSKI (1989a), S. 59ff. Ausführlicher werden NP-komplexe und -vollständige Probleme behandelt bei KARP (1972), S. 91ff.; ULLMAN (1973), S. 96ff.; SAHNI (1974), S. 28ff.; HARTMANIS (1976), S. 30ff.; SAVAGE (1976), S. 347ff., insbesondere S. 356ff.; SCHUSTER,P. (1976), S. 36ff.; LENSTRA (1977), S. 344ff.; GAREY (1978), S. 499ff.; GAREY (1979), S. 13f., 28ff., 37ff. u. 118ff.; BRUCKER (1979), S. 78ff.; BACHEM (1980), S. 822ff.; BRUCKER (1981), S. 41ff. u. 145ff.; PARKER,R. (1982a), S. 8f.; ZELEWSKI (1986a), S. 323ff.; ZELEWSKI (1989a), S. 59ff. u. 63ff.

Vgl. insbesondere zur betriebswirtschaftlichen Relevanz der NP-komplexen und -vollständigen Probleme KARP (1975b), S. 57; BENITO (1978), S. 81ff.; BACHEM (1980), S. 824, mit der Ansicht, daß "... sich fast alle praxisrelevanten Operations Research Probleme in der Klasse NP befinden. "; ZELEWSKI (1989a), S. 69 u. 98ff.

52) NP-vollständige Probleme zeichnen sich dadurch aus, daß die Ausführungszeit ihrer derzeit bekannten Lösungsalgorithmen im ungünstigsten Fall zumindest durch eine Exponentialfunktion *nach oben* beschränkt werden kann.

Dies folgt aus der polynomialen Beschränktheit der zugrundeliegenden nondeterministischen TURING-Automaten, die sich jederzeit in deterministische TURING-Automaten mit exponentiell beschränkter Ausführungszeit transformieren lassen. Daher besitzen NP-vollständige Probleme immer eine exponentiell beschränkte Zeitkomplexität. Vgl. zu dieser Transformationsmöglichkeit und der daraus folgenden exponentiellen Komplexitätsschranke COOK, S. (1971), S. 157; ULLMAN (1976), S. 145; GAREY (1979), S. 32f. (mit Beweis); LENSTRA (1979), S. 122; BACHEM (1980), S. 823 i.V.m. S. 828f.; ZELEWSKI (1989a), S. 60f.

Die exponentiellen Komplexitätsfunktionen der Lösungsalgorithmen, welche die Zeitkomplexität von NP-vollständigen Problemen bei worst case-Betrachtungen nach oben begrenzen, gehören zur Klasse der primitiv-rekursiven Funktionen. Vgl. zu dieser Funktionsklasse z.B. STETTER (1988), S. 48ff. Dagegen nimmt die Zeitkomplexität des Erreichbarkeitsproblems mindestens die Gestalt einer ACKERMANN-Funktion an. Diese Funktion gehört jedoch nicht mehr zur Klasse der primitiv-rekursiven Funktionen (vgl. die Quellen, die zur ACKERMANN-Funktion aufgeführt wurden). Daher ist die Zeitkomplexität des Erreichbarkeitsproblems notwendig größer als die aller NP-vollständigen Probleme. Besonders zu beachten ist, daß für NP-vollständige Probleme die primitiv-rekursiven Exponentialfunktionen als *obere* Schranken der Zeitkomplexität feststehen. Für das Erreichbarkeitsproblem des Petrinetz-Konzepts sind aber die nicht-primitiv-rekursiven ACKERMANN-Funktionen nur als *untere* Schranken der Zeitkomplexität bekannt. Daher kann nicht ausgeschlossen werden, daß die oberen Schranken für die Zeitkomplexität des Erreichbarkeitsproblems noch weitaus ungünstiger ausfallen.

Darüber hinaus lassen sich alle NP-vollständigen Probleme mit deterministischen TURING-Automaten lösen, deren Speicherplatzbedarf in Abhängigkeit vom Problemumfang in polynomial beschränkter Weise ansteigt; vgl. GAREY (1979), S. 170; LENSTRA (1979), S. 124; GOLDBERG (1984a), S. 46; ZELEWSKI (1989a), S. 91. Dagegen wächst der entsprechende Speicherplatzbedarf für die Lösung des Erreichbarkeitsproblems mindestens in exponentiell beschränkter Weise an. Daher übersteigt auch die Raumkomplexität des Erreichbarkeitsproblems diejenige aller NP-vollständigen Probleme.

Ein anderes Analyseproblem, das weniger komplex als das Erreichbarkeitsproblem ausfällt, ist das Erkennen von Deadlocks. Im Gegensatz zum Erreichbarkeitsproblem lassen sich für die Raumkomplexität des Deadlockproblems *obere* Schranken angeben. Darüber hinaus liegen diese oberen Schranken noch unter den unteren Schranken für die entsprechenden Komplexitätsfunktionen des Erreichbarkeitsproblems. Denn der Speicherplatzbedarf für die Bearbeitung eines Deadlockproblems nimmt in Abhängigkeit vom Problemumfang nur in polynomial beschränkter Weise zu; vgl. HOLZMANN (1987), S. 339; VALMARI (1988a), S. 106. Damit bleibt das Deadlockproblem für Petrinetze aber immer noch so komplex wie alle NP-vollständigen Probleme.

Um Mißverständnisse zu vermeiden, wird allerdings darauf hingewiesen, daß die exponentiell beschränkten Komplexitätsfunktionen die Zeitkomplexität von NP-vollständigen Problemen im schlechtest denkmöglichen Fall nur nach oben begrenzen. Dies schließt keineswegs aus, daß für NP-vollständige Probleme auch Lösungsalgorithmen mit polynomial beschränkter Zeitkomplexität aufgefunden werden. Allerdings ist dies *bis heute* für kein NP-vollständiges Problem gelungen. Darüber hinaus legt der heutige komplexitätstheoretische Kenntnisstand sogar die strengere Vermutung nahe, daß sich für NP-vollständige Probleme auch in Zukunft keine Lösungsalgorithmen finden lassen, deren Ausführungszeit im ungünstigsten Fall durch polynomiale Komplexitätsfunktionen nach oben beschränkt werden könnte. Vgl. zu dieser Mutmaßung SCHNORR (1974), S. 184; BRUCKER (1975), S. 4; HARTMANIS (1976), S. 30; BRUCKER (1976), S. 357 u. 365; LENSTRA (1977), S. 344 u. 360; MEHLHORN (1977), S. 193 u. 201; LENSTRA (1979), S. 123; GAREY (1979), S. 33; BACHEM (1980), S. 828; BRUCKER (1981), S. 25 u. 53; LENSTRA (1982), S. 202 u. 207; FRENCH, S. (1982), S. 146 u. 148; ZELEWSKI (1989a), S. 61f. u. 70ff. Im folgenden wird vom Zutreffen dieser Vermutung ausgegangen. Diese Prämisse wird insbesondere bei der anschließenden Klassifizierung der Lösungseffizienz von Problemen eine Rolle spielen. Wegen ihrer gesicherten exponentiellen, aber mutmaßlich nicht mehr gewährleisteten polynomial beschränkten worst case-Komplexität werden die NP-vollständigen Probleme auch als exponentiell oder superpolynomial beschränkte Probleme bezeichnet.

53) Vgl. ZELEWSKI (1989a), S. 68f. Das gleiche Urteil über den Schwierigkeitsgrad von NP-vollständigen Problemen fällen - ohne Bezugnahme auf Probleme des Operations Research - auch HARTMANIS (1976), S. 30; VALIANT (1978), S. 330.

54) Dabei wird als Effizienzmaßstab zugrundegelegt, Lösungsalgorithmen genau dann als (in)effizient einzustufen, wenn sie auf der Basis von deterministischen TURING-Automaten so spezifiziert werden können, daß ihre Ausführungszeit nach oben durch (keine) polynomiale Komplexitätsfunktion beschränkt wird. Entsprechend werden alle Probleme, für die (keine) polynomial beschränkte(n) Lösungsalgorithmen bekannt sind, als polynomial (un)beschränkte Probleme bezeichnet. Alle NP-vollständigen Probleme gehören - nach heutigem Kenntnisstand - zu den polynomial unbeschränkten, aber exponentiell beschränkten Problemen. Denn die Ausführungszeit ihrer derzeit bekannten Lösungsalgorithmen läßt sich auf deterministischen TURING-Automaten nicht durch polynomiale, sondern nur durch exponentielle Komplexitätsfunktionen nach oben beschränken. Daher gelten alle NP-vollständigen Probleme als ineffizient lösbar. Probleme werden dagegen als effizient lösbar betrachtet, wenn sie zur Klasse der polynomial beschränkten Probleme zählen. Vgl. zu dieser Effizienzbeurteilung von einerseits polynomial beschränkten und andererseits polynomial unbeschränkten, exponentiell beschränkten NP-vollständigen o.ä. Problemen KARP (1972), S. 87; LAWLER (1976), S. 5f.; MÜLLER-MERBACH (1976a), S. 71; ULLMAN (1976), S. 140;

ECKER (1977), S. 55; LENSTRA (1977), S. 344f.; PETERSON, J. (1977), S. 243; GAREY (1979), S. 7f.; BACHEM (1980), S. 821; LENSTRA (1982), S. 202; FRENCH, S. (1982), S. 142; PARKER, R. (1982a), S. 7; ZELEWSKI (1986a), S. 324; ZIMMERMANN, H. (1987a), S. 249; HINTZ (1987), S. 88; FREEDMAN (1988b), S. 330; ZELEWSKI (1989a), S. 51ff.

Gegenüber dem o.a. dichotomen Effizienzmaßstab der Komplexitätstheorie können zwar erhebliche Bedenken erhoben werden. Sie beziehen sich erstens darauf, daß die generelle Bevorzugung polynomialer gegenüber über-polynomialer - insbesondere exponentieller - Beschränktheit strenggenommen nur für den Grenzfall unendlicher Problemumfänge gerechtfertigt werden kann. Solche Problemumfänge spielen aber für die Lösung praktischer Probleme keine Rolle. Darüber hinaus wurden in dieser Arbeit schon mehrfach grundsätzliche Vorbehalte gegenüber Bezugnahmen auf das Unendliche geäußert. Es ist zwar möglich, daß keine generelle Präferenz polynomialer Komplexitätsfunktionen unterstellt, sondern lediglich eine bestimmte polynomiale gegenüber einer anderen speziellen nicht-polynomialen Komplexitätsfunktion vorgezogen wird. Aber selbst dann gilt jene spezielle Präferenz nur dann, wenn die betroffenen Probleme einen "hinreichenden" Problemumfang überschreiten. In welchem Ausmaß dieser Mindestumfang von praktisch relevanten Problemausprägungen überschritten wird, erfährt jedoch oftmals in komplexitätstheoretischen Untersuchungen keine Würdigung. Vgl. zu Vorbehalten, die polynomiale der über-polynomialen (exponentiellen) Beschränktheit vorzuziehen, ULLMAN (1976), S. 139; LAWLER (1976), S. 5; BLAND (1981), S. 1041; ZELEWSKI (1989a), S. 95. Zweitens kann die polynomiale Beschränktheit eines Lösungsalgorithmus so ungünstig ausfallen, daß sie nicht mehr dem intuitiven Vorverständnis effizienter Algorithmen gerecht wird; vgl. DANTZIG (1979), S. 1 ("... polynomially bounded time says little."); GAREY (1979), S. 9 u. 135; FRENKEL (1986), S. 110 (KARP zitierend); ZELEWSKI (1989a), S. 96 u. 139.

Solche Einwendungen spielen hier aber keine beachtenswerte Rolle. Denn es werden nur die Ineffizienz von Lösungsalgorithmen für NP-vollständige Probleme mit der noch größeren Ineffizienz von Lösungsalgorithmen für das Erreichbarkeitsproblem verglichen. Das *relative* (In-)Effizienzurteil würde sich auch dann nicht verändern, wenn das o.a. *absolute* Urteil über Effizienz und Ineffizienz nicht an das Vorliegen bzw. Fehlen von polynomial beschränkten Lösungsalgorithmen geknüpft würde. Daher wurde auch eingangs nicht von einer - absolut verstandenen - Ineffizienz der Erreichbarkeitsanalyse gesprochen. Statt dessen wurde in vorsichtigerer, das relative Moment hervorhebender Weise nur von ihrem Effizienzdefizit geredet.

55) Dies ist die komplexitätstheoretische Präzisierung des oben festgestellten Effizienzdefizits der Erreichbarkeitsanalyse.

56) Ebenso kann von einem besonders schwerwiegenden "Lösungsdefekt" des Erreichbarkeitsproblems gesprochen werden. Damit erfolgt ein Brückenschlag zum Phänomen der Lösungsdefektheit von Problemen, das bereits in der Rahmenlegung dieser Arbeit erörtert wurde. Dort interessierten die praktischen Schwierigkeiten, die das Lösen von Optimierungsmodellen für Maschinenbelegungsplanungen zu vereiteln drohen. Es wurde in einer ersten Annäherung von einem lösungsdefekten Problem gesprochen, wenn der Ressourceneinsatz, der zur Problemlösung erforderlich ist, mit dem Problemumfang in exponentieller Weise ansteigt. Dieser Sachverhalt wird hier durch das Konzept der NP-Vollständigkeit widerspiegelt und zugleich in zweifacher Weise präzisiert. Erstens handelt es sich nicht um irgendeinen Ressourceneinsatz, sondern um die Ausführungszeit von Lösungsalgorithmen auf deterministischen TURING-Automaten im jeweils schlechtest möglichen Fall. Zweitens wächst dieser Ressourceneinsatz nicht generell in exponentieller Art an, sondern der Ressourceneinsatz wird nach heutigem Erkenntnisstand durch keine polynomiale, wohl aber durch eine exponentielle Funktion nach oben beschränkt. Unter Berücksichtigung dieser beiden Verfeinerungen stellen alle NP-vollständigen Probleme zugleich lösungsdefekte Probleme dar. Nun verhalten sich Lösungsalgorithmen für das Erreichbarkeitsproblem des Petrinetz-Konzepts noch ineffizienter als die bereits lösungsdefekten NP-vollständigen Probleme. Daher zeichnet sich dieses Erreichbarkeitsproblem durch einen überaus gravierenden Lösungsdefekt aus.

Zur Verdeutlichung der "Überschwere" des Erreichbarkeitsproblems für Petrinetze wird auf Ermittlungsprobleme für die bekanntere Netzplantechnik verwiesen. Es läßt sich zeigen, daß schon für stochastische Netzpläne das Problem, den Erwartungswert der Projektdauer zu ermitteln, zur Klasse der NP-harten Probleme gehört. Vgl. HAGSTROM (1988), S. 139ff., insbesondere S. 146. NP-harte Probleme sind mindestens so schwierig wie die hier interessierenden NP-vollständigen Probleme - unter Umständen aber auch noch schwieriger. Das Erreichbarkeitsproblem des Petrinetz-Konzepts erweist sich aber als so schwierig, daß sein Lösungsaufwand weit über den aller NP-vollständigen - und auch den aller NP-harten Probleme - hinausreicht. Folglich leidet das Erreichbarkeitsproblem unter einem wesentlich größeren Lösungsdefekt als die Aufgabe, für stochastische Netzpläne die erwarteten Projektdauern zu bestimmen. Vgl. zu einer Präzisierung des hier nicht näher ausgeführten Verständnisses der NP-harten Probleme HARTMANIS (1976), S. 32; MEHLHORN (1977), S. 193; GAREY (1979), S. 19 u. 109ff.; LENSTRA (1979), S. 123f.; BRUCKER (1979), S. 82; FRENCH, S. (1982), S. 148f.; LENSTRA (1982), S. 202; PARKER, R. (1982b), S. 83f.; CLAUSEN, J. (1986), S. 32; ZELEWSKI (1989a), S. 7f. u. 82ff.

57) Dabei wird vorausgesetzt, daß das Erreichbarkeitsproblem als typische Untersuchungsaufgabe aus dem Bereich der Erreichbarkeitsanalyse akzeptiert wird.

58) Vgl. zur Kritik an worst case-Untersuchungen, die sich vornehmlich auf deren mangelnde praktische Bedeutung erstreckt, SCHNORR (1974), S. 182; KARP (1975b), S. 64f.; BACHEM (1980), S. 817; FISHER, M. (1982), S. 17;

COFFMAN (1982), S. 320; LENSTRA (1982), S. 205; ZELEWSKI (1989a), S. 94f. u. 97. Vgl. auch den bereits angeführten Vorbehalt gegenüber der Praxisferne von unendlich oder "hinreichend" großen Problemumfängen, die seitens der Komplexitätstheoretischen worst case-Betrachtungen stets vorausgesetzt werden.

59) Solche repräsentativen Sammlungen können direkt oder indirekt notiert werden. Im ersten Fall wird eine endliche Menge mehrerer Problemausprägungen betrachtet, deren Gesamtheit für das jeweils interessierende Problem als repräsentativ gilt. Im zweiten Fall wird das Problem mit der Hilfe von Variablen formuliert, über deren zulässigen Ausprägungen Wahrscheinlichkeitsfunktionen in der Gestalt von diskreten Verteilungs- oder kontinuierlichen Dichtefunktionen definiert sind. Dann wird unterstellt, daß die Wahrscheinlichkeitsfunktionen die relativen Auftretenshäufigkeiten aller denkmöglichen Problemausprägungen repräsentativ wiedergeben.

60) Vgl. zu solchen Komplexitätstheoretischen Durchschnittsbetrachtungen KARP (1976), S. 3ff.; GAREY (1979), S. 148ff.; COFFMAN (1982), S. 319ff.; PARKER, R. (1982b), S. 87f.; KARP (1985), S. 52ff. (annotierte Bibliographie); KARP (1986), S. 106ff.; ZELEWSKI (1989a), S. 46ff., 81, 131 u. 137ff.

61) Die Bevorzugung von average case- gegenüber worst case-Analysen läßt sich anhand eines Analogons aus dem Bereich des Operations Research verdeutlichen. Es handelt sich um das Problem der Linearen Optimierung. Der Simplex-Algorithmus, auf dessen vielfältige Varianten hier nicht weiter eingegangen wird, gilt als praktisch bedeutsamster Lösungsalgorithmus für dieses Optimierungsproblem. Dennoch konnte gezeigt werden, daß sich die Ausführungszeit dieses Algorithmus' im ungünstigsten Falle nur durch eine exponentielle Komplexitätsfunktion nach oben beschränken läßt. Dagegen existieren andere Lösungsalgorithmen - insbesondere die von KHACHIYAN und KARMARKAR -, die sich bei worst case-Analysen als polynomial beschränkt herausstellen. Dennoch vermochten sich diese alternativen Lösungsalgorithmen für die Lineare Optimierung bislang nicht durchzusetzen. Ein Grund für dieses Phänomen liegt darin, daß die Anwendung von Lösungsalgorithmen vorwiegend von ihrer Durchschnittskomplexität abhängt. Darüber sagen worst case-Betrachtungen nichts aus. Statt dessen konnte durch average case-Untersuchungen gezeigt werden, daß sich der Simplex-Algorithmus durch eine polynomial beschränkte Durchschnittskomplexität auszeichnet. Die o.a. alternativen Lösungsalgorithmen besitzen zwar ebenfalls eine polynomial beschränkte Durchschnittskomplexität, weil ihre Ausführungszeit schon für den schlechtesten denkmöglichen Fall durch polynomiale Komplexitätsfunktionen begrenzt wird. Aber die Zugehörigkeit zur gleichen Komplexitätsklasse bei average case-Betrachtungen hat bisher ausgereicht, daß sich der Simplex-Algorithmus gegenüber den alternativen Lösungsalgorithmen bei der praktischen Bewältigung von linearen Optimierungsaufgaben behaupten konnte. Darüber hinaus wurden konkrete Sammlungen "repräsentativer" Problemausprägungen aus dem Bereich der Linearen Optimierung untersucht, um die Durchschnittskomplexität der alternativen Lösungsalgorithmen - über ihre grobe Festlegung durch die polynomiale Beschränktheit hinaus - präziser festzulegen. Die Ergebnisse variieren zwar in Abhängigkeit vom jeweils zugrundegelegten Untersuchungsdesign. Aber sie stimmen weitgehend in der Grundaussage überein, daß die alternativen Lösungsalgorithmen gegenüber dem Simplex-Algorithmus keine wesentlich geringere Durchschnittskomplexität besitzen. Schwache Hinweise auf mögliche average case-Effizienzvorteile für Lösungsalgorithmen vom KARMARKAR-Typ finden sich allenfalls bei HOOKER (1986), S. 86; FRENKEL (1986), S. 111; ZELEWSKI (1989a), S. 138f. Diese Hinweise bleiben allerdings hinsichtlich ihrer praktischen Tragweite bislang umstritten.

Daher ist das Beispiel der Linearen Optimierung ein ernst zu nehmendes Indiz dafür, daß die Komplexitätsfunktionen, die aus average case-Untersuchungen resultieren, für die praktische Problemlösung weitaus bedeutsamer sind als die Ergebnisse von worst case-Betrachtungen. Zwar können bei der Penetration von Algorithmen auch Trägheitseffekte eine Rolle spielen, wie z.B. die Gewöhnung an bereits erlernte Algorithmen oder das Vorhandensein von Softwareprogrammen für etablierte Algorithmen. Doch ist seit der Vorstellung von Lösungsalgorithmen für die Lineare Optimierung, deren Ausführungszeit im ungünstigsten Fall polynomial beschränkt ist, schon mehr als ein Jahrzehnt verstrichen. Angesichts dieses erheblichen Zeitraums wirkt der Verweis auf die Erklärungskraft von Trägheitseffekten wenig glaubwürdig.

Der Verf. hat das oben skizzierte Problem der Linearen Optimierung aus Komplexitätstheoretischer Perspektive an anderer Stelle näher behandelt; vgl. ZELEWSKI (1989a), S. 129ff. Vgl. darüber hinaus zu:

- der exponentiell beschränkten worst case-Komplexität und "theoretischen" Ineffizienz des Simplex-Algorithmus KLEE, V. (1972), S. 174 i.V.m. S. 161ff.; KARP (1975a), S. 24; GOLDFARB (1979), S. 278(ff.) u. 284; BACHEM (1980), S. 833; SCHÖNLEIN (1981), S. 116f.; JOHNSON, D. (1981), S. 400; BLAND (1981), S. 1081; LENSTRA (1982), S. 204; SCHRADER (1982), S. 267; WEBER, H. (1982), S. B/229; SCHRADER (1983), S. 2; BORGWARDT (1985), S. 650; DERIGS (1985), S. 49; PAN (1986), S. 134; ZELEWSKI (1989a), S. 131ff.
- der polynomial beschränkten average case-Komplexität und "praktischen" Effizienz des Simplex-Algorithmus DANTZIG (1966), S. 185; NEWELL (1969), S. 372; KLEE, V. (1972), S. 174f.; KARP (1975a), S. 24; KARP (1975b), S. 65; DANTZIG (1979), S. 1; JOHNSON, D. (1981), S. 400; BORGWARDT (1982), S. 176 i.V.m. S. 160ff.; SCHRADER (1982), S. 267; LENSTRA (1982), S. 204 u. 206; PARKER, R. (1982a), S. 7; SCHRADER (1983), S. 2f.; SMALE (1983), S. 530ff.; BORGWARDT (1985), S. 650 u. 656ff.; PAN (1986), S. 134; ZELEWSKI (1989a), S. 130f. u. 133.

- den alternativen Lösungsalgorithmen für die Lineare Optimierung mit polynomial beschränkter worst case-Komplexität KHACHIVAN (1979), S. 191ff., insbesondere S. 191 u. 194; PADBERG (1980), S. 3ff.; BACHEM (1980), S. 833ff.; BLAND (1981), S. 1044ff., insbesondere S. 1046; SCHÖNLEIN (1981), S. 116ff., insbesondere S. 116 u. 120; GRÖTSCHEL (1981), S. 172ff.; GACS (1981), S. 62ff.; SCHRADER (1982), S. 274ff.; WEBER, H. (1982), S. B/230ff.; SCHRADER (1983), S. 4ff.; KARMARKAR, N. (1984), S. 377ff.; DERIGS (1985), S. 51ff.; SCHÖNLEIN (1986), S. 344ff.; HOOKER (1986), S. 77ff.; PAN (1986), S. 132ff. (für die nebenläufige Ausführung von KARMARKAR's Algorithmus auf Multiprozessor-Automaten); TARDOS (1986), S. 251ff. (für den Sonderfall kombinatorischer Problemformulierungen); ZELEWSKI (1989a), S. 133ff.
- den bisher wenig ermutigenden Untersuchungen, in denen die average case-Komplexität der vorgenannten alternativen Lösungsalgorithmen anhand von Sammlungen repräsentativer Problemausprägungen ermittelt wurde, SCHÖNLEIN (1981), S. 120f.; BLAND (1981), S. 1077; WEBER, H. (1982), S. B/230; LENSTRA (1982), S. 204; SCHRADER (1983), S. 12; KOLATA (1984), S. 1379; DERIGS (1985), S. 51 u. 55; SCHÖNLEIN (1986), S. 344 u. 353; HOOKER (1986), S. 77 u. 85; ZELEWSKI (1989a), S. 137ff.

62) Vgl. zu anderen Schwierigkeiten, denen average case-Betrachtungen entweder speziell unterliegen oder die sie mit worst case-Untersuchungen gemeinsam teilen, ZELEWSKI (1989a), S. 96 bzw. 94ff.

63) Vgl. zu Vorbehalten gegenüber der Repräsentativität von - direkt oder indirekt notierten - Problemausprägungssammlungen KARP (1976), S. 3; SIMON, H. (1976b), S. 298; RABIN (1976), S. 21; VALIANT (1978), S. 333; GAREY (1979), S. 150 (bezüglich des Problems, bei indirekter Notation eine geeignete Wahrscheinlichkeitsfunktion auszuwählen); BACHEM (1980), S. 817; BROWN, C.A. (1981), S. 590f.; RARDIN (1982), S. 12; GOLDBERG (1984a), S. 53; ZELEWSKI (1989a), S. 97.

64) Vgl. dazu vor allem die Kritik an den konzeptionellen und Effektivitätsdefiziten der Invariantenanalyse.

65) Schon an frühere Stelle wurde aufgezeigt, daß sich die Durchschnittskomplexität des Simplex-Algorithmus durchaus als praktisch befriedigend herausstellt, obwohl er aus der theoretischen Perspektive von worst case-Untersuchungen für das Problem der Linearen Optimierung als ineffizient gilt.

Darüber hinaus gelten noch nicht einmal die Erkenntnisse aus theoretischen worst case-Analysen in absoluter Weise. Vielmehr treffen sie immer nur relativ zu dem aktuellen Wissen über problemlösende Algorithmen zu. Dies klang bereits in einer früheren Anmerkung an, die auf die konzeptionellen Grundlagen der Komplexitätstheorie verwies. Dort wurde angedeutet, daß die Problemkomplexität im allgemeinen anhand des Ressourcenverzehr gemessen wird, der zur Ausführung der jeweils *bekanntesten* Lösungsalgorithmen erforderlich ist. Dies schließt nicht aus, daß für dasselbe Analyseproblem in Zukunft neuartige Lösungsalgorithmen entdeckt werden, deren Komplexitätsfunktionen zu einer geringeren Komplexitätsklasse gehören als alle *bisher* bekannten Lösungsalgorithmen. Es kann daher strenggenommen nicht ausgeschlossen werden, daß auch für das Erreichbarkeitsproblem des Petrinetz-Konzepts zu späterer Zeit vollkommen neuartige Lösungsalgorithmen entwickelt werden, die eine geringere worst case-Komplexität besitzen als die oben angeführte.

Dafür läßt sich abermals auf das Problem der Linearen Optimierung verweisen. Solange der Simplex-Algorithmus denjenigen Lösungsalgorithmus mit der besten bekannten, aber nur exponentiell beschränkten worst case-Komplexität für dieses Problem darstellte, gehörte das lineare Optimierungsproblem zur Klasse der schwierigen, ineffizient lösbaren, NP-vollständigen Probleme; vgl. GAREY (1979), S. 155; ZELEWSKI (1989a), S. 137 (einschließlich der Fn. 439). Sobald aber der erste Lösungsalgorithmus bekannt wurde, der das Problem der Linearen Optimierung auch im ungünstigsten Fall mit polynomial beschränkter Ausführungszeit zu bewältigen vermag, mußte es zur Klasse der leichteren, effizient lösbaren, polynomial beschränkten Probleme gerechnet werden; vgl. ZELEWSKI (1989a), S. 137. Dies zeigt, wie sich "die" Komplexität eines unveränderten Problems relativ zum Wissen über dessen Lösungsalgorithmen im Zeitablauf verändern kann.

66) Es handelt sich um das Erkennen von Deadlockmarkierungen anhand von Erreichbarkeitsgraphen.

67) Vgl. VALMARI (1988a), S. 95ff., insbesondere S. 96, 101f. u. 111. Dabei wird die Erreichbarkeitsanalyse von Netzen mit der früher vorgestellten Reduktionsanalyse so kombiniert, daß nur noch ein reduzierter Erreichbarkeitsgraph untersucht wird. Die Reduktion ist so angelegt, daß keine Informationen über die analyserelevante Netzeigenschaft - die Gesamtheit aller Deadlockmarkierungen - verloren gehen. Die dadurch erzielte Vereinfachung des Analyseproblems wird von VALMARI (1988a), S. 111 i.V.m. S. 102, anhand eines Beispiels verdeutlicht: Die Untersuchung des Synchronisationsproblems der essenden Philosophen führt im allgemeinen dazu, daß bei k vorgegebenen Philosophen als Problemumfang ein Erreichbarkeitsgraph mit insgesamt $3^k - 1$ Markierungsknoten analysiert werden muß. VALMARI's Analysealgorithmus erlaubt es dagegen, einen reduzierten Erreichbarkeitsgraph mit nur noch $3k^2 - 3k + 2$ Markierungsknoten auszuwerten. Dies bedeutet z.B. für das Problem, $k=20$ essende Philosophen zu synchronisieren, daß der Umfang des zu untersuchenden Erreichbarkeitsgraphen von knapp 3,5 Milliarden (!) auf 1.142 Knoten verringert wird.

68) Vgl. die Erörterung von ZELEWSKI (1986a), S. 321ff., insbesondere S. 325ff., Automatische Informationsverarbeitungssysteme mit nebenläufigen Architekturen zu nutzen, um die Bearbeitungseffizienz von NP-vollständigen Problemen zukünftig zu steigern.

5.2.3.5.2 Konstruktion von Erreichbarkeitsgraphen

Die Erreichbarkeitsanalyse eines Netzes besteht aus der Auswertung seines Erreichbarkeitsgraphen. Die Auswertung kann aber nur dann vollzogen werden, wenn der Erreichbarkeitsgraph bereits vorliegt. Zumindest muß er in dem Ausmaß zur Verfügung stehen, wie es für den jeweils aktuellen Vollzugszustand des Auswertungsprozesses erforderlich ist. In beiden Fällen setzt die Erreichbarkeitsanalyse eine Konstruktion des Erreichbarkeitsgraphen voraus. Der Übersichtlichkeit halber wird hier nur die vollständige Konstruktion eines Erreichbarkeitsgraphen betrachtet, die seiner anschließenden Auswertung zeitlich vorgelagert ist. Auf die zeitlich verschränkte, "schritt haltende" Konstruktion und Auswertung eines Erreichbarkeitsgraphen wird später kurz zurückgekommen¹⁾, ohne sie im Detail auszuführen²⁾.

Für die vollständige Konstruktion eines Erreichbarkeitsgraphen wird eine Konstruktionsmethode vorgeschlagen, die von der üblichen sequentiellen Konstruktionsweise³⁾ abweicht. Die hier vorgeschlagene Konstruktionsmethode zeichnet sich dadurch aus, daß sie die inhärente Nebenläufigkeit des Petrinetz-Konzepts⁴⁾ konsequent ausnutzt: Erreichbarkeitsgraphen werden durch einen nondeterministischen⁵⁾, nebenläufig ausführbaren⁶⁾ Algorithmus hervorgebracht. Es wird daher auch von einer nebenläufigen Konstruktionsmethode gesprochen⁷⁾. Sie beruht darauf, grundsätzlich alle artifiziellen Sequentialisierungen⁸⁾ von kausal unabhängigen Konstruktions-schritten⁹⁾ zu vermeiden¹⁰⁾. Dadurch wird die hier vorgestellte Konstruktionsmethode von vornherein so ausgelegt, daß sie sich später ohne größere Schwierigkeiten auf Automatischen Informationsverarbeitungssystemen mit parallelen Systemarchitekturen¹¹⁾ implementieren läßt. Allerdings erfolgt keine Festlegung auf eine bestimmte Implementierungsweise, um den Anwendungsbereich der Konstruktionsmethode nicht unnötig einzuschränken¹²⁾. Daher bleibt es hier grundsätzlich offen, in welchem Ausmaß das Nebenläufigkeitspotential der Konstruktionsmethode bei ihrer späteren Implementierung verwirklicht wird¹³⁾.

Die Geschwindigkeitsvorteile paralleler Automatenarchitekturen, die bei der Ausführung nondeterministischer Algorithmen gegenüber konventionellen sequentiellen Automatenarchitekturen des VON NEUMANN-Typs¹⁴⁾ bestehen¹⁵⁾, wachsen tendenziell mit der Anzahl von Prozessoren, die unabhängig voneinander operieren können. Die meisten der derzeit angebotenen parallel arbeitenden Informationsverarbeitungssysteme besitzen nur einige wenige Prozessoren¹⁶⁾. Ihre geringfügige Parallelität gestattet daher lediglich in bescheidenem Umfang, die inhärente Nebenläufigkeit der Konstruktion von Erreichbarkeitsgraphen konkret zu nutzen¹⁷⁾. Allerdings trifft diese Einschränkung nicht mehr zu, wenn zu Automatischen Informationsverarbeitungssystemen mit massiv parallelen Architekturen übergegangen wird. Dabei handelt es um Multiprozessorsysteme mit mehreren Tausend Einzelprozessoren.

Erste Exemplare solcher Informationsverarbeitungssysteme liegen bereits vor. Beispielsweise¹⁸⁾ verfügt die "Connection Machine"¹⁹⁾ bei vollständiger Ausschöpfung ihres Parallelitätsangebots über mehr als 64.000 unabhängig arbeitende Prozessoren. Obwohl Transputer-Systeme derzeit zumeist noch in Konfigurationen mit weniger als zwei Dutzend Einzelprozessoren (Transputern) betrieben werden²⁰⁾, läßt ihre Architektur ebenfalls die Gestaltung von Multiprozessorsystemen mit mehreren Tausend Prozessoren zu. In dieser Größenordnung wirkt sich die reale Beschränkung der Prozessorenanzahl auf die nebenläufige Konstruktion von Erreichbarkeitsgraphen kaum noch aus²¹⁾.

Die massiv parallelen Automatenarchitekturen stellen daher einen vielversprechenden Ansatz dar, um die Konstruktion von Erreichbarkeitsgraphen zu beschleunigen. Gleiches gilt für die später skizzierten Methoden für die Auswertung von Erreichbarkeitsgraphen, die sich ebenfalls nebenläufig ausgestalten lassen. Daher verliert der früher vorgetragene Vorbehalt, die Erreichbarkeitsanalyse von Netzmodellen leide unter einem beträchtlichen Effizienzdefizit, durch die Aussicht auf massiv parallele Implementierungen von nebenläufigen Konstruktions- und Aus-

wertungsmethoden an Bedeutung²²⁾. Da die Erreichbarkeitsanalyse in dieser Arbeit eine zentrale Rolle spielt, wird das Ausschöpfen ihrer Nebenläufigkeitspotentiale ausführlicher gewürdigt. Um Überschneidungen zu vermeiden, wird dies anschließend anhand der Konstruktion von Erreichbarkeitsgraphen exemplarisch dargelegt²³⁾.

Die Konstruktionsmethode für Erreichbarkeitsgraphen wird durch eine algorithmische Formulierung präzisiert. Das Resultat ist eine nebenläufige Konstruktionsprozedur²⁴⁾, die sich mit Hilfe der Automatischen Informationsverarbeitung implementieren läßt. Jeder Konstruktions-schritt wird durch eine atomare Anweisung oder eine daraus zusammengesetzte Subprozedur spezifiziert. Die Nebenläufigkeit der Konstruktionsprozedur findet in informationsverarbeitenden Agenten $AGENT_a$ mit $a \in \{1, \dots, A\}$ und $A \in \mathcal{N}_+$ ihren Ausdruck. Es müssen mindestens zwei Agenten vorhanden sein ($A \geq 2$)²⁵⁾. Jedem Agenten ist eine Aufgabe zugeordnet. Er erfüllt sie dadurch, daß er eine aufgabenspezifische Subprozedur ausführt. Die Agenten können ihre Subprozeduren im Prinzip voneinander unabhängig - also nebenläufig - ausführen²⁶⁾. Damit entsprechen die Agenten weitgehend den "Bereichsexperten" oder "Dämonen"²⁷⁾ aus dem Blackboard-Konzept.

Der Kern des Blackboard-Konzepts²⁸⁾ besteht in der Koordinierung²⁹⁾ nebenläufig operierender Bereichsexperten durch eine gemeinsame Informationsbasis. Sie stellt eine "Tafel" (Blackboard) dar, von der die Bereichsexperten die Informationen über ihre Teilaufgaben beziehen und auf die sie die Ergebnisse ihrer Aufgabenerfüllungen zurückschreiben. Die Bereichsexperten verhalten sich dabei teilautonom: Einerseits können sie in nebenläufiger Weise voneinander unabhängig agieren. Andererseits erfolgt eine "lose Kopplung"³⁰⁾ ihrer Aktivitäten mittels der gemeinsamen Informationsbasis. In dieser speziellen Koordinierung nebenläufig agierender Bereichsexperten liegt das Charakteristikum des Blackboard-Konzepts. Dieser allgemeine Ansatz wird hier in spezieller Weise als eine Agenda ausgeformt. Sie enthält nur die jeweils noch zu erfüllenden Teilaufgaben, nicht mehr aber die Ergebnisse ihrer Erfüllungen. Diese Ergebnisse werden statt dessen an andere Subprozeduren oder Subagenten weitergereicht.

Das Blackboard-Konzept wird hier der nebenläufigen Konstruktion von Erreichbarkeitsgraphen bewußt zugrundegelegt. Es handelt sich um ein Koordinierungskonzept, das seitens der Erforschung Künstlicher Intelligenz für das Studium paralleler Informationsverarbeitung breite Beachtung findet. Daher liegen bereits zahlreiche Konzeptimplementierungen vor, welche die oben skizzierte Agentenkoordinierung realisieren können. Daher brauchen sich die nachfolgenden Ausführungen zur Konstruktion von Erreichbarkeitsgraphen um die informationstechnischen Details der Agentenkoordinierung nicht weiter zu kümmern. Darüber hinaus wird eine konzeptionelle Schnittstelle zwischen Petrinetzen und Künstlicher Intelligenz etabliert. Darauf wird später noch ausführlicher zurückgekommen.

Allerdings wird das konventionelle Blackboard-Konzept durch eine besondere hierarchische Gruppierung der Agenten verfeinert³¹⁾: Eine übergeordnete Agentengruppe besteht aus den Mitgliedern $AGENT_{\bar{u}}$ mit $\bar{u} \in \{1, \dots, A\bar{u}\}$, $A\bar{u} \in \mathcal{N}_+$ und $A\bar{u} \leq 0,5 \cdot A$ ³²⁾. Jedem übergeordneten Agenten $AGENT_{\bar{u}}$ ist genau eine untergeordnete Agentengruppe zugeordnet. Sie umfaßt die Subagenten $AGENT_{\bar{u},u}$ mit $u \in \{1, \dots, A\bar{u}\}$ und $A\bar{u},u \in \mathcal{N}_+$. Die Subagenten hängen von ihrem übergeordneten Agenten insofern ab, als der übergeordnete Agent auszuführende Subprozeduren an seine Subagenten übergibt³³⁾. Diese Subprozeduren werden von den Subagenten, die zum selben übergeordneten Agenten gehören, nebenläufig ausgeführt. Auch die übergeordneten Agenten erfüllen ihre Subprozeduren voneinander unabhängig. Schließlich wird festgelegt, daß alle Agentengruppen die insgesamt vorhandene Agentenanzahl A exakt ausschöpfen³⁴⁾.

Der Konstruktionsalgorithmus wird mit der Hilfe einer Pseudocode-Variante³⁵⁾ notiert. Die Pseudocode-Notation wurde so ausgelegt, daß sie insbesondere erkennen läßt, welche Konstruktions-schritte sich nebenläufig ausführen lassen und welche sequentiell aufeinander folgen müssen³⁶⁾. Sie lehnt sich an eine Notationsvariante von SOWA³⁷⁾ und an Konstrukte aus der Programmiersprache ADA³⁸⁾ an. Als Eingabeinformationen setzt der Algorithmus ein beliebiges

Netz - z.B. ein Stelle/Transition-Netz oder ein Synthetisches Netz - mit wohldefinierter Netzdynamik voraus. Aufgrund der Netzdynamik sind die Ausgangsmarkierung M_0 und die schrittbezogene Schaltregel $SR_s^{39)}$ bekannt. Als Ausgabeinformationen werden für den zugehörigen Erreichbarkeitsgraphen $RG(M_0, SR) = (KN_{RG}, KA_{RG}; bk)$ die Knotenmenge KN_{RG} , die Kantenmenge KA_{RG} und die Erfüllungsmenge BK der Beschriftungsfunktion $bk^{40)}$ erzeugt. Der derart spezifizierte nebenläufige Algorithmus zur Konstruktion von Erreichbarkeitsgraphen nimmt für ein beliebiges Synthetisches Netz $SN = (TOP, SPEC_{MSIG}; BES, M_0; IB)^{41)}$ folgende Gestalt an⁴²⁾:

SEQ Erreichbarkeitsgraph_Konstruktion BEG

<anw₁> ≈ lies als Eingabeinformation die Definition $SN = (\dots, M_0, \dots)$ des Netzes SN ein, dessen Erreichbarkeitsgraph RG konstruiert werden soll

PAR Initialisierung BEG

TASK

<anw₂> ≈ initialisiere die Knotenmenge KN_{RG} des Erreichbarkeitsgraphen mit der Ausgangsmarkierung M_0 durch $KN_{RG} := \{M_0\}$

TASK

<anw₃> ≈ initialisiere die Kantenmenge KA_{RG} des Erreichbarkeitsgraphen durch $KA_{RG} := \emptyset$

TASK

<anw₄> ≈ initialisiere die Erfüllungsmenge BK des Erreichbarkeitsgraphen durch $BK := \emptyset$

TASK

<anw₅> ≈ initialisiere die Menge UKN der zu untersuchenden Knoten durch $UKN := \{M_0\}$

PAR Initialisierung END

LOOP globale_Knotenanalyse BEG⁴³⁾

EXIT WHEN $UKN = \emptyset$

SEQ Untersuchung_Knotenmenge BEG

<anw₆> ≈ initialisiere die Menge UKN_{AK} der aktuell zu untersuchenden Knoten durch $UKN_{AK} := UKN$

PAR singuläre_Knotenanalyse BEG⁴⁴⁾

EXIT WHEN $UKN_{AK} = \emptyset$

AGENT₁

CALL Knotenuntersuchung

reserviere die Menge UKN_{AK} für einen Zugriff, sofern sie durch keinen anderen Zugriff temporär reserviert ist⁴⁵⁾

wähle aus der Menge UKN_{AK} genau einen Knoten M_t aus

eliminiere den Knoten M_r aus der Menge UKNAK durch $UKNAK := UKNAK - \{M_r\}$

gib die Menge UKNAK für andere Zugriffe frei

AGENT _{\ddot{u}} [dito für alle $\ddot{u} \in \{2, \dots, A\ddot{U}-1\}$]

AGENT_{A \ddot{U}}

CALL Knotenuntersuchung

reserviere die Menge UKNAK für einen Zugriff, sofern sie durch keinen anderen Zugriff temporär reserviert ist

wähle aus der Menge UKNAK genau einen Knoten M_r aus

eliminiere den Knoten M_r aus der Menge UKNAK durch $UKNAK := UKNAK - \{M_r\}$

gib die Menge UKNAK für andere Zugriffe frei

TASK Knotenuntersuchung

SEQ Knotenuntersuchung BEG⁴⁶⁾

$\langle \text{anw}_7 \rangle \approx$ bestimme durch Anwenden der Schaltregel auf die Markierung M_r im zugrundeliegenden Netz die Menge SSM_r aller Schaltschritte SS_a , die unter der Markierung M_r aktiviert sind, durch $SSM_r := \{SS_a : \text{AKT}(SS_a, M_r)\}$

IF Schaltschritteanalyse BEG

COND $SSM_r \neq \emptyset$

THEN

PAR Schaltschrittuntersuchung BEG

EXIT WHEN $SSM_r = \emptyset$

AGENT _{$\ddot{u},1$}

CALL Schaltschrittausführung

reserviere die Menge SSM_r für einen Zugriff, sofern sie durch keinen anderen Zugriff temporär reserviert ist

wähle einen Schaltschritt SS_a aus der Menge SSM_r aus

eliminiere den Schaltschritt
 SS_a aus der Menge SSM_r durch
 $SSM_r := SSM_r - \{SS_a\}$

gib die Menge SSM_r für
 andere Zugriffe frei

AGENT_{ü,u} [dito für alle $u \in \{2, \dots, AU_{\bar{u}}-1\}$]

AGENT_{ü,AU \bar{u}}

CALL Schaltschrittausführung

reserviere die Menge SSM_r für
 einen Zugriff, sofern sie
 durch keinen anderen Zugriff
 temporär reserviert ist

wähle einen Schaltschritt SS_a
 aus der Menge SSM_r aus

eliminiere den Schaltschritt
 SS_a aus der Menge SSM_r durch
 $SSM_r := SSM_r - \{SS_a\}$

gib die Menge SSM_r für
 andere Zugriffe frei

TASK Schaltschrittausführung

SEQ Schaltschrittausführung BEG

$\langle \text{anw}_8 \rangle \approx$ wende die Schaltregel auf
 den Schaltschritt SS_a und
 die Markierung M_r an zur
 Berechnung der Folgemarkierung M_f
 mit $M_r[SS_a]M_f$

CASE Graphaktualisierung BEG

WHEN $M_f \in KN_{RG}$

SEQ alter_Knoten BEG

$\langle \text{anw}_9 \rangle \approx$ erzeuge die Kante $ka_{r,a,f}$
 mit $ka_{r,a,f} = (M_r, M_f)$

PAR alter_Knotenteil BEG

TASK

$\langle \text{anw}_{10} \rangle \approx \dots$
 erweitere die
 Kantenmenge KA_{RG}
 um die Kante $ka_{r,a,f}$ durch
 $KA_{RG} := KA_{RG} \cup \{ka_{r,a,f}\}$

TASK

<anw₁₁> ≈ ...
 erweitere die
 Erfüllungsmenge BK der
 Beschriftungsfunktion bk
 um das Paar aus
 der Kante $ka_{r,a,f}$
 und ihrer Beschriftung
 $bk(ka_{r,a,f}) = SS_a$ durch
 $BK := BK \cup \{(ka_{r,a,f}, SS_a)\}$

PAR alter_Knotenteil END

SEQ alter_Knoten END

WHEN $M_f \notin KN_{RG}$

SEQ neuer_Knoten BEG

<anw₁₂> ≈ erzeuge die
 Kante $ka_{r,a,f}$ mit
 $ka_{r,a,f} = (M_r, M_f)^{47}$

PAR neuer_Knotenteil BEG

TASK

<anw₁₃> ≈ ...
 erweitere die
 Kantenmenge KA_{RG} um
 die Kante $ka_{r,a,f}$ durch
 $KA_{RG} := KA_{RG} \cup \{ka_{r,a,f}\}$

TASK

<anw₁₄> ≈ ...
 erweitere die
 Erfüllungsmenge BK der
 Beschriftungsfunktion bk
 um das Paar aus der
 Kante $ka_{r,a,f}$ und
 ihrer Beschriftung
 $bk(ka_{r,a,f}) = SS_a$ durch
 $BK := BK \cup \{(ka_{r,a,f}, SS_a)\}$

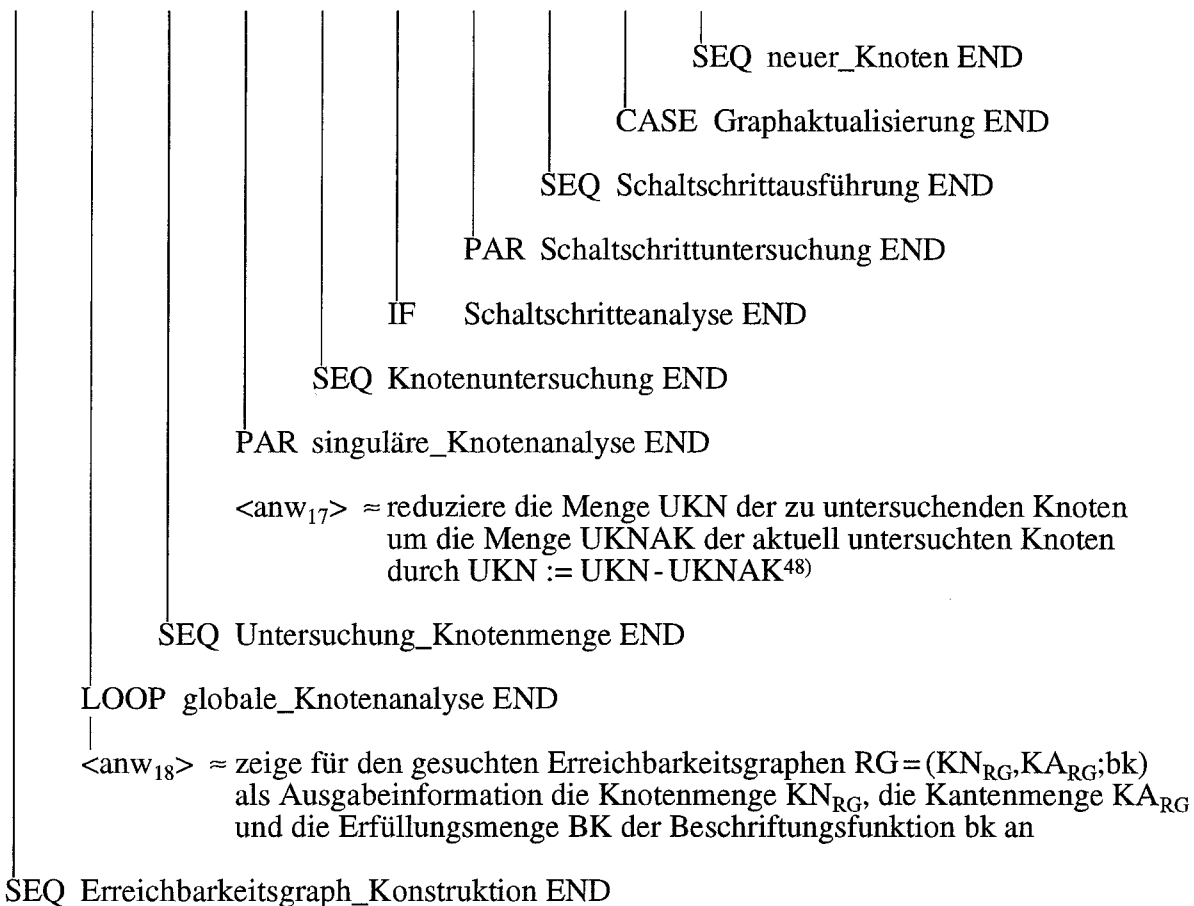
TASK

<anw₁₅> ≈ ...
 erweitere die
 Knotenmenge KN_{RG}
 um den Knoten M_f durch
 $KN_{RG} := KN_{RG} \cup \{M_f\}$

TASK

<anw₁₆> ≈ ...
 erweitere die
 Menge UKN der zu
 untersuchenden Knoten
 um den Knoten M_f durch
 $UKN := UKN \cup \{M_f\}$

PAR neuer_Knotenteil END



Die Nebenläufigkeit des voranstehenden Konstruktionsalgorithmus für die Erreichbarkeitsgraphen von Netzen läßt sich wiederum mit Hilfe des Petrinetz-Konzepts verdeutlichen. Dies unterstreicht die schon mehrfach angesprochene Selbstbezüglichkeit des Petrinetz-Konzepts: Das Verhaltenspotential eines Netzes wird zunächst durch seinen Erreichbarkeitsgraphen vollständig expliziert. Der Algorithmus, mit dessen Hilfe sich dieser Erreichbarkeitsgraph konstruieren läßt, kann wiederum als ein Netz dargestellt werden. Das algorithmusrepräsentierende Netz beschreibt, wie die Repräsentation der dynamischen Struktur eines anderen Netzes durch die Konstruktion seines Erreichbarkeitsgraphen aufgebaut wird. Innerhalb des Petrinetz-Konzepts ist es also möglich, die Untersuchung der Dynamik von Netzen, die auf der Objektebene durch die Konstruktion von Erreichbarkeitsgraphen geschieht, auf einer konzeptionellen Metaebene wiederum mit der Hilfe von Netzen abzubilden. Die Netze der Metaebene, die den Konstruktionsalgorithmus für Erreichbarkeitsgraphen von Netzen der Objektebene wiedergeben, werden fortan als algorithmische Netzmodelle thematisiert.

Zunächst wird ein Netzmodell des Konstruktionsalgorithmus präsentiert, dessen Informationsgehalt über die Pseudocode-Notation des Algorithmus - abgesehen von einer technischen Besonderheit⁴⁹⁾ - noch nicht hinausreicht. Aber die graphische Visualisierung des algorithmischen Netzmodells läßt die interne Struktur des Konstruktionsalgorithmus, vor allem seine Komposition aus hierarchisch verfeinerten Modulen, besonders transparent hervortreten. Später wird eine modifizierte Netzrepräsentation des Konstruktionsalgorithmus vorgelegt, die neue Einsichten in seine inhärente Nebenläufigkeit⁵⁰⁾ vermittelt. Dies wird zu einer Modifizierung des Algorithmus führen, die sein Nebenläufigkeitspotential konsequenter ausschöpft als die oben präsentierte Pseudocode-Darstellung⁵¹⁾.

Diese letztlich angestrebte, im folgenden schrittweise entfaltete Modifizierung des Konstruktionsalgorithmus für Erreichbarkeitsgraphen demonstriert die heuristische Kraft des Petrinetz-Konzepts⁵²⁾. Sie beruht darauf, daß durch die Rekonstruktion eines Objekts - hier eines

Algorithmus - in der Gestalt eines objektabbildenden Netzmodells interessante Einblicke in die Objektkonstitution - wie z.B. in das Nebenläufigkeitspotential eines Algorithmus - gewonnen werden können. Dies gilt insbesondere dann, wenn das Verhalten der untersuchten Erkenntnisobjekte durch ein unübersichtliches Zusammenwirken von Subobjekten geprägt ist, die teils kausal voneinander abhängen und teils nebenläufig operieren⁵³). Dann kann das Petrinetz-Konzept seine Stärke zur Geltung bringen, sowohl kausal Dependenz⁵⁴) als auch Nebenläufigkeit⁵⁵) einerseits präzise zu erfassen und andererseits anschaulich zu repräsentieren. Die präzise Erfassungsmöglichkeit begründet das analytische Erkenntnispotential des Petrinetz-Konzepts. Die transparente Repräsentation von Netzmodellen durch ihre graphische Visualisierung unterstützt vor allem die heuristische Erkenntnisbefruchtung⁵⁶). Der letzte Aspekt wird im folgenden durch ein Netzmodell des Konstruktionsalgorithmus für Erreichbarkeitsgraphen beispielhaft verdeutlicht.

Anknüpfungspunkt ist die oben benutzte Pseudocode-Notation für Algorithmen. Für jedes seiner Pseudocode-Module wird eine Netzdarstellung - ein Netzmodul - vorgelegt. Es wird hier nicht beabsichtigt, das Netzmodell des Konstruktionsalgorithmus so präzise zu formulieren, daß das algorithmische Netzmodell direkt implementiert werden könnte. Vielmehr interessiert es, einen tieferen Einblick in die Konstitution des Konstruktionsalgorithmus, vor allem in seine inhärente Nebenläufigkeit zu gewinnen. Daher beruhen die Netzdarstellungen der Pseudocode-Module "nur" auf Kanal/Instanz-Netzen. Ihre flexible Definition erlaubt es, sich auf die erkenntnisstimulierende graphische Visualisierung der Netzmodule zu konzentrieren⁵⁷). Dabei werden die jeweils wichtig erscheinenden Aspekte der algorithmischen Konstruktion von Erreichbarkeitsgraphen berücksichtigt. Diese graphische Fokussierungsleistung von Kanal/Instanz-Netzen wurde bereits an früherer Stelle gewürdigt⁵⁸). Wegen der hohen Selbsterklärungsfähigkeit von Kanal/Instanz-Netzen werden die Netzmodule - mit einer anschließend dargelegten Ausnahme - nicht im Detail erläutert. Statt dessen werden sie in den Abb. 67 bis 72 zusammen mit den jeweils zugrundeliegenden Pseudocode-Modulen präsentiert⁵⁹). Die graue Schraffur der Ausgangstransition jeder Stelle, die mit "input" beschriftet ist, und die gleiche Schraffur der Eingangstransition jeder korrespondierenden, mit "output" beschrifteten Stelle spielt später bei der hierarchischen Netzvergrößerung und -verfeinerung eine Rolle. Im Vorgriff auf diese hierarchische Moduleinbettung werden die beiden Transitionen auch schon mit zwei Stellen derart verknüpft, daß das Geviert aus Transitionen und Stellen eine Art "Modulrahmen" schafft.

Das Netzmodul, das für eine agentenorientierte parallele Prozedur in Abb. 72 wiedergegeben ist, könnte auf den ersten Blick irritieren. Seine Informationsdichte übertrifft deutliche diejenige der anderen Netzmodule. Es reicht auch inhaltlich über sein zugrundeliegendes Pseudocode-Modul hinaus. Beide Aspekte werden dadurch verursacht, daß im angesprochenen letzten Netzmodul die Synchronisation von nebenläufig operierenden Agenten explizit geleistet wird. Im o.a. Konstruktionsalgorithmus wurde diese Agentensynchronisation durch die Reservierung des Zugriffs auf die Menge UKNAK der aktuell zu untersuchenden Knoten einerseits und durch die analoge Zugriffsreservierung für die Menge SSM_r aktivierter Schaltschritte andererseits angedeutet. Es blieb allerdings offen, wie dieses Synchronisation im einzelnen realisiert werden sollte. Diese Detaillierungslücke wird durch das zuletzt angeführte Netzmodul geschlossen. Zunächst wird bei Beginn der Prozedurausführung eine Agenda errichtet. Sie ist die Menge AG aller Ausgangsinformationen "info", mit denen die auszuführende Prozedur initialisiert werden kann⁶⁰). Jede dieser Ausgangsinformationen wird durch eine Marke dargestellt, welche die Stelle "Agenda" belegt.

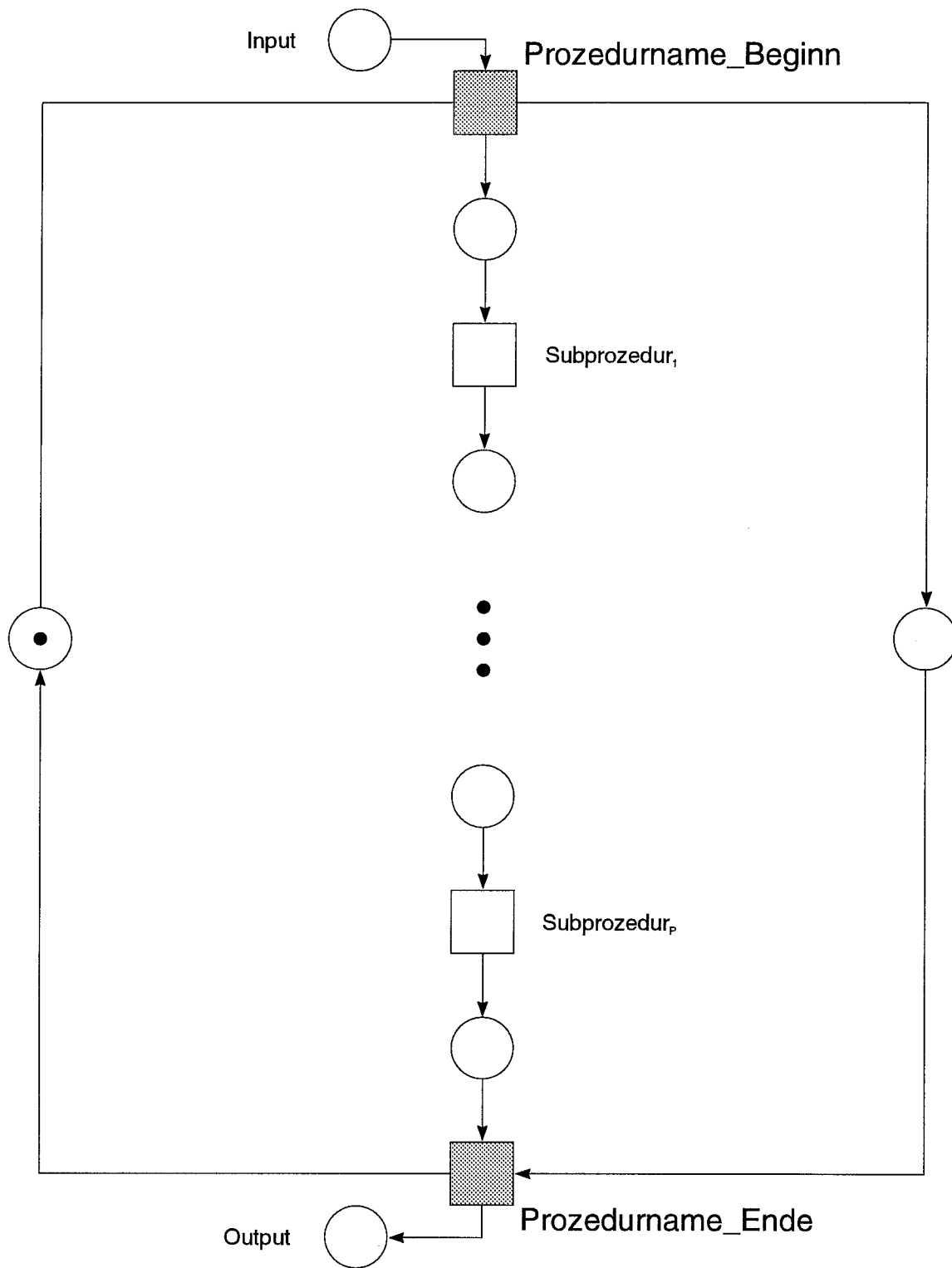


Abb. 67: Netzmodul einer sequentiellen Prozedur

SEQ <Prozedurname> BEG
 |
 <Subprozedur₁>
 ...
 <Subprozedur_p>
 |
 SEQ <Prozedurname> END

Pseudocode-Modul
 der sequentiellen Prozedur

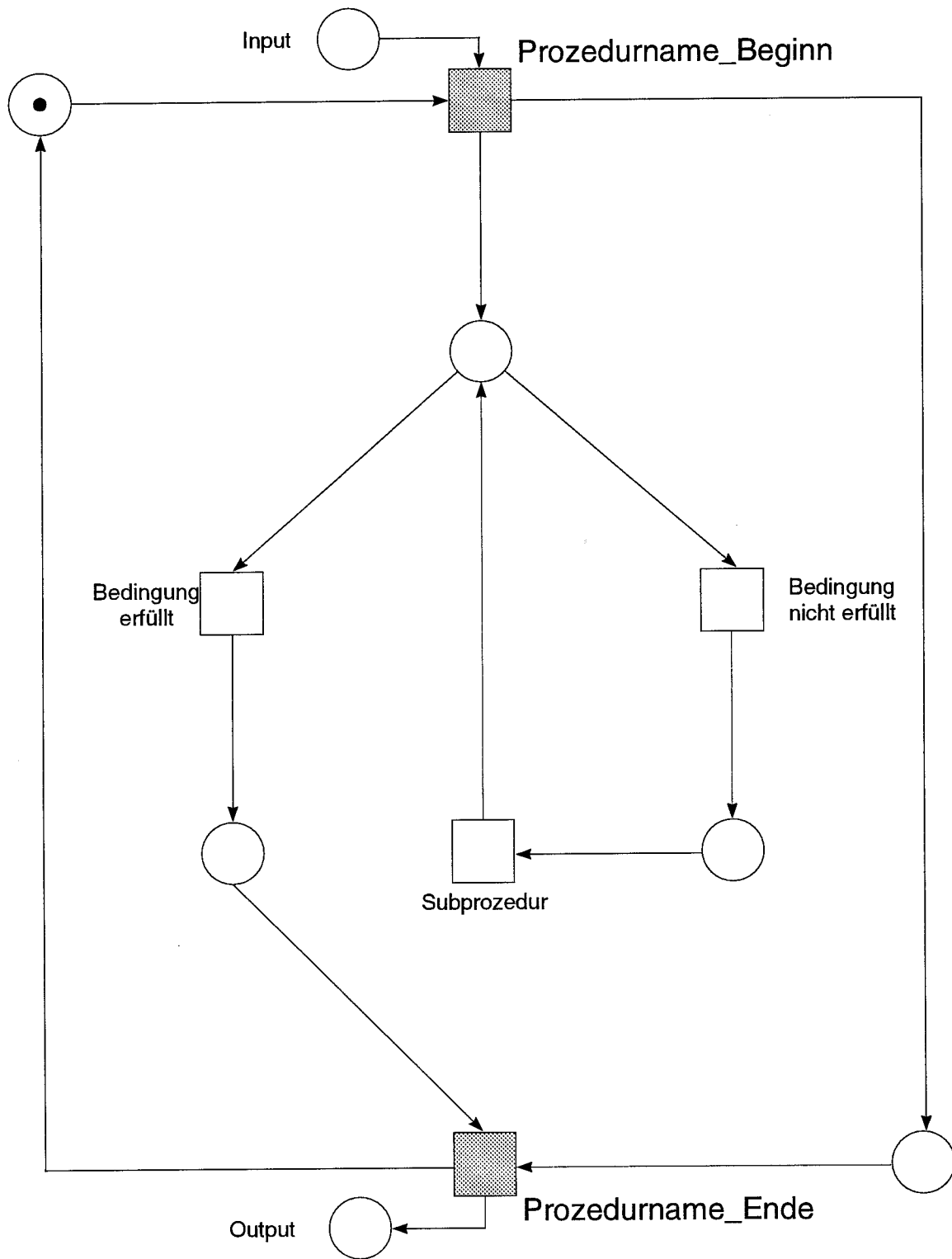


Abb. 68: Netzmodul einer iterativen Prozedur

```

LOOP    <Prozedurname> BEG
|
EXIT WHEN <Bedingung>
|
        <Subprozedur>
|
LOOP    <Prozedurname> END
    
```

Pseudocode-Modul
der iterativen Prozedur

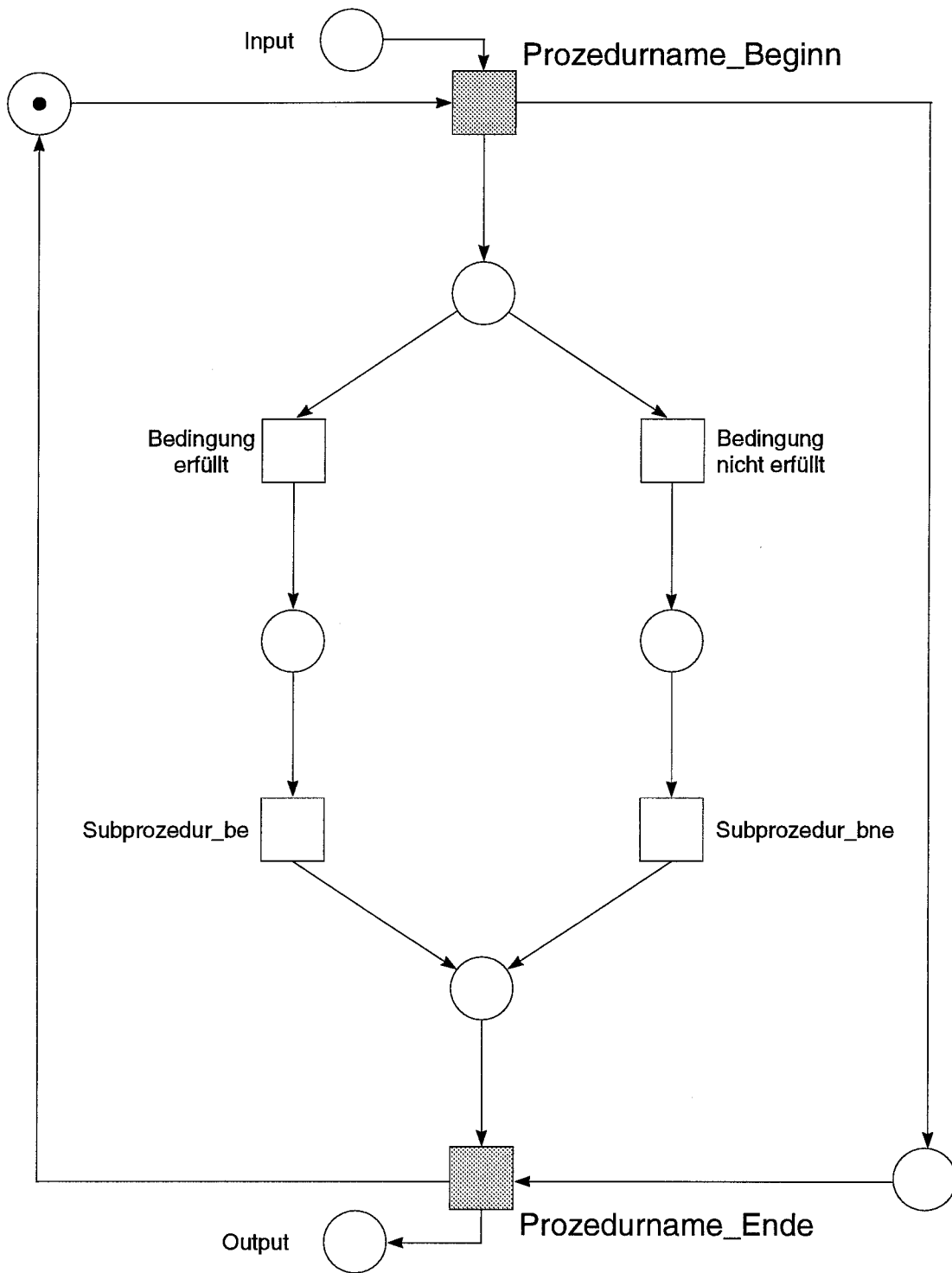


Abb. 69: Netzmodul einer einfachen Fallunterscheidung

```

IF      <Prozedurname> BEG
|
COND   <Bedingung>
|
THEN   <Subprozedur_be>
|
ELSE   <Subprozedur_bne>
|
IF     <Prozedurname> END
    
```

Pseudocode-Modul
der einfachen Fallunterscheidung

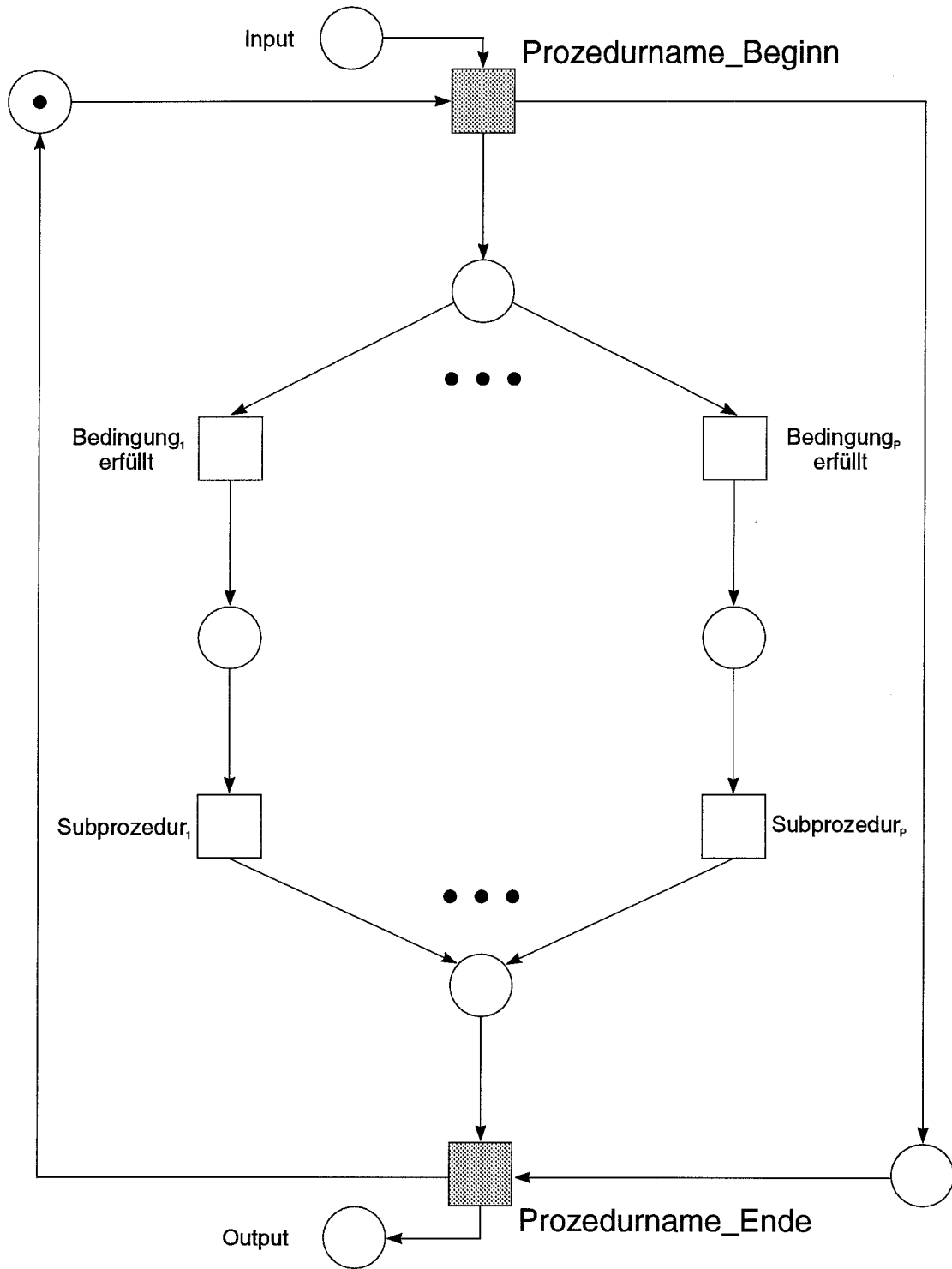


Abb. 70: Netzmodul einer komplexen Fallunterscheidung

```

CASE    <Prozedurname> BEG
|
WHEN    <Bedingung1>
        <Subprozedur1>
|
WHEN    <Bedingungp>
        <Subprozedurp>
|
CASE    <Prozedurname> END
    
```

Pseudocode-Modul
der komplexen Fallunterscheidung

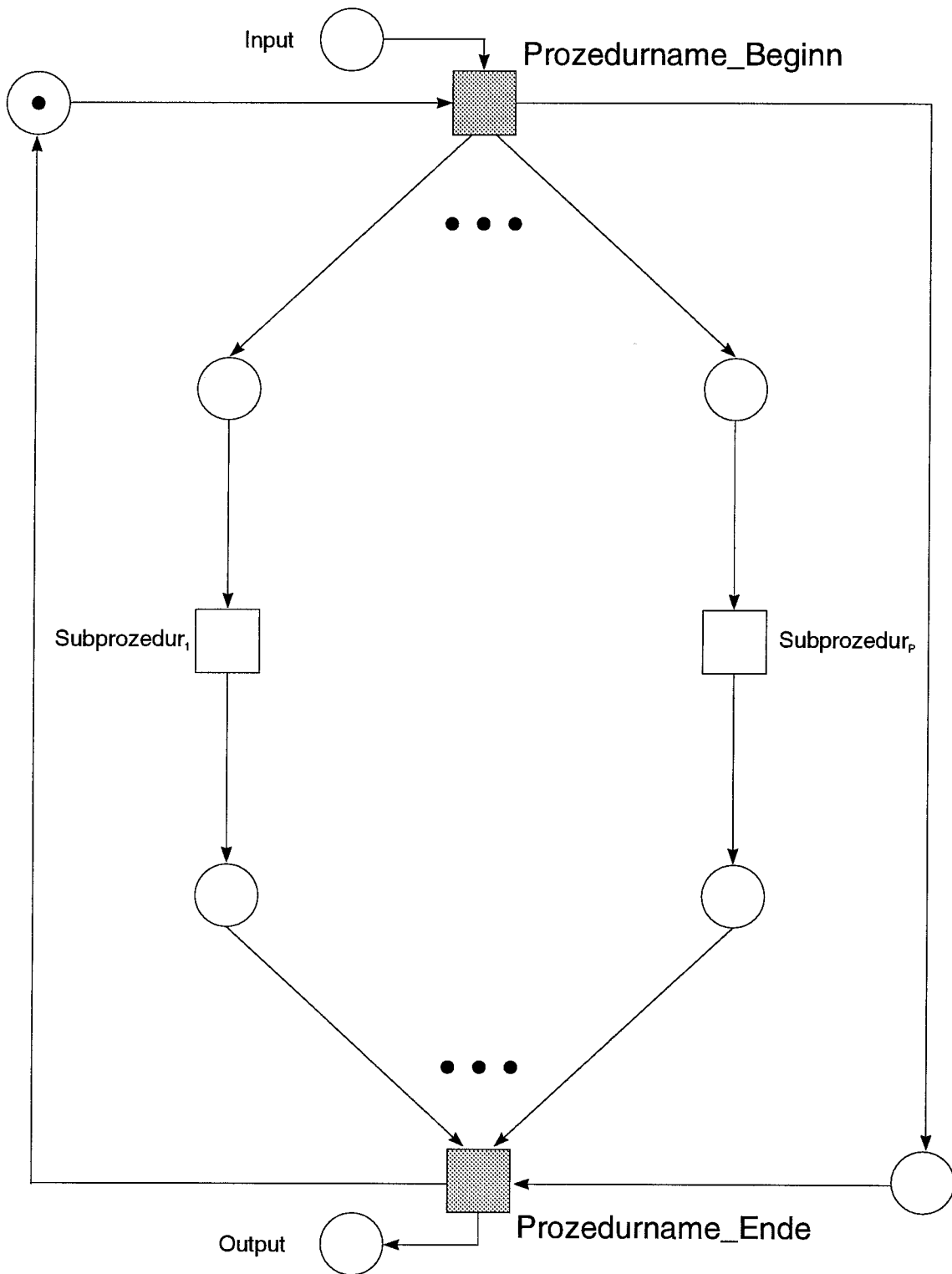


Abb. 71: Netzmodul einer aufgabenorientierten parallelen Prozedur

PAR	<Prozedurname> BEG
TASK	<Subprozedur ₁ >
TASK	...
TASK	<Subprozedur _p >
PAR	<Prozedurname> END

Pseudocode-Modul
der aufgabenorientierten
parallelen Prozedur

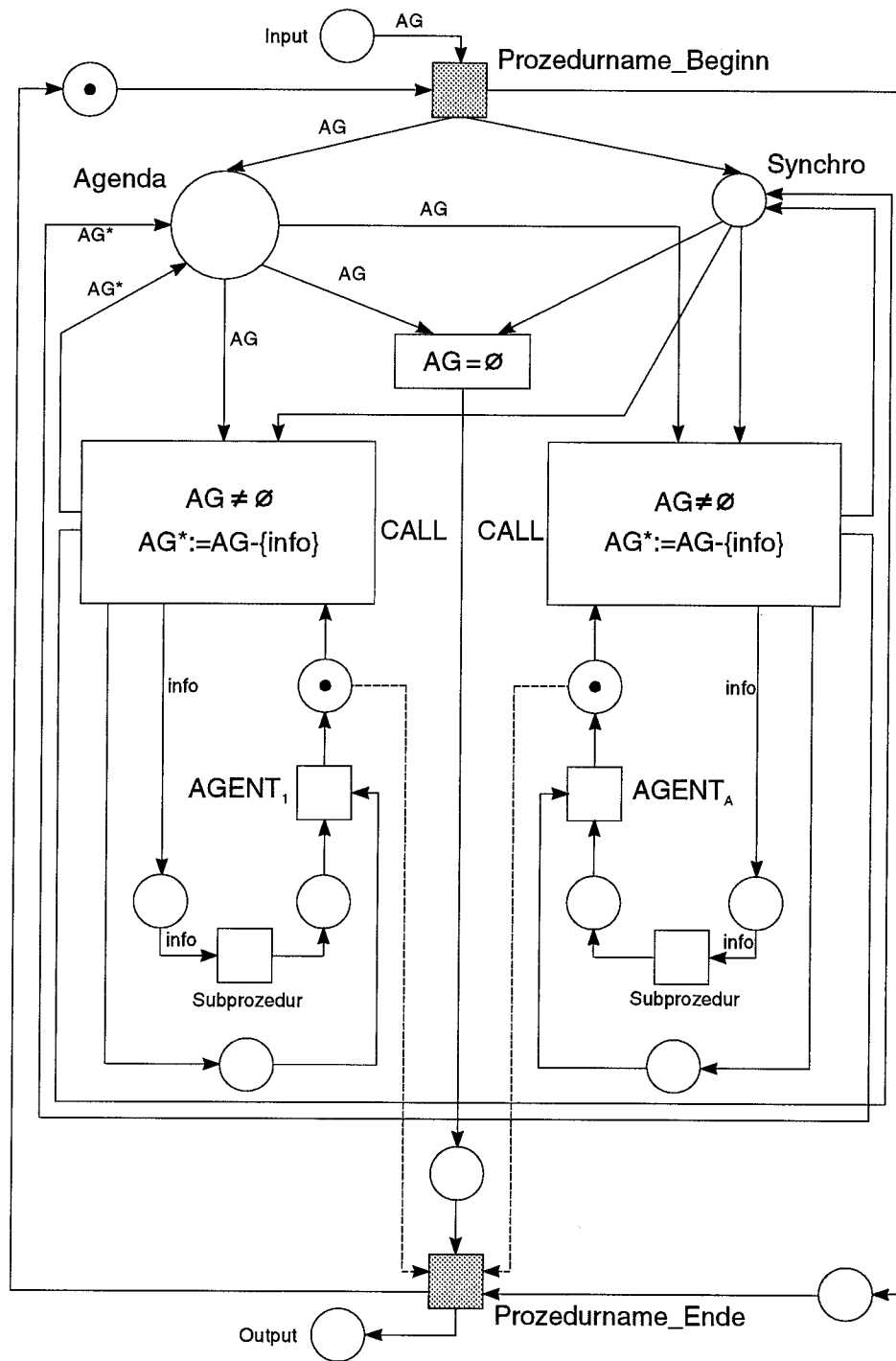


Abb. 72: Netzmodul einer agentenorientierten parallelen Prozedur

```

PAR      <Prozedurname> BEG
|
AGENT1 CALL <Subprozedurname>
|
AGENTA CALL <Subprozedurname>
|
TASK    <Subprozedurname>
|
        <Subprozedur>
|
PAR      <Prozedurname> END
    
```

Pseudocode-Modul
der agentenorientierten
parallelen Prozedur

Solange die Agenda nicht leer ist, können die Agenten diejenigen Subprozeduren ausführen, die zur Erfüllung ihrer Teilaufgaben erforderlich sind. Dabei wählen die Agenten jeweils eine initialisierende Ausgangsinformation "info" aus der Agenda aus und ersetzen die ursprünglich vorhandene Informationsmenge AG *uno actu* durch die entsprechend verkleinerte Informationsmenge AG* mit $AG^* := AG - \{info\}$. Diese Initialisierung der Subprozedurausführung wird für jeden Agenten durch eine Transition "CALL" geleistet. Sie schaltet genau dann, wenn ihr zugehöriger Agent eine Ausführung seiner Subprozedur mit einer Ausgangsinformation "info" übernimmt. Damit die nebenläufig operierenden Agenten niemals dieselbe Ausgangsinformation "info" aus ihrer gemeinsamen Agenda auswählen, werden ihre Agendazugriffe durch die Synchronisationsstelle "Synchro" koordiniert. Ihre Marke kann immer von nur *genau einer* Transition "CALL" abgezogen werden. Durch das Schalten dieser Transitionen wird zugleich die jeweils betroffene Ausgangsinformation "info" aus der Informationsmenge AG der Agenda eliminiert. Danach wird vom Agenten die abgezogene Marke auf die Synchronisationsstelle zurückgelegt. Dann kann die gleiche Subprozedur erneut initialisiert werden, allerdings nur noch mit einer Ausgangsinformation aus der nunmehr reduzierten Informationsmenge der Agenda. Sobald die Informationsmenge AG der Agenda leer ist, wird die Marke von der Synchronisationsstelle abgezogen. Wenn die Agenda leer ist und alle Agenten ihre informationsverarbeitenden Operationen abgeschlossen haben⁶¹⁾, wird die Ausführung der parallelen Prozedur beendet. Auf diese Weise wird die Subprozedur mit jeder Ausgangsinformation genau einmal zunächst initialisiert und anschließend erfüllt. Diese einfache - aber infolgedessen auch besonders übersichtliche - Synchronisationstechnik reicht schon aus, um die nebenläufigen Agentenoperationen zu koordinieren.

Die voranstehend eingeführten Netzmodule wurden grundsätzlich so ausgelegt, daß sie sich hierarchisch verfeinern und M_0 -reversibel präzisieren lassen. Die zweite Moduleigenschaft wird zunächst kurz erläutert, die erste anschließend intensiver gewürdigt.

Die Kanal/Instanz-Netze, mit deren Hilfe die Netzmodule dargestellt sind, besitzen zwar keine wohldefinierte dynamische Struktur⁶²⁾. Daher ist für sie selbst die Eigenschaft der M_0 -Reversibilität nicht definiert⁶³⁾. Aber die Kanal/Instanz-Netze der Netzmodule können durch die Zuordnung einer markenbezogenen algebraisch-prädikatenlogischen Spezifikation so präzisiert werden, daß sie die Gestalt von Synthetischen Netzen⁶⁴⁾ mit wohldefinierter Dynamik annehmen. Falls diese Präzisierung vorgenommen würde⁶⁵⁾, so ließe sich durch eine Erreichbarkeitsanalyse schnell aufzeigen, daß die Netzmodule immer wieder in ihre Ausgangsmarkierung M_0 zurückkehren können. Aufgrund dieser M_0 -Reversibilität ist es möglich, die Netzmodule in ein übergeordnetes Netz derart einzubetten, daß die Netzmodule ihr charakteristisches Schaltverhalten beliebig oft wiederholen können⁶⁶⁾.

Die hierarchische Verfeinerungsmöglichkeit von Netzmodulen wird anschließend aus zwei Gründen ausführlicher behandelt. Erstens bildet sie den konzeptionellen Kern, aus dem heraus das später präsentierte Netzmodell für den o.a. Konstruktionsalgorithmus entwickelt wurde. Zweitens spielen Netzverfeinerungen - und die komplementären Netzvergrößerungen - eine große Rolle bei der praktischen Verwendung des Petrinetz-Konzepts zur Modellierung von realen Problemen⁶⁷⁾. Denn diese Problemstellungen fallen im allgemeinen so komplex aus, daß sich ihre Modellierung bislang nur mittels der schrittweisen Modellverfeinerung beherrschen läßt⁶⁸⁾. Diese Arbeit argumentiert zwar nur vor dem *Hintergrund* der Modellierung von Realproblemen, ohne die Problemmodellierung selbst im Detail auszuführen. Aber bereits das Grobmodell für die Koordinierung Flexibler Fertigungssysteme, das später exemplarisch angeführt wird, fällt so umfangreich aus, daß auch dort mit der Technik sukzessiv verfeinerter Netzmodelle gearbeitet wird.

Die Verfeinerung von Netzen wird hier⁶⁹⁾ auf die Substitution stellenberandeter, grob strukturierter Teilnetze durch verfeinerte Teilnetze mit jeweils identischer Stellenberandung zurückgeführt. Der Rand eines Netzes ist die Menge aller seiner stellen- und transitionsartigen Knoten, die jeweils *entweder* einen leeren Vor- *oder aber* einen leeren Nachbereich besitzen⁷⁰⁾. Die

Menge aller anderen Knoten, die sowohl einen nicht-leeren Vor- als auch einen nicht-leeren Nachbereich aufweisen, bildet das Netzzinnere⁷¹⁾. Ein Netz heißt genau dann stellenberandet⁷²⁾, wenn zwei Anforderungen erfüllt sind:

- Der Netzrand ist nicht die leere Menge⁷³⁾.
- Alle Elemente des Netzrandes sind Stellen⁷⁴⁾.

Die Netzmodule erfüllen beide Postulate. Ihr Netzrand besteht jeweils aus den zwei Stellen⁷⁵⁾, die entweder mit "input" oder "output" beschriftet sind⁷⁶⁾. Es handelt sich hierbei - im doppelten Sinne - um die beiden Schnittstellen, über die sich ein Netzmodul in ein umfassenderes Netz einbetten läßt. Denn die Marken, die sich auf den In- und Outputstellen eines Netzmoduls befinden, repräsentieren Ein- bzw. Ausgabeinformationen der jeweils modellierten Pseudocode-Module.

Um zu veranschaulichen, wie sich die stellenberandeten Netzmodule zum Zwecke der Netzverfeinerung einsetzen lassen, wird zunächst gedanklich der umgekehrte Weg einer Netzvergrößerung beschrieben. Dabei wird auf die bereits oben präsentierte Visualisierung der graphisch repräsentierten Netzmodule zurückgegriffen. Dort wurden in jedem Netzmodul die Ausgangstransition seiner Inputstelle und die Eingangstransition seiner Outputstelle durch eine graue Schraffur hervorgehoben. Diese Darstellungsweise sollte die Möglichkeit verdeutlichen, ein Netzmodul so zu vergrößern, daß es nur noch aus seiner Input- und seiner Outputstelle sowie genau einer dazwischen liegenden Transition besteht. Diese Transition ist über genau eine Eingangskante mit der Input- und vermittels genau einer Ausgangskante mit der Outputstelle verknüpft. Sie wird auch als Makrotransition bezeichnet. Alle Transitionen aus dem zugrundeliegenden, noch nicht vergrößerten Netzmodul lassen sich dagegen als Mikrotransitionen ansprechen. Das Schaltverhalten der Makrotransition wird mit demjenigen Markenfluß identifiziert, den die Mikrotransitionen im ursprünglichen Netzmodul durch ihr Schalten hervorbringen können, sobald die Inputstelle des Netzmoduls markiert ist. Das Netzmodul spezifiziert daher das komplexe Schaltverhalten der einen Makrotransition, die aus der Netzvergrößerung hervorgegangen ist.

Allerdings muß bei der Spezifizierung dieses Schaltverhaltens beachtet werden, daß die Schaltakte der Makrotransition - wie die Schaltakte aller anderen Transitionen - *atomare* Ereignisse darstellen. Daher kann die Makrotransition nur entweder als Ganzes schalten - oder aber überhaupt nicht. Diese Anforderung scheint trivial zu sein. Tatsächlich ist es aber leicht möglich, Netzmodule für die Verhaltensspezifizierung einer Makrotransition zu konstruieren, die der Atomaritätsanforderung nicht gerecht werden. Beispielsweise läßt sich vorstellen, daß die Inputstelle eines Netzmoduls von mehreren Marken belegt ist oder mehrfach unmittelbar nacheinander markiert wird. Dann ist es möglich, daß die Ausgangstransition dieser Inputstelle mehrfach nacheinander schaltet, obwohl die Eingangstransition der Outputstelle noch nicht geschaltet hat. Dies entspräche einem mehrfachen Schaltbeginn der Makrotransition ohne entsprechendes Schaltende. Durch dieses Auseinanderklaffen von Schaltbeginn und Schaltende hätte die Makrotransition ihren atomaren Charakter verloren. Um solche Atomaritätsverletzungen grundsätzlich auszuschließen, wurde jedes Netzmodul von vornherein so konstruiert, daß die Ausgangstransition seiner Inputstelle und die Eingangstransition seiner Outputstelle durch zwei zusätzliche Stellen in der Art eines "Modulrahmens" verknüpft wurden. Es handelt sich um eine weitere Variante von Synchronisationsstellen. Sie garantieren, daß nach jedem Schalten der Ausgangstransition der Inputstelle zunächst die Eingangstransition der Outputstelle schalten muß, bevor die Ausgangstransition der Inputstelle erneut schalten kann⁷⁷⁾. Darüber hinaus sorgt die Ausgangsmarkierung der Synchronisationsstellen dafür, daß als erstes immer die Ausgangstransition der Inputstelle schalten darf.

Schließlich liefert ein Netzmodul nur dann eine korrekte Spezifizierung des Schaltverhaltens einer Makrotransition, wenn vier weitere Bedingungen erfüllt sind. Die Zusatzbedingungen betreffen nur das Modulinnere, das verbleibt, nachdem die Input- und die Outputstelle des Netz-

moduls samt ihrer adjazenten Kanten ausgeblendet worden sind. Erstens muß das Restmodul einen wohldefinierten Ausgangszustand besitzen. Er zeichnet sich dadurch aus, daß es sich um den einzigen zulässigen Zustand des Restmoduls handelt, von dem aus das Schaltverhalten des Netzmoduls durch Markieren seiner Inputstelle angestoßen werden kann⁷⁸). Diesen wohldefinierten Ausgangszustand bildet die Ausgangsmarkierung M_0 des Restmoduls. Sie findet sich jeweils in den Abbildungen der Netzgraphiken, die kurz zuvor für die oben eingeführten Netzmodule präsentiert wurden⁷⁹). Zweitens muß der wohldefinierte Ausgangszustand des Restmoduls wiederhergestellt werden, sobald das Netzmodul sein Schaltverhalten durch Markieren seiner Outputstelle abschließt. Denn nur so ist sichergestellt, daß das Netzmodul beliebig oft nacheinander aufgerufen werden kann und dabei den Schalttakt der Makrotransition in identischer Weise verfeinert ausführt⁸⁰). Diese gesicherte Rückkehrmöglichkeit zum Ausgangszustand liegt genau dann vor, wenn sich das Restmodul M_0 -reversibel verhält⁸¹). Drittens darf das Restmodul keine Deadlockmarkierung enthalten. Dadurch wird ausgeschlossen, daß das Netzmodul sein Schaltverhalten einstellt, bevor es durch Markieren seiner Outputstelle in der erwünschten Weise beendet worden ist. Viertens darf das Restmodul auch keine andere Livelockmarkierung als seine Ausgangsmarkierung besitzen⁸²). Diese eingeschränkte Livelockfreiheit läßt nur solche zyklischen Schaltprozesse zu, die im Restmodul von seiner Ausgangsmarkierung ausgehen und dorthin auch wieder zurückkehren. Durch den Ausschluß aller weiteren Livelocks wird verhindert, daß das Restmodul in zyklischer Weise unendlich lange weiterschaltet, ohne jemals in seinen Ausgangszustand zurückzukehren. Die voranstehenden Erläuterungen unterstreichen, daß dynamische Netzeigenschaften wie M_0 -Reversibilität, Deadlock- und eingeschränkte Livelockfreiheit für die korrekte Gestaltung dynamischer Systeme eine herausragende Bedeutung gewinnen können.

Es wird darauf verzichtet, für jedes Netzmodul nachzuweisen, daß es die vorgenannten vier Korrektheitsbedingungen erfüllt. Die Netzmodule sind so übersichtlich strukturiert, daß ihre Korrektheit durch bloße Anschauung nachvollzogen werden kann. Um jedoch eventuell aufkeimendem Zweifel zu begegnen, wird anhand des Netzmoduls für eine einfache Fallunterscheidung (vgl. Abb. 69) exemplarisch aufgezeigt, wie sich seine Korrektheit streng beweisen läßt. Zu diesem Zweck wird das Netzmodul als ein Stelle/Transition-Netz interpretiert, um ihm ein präzise definiertes Schaltverhalten zuzuordnen zu können. Dann lassen sich alle vier Korrektheitsbedingungen untersuchen, indem der Erreichbarkeitsgraph des Restmoduls ausgewertet wird. Das Restmodul des vorgenannten Netzmoduls wird in der nachfolgenden Abb. 73 in Erinnerung gerufen. Anstelle der früheren natürlichsprachlichen Knotenbeschriftungen werden nun formalsprachliche Stellen- und Transitionenbezeichnungen verwendet, die eine einfache Verknüpfung des Restmoduls mit seinem Erreichbarkeitsgraphen erlauben⁸³). Abb. 74 präsentiert den Erreichbarkeitsgraphen für das Restmodul. Die Analyse dieses Erreichbarkeitsgraphen ergibt:

- Die Transition t_B ist die Ausgangstransition der Inputstelle, die zum Netzmodul für die einfache Fallunterscheidung gehört. Diese Transition t_B kann ausschließlich unter der Ausgangsmarkierung M_0 des Restmoduls schalten. Dadurch wird die erste Korrektheitsanforderung erfüllt, daß das Restmodul genau einen zulässigen Ausgangszustand besitzen muß.
- Das Restmodul verhält sich M_0 -reversibel, weil von jeder erreichbaren Markierung, die im Erreichbarkeitsgraphen als Markierungsknoten dargestellt ist, zum Knoten der Ausgangsmarkierung M_0 zurückgekehrt werden kann. Folglich trifft auch die zweite Korrektheitsbedingung zu.
- Das Restmodul ist deadlockfrei, weil im Erreichbarkeitsgraphen jeder Markierungsknoten mindestens eine Ausgangskante besitzt. Also wird die dritte Korrektheitsbedingung erfüllt.
- Schließlich wird das Restmodul auch der vierten Korrektheitsbedingung gerecht, weil es keine anderen zyklischen Schaltwege enthält als diejenigen, die von der Ausgangsmarkierung zur Ausgangsmarkierung zurückführen. Deshalb besitzt das Restmodul die Ausgangsmarkierung als einzige - und erwünschte - Livelockmarkierung.

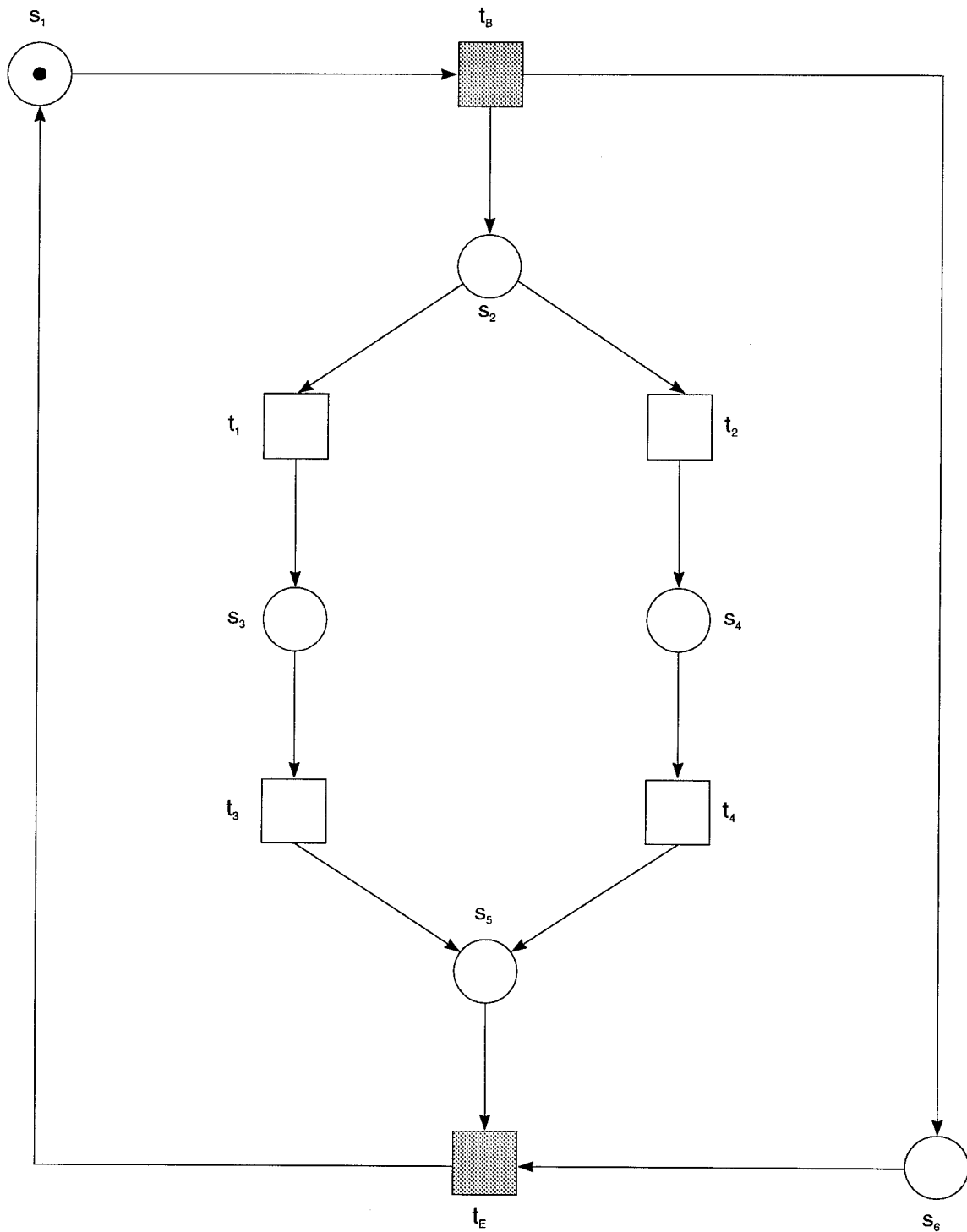


Abb. 73: Restmodul für das Netzmodul einer einfachen Fallunterscheidung

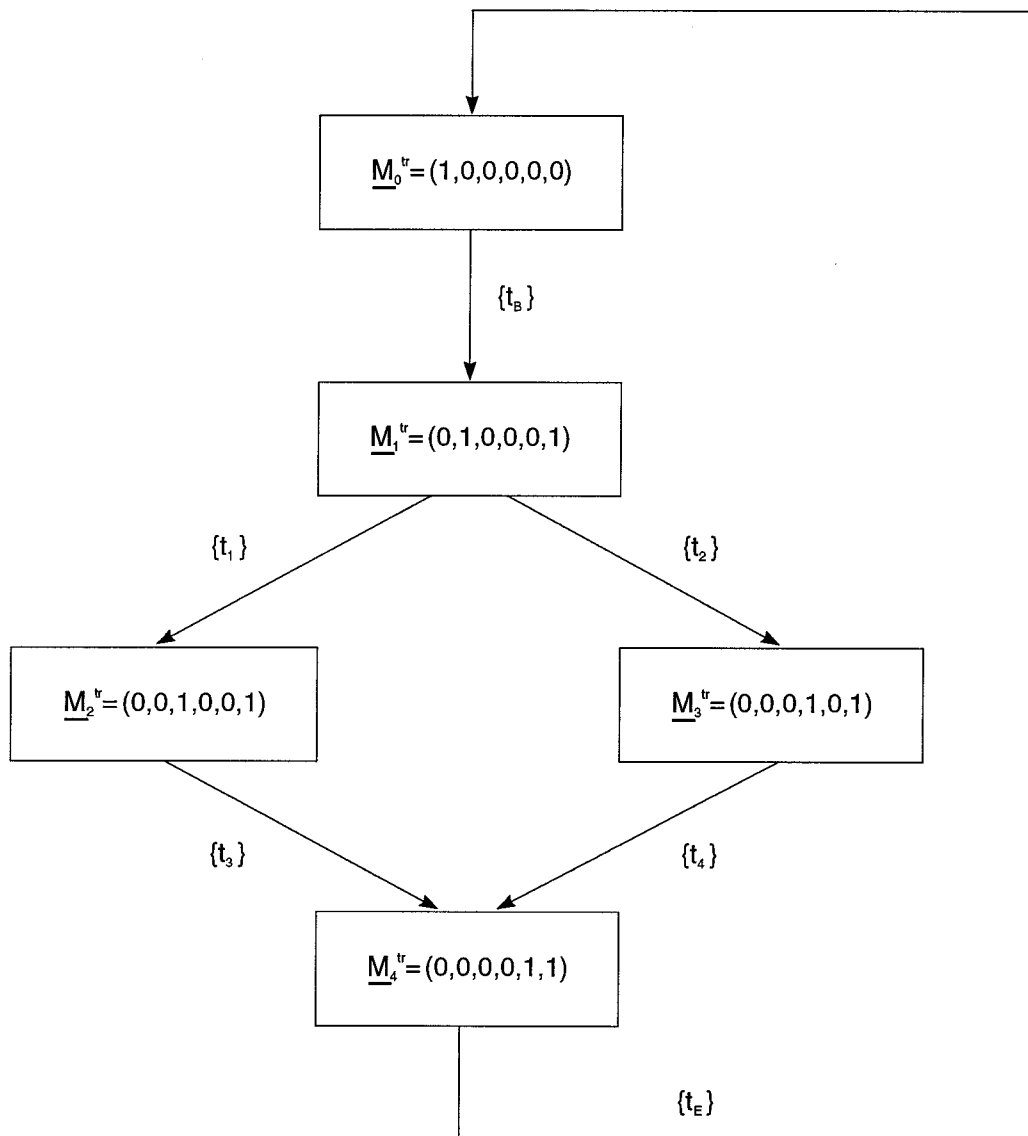


Abb. 74: Erreichbarkeitsgraph für das Restmodul aus Abb. 73

Aufgrund der voranstehenden Erläuterungen liefert jedes Netzmodul eine korrekte Spezifizierung des Schaltverhaltens einer Makrotransition. Diese Makrotransition stellt eine abstrakte Zusammenfassung aller Komponenten des vergrößerten Netzmoduls dar, die sich zwischen der Ausgangstransition seiner Inputstelle und der Eingangstransition seiner Outputstelle erstrecken⁸⁴). Diese beiden Transitionen wurden bereits durch ihre graue Schraffur als Grenzen desjenigen Modulbereichs hervorgehoben, der durch eine Makrotransition abstrahiert werden kann. Entsprechend wird die Makrotransition, die aus einer solchen Modulvergrößerung hervorgeht, durch eine Schattierung ihres graphischen Rechtecksymbols gekennzeichnet. Der "Schatten" der Makrotransition verweist dabei auf das jeweils zugrundeliegende Teilnetz, das mit seinem Netzzinnern das Schaltverhalten der Makrotransition spezifiziert.

Die voranstehend skizzierte Modulvergrößerung läßt sich ebenso in umgekehrter Richtung zur Netzverfeinerung nutzen. Dabei wird in einem Netz eine beliebige Transition betrachtet. Es wird lediglich vorausgesetzt, daß die Transition genau eine Ein- und genau eine Ausgangsstelle besitzt⁸⁵). Diese eine Transition wird als eine Makrotransition behandelt. Das Teilnetz, das diese eine Makrotransition mit ihrer einen Ein- und ihrer einen Ausgangsstelle umfaßt, ist ein stellenberandetes Teilnetz. Es stimmt exakt mit dem zuvor beschriebenen Netz überein, das aus der Vergrößerung eines Netzmoduls hervorgegangen war. Daher ist es möglich, die Vergrößerung umzukehren: Dabei wird das Teilnetz mit der Makrotransition durch ein Netzmodul mit identischem Stellenrand ersetzt. Denn per constructionem fallen die genau eine Ein- und die genau eine Ausgangsstelle der Makrotransition mit der genau einen Input- bzw. der genau einen Outputstelle des Netzmoduls zusammen. Da das stellenberandete Teilnetz durch ein *identisch berandetes* Netzmodul substituiert wird, erfährt die Umgebung des ersetzten Teilnetzes hierdurch keine Beeinflussung⁸⁶). Dafür sorgt die charakteristische Lokalität des Petrinetz-Konzepts. Denn das Teilnetz und seine Umgebung können nur über die gemeinsam geteilten Netzknoten am Rand des Teilnetzes aufeinander einwirken. Da der Stellenrand des Teilnetzes bei dessen Substitution durch das Netzmodul nicht verändert wird, wirkt das Netzmodul nach außen - also auf die Umgebung des ersetzten Teilnetzes - genau so, wie es zuvor für das Teilnetz der Fall war.

Zwar stimmen Teilnetz und Netzmodul an ihren Rändern per constructionem überein. Aber das Innere des Teilnetzes besteht nur aus seiner genau einen Makrotransition. Das Innere des Netzmoduls wird dagegen durch eine Vielfalt von Transitionen, Stellen und adjazenten Kanten geprägt. Daher führt die Ersetzung des stellenberandeten Teilnetzes durch das identisch stellenberandete Netzmodul dazu, die Makrotransition des Teilnetzes zu verfeinern. Es kann daher auch vereinfacht von einer Verfeinerung der Makrotransition gesprochen werden⁸⁷). Aus dieser transitionsbezogenen Verfeinerungsperspektive bleiben die identischen Stellenränder von Teilnetz und Netzmodul unbeachtet. Als (modulares) Subnetz wird dasjenige Restnetz bezeichnet, das vom Netzmodul verbleibt, nachdem sein Stellenrand sowie alle Kanten zwischen dem Stellenrand und dem Netzzinnern - zumindest in gedanklicher Abstraktion - entfernt worden sind. Aus diesem veränderten Blickwinkel wird die Makrotransition durch ein Subnetz ersetzt, das an seiner Oberfläche⁸⁸) nur aus Transitionen besteht. Es handelt sich einerseits um die Ausgangstransition der Inputstelle des Netzmoduls und andererseits um die komplementäre Eingangstransition der Outputstelle des Netzmoduls. Da vom Stellenrand des Netzmoduls abstrahiert wurde, bilden diese beiden Transitionen jetzt die Oberfläche desjenigen Subnetzes, das die Makrotransition verfeinert. Vereinfacht kann auch davon gesprochen werden, daß die Makrotransition durch ein "transitionenberandetes" Subnetz verfeinert worden ist⁸⁹).

Transitionen eines Netzes, die für eine Verfeinerung durch ein transitionsberandetes Subnetz vorgesehen sind, lassen sich in einem Netzmodell von vornherein als Makrotransitionen hervorheben. Zu diesem Zweck wurde oben ihre Schattierung eingeführt. Daher bietet die graphische Visualisierung eines Netzes von vornherein Aufschluß darüber, welche seiner Transitionen bereits vollständig spezifiziert sind⁹⁰) und welche Transitionen später noch verfeinert werden sollen⁹¹).

Die Verfeinerung von Makrotransitionen ist nicht auf den oben vorausgesetzten Fall beschränkt, daß die Makrotransitionen jeweils genau eine Ein- und genau eine Ausgangsstelle besitzen. Diese Voraussetzung erfolgte lediglich im Hinblick auf die früher konstruierten Netzmodule, die jeweils über genau eine korrespondierende Input- bzw. genau eine entsprechende Outputstelle verfügen. Im allgemeinen Fall kann jede Makrotransition durch ein beliebiges Subnetz verfeinert werden, das an seiner Netzoberfläche ausschließlich aus Transitionen besteht. Für ein derart transitionenberandetes Subnetz müssen dann allerdings zwei Aspekte berücksichtigt werden. Erstens ist dafür Sorge zu tragen, daß das Schaltverhalten des Subnetzes nach außen wie der atomare Schaltakt der ersetzten Makrotransition wirkt⁹²⁾. Zweitens sind besondere Konstruktionsregeln erforderlich. Sie müssen sicherstellen, daß die Transitionen aus der Subnetzoberfläche mit den Stellen aus der Nachbarschaft der verfeinerten Makrotransition korrekt verknüpft werden⁹³⁾.

In diesem Kapitel wird an der speziellen Verfeinerungstechnik festgehalten, Makrotransitionen mit jeweils genau einer Ein- und genau einer Ausgangsstelle durch die oben eingeführten stellenberandeten Netzmodule zu verfeinern. Für die Verfeinerung einer Makrotransition bestehen zwei Freiheitsgrade. Erstens kann das stellenberandete Teilnetz, das die Makrotransition enthält, durch verschiedene Netzmodule substituiert werden. Zweitens lassen sich in den eingesetzten Netzmodulen abermals stellenberandete Teilnetze identifizieren, die jeweils aus einer Makrotransition mit ihrer Ein- und ihrer Ausgangsstelle bestehen. Jede dieser modulinternen Makrotransitionen kann ihrerseits wiederum durch ein Netzmodul verfeinert werden usw. Auf diese Weise ist es möglich, eine ursprünglich vorgegebene Makrotransition schrittweise immer weiter zu verfeinern. Verschiedene Makrotransitionen lassen sich dabei unabhängig voneinander und in variierendem Ausmaß verfeinern. Dadurch kann ein ursprünglich vorliegendes Netz in verschiedenen Netzbereichen unterschiedlich intensiv verfeinert werden.

Insgesamt erweist sich die zuvor erläuterte Technik, stellenberandete Teilnetze durch Netzmodule zu ersetzen und dabei Makrotransitionen zu verfeinern, als eine flexible Gestaltungstechnik für die Entwicklung von Netzmodellen. Sie ermöglicht eine schrittweise verfeinernde Netzgestaltung⁹⁴⁾, Verfeinerungsrichtung⁹⁵⁾ und -ausmaß⁹⁶⁾ unterliegen keinen technikimmanenten Restriktionen⁹⁷⁾, sondern können vom Netzgestalter nach eigenem Bedürfnis festgelegt werden. Aufgrund der sukzessiven Netzverfeinerung besitzen die resultierenden Netzmodelle eine hierarchische Strukturierung⁹⁸⁾. Auf jeder Hierarchieebene wird das modellierte Objekt vollständig - aber auf jeweils unterschiedlichem Detaillierungsniveau⁹⁹⁾ abgebildet.

Die hierarchische Verfeinerungstechnik läßt sich mit den oben vorgestellten Netzmodulen so kombinieren¹⁰⁰⁾, daß jeder Algorithmus, der in der eingangs präsentierten Pseudocode-Notation vorliegt, in ein Kanal/Instanz-Netz transformiert wird. Es resultiert dann Netzmodell, daß den zugrundeliegenden Algorithmus in modular und zugleich hierarchisch strukturierter Weise repräsentiert. Ein solches algorithmisches Netzmodell wird anschließend für die Konstruktion der Erreichbarkeitsgraphen von Netzen vorgestellt. Es stimmt mit der früheren Pseudocode-Notation des Konstruktionsalgorithmus für Erreichbarkeitsgraphen im wesentlichen überein. Die einzige Abweichung besteht darin, daß die Synchronisation seiner nebenläufig operierenden Agenten präzisiert wird. Das gesamte algorithmische Netzmodell erstreckt sich über die hierarchisch zusammenhängenden Netzmodule, die in den Abb. 75 bis 88 auf den nachfolgenden Seiten präsentiert werden. Aufgrund seines Umfangs kann das Netzmodell nur auf der obersten Abstraktionsebene in Abb. 75 vollständig abgebildet werden. Alle feineren Detaillierungsniveaus müssen dagegen aus den abgebildeten Teilnetzen, die jeweils eine Makrotransition aus einer hierarchisch übergeordneten, gröberen Netzdarstellung verfeinern, in Gedanken zusammengesetzt vorgestellt werden.

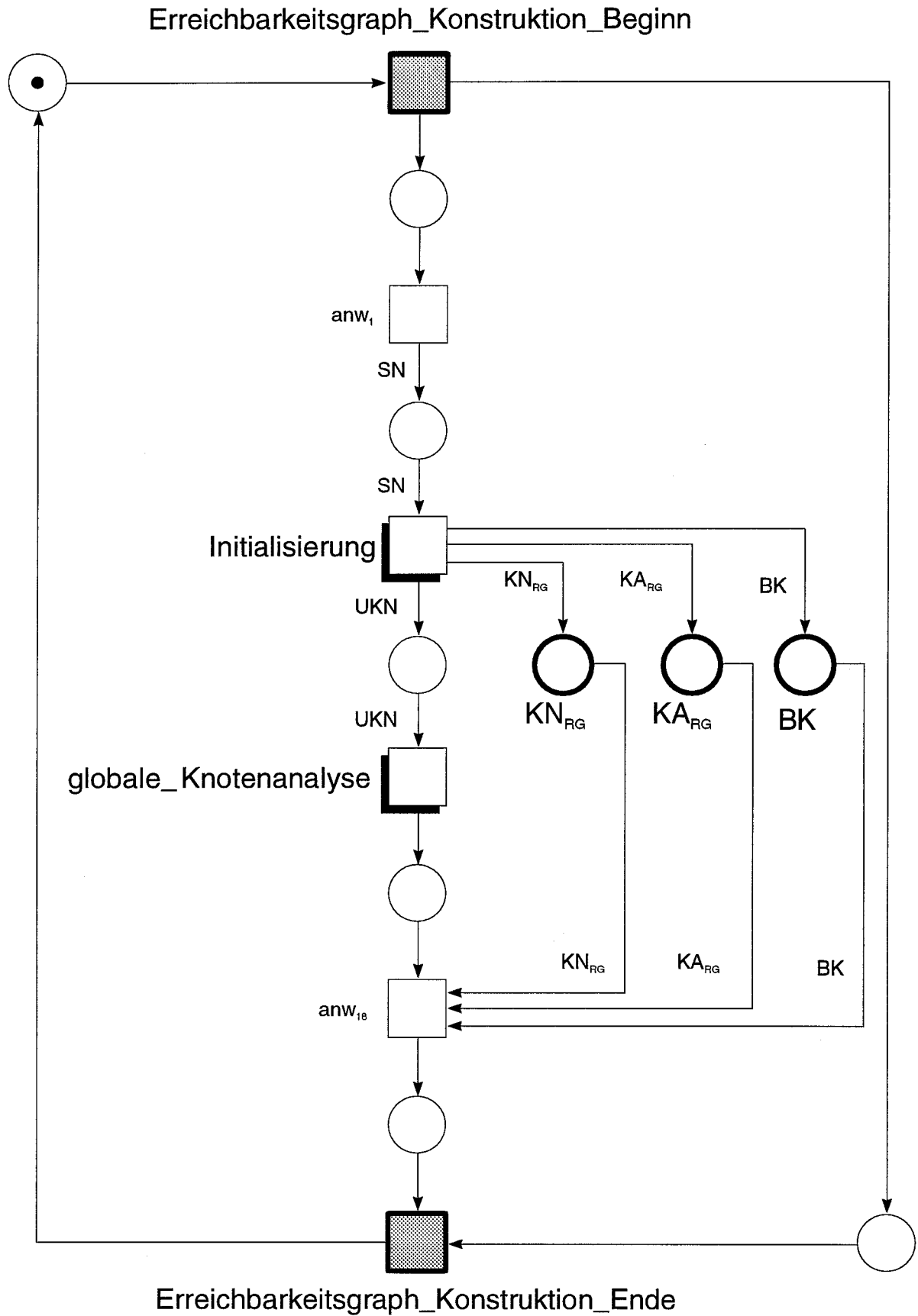


Abb. 75: Netzmodell für einen Algorithmus zur Konstruktion von Erreichbarkeitsgraphen (oberste Abstraktionsebene)

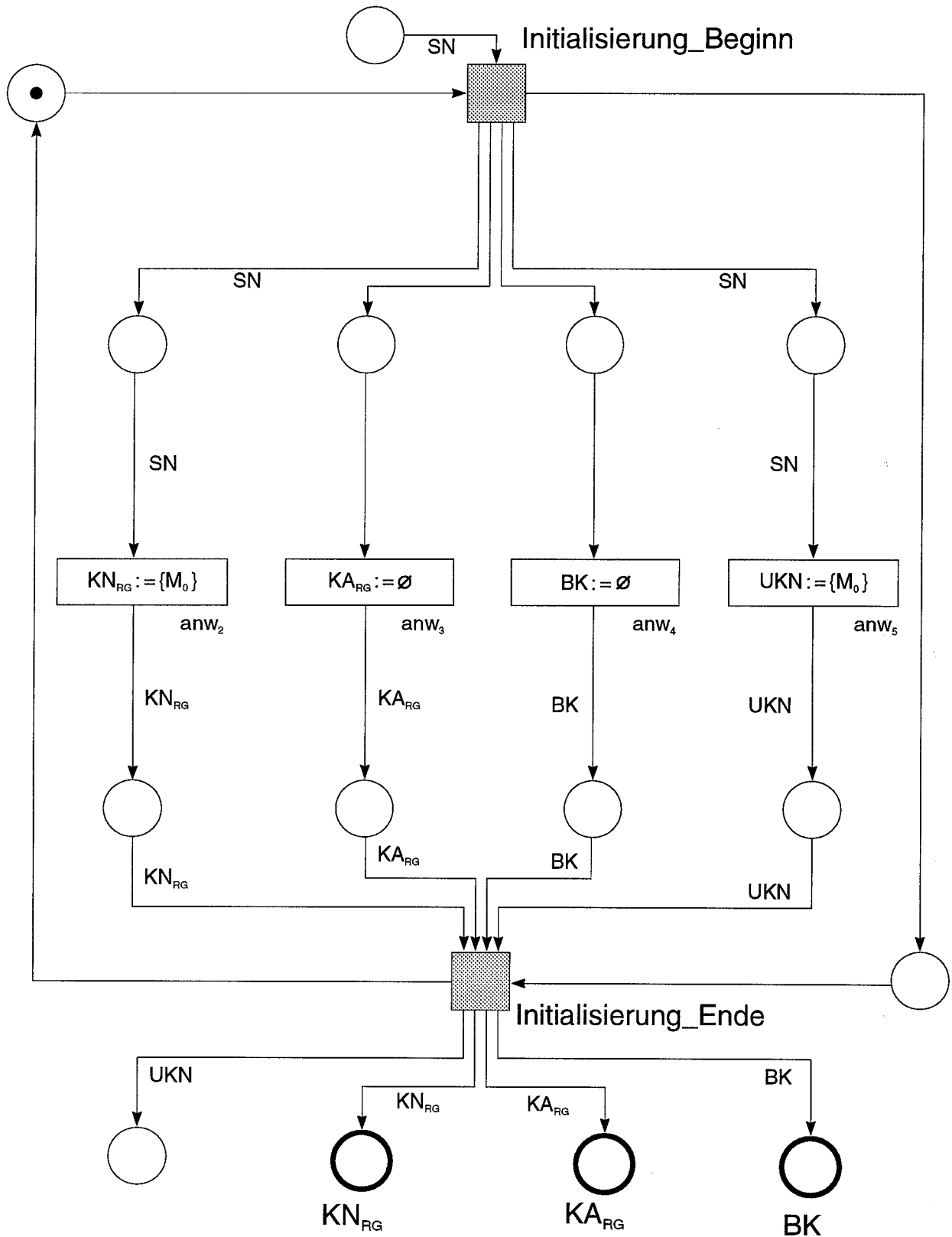


Abb. 76: Verfeinerung für die Makrotransition "Initialisierung"

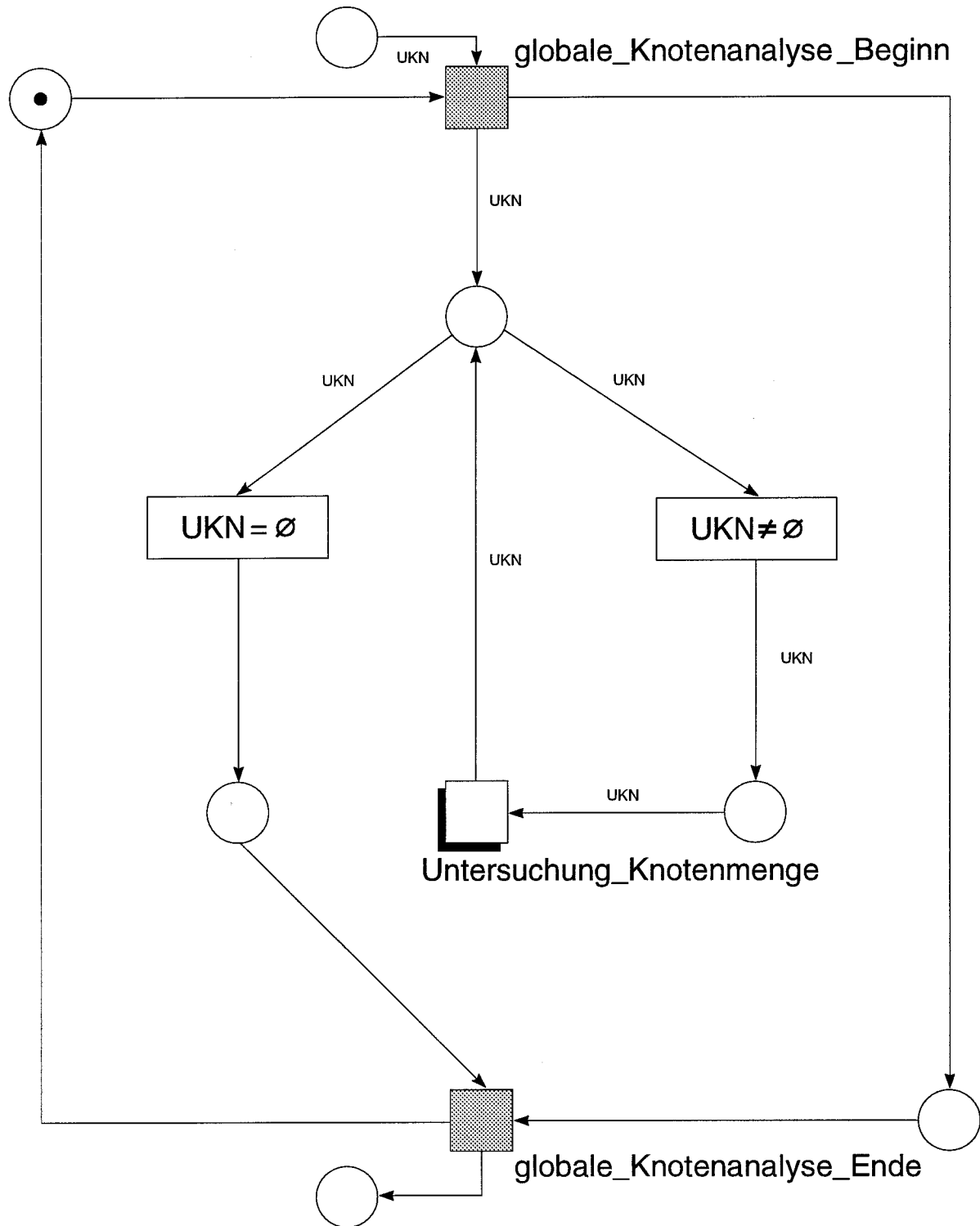


Abb. 77: Verfeinerung für die Makrotransition "globale_Knotenanalyse"

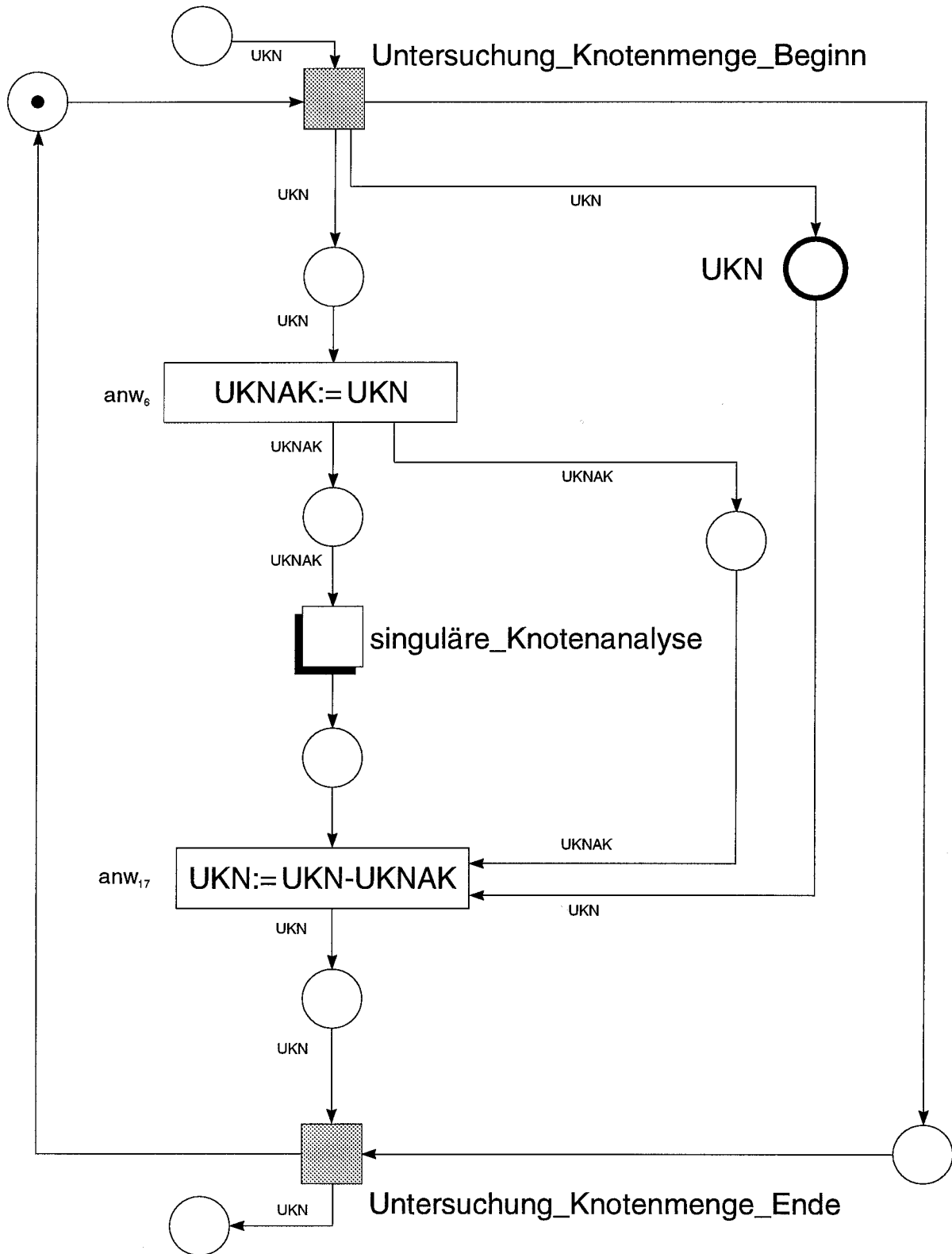


Abb. 78: Verfeinerung für die Makrotransition "Untersuchung_Knotenmenge"

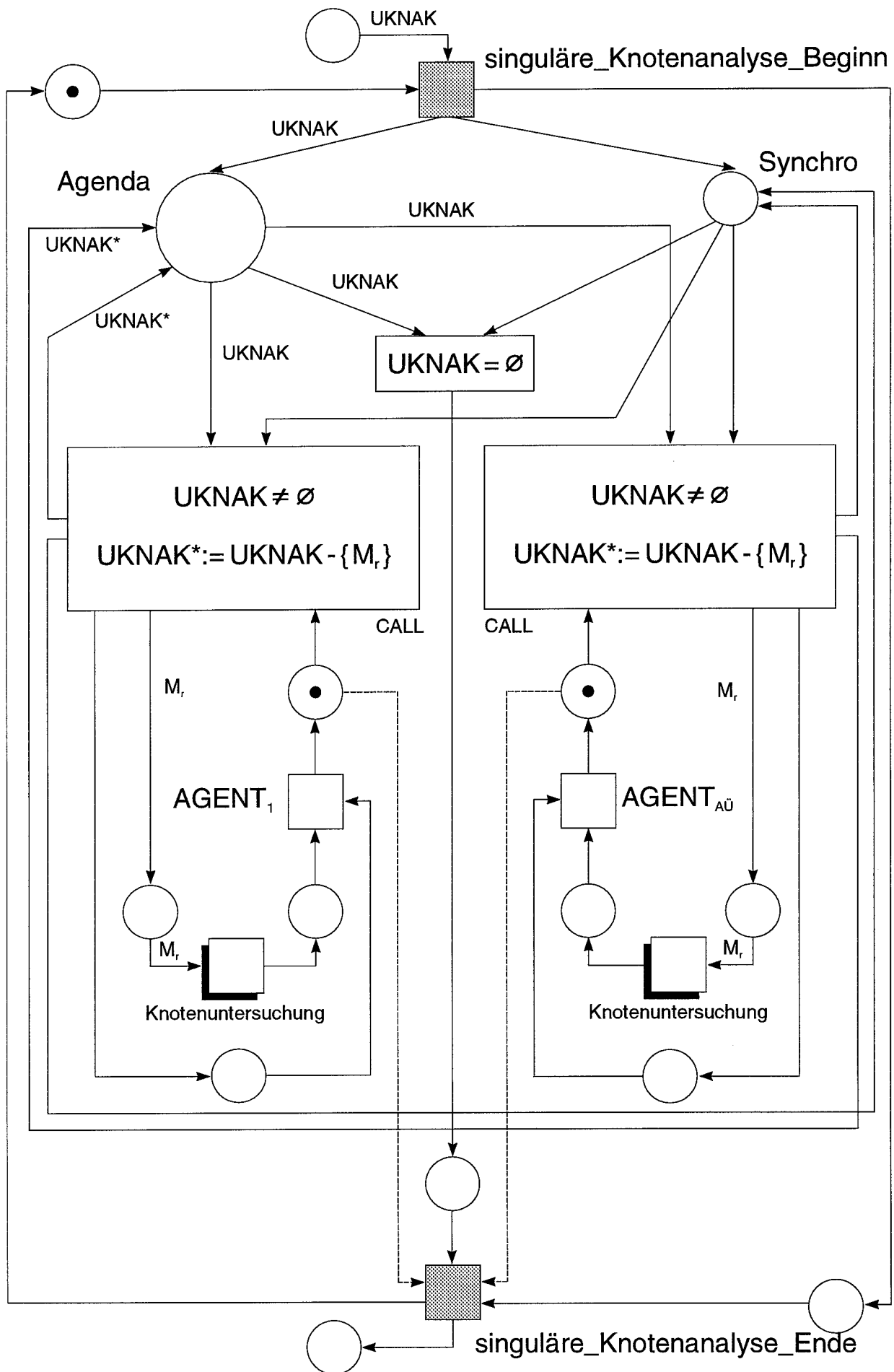


Abb. 79: Verfeinerung für die Makrotransition "singuläre_Knotenanalyse"

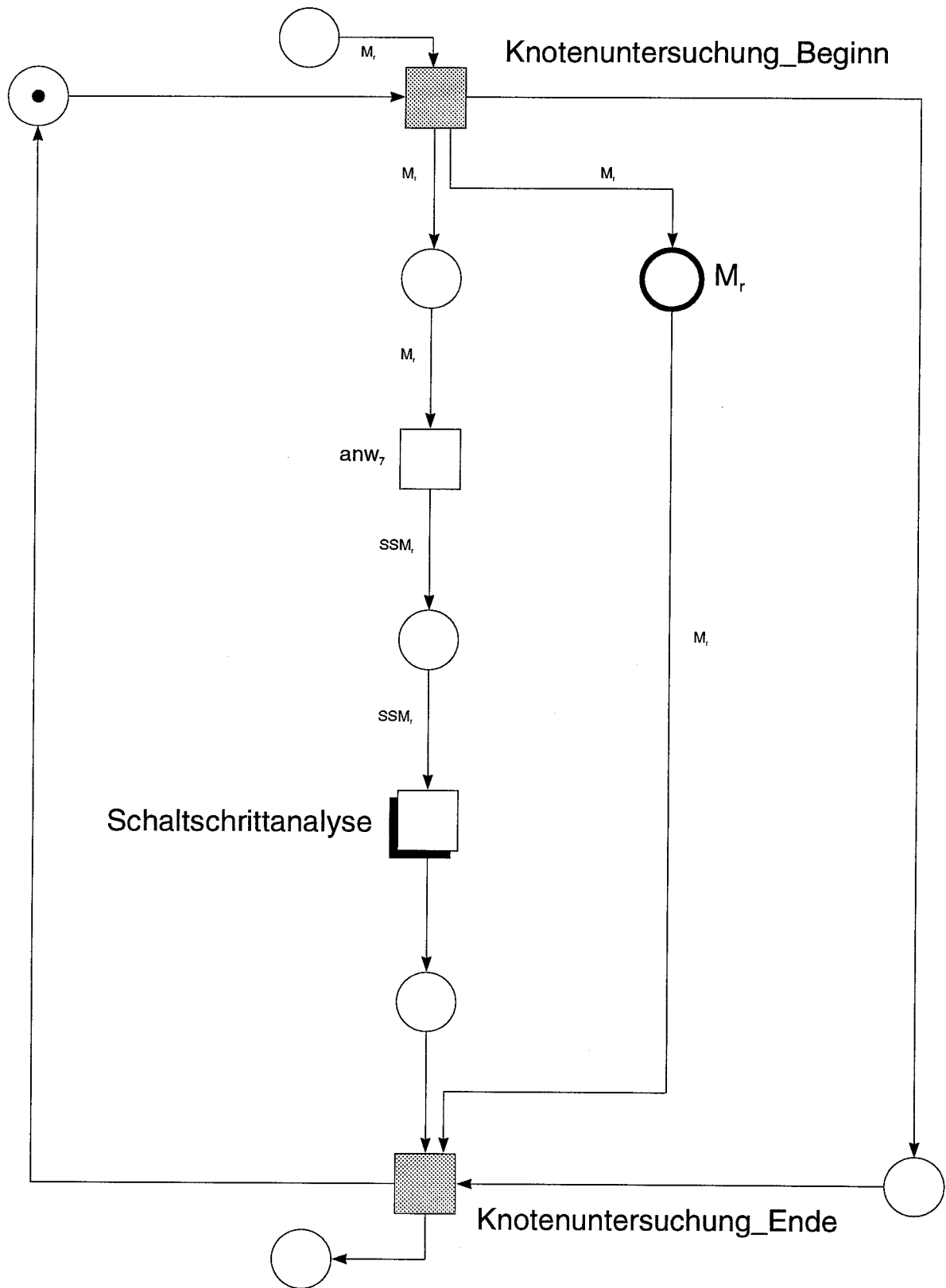


Abb. 80: Verfeinerung für die Makrotransition "Knotenuntersuchung"

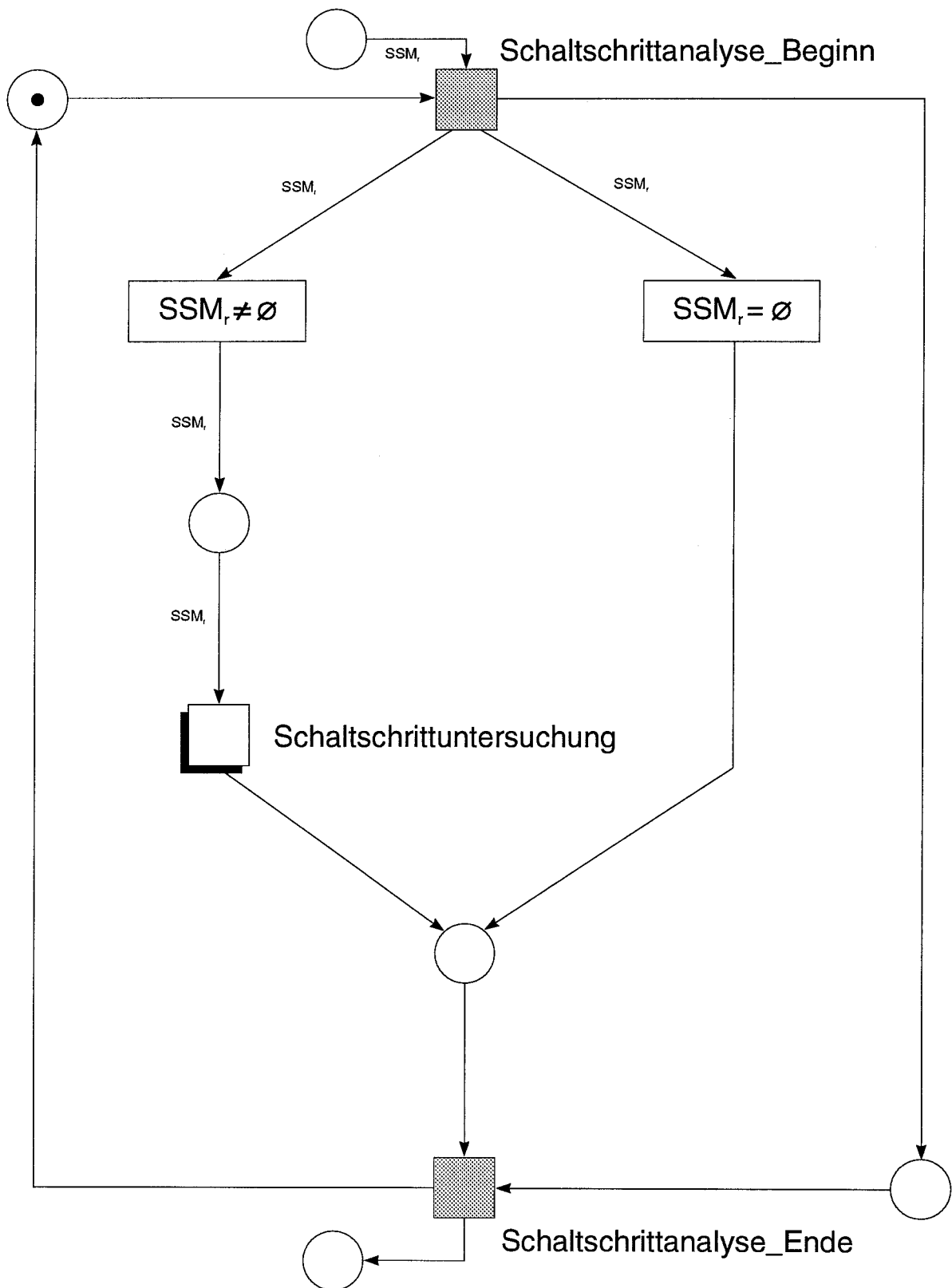


Abb. 81: Verfeinerung für die Makrotransition "Schaltschrittanalyse"

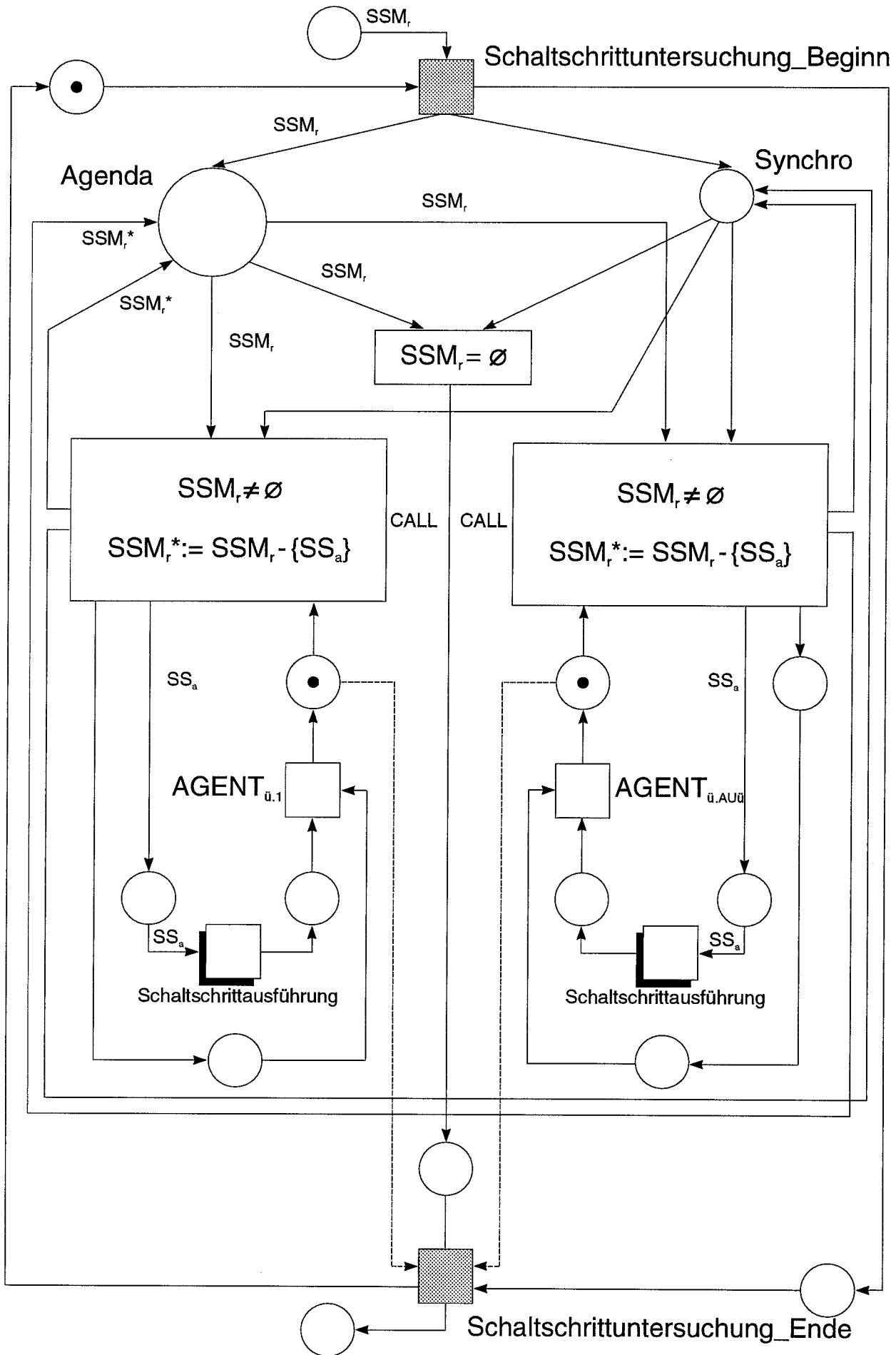


Abb. 82: Verfeinerung für die Makrotransition "Schaltschrittuntersuchung"

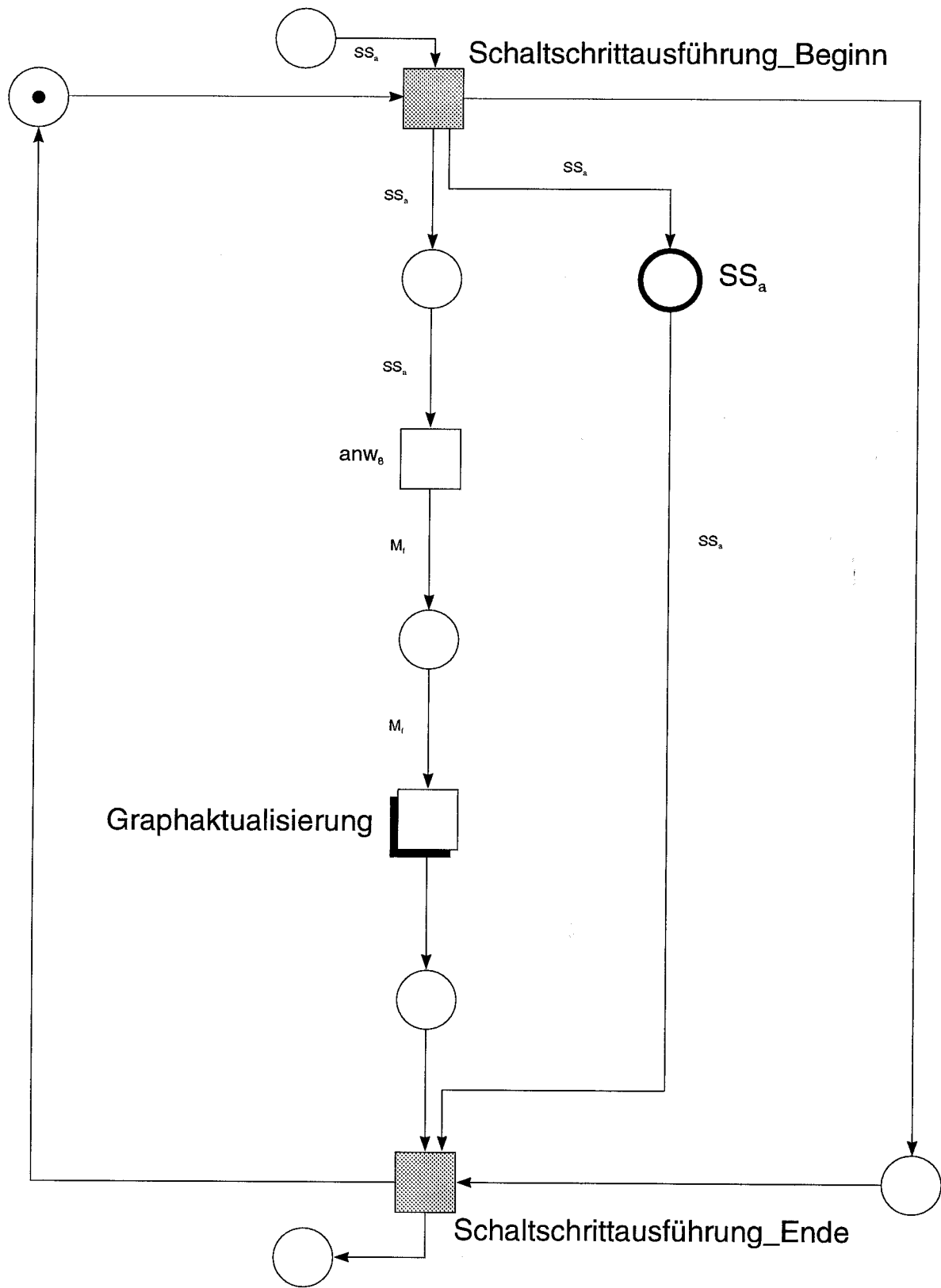


Abb. 83: Verfeinerung für die Makrotransition "Schaltschrittausführung"

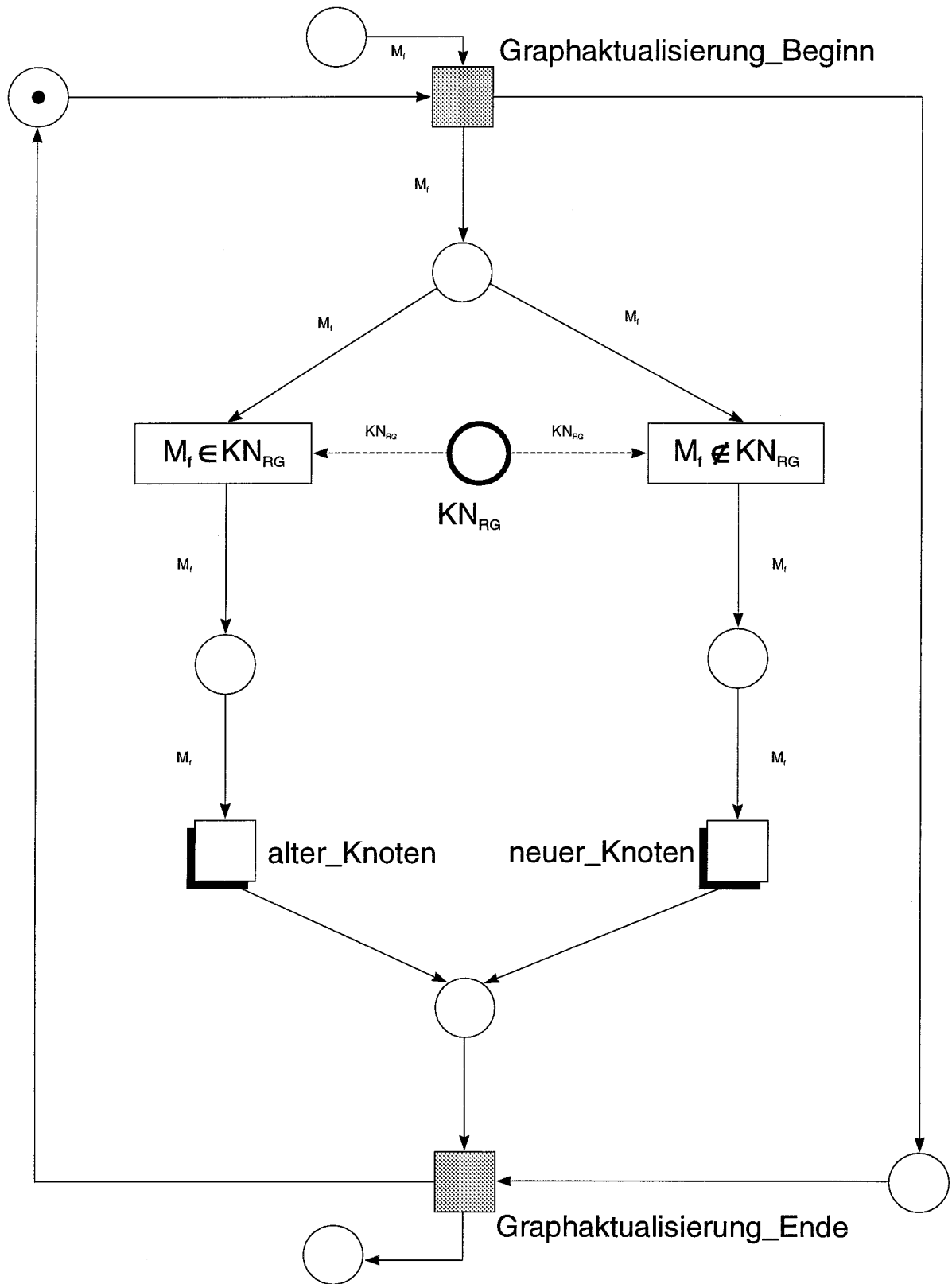


Abb. 84: Verfeinerung für die Makrotransition "Graphaktualisierung"

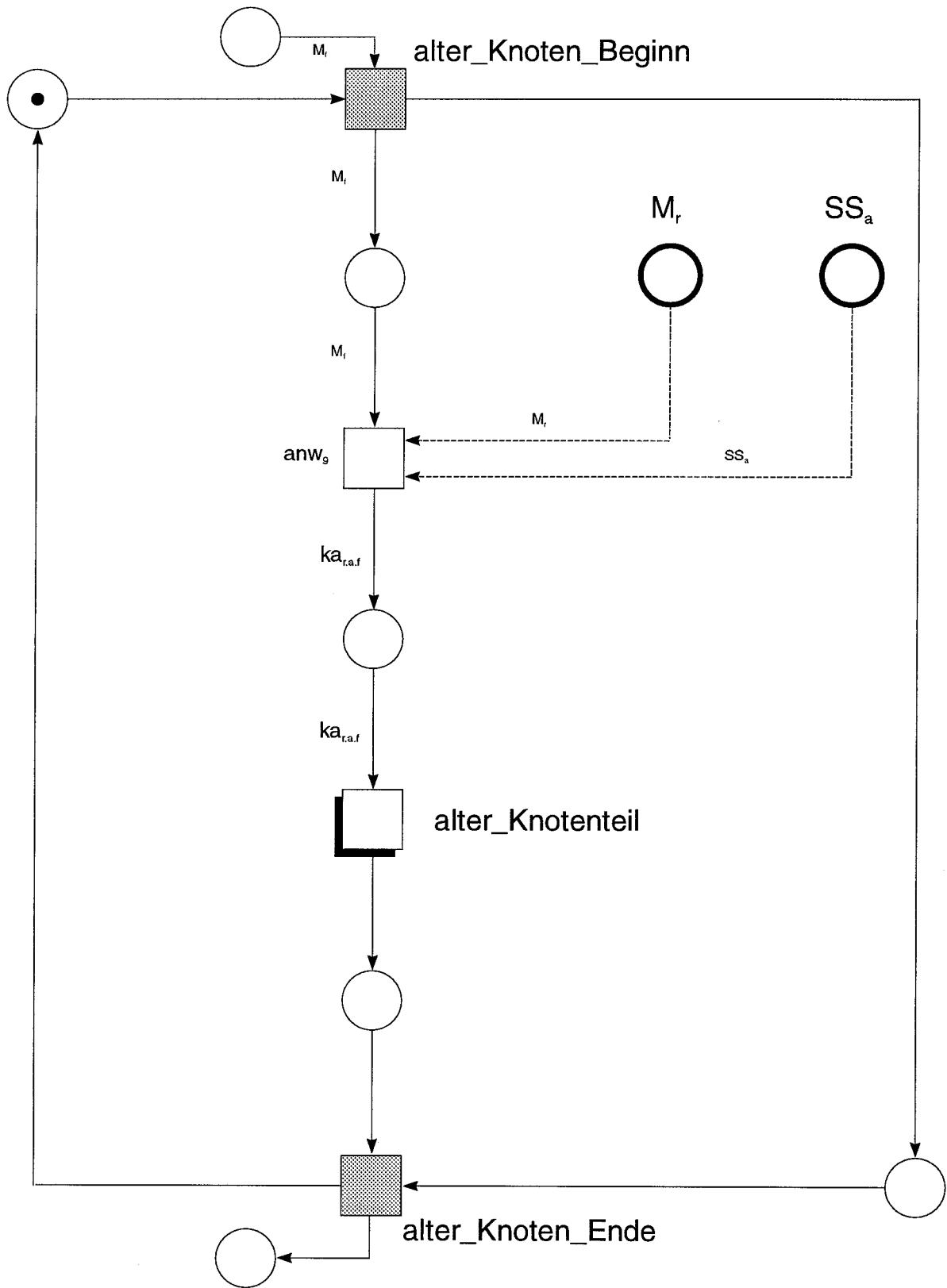


Abb. 85: Verfeinerung für die Makrotransition "alter_Knoten"

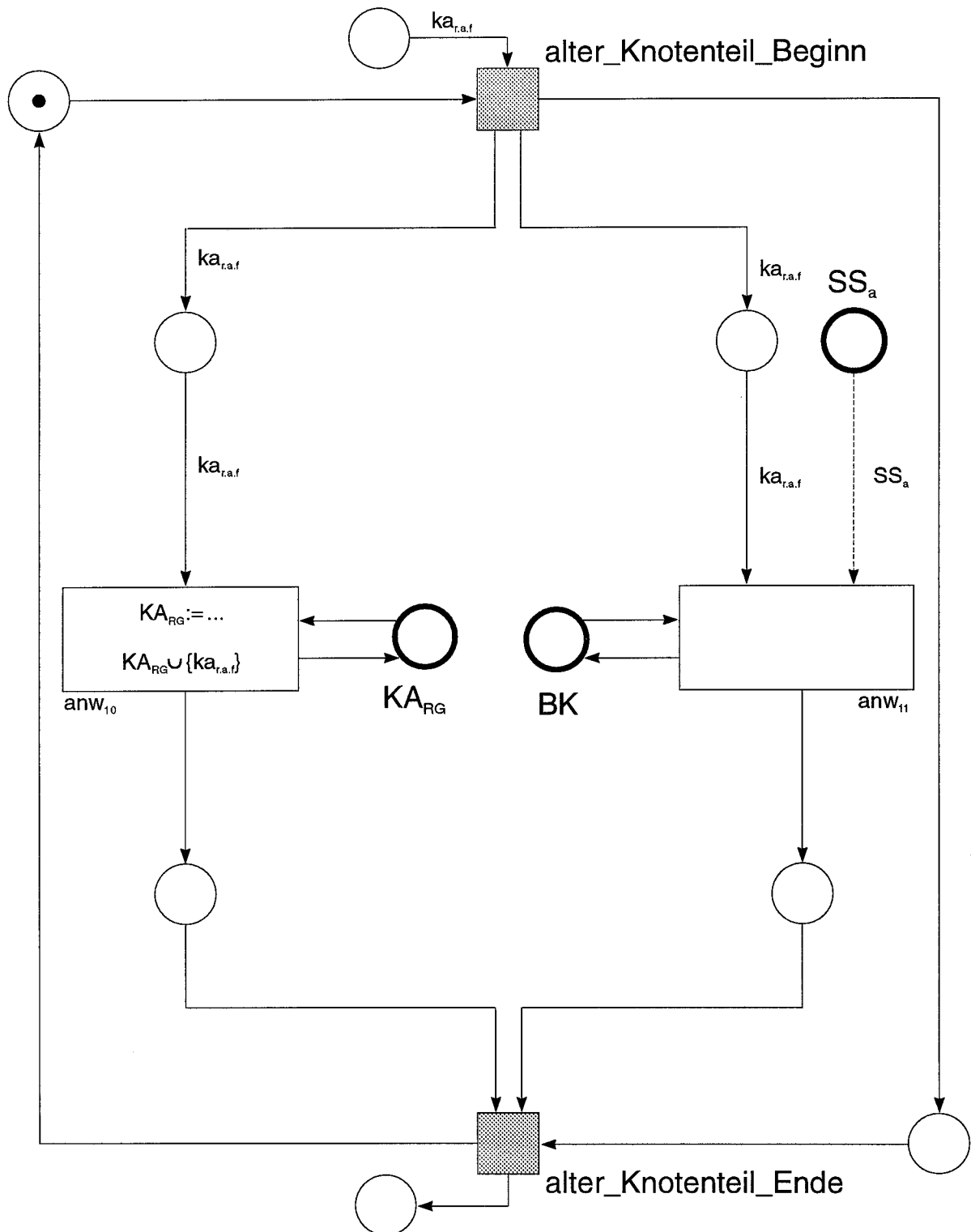


Abb. 86: Verfeinerung für die Makrotransition "alter_Knotenteil"

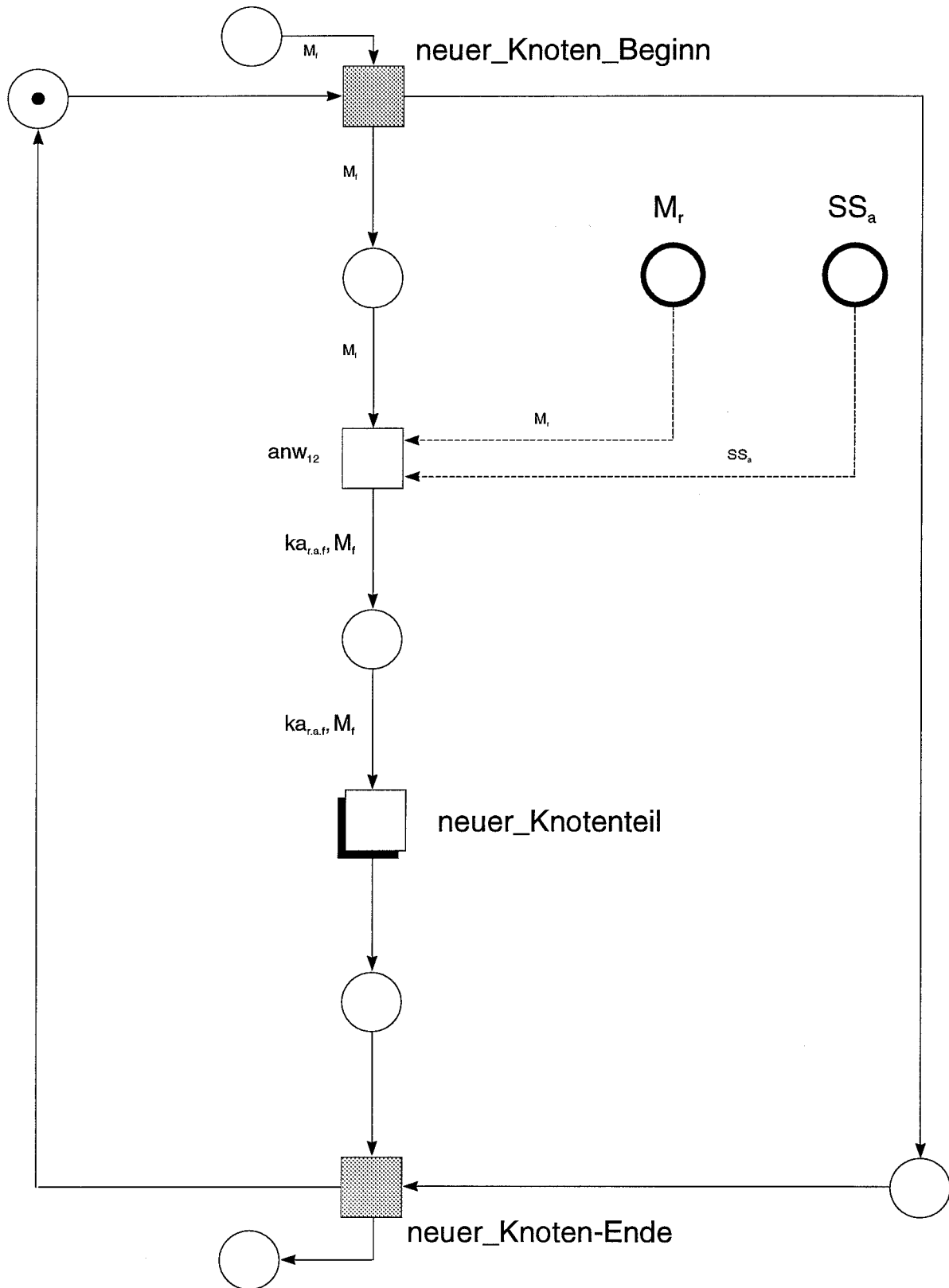


Abb. 87: Verfeinerung für die Makrotransition "neuer_Knoten"

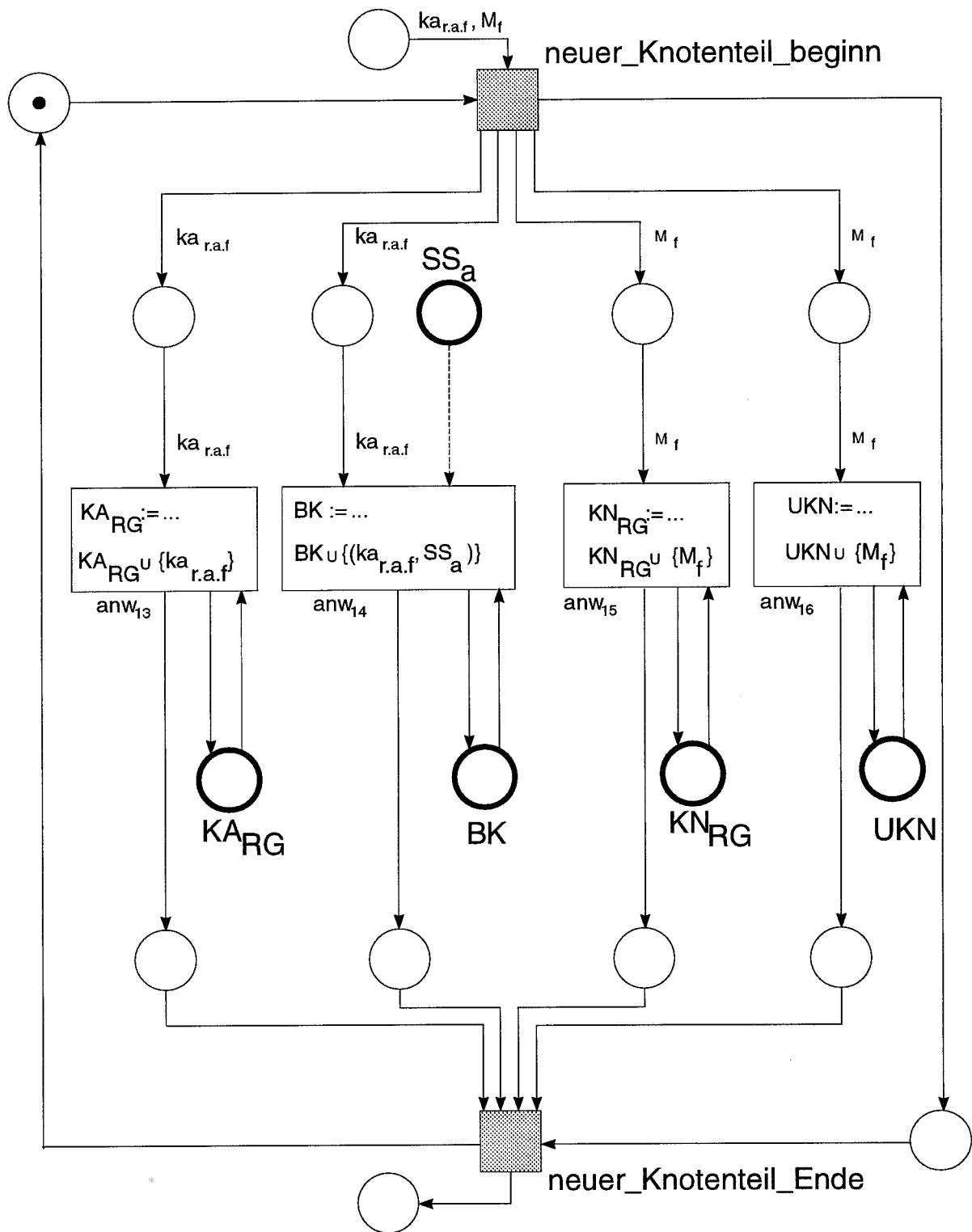


Abb. 88: Verfeinerung für die Makrotransition "neuer_Knotenteil"

Bisher wurde der eingangs vorgelegte nebenläufige Konstruktionsalgorithmus für die Erreichbarkeitsgraphen von Netzen nur durch ein nahezu gleichwertiges algorithmisches Netzmodell repräsentiert. Dadurch ließ sich die modulare Konstitution des Algorithmus besonders transparent darstellen. Aber es wurden - abgesehen von der Explizierung der Agentensynchronisation - noch keine neuartigen Einblicke in die Eigenart des Konstruktionsalgorithmus gewonnen. Dies wird nunmehr durch eine modifizierte, nicht mehr gleichwertige Netzrepräsentation des Konstruktionsalgorithmus geleistet. Anhand dieses Netzes wird später aufgezeigt, daß der Algorithmus noch zwei verborgene artifizielle Sequentialisierungen enthält¹⁰¹⁾.

Ausgangspunkt ist das Bemühen, mit der Hilfe von Petrinetzen die inhärente Nebenläufigkeit von beliebigen Algorithmen aufzudecken¹⁰²⁾. Dabei werden Algorithmen als endliche Mengen von Operatoren aufgefaßt, die Ein- in Ausgabeinformationen transformieren. Jeder Operator wird aus der Perspektive des Petrinetz-Konzepts als eine Transition dargestellt¹⁰³⁾. Seine Ein- und Ausgabeinformationen erscheinen als Marken, die zu den Einfluß- bzw. Ausgangsstellen der operatorrepräsentierenden Transition gehören. Die Transition legt bei ihrem Schalten auf ihren Ausgangsstellen diejenigen Marken ab, die von einer Anwendung des repräsentierten Operators als Ausgabeinformationen hervorgebracht werden. Bei den Einflußstellen handelt es sich um Eingangsstellen, falls die betroffenen Eingabeinformationen von einer Operatoranwendung verändert werden. Dann zieht die operatorrepräsentierende Transition die Marken, welche die Eingabeinformationen darstellen, beim Schalten von den Eingangsstellen ab. Die veränderten Informationen werden als Marken von Ausgabeinformationen auf den Ausgangsstellen abgelegt. Es ist aber auch möglich, daß durch eine Operatoranwendung eine Eingabeinformation nur zur Kenntnis genommen, aber nicht verändert wird. Dann wird die Einflußstelle, auf der sich die Marke der Eingabeinformation befindet, als eine Informationsstelle behandelt. Ihre Markierung wird durch das Schalten der operatorrepräsentierenden Transition nicht beeinflußt.

Der algorithmusspezifische Informationsverarbeitungszusammenhang zwischen den Operatoren wird durch ein Netz modelliert, das aus den operatorabbildenden Transitionen aufgebaut wird. Darin werden die operatorabbildenden Transitionen durch solche Stellen miteinander verknüpft, die zugleich die Ausgangsstelle einer Transition und die Einflußstelle einer anderen Transition darstellen. Dies ist genau dann der Fall, wenn die erste Transition einen Operator abbildet, dessen Ausgabeinformation die Eingabeinformation eines anderen Operators ist, der durch die zweite Transition repräsentiert wird. Daher gibt die topologische Struktur des Netzmodells den Informationsverarbeitungszusammenhang zwischen den Operatoren eines Algorithmus wieder. Dieser Operatorenzusammenhang wird durch den Informationsfluß konstituiert, der durch die Operatoranwendungen während der Algorithmusausführung bewirkt wird.

Allerdings muß auf die oben eingeführten Netzmodule verzichtet werden, wenn das Nebenläufigkeitspotential eines Algorithmus untersucht werden soll. Denn die Netzmodule und ihr hierarchischer Verfeinerungszusammenhang repräsentieren bereits das *Ergebnis* derjenigen Algorithmusgestaltung, die sich in der eingangs vorgestellten Pseudocode-Notation niedergeschlagen hat. Es kann nicht ausgeschlossen werden, daß dabei schon Gestaltungsentscheidungen getroffen wurden, welche die inhärente Algorithmusnebenläufigkeit nicht beachtet haben. Daher kann die Algorithmusgestaltung, die in Pseudocode-Notation erfolgte und mit Hilfe der Netzmodule rekonstruiert wurde, bereits artifizielle Sequentialisierungen enthalten. Daß dies tatsächlich der Fall ist, wird nunmehr nachgewiesen.

Die modulare und hierarchisch verfeinerte Vorstrukturierung der ursprünglichen Algorithmusformulierung wird zurückgenommen, um ihre verborgenen Sequentialisierungen aufzudecken. Der Konstruktionsalgorithmus wird jetzt durch ein modifiziertes Netzmodell ausschließlich auf der Ebene seiner atomaren Anweisungen reformuliert¹⁰⁴⁾. Jede atomare Anweisung wird wieder durch eine Transition abgebildet. Das Nebenläufigkeitspotential des derart modellierten Algorithmus wird durch alle nicht-degenerierten¹⁰⁵⁾ Schaltschritte offengelegt, die in seinem Netzmodell mindestens einmal aktiviert sind¹⁰⁶⁾. Jeder solche Schaltschritt repräsentiert mindestens zwei atomare Anweisungen, die sich im modellierten Algorithmus grundsätzlich

nebenläufig ausführen lassen. Daher läßt sich das Nebenläufigkeitspotential eines Algorithmus vollständig aus dem Erreichbarkeitsgraphen seines Netzmodells aufdecken. Es brauchen lediglich die dort enthaltenen nicht-degenerierten Schaltschritte identifiziert zu werden. Dies bereitet keine große Mühe, weil die Menge aller Kantenanschriften des Erreichbarkeitsgraphen eines algorithmischen Netzmodells genau alle Schaltschritte umfaßt, die in diesem Netzmodell jemals aktiviert sind. Aus dieser Anschriftenmenge brauchen nur noch die degenerierten Schaltschritte eliminiert zu werden. Die Restmenge charakterisiert das Nebenläufigkeitspotential des modellierten Algorithmus vollständig und explizit.

Bemerkenswert ist hier die erneute Selbstbezüglichkeit des Petrinetz-Konzepts. Denn der Algorithmus, der auf der Objektebene Erreichbarkeitsgraphen von Netzen zu konstruieren vermag, wird nicht nur auf der Metaebene durch ein Netz modelliert. Vielmehr wird jetzt auch die inhärente Nebenläufigkeit dieses Konstruktionsalgorithmus für Erreichbarkeitsgraphen selbst wieder anhand eines Erreichbarkeitsgraphen untersucht. Allerdings handelt es sich jetzt nicht mehr um die Erreichbarkeitsgraphen, die der Konstruktionsalgorithmus auf der Objektebene erzeugt. Statt dessen wird nunmehr auf der Metaebene der Erreichbarkeitsgraph des algorithmischen Netzmodells untersucht.

Das modifizierte algorithmische Netzmodell, das die modulare und hierarchische Konstitution des ursprünglich vorgelegten Konstruktionsalgorithmus durch die direkte Abbildung seiner atomaren Anweisungen ersetzt, fällt immer noch recht umfangreich aus. Es wird darauf verzichtet, es hier in seinem vollen Umfang zu präsentieren. Statt dessen werden nur zwei Modellausschnitte erörtert¹⁰⁷⁾. Sie lassen die artifiziellen Sequentialisierungen des ursprünglichen Konstruktionsalgorithmus erkennen. Die beiden Sequentialisierungen können sowohl aus den Teilnetzen selbst als auch aus ihren Erreichbarkeitsgraphen unmittelbar abgelesen werden¹⁰⁸⁾.

Die erste artifizielle Sequentialisierung betrifft die Prozedur "Erreichbarkeitsgraph_Konstruktion" und ihre Subprozedur "Initialisierung"¹⁰⁹⁾. Der relevante Ausschnitt aus dem modifizierten Netzmodell des ursprünglichen Konstruktionsalgorithmus wird durch das erste Teilnetz in Abb. 89 dargestellt. Es ist offensichtlich, daß die frühere Festlegung, die beiden Initialisierungen der Kantenmenge KA_{RG} und der Erfüllungsmenge BK müßten *zeitlich nach* dem Einlesen der netzdefinierenden Eingabeinformation erfolgen, überhaupt nicht notwendig ist. Denn beide Initialisierungsanweisungen hängen von der einzulesenden Netzdefinition nicht ab. Folglich sind ihre Ausführungen von der Ausführung der Einleseanweisung kausal - und somit auch zeitlich - unabhängig. Ihre sequentielle Anordnung im ursprünglich vorgelegten Algorithmus für die Konstruktion von Erreichbarkeitsgraphen stellt daher ein überflüssiges Artefakt der Algorithmusformulierung dar¹¹⁰⁾.

Diese intuitiv gewonnene Erkenntnis läßt sich ebenso aus dem Erreichbarkeitsgraphen des Teilnetzes systematisch ableiten. Er findet sich in Abb. 90. Seine Schaltkanten sind unmittelbar mit den Schaltschritten beschriftet, die jeweils eine Referenz- in eine Folgemarkierung transformieren. Die Definitionen seiner Markierungsknoten M_r mit $r \in \{0,1,\dots,25\}$ finden sich im Anschluß an Abb. 90.

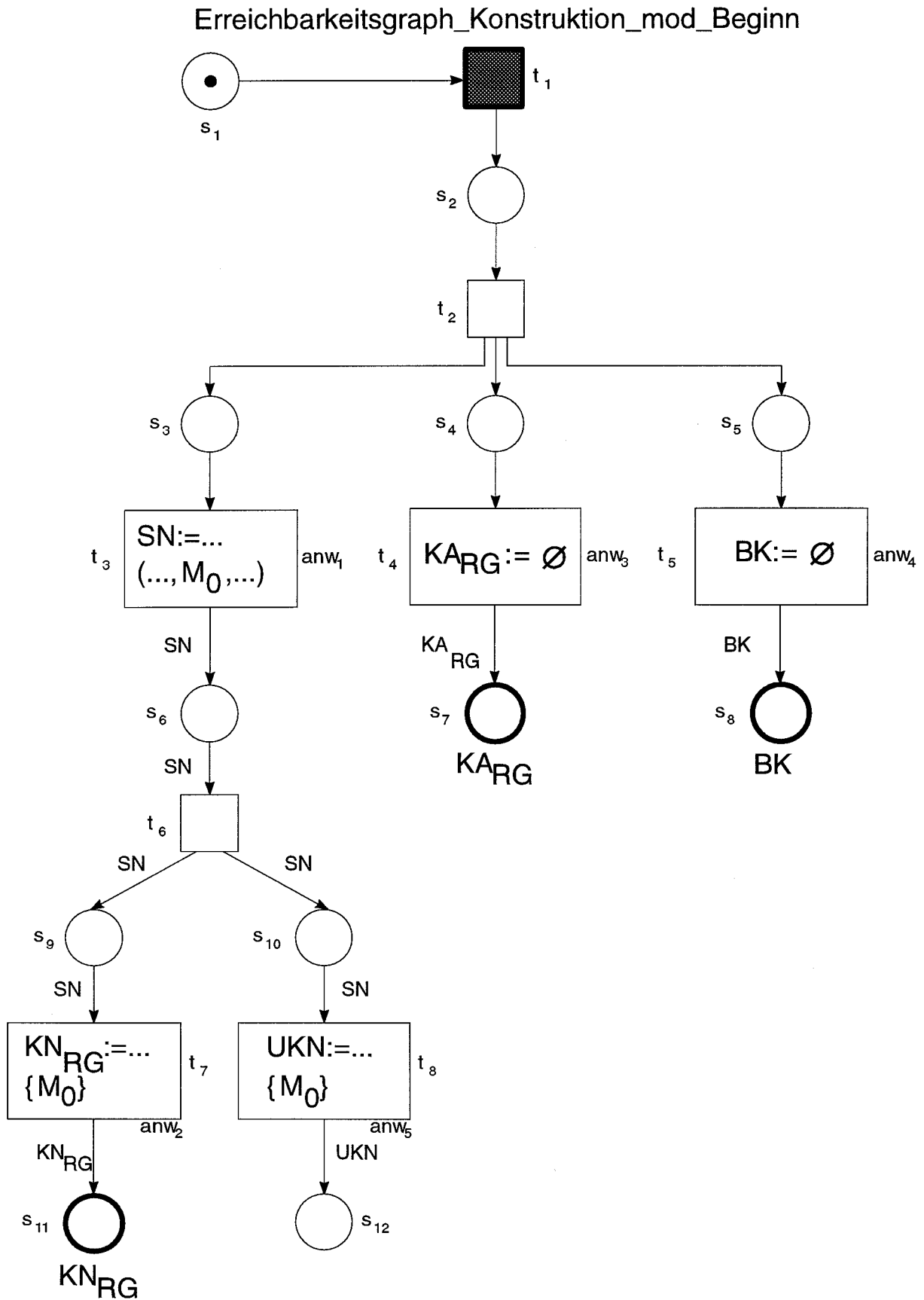


Abb. 89: erstes Teilnetz eines Ausschnitts aus dem modifizierten Netzmodell für einen Algorithmus zur Konstruktion von Erreichbarkeitsgraphen

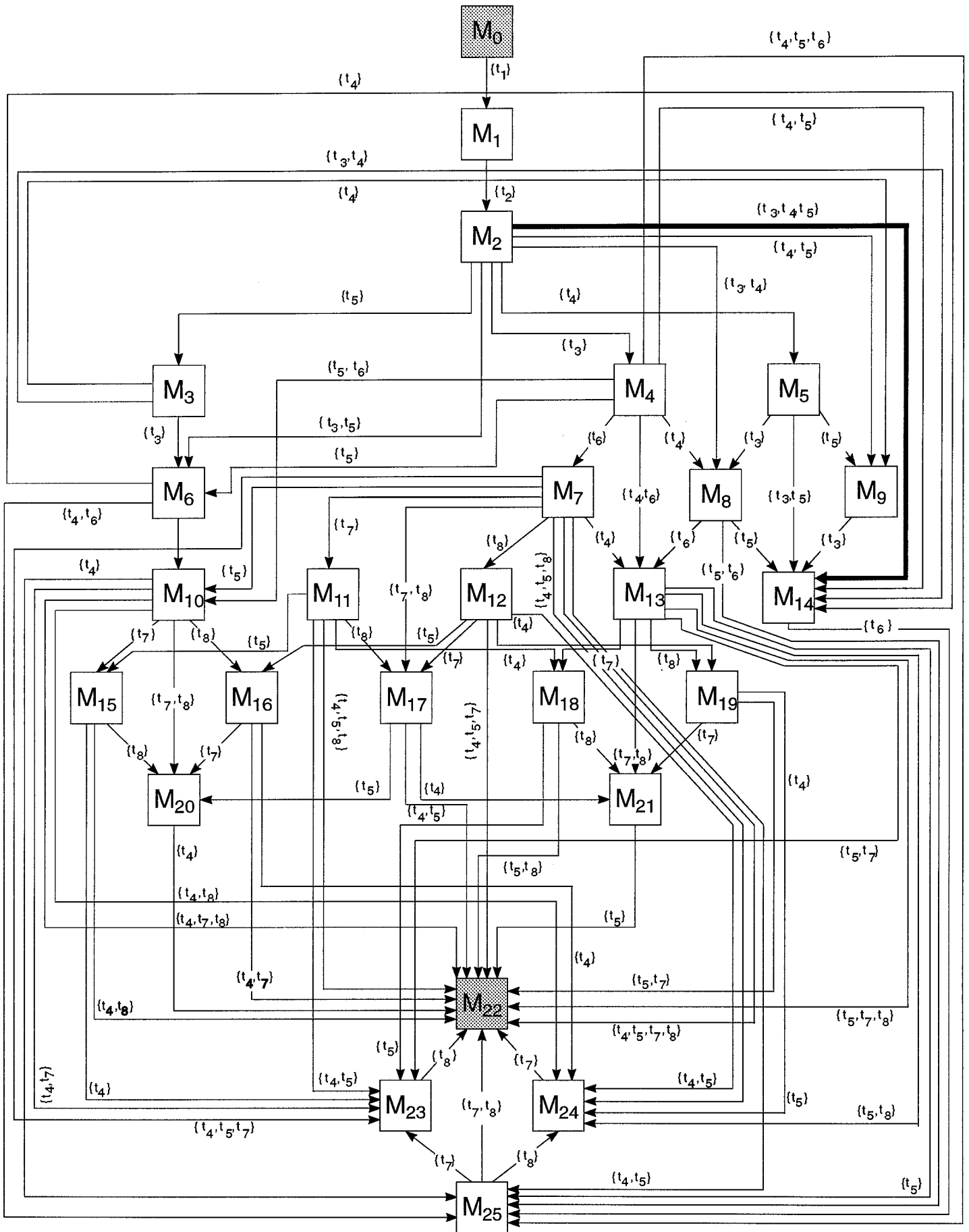


Abb. 90: Erreichbarkeitsgraph für das Teilnetz aus Abb. 89


```

TASK
  |
  |   anw5 ≈ initialisiere die Menge UKN der zu
  |           untersuchenden Knoten durch UKN := {M0}
  |
  |   PAR Ausgangsmarkierung_Folgen END
  |
  |   SEQ Ausgangsmarkierung END
  |
TASK
  |
  |   anw3 ≈ initialisiere die Kantenmenge KARG
  |           des Erreichbarkeitsgraphen durch KARG := ∅
  |
TASK
  |
  |   anw4 ≈ initialisiere die Erfüllungsmenge BK
  |           des Erreichbarkeitsgraphen durch BK := ∅
  |
  |   PAR Initialisierung_mod END
  |
  |   ...
  |
  |   SEQ Erreichbarkeitsgraph_Konstruktion_mod END

```

Die zweite artifizielle Sequentialisierung betrifft die Prozedur "neuer_Knoten". Sie gründet darin, daß die Anweisungen "erweitere die Knotenmenge KN_{RG} ..." und "erweitere die Menge UKN ..." von der einleitenden Anweisung "erzeuge die Kante ..." kausal unabhängig sind. Daher ist es überhaupt nicht notwendig, in der sequentiellen Prozedur "neuer_Knoten" *zuerst* die letztgenannte Anweisung und erst *danach* die beiden erstgenannten Anweisungen ausführen zu lassen. Dies läßt sich aus dem zweiten Teilnetz des modifizierten algorithmischen Netzmodells, das in Abb. 91 auf der nächsten Seite dargestellt wird, unmittelbar ablesen¹¹³).

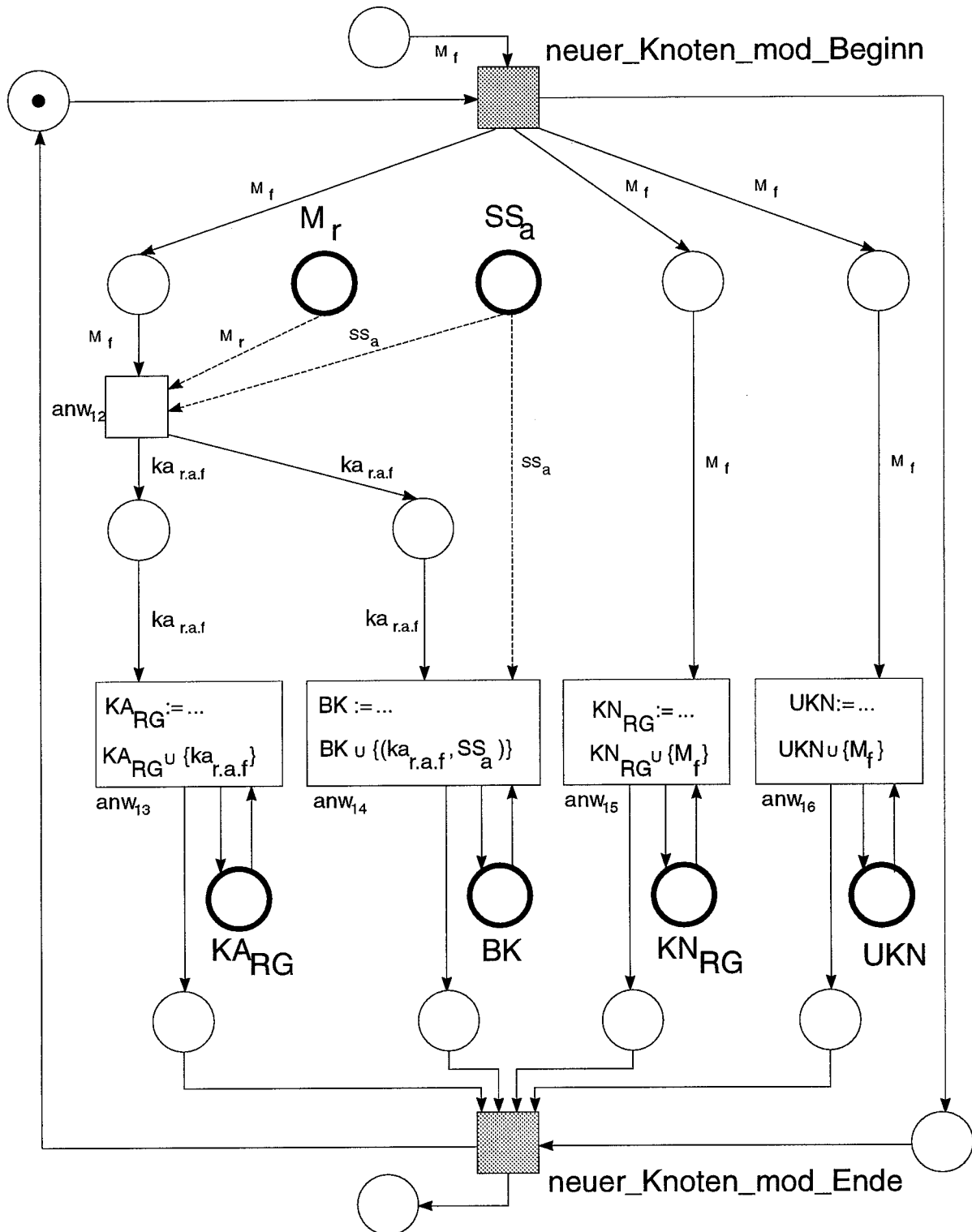


Abb. 91: zweites Teilnetz eines Ausschnitts aus dem modifizierten Netzmodell für einen Algorithmus zur Konstruktion von Erreichbarkeitsgraphen

Die überflüssige Sequentialisierung wird vermieden, wenn die vormals sequentielle Prozedur "neuer_Knoten" durch folgende modifizierte, nunmehr parallele Prozedur "neuer_Knoten_mod" ersetzt wird:

```

PAR neuer_Knoten_mod BEG
|
TASK
|   SEQ neue_Kante_mod BEG
|   |
|   |   anw12 ≈ erzeuge die Kante kar,a,f mit kar,a,f=(Mr,Mf)
|   |
|   |   PAR Erweiterung_mod BEG
|   |   |
|   |   |   TASK
|   |   |   |   anw13 ≈ erweitere die Kantenmenge KARG
|   |   |   |   |   um die Kante kar,a,f durch KARG := KARG ∪ {kar,a,f}
|   |   |   |
|   |   |   |   TASK
|   |   |   |   |   anw14 ≈ erweitere die Erfüllungsmenge BK der Beschriftungsfunktion bk
|   |   |   |   |   |   um das Paar aus der Kante kar,a,f und ihrer Beschriftung
|   |   |   |   |   |   |   bk(kar,a,f)=SSa durch BK := BK ∪ {(kar,a,f,SSa)}
|   |   |   |   |
|   |   |   |   |   PAR Erweiterung_mod END
|   |   |   |
|   |   |   |   SEQ neue_Kante_mod END
|   |   |
|   |   |   PAR Erweiterung_mod END
|   |   |
|   |   |   SEQ neue_Kante_mod END
|   |
|   |   TASK
|   |   |   anw15 ≈ erweitere die Knotenmenge KNRG
|   |   |   |   um den Knoten Mf durch KNRG := KNRG ∪ {Mf}
|   |   |
|   |   |   TASK
|   |   |   |   anw16 ≈ erweitere die Menge UKN der zu untersuchenden Knoten
|   |   |   |   |   um den Knoten Mf durch UKN := UKN ∪ {Mf}
|   |   |
|   |   |   PAR neuer_Knoten_mod END
|   |
|   |   PAR neuer_Knoten_mod END

```

Wenn die zuvor dargelegten Korrekturen in die eingangs vorgestellte Pseudocode-Notation des Konstruktionsalgorithmus für Erreichbarkeitsgraphen aufgenommen werden, enthält er keine artifiziellen Sequentialisierungen mehr. Die inhärente Nebenläufigkeit des Konstruktionsalgorithmus wird dann maximal ausgeschöpft. Dies schließt jedoch keineswegs aus, daß die algorithmische Konstruktion von Erreichbarkeitsgraphen auf der Basis *anderer* atomarer Anweisungen anw_q mit noch höherer interner Nebenläufigkeit formuliert werden kann.

Tatsächlich existiert mindestens ein solcher Konstruktionsalgorithmus mit einem noch höheren Nebenläufigkeitspotential¹¹⁴⁾. Denn die atomare Anweisung anw₇, die Menge SSM_r aller aktivierten Schaltschritte SS_a für einen Markierungsknoten M_r zu bestimmen, läßt sich in einem neuen Konstruktionsalgorithmus in andere atomare Anweisungen anw₇* zerlegen. Diese modifizierten Anweisungen anw₇* bestehen jeweils darin, genau einen potentiellen Schaltschritt hinsichtlich seiner Aktivierung unter der Markierung M_r zu untersuchen. Die anw₇*-Anweisungen können ihrerseits durch Agenten aus einer neuen, dritten Agentengruppe nebenläufig ausgeführt werden. Dadurch wird die inhärente Nebenläufigkeit der Konstruktion von Erreichbarkeitsgraphen ein erstes Mal gesteigert. Darüber hinaus kann ein Agent aus dem oben vorgestellten Konstruktionsalgorithmus, der die Subprozedur "Schaltschrittausführung" für einen aktivierten Schaltschritt SS_a aus der Menge SSM_r ausführt, bereits tätig werden, sobald einer der neu angeführten Agenten die Aktivierung dieses Schaltschritts SS_a erkannt hat. Die Agenten, die für die Subprozedur "Schaltschrittausführung" vorgesehen sind, brauchen also nicht zu warten, bis die

Gesamtheit aller aktivierten Schaltschritte für den Markierungsknoten M_r als Menge SSM_r vorliegt. Auf diese Weise wird die inhärente Nebenläufigkeit der Konstruktion von Erreichbarkeitsgraphen ein zweites Mal erhöht. Folglich besitzt ein Konstruktionsalgorithmus, der auf den modifizierten atomaren Anweisungen anw_7^* anstatt der oben benutzten atomaren Anweisung anw_7 beruht, eine größere inhärente Nebenläufigkeit als der oben präsentierte Konstruktionsalgorithmus. Der Verf. verzichtet darauf, diesen alternativen Konstruktionsalgorithmus hier im Detail darzulegen. Er hat die zugrundeliegende, soeben skizzierte Steigerung des Nebenläufigkeitspotentials bereits an anderer Stelle ausführlich behandelt¹¹⁵).

Anmerkungen zum Kapitel:

1) Sie ist weniger transparent als die vollständige Konstruktion eines Erreichbarkeitsgraphen, weil in ihr die schrittweise vorauseilende Konstruktion des Graphen mit der Auswertung des jeweils vorliegenden Teilgraphen kombiniert werden muß. Die dabei auszuführenden Konstruktionsschritte gehen nicht über diejenigen Schritte hinaus, die seitens der nachfolgend vorgestellten vollständigen Konstruktion eines Erreichbarkeitsgraphen vorgenommen werden. Daher werden durch diese vollständige Konstruktionsweise bereits alle Bausteine geliefert, mit denen sich später alle konstruktiven Teilaufgaben bei der zeitlich verschränkten Konstruktion und Auswertung eines Erreichbarkeitsgraphen erfüllen lassen.

2) Die detaillierte Spezifizierung eines Algorithmus, der die zeitlich verschränkte Konstruktion und Auswertung eines Erreichbarkeitsgraphen leistet, würde erfordern, neben dem hier vorgestellten Konstruktionsalgorithmus auch die benötigten Auswertungsalgorithmen darzulegen. Auf die Formulierung von Auswertungsalgorithmen verzichtet der Verf. in dieser Arbeit jedoch aus drei Gründen. Erstens müßte für jede interessierende dynamische Netzeigenschaft ein entsprechender Algorithmus angegeben werden, der das Vorliegen dieser Eigenschaft anhand der Auswertung des zugehörigen Erreichbarkeitsgraphen überprüfen könnte. Angesichts der Vielfalt der früher eingeführten interessanten Netzeigenschaften überstiege die Erarbeitung entsprechender Auswertungsalgorithmen den Ressourcenrahmen dieser Ausarbeitung. Zweitens liegen solche Auswertungsalgorithmen bereits in der Netzliteratur beschrieben und in Softwareprogrammen für die Netzanalyse implementiert vor. Daher erscheint ihre wiederholte Ausformulierung abundant. Dies wäre nur dann nicht der Fall, wenn *neuartige* Auswertungsalgorithmen präsentiert würden. Der Verf. sieht hierfür nur einen fruchtbaren Ansatz. Er besteht darin, die inhärente Nebenläufigkeit des Petrinetz-Konzepts in den Algorithmen für die Auswertung ihrer Erreichbarkeitsgraphen zur Entfaltung zu bringen. Dieser Aspekt liegt jedoch bereits der nachfolgend vorgestellten vollständigen Konstruktion von Erreichbarkeitsgraphen zugrunde. Daher wäre es eine redundante Wiederholung desselben konzeptionellen Ansatzes, wenn versucht würde, die inhärente Nebenläufigkeit des Petrinetz-Konzepts nicht nur bei der Konstruktion, sondern auch bei der Auswertung von Erreichbarkeitsgraphen zu berücksichtigen. Eine solche Wiederholung läßt keine neuartigen Erkenntnisse erhoffen. Dies liefert den dritten Grund, auf die detaillierte Ausarbeitung von Algorithmen für die zeitlich verschränkte Konstruktion und Auswertung von Erreichbarkeitsgraphen zu verzichten.

3) Vgl. dazu die Quellen, die bereits zur Konstruktion von Erreichbarkeitsgraphen für Stelle/Transition-Netze angeführt wurden.

4) Die inhärente Nebenläufigkeit von Netzen basiert auf der lokalen Definition ihrer Schaltregel. Diese Nebenläufigkeit läßt sich in vier Effekte differenzieren: Erstens läßt es die Schaltregel zu, unter einer gegebenen Netzmarkierung die *Aktivierung* eines jeden Schaltschritts unabhängig von allen anderen Schaltschritten zu prüfen. Folglich besteht die Möglichkeit, in jedem Markierungsknoten des Erreichbarkeitsgraphen alle potentiellen Schaltschritte hinsichtlich ihrer Aktivierung nebenläufig zu untersuchen. Zweitens können die *Schaltwirkungen* aller tatsächlich aktivierten Schaltschritte unabhängig voneinander berechnet werden. Daher lassen sich die Folgeknoten, die von einem Referenzknoten durch Ausführen jeweils eines Schaltschritts erreicht werden, nebenläufig ermitteln. Drittens kann die Schaltwirkung eines Schaltschritts bereits untersucht werden, sobald seine Aktivierung festgestellt wurde. Deswegen ist es möglich, diese *Wirkungsuntersuchung* nebenläufig zur *Aktivierungsprüfung* anderer Schaltschritte auszuführen. Dies hängt jedoch vom Zutreffen zweier Voraussetzungen ab: Einerseits muß die Aktivierung des erstgenannten Schaltschritts früher als die der letztgenannten Schaltschritte erkannt werden. Andererseits müssen alle betrachteten Schaltschritte im selben Markierungsknoten aktiviert sein. Viertens führen die Schaltwirkungen, die sich aufgrund des zweiten und dritten Effekts in nebenläufiger Weise bestimmen lassen, zu mehreren neuen Markierungsknoten, die voneinander unabhängig sind. Für diese *verschiedenen Markierungsknoten* können die Aktivierungen und Schaltwirkungen ihrer potentiellen Schaltschritte abermals in nebenläufiger Weise analysiert werden. Vgl. zu diesen Aspekten der inhärenten Nebenläufigkeit von Netzen auch ZELEWSKI (1986c), S. 17; ZELEWSKI (1988b), S. 357.

Eine Konstruktionsmethode für Erreichbarkeitsgraphen, welche die vier vorgenannten Nebenläufigkeitspotentiale vollständig ausschöpft, verwirklicht die inhärente Nebenläufigkeit des zugrundeliegenden Petrinetzes umfassend. Es liegt in der Hand des Entwicklers von algorithmischen Konkretisierungen der Konstruktionsmethode, in welchem Ausmaß das Nebenläufigkeitspotential des Petrinetz-Konzepts ausgenutzt wird. Beispielsweise kann er sich darauf beschränken, nur diejenigen Schaltschritte nebenläufig zu untersuchen, die im selben Markierungsknoten eines Erreichbarkeitsgraphen aktiviert sind. Ein weitergehender Verzicht auf potentielle Nebenläufigkeit bestünde darin, nicht alle Schaltschritte, die infolge ihrer Aktivierung ausgeführt werden können, tatsächlich auszuführen, sondern stets nur einen von ihnen auszuwählen. In diesem Freiheitsgrad variabler Realisierung des Nebenläufigkeitspotentials von Netzen liegt ein bemerkenswerter Gestaltungsspielraum des Petrinetz-Konzepts. Im folgenden wird angestrebt, bei der Konstruktion von Erreichbarkeitsgraphen die inhärente Nebenläufigkeit des Petrinetz-Konzepts in größtmöglicher Weise zu verwirklichen.

5) In einem nondeterministischen Algorithmus ist die Menge der auszuführenden Operationen - im Vergleich zu konventionellen sequentiellen Algorithmen - nicht mehr zeitlich vollständig geordnet. Daher legt die Kontrolle der

Algorithmusausführung auch keine eindeutig bestimmte Ausführungssequenz für die Operationen fest. Statt dessen unterliegt die Menge der auszuführenden Operationen nur einer kausalen Halbordnung. Sie legt fest, welche Operationausführungen abgeschlossen sein müssen, damit andere Operationausführungen beginnen können. Die kausale Halbordnung impliziert daher auch eine zeitliche Präzedenzrelation, die über der Menge der auszuführenden Operationen in der Gestalt von Normalfolgen errichtet ist. Darauf wird in einer späteren Anmerkung zurückgekommen. Dort werden kausal unabhängige Operationausführungen als Konstruktionsschritte konkretisiert. In einem Automatischen Informationsverarbeitungssystem mit paralleler Architektur können kausal unabhängige Operationausführungen eines nondeterministischen Algorithmus von jeweils einem Prozessor zeitgleich übernommen werden. Da die Prozessoren unabhängig voneinander arbeiten, liegt ex ante nicht fest, in welchem zeitlichen Verhältnis die Ergebnisse ihrer Operationausführungen vorliegen. Die Verfügbarkeit von Operationsergebnissen ist in dem Ausmaß nicht determiniert, wie keine kausalen Abhängigkeiten zwischen den Operationausführungen bestehen. Daher rührt die Bezeichnung "nondeterministische Algorithmen". Zugleich besitzen diese Algorithmen nebenläufige Qualität, weil ihre kausal unabhängigen Operationenausführungen auf mehreren Prozessoren nebenläufig erfolgen können. Deshalb werden nondeterministische Algorithmen ebenso als nebenläufig ausführbare Algorithmen bezeichnet. Da im Bereich der Informatik zumeist von parallelen anstatt von nebenläufigen Konzepten gesprochen wird, läßt sich schließlich auch das Synonym "parallele Algorithmen" verwenden.

Nähere Erläuterungen von nondeterministischen, nebenläufig ausführbaren oder parallelen Algorithmen finden sich bei COOK, S. (1971), S. 155 u. 157; KARP (1975b), S. 45, 47 u. 58; IMAI (1979), S. 416ff.; NAKAGAWA (1979), S. 628ff.; KORNFELD (1981a), S. 575ff.; KANAL (1981), S. 570f.; GEORGEFF (1981), S. 566; FUCHI (1983), S. 6f.; SIMON, H. (1983a), S. 13; KINDERVATER (1985), S. 113ff.; HERMAN (1985), S. 130ff.; PAN (1986), S. 130ff.; ZELEWSKI (1986a), S. 84f., 283 (Fn. 1), 314ff., 322ff. u. 673f.; ZELEWSKI (1986c), S. 1ff., insbesondere S. 33ff. u. 44ff.; IBARAKI (1987), S. 540ff.; VALMARI (1988a), S. 107ff.; ZELEWSKI (1988b), S. 352ff.

Eine abstrakte Variante paralleler Automatenarchitekturen für nondeterministische Algorithmen stellen nondeterministische TURING-Automaten dar. Sie wurden bereits im Zusammenhang mit NP-vollständigen Problemen eingeführt.

Vgl. zur Möglichkeit, nondeterministische Algorithmen auf Automatischen Informationsverarbeitungssystemen mit parallelen Architekturen zu implementieren, BOLEY (1981), S. 151f.; KORNFELD (1981a), S. 575; BIBEL (1982a), S. 266f.; MINKER (1983), S. 142; ZELEWSKI (1986a), S. 321f., 325 u. 685; IBARAKI (1987), S. 538f.

6) Nebenläufig ausführbare Algorithmen werden fortan auch kurz als nebenläufige Algorithmen angesprochen.

7) Dabei handelt es sich um eine sprachliche Vereinfachung, die von zwei Aspekten abstrahiert. Erstens wird nicht weiter beachtet, daß keine Methode als Ganzes nebenläufig sein kann. Statt dessen lassen sich nur einzelne Komponenten der Methode zueinander nebenläufig ausführen. Zweitens wird davon abgesehen, ob die nebenläufig ausführbaren Komponenten der Konstruktionsmethode bei einer Implementierung des methodenzugehörigen Konstruktionsalgorithmus auch tatsächlich nebenläufig ausgeführt werden. Denn jeder nebenläufig konzipierte Algorithmus läßt sich durch nachträgliche Sequentialisierungen auf rein sequentiell arbeitenden Automatischen Informationsverarbeitungssystemen ausführen.

8) Von der artifiziellen Sequentialisierung zweier Operationen aus einer halbgeordneten Operationenmenge wird gesprochen, wenn die beiden Operationen zwar in der zugrundeliegenden Operationenmenge nicht angeordnet sind, aber dennoch nachträglich in einer von ihren zwei denkmöglichen Reihenfolgen ordiniert werden. Da diese Reihenfolgen in der halbgeordneten Operationenmenge nicht vorgesehen sind, handelt es sich um eine Überlagerung der Halbordnung der Operationenmenge durch eine zusätzliche - "künstliche" - Anordnungsbeziehung. Im oben thematisierten Kontext der Konstruktion von Erreichbarkeitsgraphen werden die Operationen als Konstruktionsschritte konkretisiert.

Artifizielle Sequentialisierungen stellen keineswegs eine exotische Randerscheinung dar. Vielmehr handelt es sich um den Normalfall, der den Entwurf von Algorithmen dominiert. Hierin manifestiert sich, daß algorithmisches Denken weithin in sequentiellen Kognitionsstrukturen eingespielt ist. Die sequentielle VON NEUMANN-Architektur konventioneller Automatischer Informationsverarbeitungssysteme mag dazu wesentlich beigetragen haben. Vgl. auch ZELEWSKI (1986c), S. 1; ZELEWSKI (1988b), S. 352.

Es überstiege den Erkenntnisrahmen der hier vorgelegten Untersuchungen, die voranstehenden Anmerkungen zu sequentiellen Denkweisen näher zu belegen und argumentativ abzustützen. Statt dessen wird nur - als pars pro toto - auf die Ansicht von ALDINGER (1985a), S. 107, verwiesen, daß im Bereich der Maschinenbelegungsplanung "... jeder Algorithmus nur sequentiell arbeitet". Diese kategorische Behauptung universeller Sequentialität wird durch die nebenläufige Konstruktion und durch die ebenso nebenläufige Auswertung der Erreichbarkeitsgraphen von Netzen, die etwa Maschinenbelegungsprobleme in Flexiblen Fertigungssystemen modellieren können, widerlegt. Weitere Hinweise auf künstliche Sequentialisierungen und ihre - zumeist unerwünschten - Nebenwirkungen - finden sich bei ZELEWSKI (1986a), S. 314, 322, 933 u. 946 (Fn. 2).

9) Zwei Konstruktionsschritte heißen kausal voneinander abhängig, wenn die Ausführung eines von beiden erst dann begonnen werden kann, nachdem die Ausführung des jeweils anderen Konstruktionsschritts beendet worden

ist. Zwei Konstruktionsschritte sind dagegen kausal unabhängig, wenn jeder von ihnen ausgeführt werden kann, ohne den Ausführungsbeginn oder das Ausführungsende des jeweils anderen Konstruktionsschritts vorauszusetzen. Andere Ereignisse als Ausführungsbeginn und -ende von Konstruktionsschritten brauchen nicht berücksichtigt zu werden. Falls ein dazwischen liegendes Ereignis für die kausale Abhängigkeit von Konstruktionsschritten relevant sein sollte, wird derjenige Konstruktionsschritt, zu dem das Ereignis gehört, durch dieses Ereignis in zwei Teilschritte aufgespalten. Das Ereignis stellt dann zugleich das Ausführungsende des zeitlich früheren Teilschritts und den Ausführungsbeginn des zeitlich späteren Teilschritts dar. Andere kausale Abhängigkeiten als die oben dargelegte kausale Normalfolge brauchen nicht berücksichtigt zu werden. Denn der unten vorgestellte Konstruktionsalgorithmus besteht ausschließlich aus solchen Konstruktionsschritten, die in der Beziehung einer kausalen Normalfolge stehen, falls sie kausal voneinander abhängen.

10) Der unten vorgestellte Konstruktionsalgorithmus enthält allerdings zunächst noch zwei artifizielle Sequentialisierungen. Sie wurden bewußt vorgenommen, um solche Sequentialisierungen und ihre Aufdeckung exemplarisch verdeutlichen zu können. Vgl. dazu die Erläuterung dieser beiden Sequentialisierungen an früherer Stelle. Dort werden auch entsprechende partielle Modifizierungen des Konstruktionsalgorithmus vorgelegt, welche die artifiziellen Sequentialisierungen aufheben.

11) Von parallelen Systemarchitekturen wird hier in einer groben Annäherung gesprochen, wenn Automatische Informationsverarbeitungssysteme über mehrere Prozessoren verfügen, die ihre Operationen zeitgleich ausführen können (aber nicht müssen). Die Operationsausführungen können dabei entweder durch eine globale Zeittaktung aller Prozessoren synchronisiert oder aber ohne Bezug auf eine globale Systemuhr asynchron verlaufen. Der erste Fall herrscht heute bei Automatischen Informationsverarbeitungssystemen mit parallelen Architekturen vor. Der zweite Fall entspricht konzeptionell der Nebenläufigkeit des Schaltens ("Ausführens") von Transitionen ("Operationen") in Petrinetzen. Darauf wird noch zurückgekommen.

Verfeinernde Übersichten über Konzepte für parallele Architekturen von Automatischen Informationsverarbeitungssystemen finden sich z.B. bei GILOI (1981), S. 101ff.; GILOI (1982), S. 6ff.; AMAMIYA (1982), S. 180 u. 183ff.; TANAKA, H. (1982), S. 209ff.; MINKER (1983), S. 142ff.; GILOI (1984), S. 58ff.; DIEL (1984), S. 373ff.; SHAPIRO, E. (1984), S. 458ff.; MOTO-OKA (1984), S. 479ff.; SAKAI (1984), S. 507ff.; BIC (1984), S. 517ff.; YAMAGUCHI, Y. (1984), S. 524ff.; AMAMIYA (1984), S. 105ff.; ZELEWSKI (1986a), S. 314ff. u. 326ff.; IBARAKI (1987), S. 539f. Vgl. darüber hinaus zum Begriff der Automatenarchitektur, der hier nicht weiter erörtert wird, GILOI (1978), S. 275, 277f. u. 298; GILOI (1981), S. 1ff.; GILOI (1984), S. 60f.

12) Dies entspricht dem Postulat, Konzepte unabhängig von ihrer späteren Implementierung zu formulieren.

13) Folglich wird die Konstruktionsmethode so formuliert, daß sie nicht nur auf Automatischen Informationsverarbeitungssystemen mit verschiedenen parallelen, sondern ebenso auf solchen mit sequentiellen Systemarchitekturen implementiert werden kann. Vgl. dazu die Anmerkung hinsichtlich des Freiheitsgrads, die inhärente Nebenläufigkeit des Petrinetz-Konzepts in unterschiedlichem Ausmaß zu verwirklichen. Falls die Konstruktion von Erreichbarkeitsgraphen auf Informationsverarbeitungssystemen mit sequentiellen Architekturen implementiert würde, ließe sich das Nebenläufigkeitspotential der Konstruktionsmethode nur minimal - nämlich überhaupt nicht - ausnutzen. Es würde dagegen maximal ausgeschöpft, wenn ein Automatisches Informationsverarbeitungssystem mit mindestens so vielen asynchron operierenden Prozessoren zur Verfügung stände, die ausreichten, um in dem anschließend vorgestellten Konstruktionsalgorithmus alle nebenläufig auszuführenden Subprozeduren durch jeweils einen Prozessor abarbeiten zu lassen.

14) Vgl. zum VON NEUMANN-Architekturprinzip sequentieller Informationsverarbeitung VON NEUMANN (1958), S. 17ff.; BURKS (1963), S. 35ff., insbesondere S. 38; PYLYSHYN (1970), S. 6; GILOI (1978), S. 282 u. 299; GILOI (1981), S. 33ff.; GILOI (1984), S. 55f. u. 61f.; SCHMITZ, P. (1985), S. 80f.

15) Vgl. zu den Beschleunigungspotentialen, die in paralleler (nebenläufiger) gegenüber sequentieller Informationsverarbeitung liegen, IMAI (1979), S. 418; GILOI (1981), S. 128ff. u. 141ff.; KORNFELD (1981a), S. 579; BOLEY (1981), S. 152f.; KRAUSE, F. (1985), S. 65f.; ZELEWSKI (1986a), S. 318 u. 325f.; ZELEWSKI (1986c), S. 2; IBARAKI (1987), S. 538 u. 545; ZELEWSKI (1988b), S. 352; BARR, R. (1989), S. 31f.; vgl. aber auch die Diskussion der bemerkenswerten Beschleunigungsanomalien bei IBARAKI (1987), S. 549ff., und BARR, R. (1989), S. 32 i.V.m. S. 31.

16) Die Prozessoranzahl liegt zumeist bei weniger als einem halben Dutzend. Vgl. z.B. BARR, R. (1989), S. 30f. (nur maximal 6 Prozessoren).

17) Vgl. zu Begrenzungen, das Nebenläufigkeitspotential nondeterministischer Algorithmen durch reale Automatische Informationsverarbeitungssysteme mit geringen Prozessoranzahlen zu verwirklichen, SIMON, H. (1979b), S. 1091; BOLEY (1981), S. 153; HART, P. (1982), S. 14; ZELEWSKI (1986a), S. 322 u. 325; IBARAKI (1987), S. 545; BARR, R. (1989), S. 30.

18) Auch im japanischen Projekt, "Computer der 5. Generation" zu entwickeln, wurden parallele Automatenarchitekturen angestrebt, in denen mehrere Tausend Prozessoren zusammenwirken sollten. Dies hat der Verf. an anderer Stelle dargelegt; vgl. ZELEWSKI (1986a), S. 327f., und die dort angeführten Quellen. Allerdings ist dieses Projekt

mittlerweile ohne bemerkenswerte Realisierungen seiner hoch gesteckten Ziele eingestellt worden. Die nachfolgend vorgestellten Beispiele für parallel arbeitende Informationsverarbeitungssysteme erstrecken sich dagegen auf Automatenarchitekturen, die bereits verwirklicht wurden.

19) Vgl. zu diesem Automaten, der schon das Stadium kommerziellen Vertriebs erreicht hat, HILLIS (1986b).

20) Typische Konfigurationen mit relativ "hoher" Parallelität bestehen derzeit aus 16 Transputern; vgl. WÜRTZ (1990), S. 39; MIGLIORE (1990), S. 292.

21) Dies kann zwar ohne ausführliche Untersuchungen, in denen "repräsentative" Konstruktionsaufgaben erfüllt werden, nicht streng belegt werden. Aber in diese Richtung weisen immerhin die Erfahrungen des Verf. mit denjenigen Erreichbarkeitsgraphen, die er in dieser Arbeit vorlegt. Dort bleiben die Anzahlen nebenläufig ausführbarer Konstruktionsschritte auf einige Dutzend beschränkt. Es zeigt sich dabei auch, daß die Anzahl nebenläufig ausführbarer Konstruktionsschritte nur geringfügig zunimmt, wenn die Größe der betrachteten Erreichbarkeitsgraphen ansteigt. Daher ist auch nicht zu erwarten, daß sich beim Übergang zu wesentlich umfangreicheren Netzmodellen das Nebenläufigkeitspotential der Konstruktion ihrer Erreichbarkeitsgraphen nicht mehr realisieren ließe, wenn mehrere Tausend Prozessoren zur Verfügung stehen.

22) Dies gilt allerdings nur im Hinblick auf die *Geschwindigkeit* der Erreichbarkeitsanalyse. Effizienzbetrachtungen können zwar auf diesen Geschwindigkeitsaspekt beschränkt werden. Sie brauchen es jedoch keineswegs. Statt dessen läßt sich auch ein weiter gefaßter Effizienzbegriff zugrundelegen, der auch den Ressourceneinsatz für Bereitstellung und Betrieb der Automatischen Informationsverarbeitungssysteme in Rechnung stellt, auf denen die Konstruktions- und Auswertungsmethoden in nebenläufiger Weise implementiert werden. Dabei spielt erstens derjenige Ressourcemeheinsatz eine Rolle, der für die Realisierung eines Multiprozessorsystems gegenüber einem konventionellen Informationsverarbeitungssystem mit nur einem Prozessor zusätzlich anfällt. Zweitens gilt es denjenigen Ressourcenverzehr zu beachten, der für die Koordinierung der nebenläufig geschehenden Prozessoroperationen anfällt. Aus der Perspektive eines derart erweiterten Effizienzbegriffs wären die Effizienzgewinne infolge beschleunigter Methodenanwendung und die Effizienzverluste aufgrund zusätzlichen Ressourceneinsatzes gegeneinander aufzuwiegen. Die Gegenläufigkeit dieser beiden effizienzbeeinflussenden Aspekte klingt auch bei LOWAS (1986b), S. 1692, an, der zwei komplementäre Optimierungsprobleme formuliert: Entweder gilt es, eine Methode bei vorgegebener Ausführungsgeschwindigkeit mit minimalem Ressourceneinsatz zu implementieren. Oder aber es wird angestrebt, bei festliegendem Ressourceneinsatz die damit erzielbare Ausführungsgeschwindigkeit einer Methode zu maximieren.

Schließlich müßten bei einer umfassend angelegten Effizienzbetrachtung auch mögliche Veränderungen des Analysenutzens Berücksichtigung finden. Diese können bei der Auswertung von Erreichbarkeitsgraphen vor allem dadurch eintreten, daß eine Obergrenze maximaler Auswertungsdauer - ein "timeout" - vorgegeben wird. Dann kann die nebenläufige Implementierung einer Auswertungsmethode aufgrund ihres Geschwindigkeitsvorteils für ein Auswertungsproblem tendenziell mehr Problemlösungen (Auswertungsergebnisse) generieren, als es mit der sequentiellen Implementierung der gleichen Auswertungsmethode möglich wäre. Diese Tendenz wirkt sich um so stärker aus, je knapper die maximale Auswertungsdauer bemessen ist. Die Nutzenvariabilität trifft allerdings nur zu, falls erstens für ein Auswertungsproblem *mehrere* Lösungen existieren und zweitens ein Interesse daran besteht, möglichst viele dieser Problemlösungen als Auswertungsergebnisse zu erhalten. Beispielsweise läßt sich an die Bestimmung von Netzoptima \underline{sw}^* denken, die durchaus durch mehrere verschiedene \underline{sw}^* -optimale Schaltwege im Erreichbarkeitsgraphen eines Netzmodells realisiert werden können. Die voranstehend skizzierten Vorbehalte gegenüber der eingeschränkten Perspektive, die Effizienz einer Methodenimplementierung mit ihrer Ausführungsgeschwindigkeit gleichzusetzen, gelten für alle nachfolgenden Effizienzbetrachtungen. Gelegentlich wird darauf explizit zurückgekommen.

Der effizienzmindernde Einfluß der Operatorenkoordinierung wird von GILOI (1978), S. 295f.; IMAI (1979), S. 418; DENNIS, J. (1980), S. 55; BOLEY (1981), S. 153; KORNFELD (1981a), S. 579; DEERING (1984), S. 77; FORGY (1984), S. 118f.; ZELEWSKI (1986c), S. 2; IBARAKI (1987), S. 545, herausgestellt. Vgl. dazu auch die Anmerkungen bei ZELEWSKI (1986a), S. 329f., in denen die Koordinierung von Bereichsexperten einer Blackboard-Architektur als ein betriebswirtschaftliches Ressourcenzuordnungs- und Terminplanungsproblem dargestellt wird. Diese Blackboard-Architektur stellt einen Sonderfall paralleler Automatenarchitekturen dar, da auch ihre Bereichsexperten nebenläufig operieren können. Darauf wird später zurückgekommen.

23) Auf die entsprechenden Nebenläufigkeitspotentiale der Auswertung von Erreichbarkeitsgraphen wird dann nur noch kurz hingewiesen.

24) Die Begriffe "Konstruktionsprozedur" und "Konstruktionsalgorithmus" werden in dieser Arbeit synonym verwendet. Allenfalls überwiegt der erste Begriff in Kontexten der Automatischen Informationsverarbeitung, während der zweite in implementierungsunabhängigen Zusammenhängen bevorzugt wird. Da kein Unterschied zwischen nebenläufigen und nondeterministischen Algorithmen besteht, kann wegen der vorgenannten Synonymität eine nebenläufige Konstruktionsprozedur ebenso als nondeterministische Konstruktionsprozedur angesprochen werden.

25) Die Bedingung $A \geq 2$ muß ergänzt werden. Denn die nachfolgend vorgestellte zweistufig-hierarchische Konstitution von Agentengruppen ist nur dann wohldefiniert, wenn mindestens zwei Agenten definiert sind. Der erste Agent bildet allein die "Gruppe" der übergeordneten Agenten, während der zweite Agent die "Subgruppe" derjenigen Agenten bildet, die dem ersten Agenten untergeordnet sind. Mit der Agentenanzahl $A=1$ wäre diese hierarchische Agentengruppierung nicht mehr aufrechtzuerhalten.

26) Damit wird nur die potentielle Nebenläufigkeit der Agentenoperationen ausgedrückt, die aufgrund des Konstruktionsalgorithmus für Erreichbarkeitsgraphen "im Prinzip" zutrifft. In welchem Ausmaß dieses Nebenläufigkeitspotential tatsächlich verwirklicht wird, hängt von der jeweils realisierten Implementierung des Algorithmus ab. Darauf wurde schon hingewiesen. Insbesondere können alle Agenten durch Softwaremodule realisiert werden, die auf einem konventionellen Automatischen Informationsverarbeitungssystem mit einer Monoprozessor-Architektur implementiert wurden. Auch dies wurde schon angesprochen. Daher ist es u.a. auch möglich, die mindestens zwei Agenten ($A \geq 2$) mit der Hilfe von nur einem Prozessor zu implementieren.

27) Die Terminologie ist uneinheitlich. Von Dämonen sprechen beispielsweise PACINI (1983), S. 862ff.; ZELEWSKI (1986c), S. 20 u. 33ff.; INTELLICORP (1989), S. 17, und VON BECHTOLSHEIM (1991), S. 38. Oftmals werden die Bereichsexperten auch als "knowledge sources" bezeichnet. Diese "Wissensquellen" stellen bereichsspezifisches Anwendungswissen dar, über das die Bereichsexperten verfügen.

28) Das Blackboard-Konzept wurde bereits kurz im Zusammenhang mit der opportunistischen Prozeßkoordinierung vorgestellt.

29) Die Koordinierung der Bereichsexperten wird oftmals auch als deren Synchronisation angesprochen.

30) Von einer losen Kopplung wird gesprochen, wenn Aktoren im Prinzip frei operieren können, jedoch in ihren Operationsspielräumen durch gemeinsame Restriktionen eingeschränkt werden. Der jeweils aktuelle Inhalt der Informationsbasis stellt im Blackboard-Konzept eine solche gemeinsame, im Zeitablauf veränderliche Restriktion dar. Denn ihre aktuellen Informationsinhalte legen fest, welche Aktoren welche Operationen in einem Zustand der Informationsbasis ausführen können. Eine starke Kopplung von Aktoren liegt dagegen vor, wenn den Aktoren unmittelbar vorgeschrieben wird, welche Operationen sie als jeweils nächste ausführen sollen. Die Operationsanweisungen können von einer übergeordneten Koordinierungsinstanz stammen oder auch durch wechselseitigen Aktorenaufruf ausgetauscht werden.

Ein detailliert ausgearbeitetes Konzept der losen Kopplung hat z.B. WEDDE vorgelegt; vgl. dazu WEDDE (1973), S. 372ff.; WEDDE (1975); WEDDE (1977), S. 549ff.; WEDDE (1980a), S. 97ff.; WEDDE (1980b), S. 5ff.; WEDDE (1980d), S. 19ff.

Es existieren auch Berührungspunkte zwischen WEDDE's Konzept lose gekoppelter Systeme und dem Petrinetz-Konzept. Darauf wird hier aber nicht weiter eingegangen, weil es keinen Beitrag zur Konstruktion von Erreichbarkeitsgraphen erkennen läßt. Das Konzept der losen Kopplung liegt auch den Kontraktnetzen zugrunde, die später im Zusammenhang mit Optimierungsnetzen angesprochen werden. Vgl. zur losen Kopplung der informationsverarbeitenden Knoten von Kontraktnetzen DAVIS,R. (1981), S. 5, 7 u. 12ff.; DAVIS,R. (1983), S. 65, 67 u. 75ff.

31) Wie viele Agenten tatsächlich zur Verfügung stehen und wie diese auf die Agentengruppen verteilt werden, hängt davon ab, wie der Konstruktionsalgorithmus konkret implementiert wird. Solche Implementierungsspezifika werden hier jedoch nicht weiter berücksichtigt.

32) Die Restriktion $AU_{\bar{i}} \leq 0,5 \cdot A$ könnte wegen $AU_{\bar{i}} \in \mathcal{N}_+$ bei $A=1$ überhaupt nicht erfüllt werden. Dies verdeutlicht noch einmal, daß die o.a. Bedingung $A \geq 2$ für die hierarchische Agentengruppierung erforderlich ist.

33) In dieser Hinsicht wird die zuvor erfolgte Feststellung, die Agenten seien "im Prinzip" voneinander unabhängig, nachträglich eingeschränkt.

34) Daher gilt: $A\bar{U} + (\sum_{\bar{i} \in \{1, \dots, A\bar{U}\}}: AU_{\bar{i}}) = A$. Diese Restriktion stellt einerseits sicher, daß in den Agentengruppen nicht mehr Agenten vorgesehen werden, als insgesamt gemäß der Agentenanzahl A definiert sind. Andererseits wird aber auch dafür gesorgt, daß alle vorhandenen Agenten bei der Konstruktion eines Erreichbarkeitsgraphen in den Agentengruppen zum Einsatz gelangen können. Von vornherein "tote" Agenten werden also ausgeschlossen.

35) Eine Pseudocode-Notation wurde bereits für die formale Definition des allgemeinen Übergangsschemas $\bar{U}\bar{S}$ eingeführt. Vgl. dazu auch die Quellen, die in der zugehörigen Anmerkung angeführt wurden. Im folgenden wird jedoch eine abweichende Notation benutzt. Für die hier vorgenommenen Modifizierungen sprechen im wesentlichen vier Gründe.

Erstens werden die Pseudocode-Module durch reservierte Ausdrücke und Modulnamen ausführlicher gekennzeichnet, um den Pseudocode für die Konstruktion von Erreichbarkeitsgraphen übersichtlicher zu gestalten. Zweitens wird die frühere Anweisungsaufzählung durch das Codemodul "DO ... ENDDO" ausdifferenziert in einerseits sequentiell und andererseits nebenläufig auszuführende Anweisungsmengen. Drittens werden neuartige Konstrukte wie das Schleifen- und das Fallunterscheidungsmodul eingeführt, um die Ausdruckskraft des Pseudocodes zu erhöhen. Viertens trägt die Anlehnung an die Programmiersprache ADA tendenziell dazu bei, die Verständlichkeit des

Pseudocodes für Rezipienten zu erhöhen, die sich vornehmlich mit der Gestaltung hockkomplexer, intern nebenläufiger Softwaresysteme beschäftigen. Denn die Sprache ADA wurde speziell für die Programmierung solcher Softwaresysteme entwickelt. Infolge ihrer Unterstützung durch das U.S.-Verteidigungsministerium kann erwartet werden, daß die Verbreitung dieser Programmiersprache in Zukunft stark anwachsen wird. Vgl. SAMMET (1987), S. 191 u. 197ff., zu einer ausführlicheren Würdigung der Sprache ADA.

Die formalsprachlichen Konstrukte und reservierten Ausdrücke, die in der hier vorgelegten Pseudocode-Notation Verwendung finden, werden im folgenden kurz erläutert. Sie beruhen gemeinsam auf Anweisungen, die selbst nicht im Pseudocode definiert sind. Eine Anweisung ist entweder eine atomare oder aber eine zusammengesetzte Anweisung. Die atomaren Anweisungen werden durch selbsterklärende, formal- oder natürlichsprachlich formulierte Anweisungsvorschriften spezifiziert. Dabei wird vorausgesetzt, daß sich alle atomaren Anweisungen in einer beliebigen Programmiersprache mit imperativer Semantik implementieren lassen. Die zusammengesetzten Anweisungen stellen Prozeduren dar, die aus den atomaren Anweisungen aufgebaut sind. Jede zusammengesetzte Anweisung wird durch ein Pseudocode-Modul definiert. Das Pseudocode-Modul besitzt einen eindeutig identifizierenden, natürlichsprachlichen Namen "prozedurname". Die Moduldefinition geschieht in rekursiver Weise: Jedes Modul enthält mindestens eine Subprozedur. Jede Subprozedur besteht entweder aus einer atomaren Anweisung, oder sie stellt ein beliebiges der anschließend definierten Pseudocode-Module dar.

"SEQ <prozedurname> BEG; <subprozedur₁>; ...; <subprozedur_p>; SEQ <prozedurname> END" kennzeichnet eine sequentielle Prozedur. Der Prozedurrumpf besteht aus den Subprozeduren "subprozedur_p" mit $p \in \{1, \dots, P\}$, $P \in \mathcal{N}_+$ und $P \geq 2$. Die Subprozeduren werden bei der Abarbeitung der sequentiellen Prozedur jeweils genau einmal ausgeführt. Dabei wird die Ausführungsreihenfolge durch die Subprozedurliste des Prozedurrumpfes - beginnend mit der ersten "subprozedur₁" - eindeutig festgelegt.

"LOOP <prozedurname> BEG; EXIT WHEN <bedingung>; <subprozedur>; LOOP <prozedurname> END" kennzeichnet eine iterative Prozedur. Die Subprozedur "subprozedur" aus dem Prozedurrumpf wird so lange ausgeführt und dabei beliebig oft wiederholt, bis die Abbruchbedingung "bedingung" im EXIT WHEN-Submodul erfüllt ist. Falls diese Abbruchbedingung von Anfang an erfüllt ist, wird die Subprozedur überhaupt nicht ausgeführt.

"IF <prozedurname> BEG; COND <bedingung>; THEN <subprozedur_{be}>; ELSE <subprozedur_{bne}>; IF <prozedurname> END" kennzeichnet eine einfache Fallunterscheidung. Falls die Bedingung "bedingung" des COND-Submoduls erfüllt ist, wird die Subprozedur "subprozedur_{be}" aus dem THEN-Submodul ausgeführt. Wenn die Bedingung "bedingung" des COND-Submoduls dagegen nicht erfüllt ist, wird die Subprozedur "subprozedur_{bne}" aus dem ELSE-Submodul ausgeführt. Auf das ELSE-Submodul läßt sich verzichten, sofern keine Anweisung ausgeführt werden soll, falls die Bedingung "bedingung" nicht erfüllt ist.

"CASE <prozedurname> BEG; WHEN <bedingung₁> <subprozedur₁>; ...; WHEN <bedingung_p> <subprozedur_p>; CASE <prozedurname> END" kennzeichnet eine komplexe Fallunterscheidung. Dabei wird zwischen P Unterfällen mit $P \in \mathcal{N}_+$ und $P \geq 2$ (im Regelfall $P \geq 3$) differenziert. Jeder Unterfall wird durch die Bedingung "bedingung_p" mit $p \in \{1, \dots, P\}$ so spezifiziert, daß sich alle Unterfälle wechselseitig ausschließen. Wenn seine Bedingung erfüllt ist, dann wird die unterfallsspezifische Subprozedur "subprozedur_p" ausgeführt.

"PAR <prozedurname> BEG; TASK <subprozedur₁>; ...; TASK <subprozedur_p>; PAR <prozedurname> END" kennzeichnet eine parallele Prozedur. Sie wird auch als aufgabenorientierte parallele Prozedur bezeichnet, um sie von einer zweiten, weiter unten vorgestellten parallelen Prozedur abzugrenzen. Die aufgabenorientierte parallele Prozedur setzt sich aus mehreren Subprozeduren "subprozedur_p" mit $p \in \{1, \dots, P\}$, $P \in \mathcal{N}_+$ und $P \geq 2$ zusammen. Jede Subprozedur wird bei der Abarbeitung der parallelen Prozedur genau einmal ausgeführt. Soweit entspricht die aufgabenorientierte parallele der o.a. sequentiellen Prozedur. Der wesentliche Unterschied zwischen beiden Prozedurarten besteht jedoch darin, daß die Subprozeduren der parallelen Prozedur in keiner festen Reihenfolge angeordnet sind. Statt dessen sind die Ausführungen der Subprozeduren der parallelen Prozedur in keiner festen Reihenfolge angeordnet. Die Subprozedurausführungen können daher nebenläufig geschehen. Solche nebenläufig ausführbaren Subprozeduren werden der Klarheit halber auch als "Aufgaben" bezeichnet. Das Ausführen einer Subprozedur stellt dann eine Aufgabenerfüllung dar.

Jede aufgabenorientierte parallele Prozedur entspricht aus der Perspektive des Petrinetz-Konzepts einem Schaltschritt, dessen nebenläufig aktivierten Transitionen ("Subprozeduren") ebenso nebenläufig geschaltet ("ausgeführt") werden. Da sich der Begriff der Nebenläufigkeit in den Notationen von Pseudocodes und Programmiersprachen noch nicht durchzusetzen vermochte, wird der reservierte Ausdruck "PAR" für das "parallele" Ausführen der Subprozeduren benutzt. Parallelität der Subprozedurausführung wird hier aber im Sinne der Nebenläufigkeit des Petrinetz-Konzepts ausgelegt: Die Subprozeduren dürfen gleichzeitig (auf entsprechend vielen Prozessoren), zeitlich beliebig überlappt (auf mehreren Prozessoren), zeitlich verschachtelt (bei Time-sharing-Betrieb) oder auch in jeder beliebigen Reihenfolge (auf einem Prozessor) ausgeführt werden. Die Parallelität der Subprozedurausführung hängt von der Architektur des jeweils zugrundeliegenden Automatischen Informationsverarbeitungssystems ab. Insbesondere spielt die Anzahl der Prozessoren eine Rolle, auf denen in jedem Zeittakt an der Ausführung von jeweils einer Anweisung gearbeitet werden kann.

"PAR <prozedurname> BEG; EXIT WHEN <bedingung>; AGENT₁ CALL <subprozedurname>; ...; AGENT_A CALL <subprozedurname>; TASK <subprozedurname> <subprozedur>; PAR <prozedurname> END" ist eine spezielle Variante paralleler Prozeduren. Sie wird hier als agentenorientierte parallele Prozedur bezeichnet. Zwar findet sie

sich weder in der Programmiersprache ADA noch in den anderen, oben belegten Pseudocode-Notationen wieder. Aber sie erlaubt es, die Konstruktion von Erreichbarkeitsgraphen so zu formulieren, daß zwei Aspekte Berücksichtigung finden. Erstens lehnt sich die Prozedurvariante eng an das Blackboard-Konzept an, dessen Bedeutung für die Erforschung Künstlicher Intelligenz bereits oben anlässlich der Einführung informationsverarbeitender Agenten angesprochen wurde. Daher bildet die agentenorientierte Prozedurvariante einen zentralen Anknüpfungspunkt, um in die Erreichbarkeitsanalyse von Netzen befruchtende Impulse der KI-Forschung einfließen zu lassen. Zweitens wird die nebenläufige Implementierung der Konstruktionsmethode für Erreichbarkeitsgraphen unterstützt, die bereits als wünschenswert angesprochen wurde. Dabei weist das Unterstützungspotential über die nebenläufige Ausführung von Subprozeduren hinaus, die bereits durch die voranstehend erläuterten, aber aufgabenorientiert definierten parallelen Prozeduren vorgesehen wird. Denn im aufgabenorientierten Fall werden *verschiedene* Subprozeduren nebenläufig - und jeweils genau einmal - ausgeführt. Dagegen wird in der hier präsentierten agentenorientierten Variante die *gleiche* Subprozedur "subprozedur" durch unterschiedliche informationsverarbeitende Einheiten beliebig oft - d.h. keimmal, einmal oder mehrmals - ausgeführt. Falls sie mehrere Male ausgeführt wird, so geschieht dies wiederum in nebenläufiger Weise.

Bei den Informationsverarbeitungs-Einheiten $AGENT_a$ mit $a \in \{1, \dots, A\}$ und $A \in \mathcal{N}_+$ handelt es sich um teilautonome "Agenten". Sie verhalten sich insofern autonom, als sie ihre Operationen kausal unabhängig voneinander ausführen. Dabei ruft jeder Agent die eine auszuführende Subprozedur zunächst durch den reservierten Ausdruck CALL und ihren Namen "subprozedurname" auf. Danach führt er sie durch Abarbeitung aller Anweisungen aus, die in dieser Subprozedur enthalten sind. Die Autonomie der Agenten wird jedoch dadurch eingeschränkt, daß sie auf eine gemeinsame Informationsbasis - die subprozedurspezifische Agenda - zugreifen. Die Agenda ist eine Informationsmenge AG. Sie enthält alle Ausgangsinformationen "info", die zur Initialisierung der Subprozedurausführung in Betracht kommen. Wird die gleiche Subprozedur mit unterschiedlichen Ausgangsinformationen initialisiert, so werden die Resultate der Agentenoperationen im allgemeinen voneinander abweichen. Dabei ist es zunächst unerheblich, ob die Subprozedurausführungen von verschiedenen Agenten oder von demselben Agenten vorgenommen werden. Allerdings können die Subprozedurausführungen nur in dem Ausmaß nebenläufig erfolgen, wie verschiedene Agenten beteiligt sind. Denn jeder Agent vermag zur selben Zeit immer nur eine Subprozedur auszuführen. Immerhin ist es jedoch möglich, daß derselbe Agent die gleiche Subprozedur mehrmals nacheinander - jeweils mit unterschiedlichen Ausgangsinformationen - ausführt. Die Agenda kann auch nur eine oder überhaupt keine Ausgangsinformation enthalten. Dann wird die Subprozedur im ersten Fall durch einen Agenten mit der einen Ausgangsinformation genau einmal ausgeführt. Im zweiten Fall wird dagegen keiner der Agenten aktiv, so daß die Subprozedur keimmal ausgeführt wird.

Eine Synchronisationstechnik sorgt dafür, daß die Subprozedur mit jeder ihrer Ausgangsinformationen genau einmal initialisiert und dann mit dieser Ausgangsinformation genau einmal ausgeführt wird. Wie die Synchronisation im Detail ausgestaltet wird, interessiert hier nicht näher; vgl. dazu die Erläuterungen zur Reservierung des Zugriffs auf die Knotenmenge UKNAK und zur Synchronisationsstelle "Synchro". Statt dessen reicht es hier aus, die Funktionsweise der Synchronisationstechnik zu skizzieren. Mit ihrer Hilfe wird der Zugriff auf die Agenda in jedem Zeitpunkt nur höchstens einem Agenten gestattet, damit nicht mehrere Agenten die gleiche Subprozedur mit derselben Ausgangsinformation initialisieren können. Falls ein Agent tatsächlich zugreift, wählt er eine Ausgangsinformation "info" aus der Agenda aus und eliminiert sie aus ihrer Informationsmenge AG. Danach gibt er den Zugriff auf die Agenda für die anderen Agenten wieder frei und fährt mit der Ausführung der Subprozedur fort. Nach endlich vielen Agentenzugriffen auf die Agenda enthält sie keine Ausgangsinformationen mehr. Die Abbruchbedingung "bedingung" aus dem EXIT WHEN-Submodul wird so formuliert, daß sie genau dann erfüllt wird, wenn die Informationsmenge der Agenda leer ist. Dies gewährleistet u.a., daß die parallele Prozedur sofort wieder verlassen wird, falls ihre Agenda von vornherein - zufällig - keine Ausgangsinformationen enthält. Wenn die Agenda zunächst nicht leer war, gilt dagegen: Sobald die Abbruchbedingung "bedingung" eingetreten ist, führen alle Agenten, die in diesem Zeitpunkt an einer Ausführung der Subprozedur arbeiten, ihre informationsverarbeitenden Operationen noch zu Ende. Andere Agenten können keine Ausführung der Subprozedur mehr übernehmen, weil die leere Agenda keine Ausgangsinformationen enthält, die zur Initialisierung der Subprozedurausführung erforderlich wären. Haben alle Agenten ihre Subprozedurausführungen abgeschlossen, so ist die parallele Prozedur beendet.

Beispielsweise kann eine Subprozedur aus der Aufgabe bestehen, einen aktivierten Schaltschritt auszuführen. Für die Untersuchung eines Markierungsknoten M_i in einem Erreichbarkeitsgraphen läßt sich eine agentenorientierte parallele Prozedur spezifizieren, die alle dort aktivierten Schaltschritte hypothetisch ausführen soll, um die direkt erreichbaren Folgeknoten zu ermitteln. Wenn SSM_i die Menge aller Schaltschritte SS_a ist, die im Markierungsknoten M_i aktiviert sind, so besteht die Agenda der parallelen Prozedur aus der Schaltschrittmenge SSM_i . Sie stellen diejenigen Ausgangsinformationen dar, mit denen die Subprozedur, einen aktivierten Schaltschritt SS_a unter der Markierung M_i auszuführen, initialisiert werden kann. Ein Agent übernimmt eine Ausführung dieser Subprozedur, indem er sie mit der Ausgangsinformation über einen Schaltschritt initialisiert und diesen Schaltschritt aus der Agenda streicht. Anschließend führt der Agent die Anweisungen aus, die für die Subprozedur, einen Schaltschritt auszuführen, in Pseudocode-Notation festgelegt wurden. Wenn alle aktivierten Schaltschritte aus der Menge SSM_i durch Agenten ausgeführt worden sind, ist die Agenda leer. Dann ist die zugehörige parallele Prozedur abgeschlossen.

Die Agenten können jeweils durch einen Prozessor in einem Automatischen Informationsverarbeitungssystem mit Multiprozessor-Architektur verwirklicht werden. Dies muß jedoch keineswegs der Fall sein. Statt dessen können die Agenten auch als Softwaremodule realisiert sein, die auf einem konventionellen Informationsverarbeitungssystem mit Monoprozessor-Architektur (VON NEUMANN-Architektur) sequentiell nacheinander abgearbeitet werden. In Übereinstimmung mit dem bereits erwähnten Postulat, Konzepte implementierungsunabhängig zu formulieren, erfolgt hier keine Festlegung darauf, wie die Agenten später implementiert werden. Darüber hinaus wird die Agentenanzahl A nicht konkret bestimmt, sondern offengelassen. Daher ist es möglich, die Agentenanzahl flexibel an die Anzahl der tatsächlich vorhandenen Prozessoren anzupassen, falls die Agenten tatsächlich durch jeweils einen Prozessor implementiert werden.

Abschließend wird darauf hingewiesen, daß die zuvor erläuterte Variante paralleler Prozeduren die Gestaltungsmächtigkeit des Blackboard-Konzepts bei weitem nicht ausschöpft. Denn hier wurden Agenten nur zur nebenläufigen Ausführung der *gleichen* Subprozedur mit jeweils unterschiedlichen Ausgangsinformationen eingesetzt. Die Agenten des Blackboard-Konzepts können dagegen ebenso *verschiedene* Subprozeduren nebenläufig ausführen. Daher decken die Agenten des Blackboard-Konzepts ebenso die zuerst angeführten, aufgabenorientierten parallelen Prozeduren ab. Folglich lassen sie sich nutzen, um *alle* hier vorgestellten parallelen Prozeduren auf Automatischen Informationsverarbeitungssystemen mit Blackboard-Architekturen zu implementieren. Dies unterstreicht nochmals den oben erfolgten Hinweis, daß die Agenten einen zentralen Ansatzpunkt bilden, um in die Erreichbarkeitsanalyse von Netzen Erkenntnisse aus der Erforschung Künstlicher Intelligenz einzubinden.

Die beiden o.a. Varianten paralleler Prozeduren könnten in einer komplexer formulierten, aber auch ausdrucksreicheren parallelen Prozedur zusammengefaßt werden. Sie nähme die Gestalt an: "PAR <prozedurname> BEG; EXIT WHEN <bedingung>; AGENT₁ CALL <subprozedurname_{p(1)}>; ...; AGENT_{AZ} CALL <subprozedurname_{p(A)}>; TASK <subprozedurname₁> <subprozedur₁> ; ... ; TASK <subprozedurname_p> <subprozedur_p> END; PAR <prozedurname> END". Dabei müßte für jede der insgesamt P verschiedenen Subprozeduren mit den Namen "subprozedurname_p", $p \in \{1, \dots, P\}$ und $P \in \mathcal{N}_+$ eine subprozedurspezifische Agenda vorgesehen werden. Die Anzahl P der unterschiedlichen Subprozeduren und die Agentenanzahl A brauchen nicht übereinzustimmen. Zwei unterschiedlichen Agenten AGENT_{a1} und AGENT_{a2} mit den Subprozeduren "subprozedurname_{p(a1)}" bzw. "subprozedurname_{p(a2)}" dürfte mit $p(a_1) = p(a_2)$ durchaus die gleiche Subprozedur zugeordnet werden. Die Abbruchsbedingung "bedingung" wäre so zu formulieren, daß sie genau dann erfüllt ist, wenn *alle* subprozedurspezifischen Agenden leer sind. Hierdurch ließe sich das Gestaltungspotential des Blackboard-Konzepts vollständig ausschöpfen. Dennoch wird auf den Gebrauch einer derart verallgemeinerten parallelen Prozedur in dieser Arbeit verzichtet, weil sie an keiner Stelle erforderlich ist.

36) Die nebenläufige Erfüllung von Konstruktionsschritten (Subprozeduren) erlaubt es, alle artifizialen Sequentialisierungen von kausal unabhängigen Konstruktionsschritten zu vermeiden. Dadurch werden Programme, die in der hier vorgestellten Pseudocode-Notation konzipiert wurden, für eine spätere Implementierung auf Automatischen Informationsverarbeitungssystemen mit einer nebenläufigen ("parallelen") Architektur vorbereitet. Da jedoch keine Festlegung auf eine bestimmte Systemarchitektur erfolgt, wird auch keine Implementierungsweise fixiert. Statt dessen wird nur das Potential offengehalten, die nebenläufige Erfüllung einer Gruppe von kausal unabhängigen Konstruktionsschritten später jeweils so zu implementieren, wie es die Architektur des zugrundeliegenden Automatischen Informationsverarbeitungssystems zuläßt. Dies kann von der konventionellen sequentiellen Abarbeitung der Konstruktionsschritte bis zu ihrer zeitgleichen Erfüllung durch Multiprozessor-Systeme reichen.

37) Vgl. SOWA (1984), S. 398ff.

38) Die Bezeichnung ADA erinnert an eine Pionierin der automatischen Informationsverarbeitung: Auguste Ada Byron, Countness of Lovelace (1815-1852). Nähere Beschreibungen dieser Programmiersprache finden sich bei SAMMET (1986), S. 722ff.; SCHWÄRTZEL (1986), S. 349ff.; SAMMET (1987), S. 191ff.; POMBERGER (1987), S. 181ff.

39) Darauf wird beim u.a. Konstruktionsalgorithmus in der Anweisung zurückgegriffen, für eine gegebene Netzmarkierung M_r die Menge aller Schaltschritte zu bestimmen, die unter dieser Markierung aktiviert sind. Diese Anweisung wird im Algorithmus als atomar behandelt. Sie müßte aber bei einer konkreten Algorithmusimplementierung durch einen Anweisungskomplex ersetzt werden, der die Schaltregel des zugrundeliegenden Netzes implementiert. Dabei könnte der Anweisungskomplex einen Generierungs- und Test-Zyklus umfassen, in dem alle denkmöglichen Schaltschritte erzeugt und hinsichtlich ihrer Aktivierung unter der aktuellen Markierung überprüft werden. Statt dessen ließe sich der Anweisungskomplex aber auch so auslegen, daß er von vornherein genau alle aktivierten Schaltschritte erzeugt. Von diesen implementierungstechnischen Details wird jedoch in der eingangs angesprochenen Anweisung des Konstruktionsalgorithmus abstrahiert.

40) Die Beschriftungsfunktion bk mit $bk: KA_{RG} \rightarrow SSM$ wird hier nicht in der üblichen Weise durch die explizite Angabe ihrer Abbildungsvorschrift ausgegeben. Statt dessen wird ihre Abbildungsvorschrift implizit festgelegt durch die Erfüllungsmenge BK der Beschriftungsfunktion. Diese Menge umfaßt genau alle Paare, welche die Abbildungsvorschrift der Beschriftungsfunktion bk erfüllen. Die Paare bestehen in ihrer ersten Komponente jeweils aus einer Kante $ka_{r,a,f} = (M_r, M_f)$, die zur Kantenmenge KA_{RG} des Erreichbarkeitsgraphen gehört. In ihrer zweiten Komponente umfassen sie jeweils denjenigen Schaltschritt SS_a aus der Schaltschrittmenge SSM , mit dem die Kante $ka_{r,a,f}$

im Erreichbarkeitsgraphen gemäß $bk(ka_{r,a,f}) = SS_a$ beschriftet ist. Mit Hilfe ihrer Erfüllungsmenge BK ist die Beschriftungsfunktion bk definiert durch:

$$\begin{aligned} bk: & \quad KA_{RG} \rightarrow SSM \\ ka_{r,a,f} & \rightarrow bk(ka_{r,a,f}) = SS_a \Leftrightarrow (ka_{r,a,f}, SS_a) \in BK \end{aligned}$$

41) Der Konstruktionsalgorithmus gilt für alle anderen Netze mit wohldefinierter dynamischer Netzstruktur in der gleichen Weise. Es braucht lediglich in der ersten Anweisung eine andere Netzdefinition - für Stelle/Transition-Netze etwa das Tupel $STN = (\dots, M_0)$ - eingelesen zu werden.

42) Alle atomaren Anweisungen werden durch den identifizierenden Zusatz " $anw_q \approx$ " mit $q \in \{1, \dots, 18\}$ gekennzeichnet. Dies bereitet die spätere Repräsentation des Konstruktionsalgorithmus durch ein algorithmisches Netzmodell vor. In ihm werden die Transitionen, die atomare Anweisungen darstellen, mit den anweisungsspezifischen Zusätzen anw_q beschriftet.

43) Die Iteration "globale_Knotenanalyse" kann auf der Abstraktionsebene des hier zugrundegelegten Pseudocodes nicht durch eine parallele Prozedur ersetzt werden. Denn es handelt sich um eine induktive Prozedur, deren (i+1)-te Ausführung (mit $i \in \mathcal{N}$) vom Ergebnis ihrer i-ten Ausführung abhängen kann. Denn in der i-ten Ausführung ist u.U. ein Knoten neu erzeugt und zur Menge UKN der noch zu untersuchenden Knoten ergänzt worden. Dann ist die (i+1)-te Prozedurausführung von der i-ten Prozedurausführung abhängig, falls die erste auf den neuerzeugten Knoten der zweiten zugreift.

Die Iteration könnte nur dann in eine parallele Prozedur transformiert werden, wenn die "EXIT WHEN UKN = \emptyset "-Klausel erweitert würde zu: "EXIT WHEN UKN = \emptyset und keine der parallel abgearbeiteten Subprozeduren 'SEQ Knotenanalyse ...' befindet sich in Bearbeitung". Dieser Zusatz würde sicherstellen, daß eine nur vorübergehende Leerung der Knotenmenge UKN nicht zum vorzeitigen - fehlerhaften - Abbruch der globalen Knotenanalyse führte. Eine solche temporäre Leerung durch UKN = \emptyset träte ein, wenn durch noch mindestens eine in Abarbeitung befindliche Subprozedur "SEQ Knotenanalyse ..." ein neuer zu untersuchender Knoten generiert wird, so daß nach ihrem Ende $\#(UKN) \geq 1$ gilt. Der o.a. Zusatz zur Abbruchbedingung der globalen Knotenanalyse ist aber innerhalb des hier verwendeten Pseudocodes nicht definiert, weil die Zulässigkeit eines solchen Zusatzes von der speziellen Implementierungsumgebung des Algorithmus abhängt. Von diesen Implementierungseinflüssen wird jedoch auf der Pseudocode-Ebene des hier formulierten Algorithmus bewußt abstrahiert, um das o.a. Postulat der Implementierungsunabhängigkeit zu erfüllen. Die Möglichkeit einer nur temporären Leerung der Menge UKN erfordert es auch, zwischen dieser Menge UKN, die jedem Iterationsschritt neu zugrundegelegt wird, und der Menge UKNAK, die jeweils in einem Iterationsschritt aktuell untersucht wird, zu unterscheiden.

44) Die parallele Prozedur "PAR Knotenanalyse ..." ist in den algorithmischen Darstellungen der Netzliteratur im allgemeinen nicht vorgesehen. Denn dort wird gewöhnlich von der impliziten Prämisse ausgegangen, die Konstruktionsalgorithmen würden auf Automatischen Informationsverarbeitungssystemen mit sequentiellen Architekturen implementiert. Hierdurch wird aber eine artifizielle Sequentialisierung vorgenommen, die nicht dem zu lösenden Konstruktionsproblem selbst innewohnt, sondern dem sequentiellen und implementierungsorientierten Denken der Algorithmus-Entwickler entspringt. Von solchen problemfremden Verzerrungen und - wie später aufgezeigt wird - hinderlichen Einschränkungen der Algorithmusformulierung wird hier jedoch Abstand genommen. Statt dessen wird die inhärente Nebenläufigkeit der Aufgabe, mehrere Knoten eines Erreichbarkeitsgraphen hinsichtlich ihrer Folgeknoten zu untersuchen, von vornherein durch die verzerrungsfreie parallele Prozedur "PAR Knotenanalyse ..." wiedergegeben. Sie geht auf einen Ansatz von ZELEWSKI (1986c), S. 45ff., zurück.

Eine rein sequentielle Analyse würde dagegen dann erfolgen, wenn die Prozedur "PAR Knotenanalyse BEG; EXIT WHEN <bedingung>; ... ; PAR Knotenanalyse END" durch eine sequentiell abzuarbeitende Iteration "LOOP Knotenanalyse BEG; EXIT WHEN <bedingung>; ... ; LOOP Knotenanalyse END" derart ersetzt würde, daß beide Prozeduren in ihren Rümpfen "..." jeweils dieselben Anweisungen umfassen.

Im weiteren Verlauf des hier dargestellten Algorithmus wird noch einmal die inhärente Nebenläufigkeit des Problems, einen Erreichbarkeitsgraphen zu konstruieren, benutzt, um die parallele Prozedur "PAR globale_Schrittanalyse ..." einzuführen.

45) Die anschließende Zugriffsreservierung durch jeden Agenten gehört zu der Synchronisationstechnik für nebenläufig operierende Agenten, die bereits in einer früheren Anmerkung angesprochen wurde. Sie wird daher nicht als eine eigenständige Subprozedur, sondern nur als informationstechnisches Artefakt behandelt. (Gleiches gilt für die später angeführte Prozedur "Schaltschrittanalyse".) Die Zugriffsreservierung läßt sich beispielsweise auf der Basis der "locking/unlocking"-Technik der transaktionsorientierten Informationsverarbeitung realisieren. Dort werden Protokolle implementiert, die sicherstellen, daß ein Informationsobjekt für die Ausführung einer Transaktion temporär reserviert und am Ausführungsende für die Zugriffe anderer Transaktionen wieder freigegeben wird. Vgl. zu dieser "locking/unlocking"-Technik ALAGIC (1986), S. 179ff. u. 191ff.

Die Subprozedur "SEQ Knotenanalyse ..." kann hier durch die übergeordneten Agenten $AGENT_i$ mit $i \in \{1, \dots, A\}$ über ihre CALL-Anweisungen mehrfach aufgerufen werden. Bei jedem Aufruf der Subprozedur durch einen Agenten wird sie mit einem Knoten aus der Menge UKNAK initialisiert. Ohne die o.a. Reservierung wäre es möglich, daß sich die Subprozeduraufrufe mehrerer Agenten dadurch überlagern, daß sie die Subprozedur jeweils mit demselben Knoten initialisieren. Dann würde derselbe Knoten mehrfach analysiert. Dies bedeutet auf jeden Fall eine Verschwendung der knappen Ressourcen, die für die Subprozedurausführungen zur Verfügung stehen. Daher werden solche Mehrfachanalysen durch die o.a. Reservierungstechnik von vornherein unterbunden. Deshalb braucht hier auch nicht untersucht zu werden, ob die Mehrfachanalysen darüber hinaus zu inkonsistenten Algorithmusausführungen führen könnten.

Auf welche Weise die Zugriffsreservierung und die zugehörige spätere Zugriffsfreigabe realisiert werden, bleibt der konkreten Algorithmusimplementierung überlassen. Insbesondere muß eine solche Implementierung das Synchronisationsproblem lösen, das entsteht, wenn mehrere der nebenläufig ausgeführten Subprozeduren gleichzeitig eine Zugriffsreservierung versuchen. Dazu kann beispielsweise auf das Semaphoren-Konzept zurückgegriffen werden. Vgl. dazu die Quellen, die zur Behandlung von Synchronisationsproblemen mit der Hilfe von Petrinetzen angeführt wurden.

In der Netzrepräsentation des Konstruktionsalgorithmus, die in Kürze vorgestellt wird, ist das Synchronisationsproblem auf besonders einfache und transparente Weise gelöst: Die Zugriffsreservierung geschieht durch die Marke auf der Synchronisationsstelle, die jeweils nur genau einem Agenten zugeordnet werden kann.

46) Da die Menge SSM_i alle Schaltschritte umfaßt, die unter der Markierung M_i aktiviert sind, wird die universelle Schaltschritt-Strategie angewendet. Diese Schaltstrategie wurde allen Netzen, die in dieser Arbeit behandelt werden, zugrundegelegt.

47) Die atomare Anweisung anw_{12} stimmt mit der o.a. atomaren Anweisung anw_9 überein. Sie können daher bei einer Implementierung des Konstruktionsalgorithmus durch ein gemeinsames Konstrukt berücksichtigt werden. Sie werden hier dennoch als unterschiedliche Anweisungen behandelt, weil sie zu unterschiedlichen Prozeduren gehören. Gleiches gilt für die übereinstimmenden Anweisungen anw_{13} und anw_{10} sowie die ebenso übereinstimmenden Anweisungen anw_{14} und anw_{11} .

48) Die Menge UKN der zu untersuchenden Knoten des Erreichbarkeitsgraphen muß durch diese Anweisung keineswegs auf die leere Menge $UKN = \emptyset$ reduziert werden. Dies mag auf den ersten Blick erstaunen, weil oben die Menge UKNAK der aktuell untersuchten Knoten wegen $UKNAK := UKN$ mit der Menge UKN der zu untersuchenden Knoten identifiziert worden war. Es ist jedoch zu beachten, daß durch die zwischenzeitlich erfolgten Schrittausführungen die Menge UKN der zu untersuchenden Knoten um neue Knoten M_i erweitert worden sein kann. Dies ist immer dann der Fall, wenn die Anweisung $UKN := UKN \cup \{M_i\}$ ausgeführt worden ist.

49) Sie erstreckt sich auf die explizite Agentensynchronisation in parallelen Prozeduren mittels einer Synchronisationsstelle.

50) Die synonymen Begriffe der inhärenten Nebenläufigkeit und des Nebenläufigkeitspotentials werden hier als Grenzbegriffe verstanden. Jeder Algorithmus besitzt zunächst eine realisierte Nebenläufigkeit. Sie besteht in dem Ausmaß, wie es die konkret vorliegende Algorithmusformulierung zuläßt, die algorithmuskonstituierenden atomaren Anweisungen (Operationen) tatsächlich nebenläufig auszuführen. Dies kann bis zur rein sequentiellen Ausführung aller Anweisungen reichen (degenerierte Nebenläufigkeit). Es läßt sich jedoch vorstellen, den Algorithmus mit den gleichen atomaren Anweisungen so zu reformulieren, daß die nebenläufigen Ausführungsmöglichkeiten seiner Anweisungen vergrößert werden. Dabei wird nur der Verarbeitungszusammenhang der atomaren Anweisungen verändert, nicht aber die algorithmusspezifische Menge aller auszuführenden atomaren Anweisungen. Die inhärente Nebenläufigkeit eines Algorithmus drückt das höchste Nebenläufigkeitsausmaß aller Anweisungsausführungen aus, das für einen Algorithmus mit invarianter Anweisungsmenge erzielt werden kann. Dies begründet die Grenzbegriffeigenschaft des Nebenläufigkeitspotentials. Darüber hinaus handelt es sich um einen relativen Grenzbegriff. Denn er gilt immer nur in bezug auf das Wissen, das über alle Reformulierungsmöglichkeiten eines Algorithmus mit unveränderter Anweisungsmenge aktuell zur Verfügung steht. Es wird also nicht ausgeschlossen, daß nachträglich neuartige Reformulierungsoptionen entdeckt werden, die erkennen lassen, daß das Nebenläufigkeitsausmaß von Anweisungsausführungen noch größer als vormals vorgestellt ist.

Darüber hinaus läßt sich der Grenzbegriff inhärenter Nebenläufigkeit von der Betrachtung eines Algorithmus auf die Untersuchung desjenigen Problems übertragen, zu dessen Bewältigung der Algorithmus formuliert wurde. Das Nebenläufigkeitspotential eines Problems ist dann die maximale inhärente Nebenläufigkeit aller Algorithmen, mit denen das Problem überhaupt gelöst werden kann; vgl. ZELEWSKI (1986c), S. 3; ZELEWSKI (1988b), S. 353. Die Gesamtheit dieser Algorithmen ist aber im allgemeinen nicht unbekannt. Selbst wenn die zugehörigen Algorithmen bekannt wären, so wäre immer noch ein "intellectus infinitus" erforderlich, um zu erkennen, daß die Menge dieser Algorithmen tatsächlich die Gesamtheit aller denkmöglichen Algorithmen für das Problem darstellt. Aus beiden Gründen bleibt diese Algorithmen-Gesamtheit im allgemeinen unbekannt. Folglich kann das Nebenläufigkeitspotential eines Problems auch nur *vorläufig* mit der größten inhärenten Nebenläufigkeit aller aktuell bekannten problemlösenden Algorithmen identifiziert werden. Dieses vorläufige Nebenläufigkeitspotential läßt sich nachträglich

immer noch steigern, falls später Algorithmen mit noch größerer inhärenter Nebenläufigkeit bekannt werden. Für das hier erörterte Problem der Konstruktion von Erreichbarkeitsgraphen folgt daraus: Das Nebenläufigkeitspotential der Aufgabe, für ein Netzmodell dessen Erreichbarkeitsgraphen zu konstruieren, wird vorläufig mit der inhärenten Nebenläufigkeit desjenigen Konstruktionsalgorithmus gleichgesetzt, der unter allen bekannten Konstruktionsalgorithmen die größte inhärente Nebenläufigkeit besitzt. Dieser Konstruktionsalgorithmus wird im folgenden entwickelt. Abschließend wird angedeutet, daß sich seine inhärente Nebenläufigkeit durch Übergang zu neuartigen Konstruktionsalgorithmen mit veränderten atomaren Anweisungen durchaus noch erhöhen läßt.

51) Daneben hat die Modellierung des Konstruktionsalgorithmus für Erreichbarkeitsgraphen durch ein Netz auch weitere Erkenntnisse stimuliert, die bereits in seine eingangs vorgestellte Pseudocode-Notation eingeflossen sind. Denn diese pseudocodegestützte Algorithmusformulierung stellt die Fortentwicklung zweier früherer Entwürfe der gleichen nebenläufigen Konstruktion von Erreichbarkeitsgraphen dar. Sie sind in ZELEWSKI (1986c), S. 33ff., insbesondere S. 36ff. u. 45ff., dokumentiert. Ein Vergleich der Pseudocode-Formulierungen läßt unmittelbar die notationellen und algorithmischen Detailverbesserungen erkennen. Als pars pro toto wird nur auf die klarere Notation des Moduls für agentenorientierte parallele Prozeduren verwiesen.

52) Ein anderer Aspekt der heuristischen Konzeptkraft wurde bereits in einer früheren Anmerkung erörtert. Er betraf die Stimulierung einer kohärenten Terminologie für Netzzeigenschaften.

53) Dies wurde schon eingangs in der Rahmenlegung dieser Arbeit angedeutet. Es wird nunmehr anhand des Konstruktionsalgorithmus für Erreichbarkeitsgraphen verdeutlicht. In ihm übernehmen die Subprozeduren die Rolle von Subobjekten, die sowohl kausal voneinander abhängen als auch nebenläufig ausgeführt werden können.

54) Kausale Abhängigkeiten werden im Petrinetz-Konzept zunächst in lokaler Weise durch die Schaltvoraussetzungen und Schaltwirkungen einzelner Transitionen erfaßt. Durch die Flußrelation werden die Transitionen vermittels ihrer Ein- und Ausgangsstellen so miteinander verknüpft, daß die Netztopologie als Ganzes den globalen Kausalzusammenhang der Schaltvoraussetzungen und -wirkungen aller Transitionen widerspiegelt. Vgl. dazu die Ausführungen zur kausalen Interpretation von Synthetischen Netzen.

55) Nebenläufigkeit schlägt sich im Petrinetz-Konzept u.a. als nebenläufige Aktivierung und nebenläufiges Schalten von Transitionen in mehrelementigen Schaltschritten wieder.

56) Auf die kognitiven Vorzüge graphischer Repräsentationsweisen wurde bereits im Zusammenhang mit Kanal/Instanz-Netzen näher eingegangen.

57) Dagegen spielt der Aspekt der informationstechnischen Implementierung hier keine dominierende Rolle. Infolgedessen werden auch nicht die notationellen Konventionen berücksichtigt, die früher für Synthetische Netze eingeführt wurden, um ihre Implementierung durch die Programmiersprache Turbo-PROLOG vorzubereiten. Dies gilt insbesondere hinsichtlich der Notation von Variablen und Nicht-Variablen. In der vorgenannten Programmiersprache müssen ihre Bezeichnungen mit einem Groß- bzw. einem Kleinbuchstaben beginnen. Eine solche Restriktion bleibt jedoch in den nachfolgenden Kanal/Instanz-Netzen unberücksichtigt. Statt dessen wird auf die intuitive Verständlichkeit der Netzbeschriftungen großer Wert gelegt. Daher werden die Anschriften von Netzknoten und -kanten - wie schon bei der zugrundeliegenden Pseudocode-Notation - an die Orthographie der gewöhnlichen Umgangssprache angelehnt.

58) Darüber hinaus wird auch auf die Verfeinerungs- und Vergrößerungskonzepte für Kanal/Instanz-Netze zurückgegriffen. Sie werden intensiv genutzt, um durch die Kombination von Netzmodulen den Konstruktionsalgorithmus für Erreichbarkeitsgraphen in netzartiger Weise zu reformulieren.

59) Da es sich um Kanal/Instanz-Netze handelt, wird die Eigenart ihrer Marken nicht näher präzisiert. Sie können sowohl unstrukturierte als auch beliebig komplex strukturierte Objekte darstellen. Durch die Kantenanschriften wird aber jeweils verdeutlicht, welche Markenarten jeweils gemeint sind. Vgl. dazu vor allem das letzte Netzmodul in Abb. 72. Dort zeigen die Kantenanschriften "AG", daß Marken gemeint sind, die jeweils die aktuelle Informationsmenge einer Agenda "AG" repräsentieren. Dies wird im anschließenden Abschnitt näher erläutert. Solche Kantenanschriften werden im später vorgestellten Netzmodell des Konstruktionsalgorithmus für Erreichbarkeitsgraphen intensiv genutzt, um seinen Marken jeweils einen verarbeitungsbezogenen Informationsgehalt zuzuordnen.

Abgesehen von den Kantenanschriften handelt es sich um eine besonders einfache Ausprägung von Kanal/Instanz-Netzen. Ihre Kanten besitzen das universelle Einheitsgewicht "Eins". Die Markenskapazitäten aller Stellen kann offengelassen werden. Die Netzmodule verhalten sich per constructionem in der gleichen Weise unabhängig davon, ob ihre Stellen unbeschränkte Markenskapazitäten oder aber z.B. das universelle Einheitsgewicht "Eins" zugeordnet erhalten. Die Ausgangsmarkierungen werden durch die Markenverteilungen in den visualisierten Graphen der Netzmodule festgelegt.

Die Konstruktionsweise der Netzmodule stellt sicher, daß in einem Netzmodul genau dann ein Markenfluß einsetzen kann, wenn die Modulstelle, die mit "Input" beschriftet ist, mit einer Marke belegt wird. Der Markenfluß kann in keinem der Netzmodule durch einen Deadlock gestoppt werden oder sich in einer Endlosschleife verfangen, bevor eine Marke auf der Stelle abgelegt wird, die mit "Output" beschriftet ist. Daher führt der Markenfluß, der mit einer

Markierung der "Input"-beschrifteten Stelle startet, immer nach endlich vielen Schaltakten der modulzugehörigen Transitionen zur Markierung der "Output"-beschrifteten Stelle. Dies entspricht einer Verarbeitung von Eingabe- in Ausgabeinformationen, deren Ausführung im Netzmodul durch das Schalten der Transitionen und den hiervon bewirkten Markenfluß repräsentiert wird.

Die Ausgestaltung der Netzmodule garantiert darüber hinaus, daß ein erneuter Markenfluß im Netzmodul durch eine weitere Marke auf seiner Stelle "Input" frühestens dann ausgelöst werden kann, wenn der vorangehende Markenfluß durch Ablegen einer Marke auf der Stelle "Output" abgeschlossen worden ist. Daher werden in den Netzmodulen nur solche Informationsverarbeitungsprozesse abgebildet, die eine Eingabeinformation stets vollständig in eine Ausgabeinformation transformieren, bevor sie mit der Verarbeitung einer neuen Eingabeinformation fortgesetzt werden können. Auf diese Weise werden alle Komplikationen verhindert, die daraus resultieren können, daß die Verarbeitung mehrerer Eingabeinformationen innerhalb derselben Verarbeitungsausführung vermengt werden. Eine solche Konfundierung von Eingabeinformationen würde z.B. im Netzmodul für die agentenorientierte parallele Prozedur (vgl. Abb. 72) stattfinden, wenn nacheinander zwei Marken "AG" auf die Stelle gelangen würden, die mit "Agenda" beschriftet ist. Sie könnten z.B. für zwei *unterschiedliche* Knoten eines Erreichbarkeitsgraphen die Mengen der dort jeweils aktivierten Schaltschritte darstellen. Falls die Agenten auf beide Marken zugleich zugreifen könnten, drohte die Gefahr, daß "erreichbare" Folgeknoten für falsche Referenzknoten ausgewiesen werden. Solche Konfusionen werden jedoch dadurch ausgeschlossen, daß die Ausgangstransition der Stelle "Input" wegen ihrer zweiten Eingangsstelle nach ihrem Schalten so lange nicht wieder aktiviert sein kann, wie nicht die Eingangstransition der Stelle "Output" geschaltet hat. Denn erst durch diesen letztgenannten Schaltakt erhält die zweite Eingangsstelle der Ausgangstransition der Stelle "Input" wieder eine Marke. Erst diese Marke läßt es zu, die Ausgangstransition der Stelle "Input" durch eine neue Marke auf der Stelle "Input" wiederholt zu aktivieren. Die gleiche Wirkung läßt sich auch dadurch erzielen, daß allen Stellen die feste Markenkapazität "Eins" zugeordnet wird. Diese Lösung setzte aber eine genauere Betrachtung des Schaltverhaltens voraus. Dieses Schaltverhalten ist jedoch bei Kanal/Instanz-Netzen nicht wohldefiniert. Daher bevorzugt der Verf. die oben erläuterte erste Vorgehensweise. Darüber hinaus hat sie den Vorteil, daß sich die zuvor erläuterte Funktionsweise der Netzmodule aus den lokalen Umgebungen der Stellen "Input" und "Output" unmittelbar ablesen läßt. Eine Bezugnahme auf intuitiv weniger anschauliche Markenkapazitäten wird dadurch vermieden.

60) Vgl. dazu die Erläuterung der agentenorientierten parallelen Prozedur.

61) Dies wird durch Informationskanten überprüft. Sie sind in der Abb. 72 mit unterbrochenen Linien dargestellt, wie es für die Visualisierung von Synthetischen Netzen vereinbart wurde. Aufgrund dieser Informationskanten kann die Transition "Prozedurname_Ende" erst dann aktiviert sein, wenn die Stellen im Ursprung aller ihrer Informationskanten eine Marke tragen. Jede dieser Stellen ist per constructionem genau dann markiert, wenn der Agent, zu dessen Teilnetz sie gehört, seine Subprozedurausführung beendet hat.

62) Da in Kanal/Instanz-Netzen der Übersichtlichkeit halber auf eine Präzisierung ihrer Marken und ihrer Schaltregeln verzichtet wurde, weisen sie kein wohldefiniertes Schaltverhalten auf. Das wurde bereits dargelegt und gerechtfertigt. Allerdings reicht die Selbsterklärungsfähigkeit von Kanal/Instanz-Netzen aus, um ihre Funktionsweise durch einen intuitiv erfaßten Markenfluß zu veranschaulichen. Genau darauf zielen die oben vorgestellten Netzmodule ab.

63) Vgl. dazu die Voraussetzung wohldefinierter Netzdynamiken für die Erreichbarkeitsanalyse von Netzen. Da die M_0 -Reversibilität auf dem Konzept der Erreichbarkeit beruht, kann sie nicht präzise festgelegt werden, solange eine wohldefinierte Schaltregel fehlt.

64) Die Marken in den Netzmodulen repräsentieren Informationen. Der Markenfluß durch die Netzmodule hängt vom Informationsgehalt der Marken ab. Dies wird besonders deutlich anhand der Fallunterscheidungsmodule. Gleiches gilt für das Modul der agentenorientierten parallelen Prozedur. Dort hängt es vom Informationsgehalt der Marke auf der Stelle "Agenda" ab, ob ein Agent mit einer Ausgangsinformation "info" eine Subprozedurausführung initialisieren kann oder ob die Prozedurausführung an ihr Ende gelangt, weil die Marke auf jener Agendastelle die leere Informationsmenge repräsentiert. Daher müssen die Marken jeweils mit einer internen Struktur vorgestellt werden, die in der Lage ist, Informationen auszudrücken. Solche intern strukturierten Marken lassen sich nur in Höheren Netzen - wie z.B. den hier entwickelten Synthetischen Netzen - erfassen. Daher würde eine Präzisierung des Informationsaspekts notwendig zur Verwendung Höherer Netze führen.

65) Diese Präzisierung wird im Rahmen der hier vorgelegten Untersuchungen nicht vorgenommen, weil kein Interesse an einer vollständigen Ausarbeitung aller informationstechnischen Details besteht. Darauf wurde bereits hingewiesen. Es reicht hier aus, die Funktionsweise des Konstruktionsalgorithmus für Erreichbarkeitsgraphen durch Kanal/Instanz-Netze grob zu veranschaulichen. Denn schon auf diesem Darstellungsniveau lassen sich die hier interessierenden Aspekte herausarbeiten: die Selbstbezüglichkeit des Petrinetz-Konzepts, die interne Nebenläufigkeit des Konstruktionsalgorithmus und die heuristische Kraft von Netzen, die Erkenntnis artifizieller Sequentialisierungen zu stimulieren.

66) Diese Moduleigenschaft mag auf den ersten Blick unerheblich und "technisch" erscheinen. Sie ist aber unverzichtbar, um innerhalb eines Algorithmus eine Prozedur mehrmals ausführen zu können und dabei die Prozedur jedes Mal wieder in einen wohldefinierten Ausgangszustand zu versetzen. Ohne auf diese Eigenschaft zurückzugreifen, lassen sich anspruchsvollere Algorithmen kaum formulieren. Darüber hinaus trifft diese reversible Prozedurausführung auch auf jeden Algorithmus als Ganzes zu, weil er auch selbst eine Prozedur darstellt. M_0 -Reversibilität bedeutet daher ebenso, einen Algorithmus beliebig oft wiederholt anwenden zu können. Dies erscheint so naheliegend, daß die beliebige Wiederholbarkeit von Algorithmusanwendungen im allgemeinen nicht explizit thematisiert wird. Aber die dynamische Netzeigenschaft der M_0 -Reversibilität lenkt den Blick auf diese herausragende Algorithmuseigenschaft. Sie wird durch das nachfolgend vorgestellte Netzmodell des Konstruktionsalgorithmus für Erreichbarkeitsgraphen *explizit* gewährleistet. Denn alle Netzmodule verhalten sich - bei entsprechender Präzisierung als Synthetische oder andere Höhere Netze - per constructionem M_0 -reversibel. Die Modulverknüpfung durch die anschließend erörterte hierarchische Netzverfeinerung beeinflußt diese Reversibilität nicht. Daher verhält sich auch jedes modular aufgebaute Netzmodell in seiner Gesamtheit M_0 -reversibel.

67) Vgl. zu ausführlicheren Thematisierungen von Netzverfeinerungen und -vergrößerungen die Vielzahl der Quellen, die später bei der Beurteilung des Petrinetz-Konzepts angeführt wird.

68) Damit wird nicht behauptet, daß hierarchische Verfeinerungstechniken *immer* gestatteten, die Komplexität von Realproblemen zu bewältigen. Ebenso wenig wird der Anspruch erhoben, eine solche Komplexitätsbeherrschung sei - wenn sie überhaupt möglich ist - *grundsätzlich nur* mittels hierarchischer Modellverfeinerung zu erzielen. Statt dessen wird das viel schwächere Tendenzurteil vertreten, daß sich *bis heute* die hierarchischen Verfeinerungstechniken bei der Behandlung komplexer Problemstellungen in der Modellierungspraxis gegenüber allen anderen *bekannt*en Modellierungstechniken als überlegen erwiesen haben. Es liegt außerhalb des Erkenntnisinteresses dieser Arbeit, alternative Modellierungstechniken näher zu untersuchen oder gar einen operationalen Maßstab für die relative Vorteilhaftigkeit solcher Techniken vorzulegen. Vielmehr wird auf einen kleinen Ausschnitt aus der umfangreichen Literatur verwiesen, die bei der Modellierung komplexer Problemstellungen einhellig die hierarchischen Verfeinerungstechniken bevorzugt.

69) Es lassen sich auch andere Verfeinerungsansätze vorstellen. Vgl. dazu den Hinweis zur Substitution von transitionenberandeten Teilnetzen. Vgl. ebenso die späteren Ausführungen zur Verfeinerung von Makrotransitionen durch "transitionenberandete" Subnetze. Vgl. schließlich ABEL, D. (1990), S. 55. Er stellt den Freiheitsgrad heraus, in einem Netz sowohl dessen Stellen als auch dessen Transitionen durch (entsprechend berandete) Subnetze verfeinern zu können.

70) Die hier benutzte Definition des Netzrandes ist nicht mit der Definition zu verwechseln, die DITTRICH, G. (1989b), S. 6, als Oberfläche, und FEHLING (1990a), S. 4, als Rand eines Netzes thematisieren. Da der Begriff "Netzrand" bereits belegt ist, wird dasjenige Konstrukt, das DITTRICH - zusammen mit EVERTZ-JÄGERS - und FEHLING definieren, der Klarheit halber nur als Netzoberfläche angesprochen.

Zwar dienen Netzrand und Netzoberfläche dem gleichen Zweck, eine systematische Konstruktion für hierarchische Netzverfeinerungen anzubieten. Aber das Konstrukt der Netzoberfläche verfolgt einen vollkommen anderen Verfeinerungsansatz. Er definiert eine Netzoberfläche stets *von außen* aus der Sicht eines umgebenden Netzes, in welches das Netz mit der betrachteten Oberfläche eingebettet ist. Aus dieser Außenperspektive ist die Oberfläche des eingebetteten Netzes die Gesamtheit aller Knoten, die zwei Bedingungen erfüllen: Sie müssen erstens zum eingebetteten Netz gehören. Zweitens muß jeder Knoten aus der Netzoberfläche mit mindestens einem Knoten aus dem umgebenden Netz verknüpft sein. Aufgrund der zweiten Bedingung ist es unmöglich, die Oberfläche eines Netzes "an sich" - also ohne Einbettung in ein umgebendes Netz - zu bestimmen. Die nachfolgend vorgestellte Definition des Netzrandes erlaubt es dagegen, für jedes Netz unabhängig von dessen Einbettung in umfassendere Netze den Netzrand zu bestimmen. Dabei wird ausschließlich auf jene Knoten Bezug genommen, die zum Netz gehören, dessen Rand bestimmt werden soll. Daher wird der Rand eines Netzes von seinem *Innern* her definiert. Aufgrund dieser Innenperspektive verhält sich die Definition des Netzrandes komplementär zur außenperspektivischen Definition der Netzoberfläche.

Die beiden komplementären Definitionen führen selbst dann, wenn eingebettete Netze betrachtet werden, keineswegs zu gleichen Ergebnissen. So besitzt das Beispielnetz, das DITTRICH, G. (1989b), S. 6, präsentiert, die Knotenmenge $\{t_2, s_5, t_4\}$ als Oberfläche. Sein Netzrand besteht dagegen nur aus der einelementigen Knotenmenge $\{t_4\}$. Welche von den beiden komplementären Definitionen bevorzugt wird, hängt von der jeweils verfolgten Verfeinerungsstrategie ab. Im Rahmen dieser Arbeit wird aufgrund ihrer spezifischen Konstruktion von Netzmodulen lediglich die Definition von Netzrändern benötigt. Die Netzmodule sind so ausgelegt, daß ihre Ränder ausschließlich aus Stellen bestehen. Darüber hinaus verhält sich jede dieser Stellen per constructionem eindeutig entweder als Input- oder aber als Outputstelle des betroffenen Netzmoduls. Auf diese Weise läßt sich eine besonders klare Schnittstellenstruktur der Netzmodule verwirklichen. Netzoberflächen lassen dies nicht zu. Zwar können sie so festgelegt werden, daß sie ausschließlich Stellen umfassen; vgl. dazu die "S-Menge" bei DITTRICH, G. (1989b), S. 6. Aber die Definition der Netzoberflächen gestattet es nicht, eine eindeutige Trennung zwischen Stellen mit entweder Input- oder aber Outputcharakter zu erzwingen. Dagegen spielen Netzoberflächen eine Rolle, wenn Netzverfeinerungen

auf der Basis von Netzmorphismen eingeführt werden; vgl. FEHLING (1990a), S. 5. In dieser Arbeit werden Netzmorphismen zwar am Rande erwähnt. Doch sie werden wegen der erheblichen Einschränkungen, denen ihre Anwendung unterliegt, hier nicht zur modularen Netzverfeinerung benutzt. Aus den beiden vorgenannten Gründen wird auf Netzoberflächen vorerst nicht näher eingegangen. Sie werden aber in Kürze verwendet, um die Verfeinerung von Makrotransitionen zu erklären. Später werden sie noch einmal angesprochen, und zwar im Zusammenhang mit der Vergrößerung von Netzen für eine zentrale Systemuhr.

71) Da alle Petrinetze die Verknüpftheitsbedingung ihrer Knoten per definitionem erfüllen, sind sie immer zusammenhängend. Isolierte stellen- oder transitionenartige Knoten können daher nicht vorkommen. Folglich kann es in ihnen keine Knoten mit sowohl leerem Vor- als auch mit leerem Nachbereich geben. Dieser denkmögliche Fall braucht daher hier nicht mehr beachtet zu werden.

72) Der Netzrand eines stellenberandeten Netzes wird auch als Stellenrand bezeichnet.

73) Der Rand eines Netzes kann grundsätzlich auch leer sein. Dies ist genau dann der Fall, wenn jede Stelle und jede Transition jeweils mindestens eine Ein- und mindestens eine Ausgangskante besitzt. Das einfachste Netz mit einem leeren Rand besteht aus genau einer 1-Schleife.

74) Folglich gehören alle Transitionen eines stellenberandeten Netzes zum Netzzinnern. Dies ist aber nur eine Implikation, jedoch keine Definition der Stellenberandetheit eines Netzes. Denn die Forderung, daß in einem stellenberandeten Netz der Netzrand keine Transitionen enthalten dürfte, wird auch durch jedes Netz mit leerem Netzrand erfüllt. In diesem Fall von einem "stellenberandeten" Netz zu sprechen, widerspräche jedoch dem intuitiven Begriffsverständnis.

75) Die oben vorgestellten Netzmodule enthalten der Übersichtlichkeit halber jeweils nur genau eine "Input"- und genau eine "Output"-beschriftete Stelle. Jede dieser beiden Stellen kann aber bei Bedarf beliebig oft dupliziert werden. Dies empfiehlt sich immer dann, wenn hervorgehoben werden soll, daß ein Netzmodul mehrere verschiedenartige Inputs voraussetzt oder Outputs hervorzubringen vermag. Von dieser Modellierungsfreiheit wird beim später vorgelegten Netzmodell des Konstruktionsalgorithmus für Erreichbarkeitsgraphen Gebrauch gemacht; vgl. dazu die Abb. 76 für das Netzmodul der Prozedur "Initialisierung".

76) Entsprechend ihrer Beschriftung werden die beiden Stellen als Input- bzw. Outputstellen ihrer Netzmodule angesprochen. Unter Anspielung auf ihre Eigenschaft, den Netzrand der Module zu konstituieren, kann auch von Input- bzw. Outputschnittstellen geredet werden.

77) Eine ähnliche Synchronisationstechnik sieht ABEL,D. (1990), S. 46f., bei seiner Verfeinerung von (Makro-)Transitionen vor. Allerdings verwendet er nicht zwei Synchronisationsstellen, die einen Modulrahmen bilden, sondern nur eine "Komplementstelle". Sie besitzt die gleiche Wirkung wie die hier bevorzugten zwei Synchronisationsstellen. Insofern erweist sich die eine Komplementstelle als "technisch" ausreichend. Dennoch hat der Verf. eine aufwendigere Konstruktion gewählt, weil sie den Synchronisationsgedanken intuitiv anschaulicher ausdrückt. Denn die beiden Synchronisationsstellen lassen durch ihre aktuelle Markierung unmittelbar erkennen, ob das betrachtete Netzmodul entweder "arbeitsbereit" oder aber "in Arbeit" ist. Der erste Modulzustand liegt genau dann vor, wenn diejenige Synchronisationsstelle markiert ist, die mit der Inputstelle die gemeinsame Ausgangstransition teilt. Der zweite Modulzustand herrscht dagegen genau so lange vor, wie die andere Synchronisationsstelle markiert ist, die mit der Outputstelle die gemeinsame Eingangstransition aufweist.

Darüber hinaus mutet es seltsam an, daß ABEL,D. (1990) - wie zuvor skizziert wurde - zunächst einen Synchronisationsmechanismus einführt, der die oben dargelegte Atomaritätsverletzung einer Makrotransition zu verhindern vermag. Aber wenige Seiten darauf stigmatisiert er diesen Mechanismus als zu restriktiv (S. 57 u. 59). Statt dessen plädiert er am selben Ort dafür, das zeitlich überlappende Mehrfachschalten eines Netzmoduls ("Unternetzes") zuzulassen. Dabei soll dasselbe Netzmodul mit der Ausführung eines neuen Schaltprozesses beginnen dürfen, noch bevor die Ausführung des unmittelbar vorangehenden Schaltprozesses geendet hat. Die modulinternen Schaltprozeßausführungen überlappen daher einander. Wie ABEL dies mit den *atomaren* Schaltakten der Makrotransitionen, die aus der Vergrößerung der Netzmodule hervorgehen kann, vereinbaren möchte, läßt er aber vollkommen unberührt. Anscheinend verfolgt er auf S. 57 u. 59 nur noch die Verfeinerung von Transitionen durch Netzmodule. Dagegen spielt die umgekehrte Richtung der Netzmodulvergrößerung offensichtlich überhaupt keine Rolle mehr. Zwar wirkt das Beispiel, daß ABEL auf S. 59 zur Rechtfertigung seiner Überlappungsforderung heranzieht, aus produktionswirtschaftlicher Perspektive interessant: Er beruft sich auf die überlappende Produktion eines Loses, dessen Werkstücke an mehreren aufeinander folgenden Maschinen zur gleichen Zeit bearbeitet werden. Dieses Beispiel ist für die Demonstration nebenläufigen Schaltverhaltens gut gewählt. Aber die entscheidende Behauptung ABEL's, die nebenläufige Werkstückbearbeitung an mehreren Maschinen lasse sich *nur* durch Netzmodule mit zeitlich überlappendem Mehrfachschalten erfassen (S. 59), trifft nicht zu. Sie bricht schon dann in sich zusammen, wenn die betroffenen Maschinen jeweils durch ein eigenes Netzmodul repräsentiert werden (vgl. dazu die Modellierung von Bearbeitungsstationen in der späteren Fallstudie). Denn in diesem Fall kann die überlappende Losproduktion auf mehreren Maschinen einfach dadurch modelliert werden, daß die Schaltprozesse der maschinenspezifischen Netzmodule nebenläufig zueinander ausgeführt werden. Dies betrifft aber nur das Verhältnis *zwischen* den Netzmodulen.

Dagegen erfolgt *modulintern* weiterhin kein zeitlich überlappendes Mehrfachschalten: Jede modellierte Maschine muß die Ausführung einer zeitlich vorangehenden Bearbeitungsoperation abgeschlossen haben, bevor sie eine nachfolgende Operationsausführung beginnen kann.

78) Das Netzmodul muß mindestens einen solchen zulässigen Modulzustand besitzen, damit sich sein Schaltverhalten durch Markieren der Inputstelle überhaupt einleiten läßt. Es darf aber auch nur höchstens einen solchen zulässigen Modulzustand aufweisen, um unzulässige Verhaltensüberlagerungen zu vermeiden. Solche Verhaltensüberlagerungen würden drohen, wenn das Schaltverhalten des Moduls von mehreren zulässigen Modulzuständen aus angestoßen werden könnte. Dann ließe sich der bereits erwähnte Fall nicht ausschließen, daß das Schaltverhalten eines Netzmoduls ein wiederholtes Mal angestoßen wird, bevor es sein zuletzt eingeleitetes Schaltverhalten beendet hat. Die Synchronisationsstellen sorgen bei den hier vorgestellten Netzmodulen dafür, daß dieser Fall niemals eintreten kann.

79) Vgl. die Netzgraphiken der Abb. 75 bis 88. Dort sind zwar die Ausgangsmarkierungen der gesamten Netzmodule abgebildet. Aber die Markierungen der Input- und Outputstellen hängen auch vom zufälligen Schaltverhalten der Netzumgebungen ab, in denen die Netzmodule eingebettet sind. Daher erstrecken sich die hier diskutierten Ausgangsmarkierungen nur auf die Restmodule, die nach dem Ausblenden ihrer Input- und Outputstellen verbleiben.

Unter diesen Ausgangsmarkierungen ist in jedem Netzmodul diejenige Synchronisationsstelle markiert, die mit der Inputstelle des Netzmoduls deren Ausgangstransition teilt. Daher läßt sich einerseits das Schaltverhalten eines Netzmoduls von seiner Ausgangsmarkierung aus jederzeit durch das Schalten der Ausgangstransition starten, falls sich auf der Inputstelle eine Marke befindet. Andererseits sind die Netzmodule so konstruiert, daß keine andere Markierung als die Ausgangsmarkierung erreicht werden kann, unter der die eingangs erwähnte Synchronisationsstelle ebenso markiert wäre. Da die erreichbaren Markierungen des Netzmoduls dessen zulässige Modulzustände sind, gilt ebenso: Die Ausgangsmarkierung ist der einzige zulässige Modulzustand, von dem aus das Schaltverhalten des Netzmoduls angestoßen werden kann.

80) Andernfalls wäre es möglich, daß ein Netzmodul zwar sein Schaltverhalten beginnt, aber nicht in seinen Ausgangszustand zurückkehrt. Der Ausgangszustand wurde aber so definiert, daß nur von ihm aus das Schaltverhalten des Netzmoduls angestoßen werden kann. Daher würde die fehlende Rückkehr in den Ausgangszustand bedeuten, daß das Netzmodul niemals mehr sein Schaltverhalten wiederholen könnte. Damit wäre seine beliebige häufige Wiederverwendung nicht mehr gewährleistet.

81) Die Reversibilitätsforderung wird von ABEL, D. (1990), S. 49, klar herausgearbeitet, auch wenn er sich nicht ausdrücklich auf Netzmodule bezieht.

82) Die Ausgangsmarkierung des Restmoduls stellt immer eine Livelockmarkierung dar, falls die postulierte M_0 -Reversibilität des Restmoduls erfüllt ist. Denn in jedem M_0 -reversiblen Netz stellt die Ausgangsmarkierung M_0 eine Livelockmarkierung dar.

83) Die Schaltkanten des Erreichbarkeitsgraphen sind mit den Transitionen beschriftet, die den Schaltschritt einer Schaltkante bilden. Die Markierungsknoten des Erreichbarkeitsgraphen geben die Netzmarkierungen wieder, deren Komponenten - wie auch sonst üblich - in der Reihenfolge der Stellenindizes angeordnet sind.

84) Die beiden ausgezeichneten Transitionen gehören selbst auch zu dieser abstrakten Komponentenzusammenfassung.

85) Falls eine Transition mehrere Ein- oder mehrere Ausgangsstellen besitzt, lassen sich die Input- bzw. Outputstellen der o.a. Netzmodule entsprechend oft duplizieren. Daher kann man die oben erläuterte Verfeinerung einzelner Transitionen im Prinzip auf alle Transitionen anwenden, die zwei Bedingungen erfüllen. Erstens muß es sich um Transitionen handeln, die das Netzzinnere eines stellenberandeten Teilnetzes bilden. Zweitens kommen nur solche Transitionen in Betracht, die sowohl einen nicht-leeren Vorbereich als auch einen nicht-leeren Nachbereich besitzen. Der Übersichtlichkeit halber wird die Verfeinerung solcher Transitionen hier aber nur anhand des einfachen Falles von Transitionen verdeutlicht, deren Vor- und Nachbereiche aus jeweils genau einer Stelle bestehen. An anderer Stelle wird dieser Ansatz zu Transitionen mit beliebigem Stellenrand verallgemeinert.

86) Dies gilt strenggenommen nur so lange, wie noch keine globalen Variablen vorhanden sind, die Fernwirkungen von Modulen über ihre Modulgrenzen hinaus bewirken können. Solche globalen Variablen werden an anderer Stelle eingeführt. In der zugehörigen Anmerkung wird allerdings auch angedeutet, daß sich diese globalen Variablen - bei wesentlich größerem Modellierungsaufwand - grundsätzlich vermeiden lassen. Daher wird ihr modulgrenzenüberschreitender Einfluß auf die Vergrößerung und Verfeinerung von algorithmischen Netzmodellen hier nicht eingehender diskutiert. Im übrigen verletzen die globalen Variablen aber nicht die grundsätzliche Gültigkeit der nachfolgenden Argumentation. Denn die Informationsstellen, auf denen die Ausprägungen globaler Variablen abgefragt werden können, werden stets so angelegt, daß auch sie zum Stellenrand eines Netzmoduls gehören.

87) Es handelt sich um eine vereinfachte Diktion, weil tatsächlich eine Teilnetzsubstitution erfolgt. Ihre wesentliche Bedeutung erstreckt sich aber nur auf die Verfeinerung der Makrotransition.

88) Dabei liegt die Definition der Netzoberfläche zugrunde, die bereits als Variante für den Begriff des Netzrandes vorgestellt wurde. Dort wurde die Netzoberfläche als Menge aller Netzknoten definiert, über die ein Netz mit seiner Umgebung verknüpft ist. Genau dies trifft auf die hier betrachteten Transitionen zu, nachdem vom Stellenrand des zugrundeliegenden Netzmoduls abstrahiert wurde. Denn das verbleibende Subnetz ist mit seiner Umgebung - dem Stellenrand des Netzmoduls - nur durch die Ausgangstransition der Inputstelle und durch die Eingangstransition der Outputstelle verknüpft. Input- und Outputstelle bilden exakt den Stellenrand des Netzmoduls, der ausgeblendet wurde.

89) Es erfolgt eine sprachliche Vereinfachung, weil die beiden Transitionen aus der Subnetzoberfläche keinen Subnetzrand bilden, der die o.a. Definition von Netzrändern erfüllt. Denn diese Netzränder umfassen nur solche Knoten die entweder einen leeren Vor- oder aber einen leeren Nachbereich besitzen. Die beiden Transitionen aus der Subnetzoberfläche verfügen aber in den oben eingeführten Netzmodulen stets über mindest eine Eingangs- und mindestens eine Ausgangsstelle. Es handelt sich dabei um jeweils eine von den zwei charakteristischen Synchronisationsstellen. Folglich besitzen die Subnetze, die aus den Netzmodulen nach Abstraktion von ihren Stellenrändern verbleiben, jeweils einen *leeren* Netzrand. Allerdings wurde schon darauf hingewiesen, daß Netzoberflächen in der Netzliteratur oftmals ebenso als Netzränder bezeichnet werden. Daher wird hier zugelassen, Subnetze mit Oberflächen, die nur aus Transitionen bestehen, als "transitionenberandete" Subnetze anzusprechen. Diese Vereinbarung gestattet es, auf die präzisere, aber sprachlich kompliziert anmutende Formulierung "transitionenoberflächige Subnetze" zu verzichten.

90) Das sind alle Transitionen ohne eine Schattierung ihrer graphischen Rechtecksymbole. Sie werden in Anlehnung an die oben skizzierte Modulvergrößerung fortan als Mikrotransitionen bezeichnet.

91) Dabei handelt es sich um alle Makrotransitionen, die schattiert dargestellt werden.

92) Dabei müssen nicht unbedingt die Synchronisationsstellen verwendet werden, die in den o.a. Netzmodulen auftraten. Darauf wurde schon anhand der Komplementstelle ABEL's aufmerksam gemacht.

93) Die benachbarten Stellen der Makrotransition sind allesamt nur mit dieser Transition verknüpft. Sobald die Makrotransition aber durch ein verfeinerndes Subnetz ersetzt wird, muß feststehen, mit *welchen* Eingangsstellen der früheren Makrotransition *welche* Mikrotransitionen aus der Subnetzoberfläche verknüpft sind. Ebenso ist zu klären, *welche* Mikrotransitionen mit *welchen* Ausgangsstellen der früheren Makrotransition verbunden werden.

94) Da die Verfeinerungstechnik konzeptionell aus einer Vergrößerung von Netzmodulen hergeleitet wurde, läßt sie sich ebenso einsetzen, um Netzvergrößerungen vorzunehmen. Dazu brauchen lediglich die Verfeinerungsoperationen in komplementäre Vergrößerungsoperationen umgekehrt zu werden. Aus stellenberandeten Netzmodulen werden auf diese Weise vergrößerte Teilnetze, deren Stellenrand unverändert ist, deren Teilnetzinneres aber nur noch aus einer Makrotransition besteht. Dies folgt unmittelbar aus der Modulvergrößerung, die an früherer Stelle skizziert wurde. Solche Netzvergrößerungen interessieren in dieser Arbeit jedoch nicht weiter.

95) Die Verfeinerungsrichtung wird durch diejenigen Netzbereiche festgelegt, in denen Transitionen als Makrotransitionen behandelt und entsprechend verfeinert werden.

96) Das Verfeinerungsausmaß betrifft die Anzahl der Verfeinerungsschritte, die nacheinander auf dieselbe ursprüngliche - noch nicht verfeinerte - Makrotransition angewendet werden. Diese Schrittzahl wird auch als Verfeinerungstiefe der jeweils betroffenen Transition bezeichnet. Beträgt diese Schrittzahl für alle Transitionen eines Netzes Null, so erfolgt überhaupt keine Netzverfeinerung. Es wird dann von einem "flachen" Netzmodell gesprochen. Ein "tiefes" Netzmodell liegt dagegen vor, falls mindestens eine seiner Transitionen eine von Null verschiedene Verfeinerungstiefe besitzt. Die Modell- oder Modellierungstiefe wird mit der maximalen Verfeinerungstiefe aller Transitionen eines Netzmodells gleichgesetzt.

97) Eine Restriktion könnte darin gesehen werden, daß ausschließlich Transitionen - aber keine Stellen verfeinert werden. Darin besteht aber aus zwei Gründen keine wesentliche Beschränkung. Erstens zeigt sich bei der praktischen Anwendung des Petrinetz-Konzepts für Modellierungszwecke immer wieder, daß sich die Verfeinerungsbedürfnisse auf die Präzisierung des *Verhaltens* von Netzmodellen erstrecken. Daher reicht es aus, das Schaltverhalten von Transitionen durch die o.a. Technik beliebig verfeinern zu können. Zweitens bietet das Petrinetz-Konzept auch komplementäre Ansätze, bei denen transitionsberandete Teilnetze durch Netzmodule substituiert werden. Auf diese Weise lassen sich (Makro-)Stellen in genau der gleichen Weise verfeinern, wie es zuvor für Makrotransitionen beschrieben wurde. Diese komplementäre Verfeinerungstechnik spielt jedoch aus dem erstgenannten Grund für die hier vorgelegten Untersuchungen keine Rolle. Sie wird daher nicht weiter betrachtet.

98) Es wird hier vereinbart, die oberste Hierarchieebene mit demjenigen Netz zu identifizieren, in dem überhaupt keine Transition verfeinert ist. Jede tiefere Hierarchieebene geht aus der jeweils höheren Ebene dadurch hervor, daß

mindestens eine Transition des Netzes der höheren Hierarchieebene verfeinert wird. Die unterste Hierarchieebene stellt jenes Netz dar, dessen Transitionen nicht mehr weiter verfeinert werden. Als ein Netzmodell wird stets die *Gesamtheit* aller Netze verstanden, die dasselbe Modellierungsobjekt auf verschiedenen Hierarchieebenen abbilden. Das einzelne Netz, das dieses Modellierungsobjekt auf einer Hierarchieebene darstellt, wird auch als eine Variante des Netzmodells bezeichnet. Nur ein flaches Netzmodell besteht aus genau einem Netz, da es per definitionem nur genau eine "Hierarchieebene" umfaßt.

99) Das Detaillierungsniveau wird auch - mit jeweils entgegengesetzter Blickrichtung - als Verfeinerungs- oder Vergrößerungsniveau bezeichnet. Auf der obersten (untersten) Hierarchieebene sind das Detaillierungs- und Verfeinerungsniveau minimal (maximal), während das Vergrößerungsniveau maximal (minimal) ausfällt.

100) Die Kombination der hierarchischen Verfeinerungstechnik mit den Netzmodulen wird hier nicht im Detail beschrieben. Denn sie erfolgt im wesentlichen genau so, wie es schon anlässlich der Verfeinerung von Makrotransitionen aus stellenberandeten Teilnetzen dargelegt wurde:

- Zunächst wird der zu transformierende Algorithmus auf der obersten Abstraktionsebene durch genau eines der o.a. Netzmodule grob repräsentiert.
- Dann werden einzelne Transitionen als Makrotransitionen ausgezeichnet. Sie werden einschließlich ihrer Ein- und ihrer Ausgangsstelle(n) durch ein verfeinerndes Netzmodul substituiert.
- Die Transitionsverfeinerungen lassen sich beliebig weit fortsetzen, bis der Modellgestalter das zuletzt erzielte Detaillierungsniveau für ausreichend erachtet.
- Jede Mikrotransition, die nicht weiter verfeinert wird, repräsentiert eine atomare Anweisung. Die Transition wird mit der Anweisungskennzeichnung "anw_q" beschriftet, die bereits in der früheren Pseudocode-Notation des Konstruktionsalgorithmus verwendet wurde. Im Interesse der Netztransparenz kann die Verarbeitungsleistung einer solchen atomaren Anweisung in die anweisungsdarstellende Mikrotransition eingetragen werden.

Lediglich auf zwei Besonderheiten ist hinzuweisen. Erstens werden aus dem Netzmodul, das den Algorithmus auf der obersten Abstraktionsebene repräsentiert, die Stellen "Input" und "Output" samt ihrer Aus- bzw. Eingangskanten entfernt. Denn diese beiden Stellen spielen nur *innerhalb* eines algorithmischen Netzmodells eine Rolle, um die Informationsübergabe an den Schnittstellen zwischen einem Netzmodul und seiner Netzumgebung zu modellieren. Das Netzmodell als Ganzes besitzt jedoch keine weiterreichende Netzumgebung, so daß auch die vorgenannten Schnittstellen nicht mehr definiert sind. Zweitens kann jedes Netzmodul über seine Transitionsverfeinerung hinaus auch noch um globale Variablen erweitert werden. Dabei handelt es sich um Variablen, deren Ausprägungen von mehreren Modulen gemeinsam festgelegt oder abgefragt werden. Die globalen Variablen werden in den algorithmusdarstellenden Netzmodulen der Abb. 75 bis 88 jeweils durch Stellen repräsentiert, deren Graphiksymbole durch Fettdruck und fette Stellenanschriften hervorgehoben werden.

Über die Verwendung globaler Variablen in einer modular angelegten Algorithmusdefinition ließe sich streiten. Denn solche globalen Variablen vermitteln grundsätzlich Fernwirkungen, die über das Modul, in dem sie jeweils enthalten sind, hinausreichen. Solche Fernwirkungen wurden an anderer Stelle kritisiert. Sie könnten vermieden werden, indem alle erforderlichen Variablenwerte an die jeweils betroffenen Module weitergereicht werden. Dies entspräche dem Persistenz-Prinzip, das früher für die Modellierung von Realproblemen aufgestellt wurde. Dann müßten aber einige Kanten aus dem algorithmischen Netzmodell wesentlich aufwendiger beschriftet werden, als es für die nachfolgenden Teilnetze der Fall ist. Darauf wird jedoch verzichtet. Denn diese Arbeit zielt nicht auf den informationstechnischen Aspekt ab, informationsverarbeitende Algorithmen bis ins letzte Detail auszuarbeiten. Daher wurde für die Veranschaulichung des Konstruktionsalgorithmus nur das Konzept der Kanal/Instanz-Netze - und nicht das wesentlich präziser ausgearbeitete Konzept der Synthetischen Netze ausgewählt.

101) Vgl. die Kritik an artifiziellen Sequentialisierungen, die im Kontext von Arbeitsplänen erfolgte.

102) Beiträge, welche die inhärente Nebenläufigkeit von Algorithmen im Rahmen des Petrinetz-Konzepts untersuchen, finden sich z.B. bei LOWAS (1986b), S. 1692ff.

103) Vgl. z.B. LOWAS (1986b), S. 1693.

104) Dies entspricht genau der grenzbegriffsartigen Definition des Nebenläufigkeitspotentials von Algorithmen. Dort wurde die inhärente Nebenläufigkeit eines Algorithmus mit dem Ausmaß identifiziert, in dem sich seine atomaren Anweisungen nebenläufig ausführen lassen, wenn die Menge dieser Anweisungen unverändert bleibt, aber ihr Verarbeitungszusammenhang beliebig variiert werden darf.

105) Degenerierte Schaltschritte, die jeweils nur eine Transition umfassen, lassen kein nebenläufiges Schalten mehrerer Transitionen zu. Genau dieses Nebenläufigkeitspotential interessiert hier aber.

106) Damit diese Schaltschritte - und die daraus abgeleiteten Erreichbarkeitsgraphen - überhaupt angegeben werden können, muß das modifizierte algorithmische Netzmodell eine wohldefinierte dynamische Struktur besitzen. Dies ist jedoch für die oben benutzten Kanal/Instanz-Netze nicht der Fall. Daher werden alle modifizierten Netzmodelle nachfolgend als Stelle/Transition-Netze vorausgesetzt. Sie besitzen wohldefinierte Schaltschritte und Erreichbarkeitsgraphen. Die Beschriftungen der Netzknoten und -kanten werden aus den früher vorgelegten Kanal/Instanz-

Netzen in die graphische Netzvisualisierung übernommen, weil sie deren Selbsterklärungsfähigkeit unterstützen. Diese Beschriftungen sind jedoch in Stelle/Transition-Netzen nicht definiert. Daher werden diese Knoten- und Kantenanschriften bei der Untersuchung der Schaltschritte und Erreichbarkeitsgraphen des modifizierten algorithmischen Netzmodells nicht berücksichtigt.

107) Nur der erste Modellausschnitt wird ausführlicher besprochen. Hinsichtlich des zweiten Modellausschnitts beschränkt sich der Verf. darauf, die daraus ersichtliche Sequentialisierung aufzuzeigen. Auf die aufwendige Konstruktion des zugehörigen zweiten Erreichbarkeitsgraphen wird jedoch verzichtet. Sie würde lediglich die grundsätzlich Vorgehensweise, die anhand des Erreichbarkeitsgraphen des Modellausschnitts verdeutlicht wird, nochmals reproduzieren.

108) Aus kognitionspsychologischer Perspektive böte es sich an, die heuristische Kraft des Petrietz-Konzepts durch folgendes Experiment zu untersuchen: Eine erste Probandengruppe wird mit der ursprünglichen Pseudocode-Notation des Konstruktionsalgorithmus für Erreichbarkeitsgraphen konfrontiert und gebeten, nach darin enthaltenen artifiziellen Sequentialisierungen zu suchen. Eine zweite Probandengruppe, die zuvor in die Grundlagen des Petrietz-Konzepts eingewiesen wurde, erhält die gleiche Pseudocode-Notation und zusätzlich die nachfolgend abgebildeten Teilnetze, eventuell auch deren Erreichbarkeitsgraphen. Die zweite Gruppe soll ebenso die Aufgabe erfüllen, artifizielle Sequentialisierungen in der Pseudocode-Notation aufzudecken. Der Verf. wagt die Vermutung, daß die zweite Probandengruppe wesentlich erfolgreicher abschneiden wird.

109) Daß sich die Sequentialisierung über ein Netzmodul *und* dessen Submodul erstreckt, verdeutlicht die oben erfolgte Feststellung, daß bereits in der modularen und hierarchisch verfeinerten Algorithmusformulierung unnötige Sequentialisierungen verborgen sein können.

110) Tatsächlich enthält die erste Algorithmusformulierung in ihrem Verarbeitungszusammenhang der atomaren Anweisungen anw_q mit $q \in \{1, \dots, 5\}$ eine Vielzahl von artifiziellen Sequentialisierungen. Beispielsweise besitzt auch die Sequenz, zunächst die Anweisung anw_1 und dann die Anweisung anw_4 auszuführen, artifiziellen Charakter. Dies wird ebenfalls aus dem ersten Teilnetz der Abb. 89 und aus seinem Erreichbarkeitsgraphen deutlich. Denn der Erreichbarkeitsgraph enthält ebenso den Schaltschritt $\{t_3, t_5\}$, dessen Transitionen im zugrundeliegenden Teilnetz mit den Anweisungen anw_1 und anw_4 beschriftet sind. Es wird jedoch darauf verzichtet, diese Sequentialisierungen vollständig aufzulisten. Denn sie gleichen sich alle in der Hinsicht, daß sie auf derselben Sequentialisierungsursache beruhen. Die gemeinsame Quelle ihrer künstlichen, kausal überflüssigen Sequentialisierung liegt darin, die Anweisung anw_1 für das Einlesen der ursprünglichen Netzdefinition *zeitlich vor* den Initialisierungsanweisungen anw_3 und anw_4 vorzunehmen.

111) Ab dem Beginn der iterativen Prozedur "LOOP globale_Knotenanalyse ..." stimmen die ursprüngliche Prozedur "Erreichbarkeitsgraph_Konstruktion" und ihre Modifizierung "Erreichbarkeitsgraph_Konstruktion_mod" wieder überein.

112) Auch alle anderen künstlichen Sequentialisierungen, die in einer früheren Anmerkung angesprochen wurden, gehen hierdurch unter. Denn ihre *gemeinsame* Ursache, die zeitliche Präzedenz zwischen der Einleseanweisung anw_1 einerseits sowie den beiden Initialisierungsanweisungen anw_3 und anw_4 andererseits, wird aufgehoben.

113) Auf den Erreichbarkeitsgraphen dieses Teilnetzes wird hier verzichtet.

114) Dies unterstreicht den Hinweis, daß das Nebenläufigkeitspotential des Problems, den Erreichbarkeitsgraphen eines Netzmodells zu konstruieren, immer nur vorläufigen Charakter besitzt: Es gilt nur im Hinblick auf alle Konstruktionsalgorithmen, die für die Lösung des Konstruktionsalgorithmus aktuell bekannt sind. Der nachfolgend skizzierte neuartige Konstruktionsalgorithmus mit vergrößerter inhärenter Nebenläufigkeit läßt auch das vorläufige Nebenläufigkeitspotential des vorgenannten Konstruktionsproblems anwachsen.

115) Vgl. dazu das "schaltphasenorientierte Konzept" in ZELEWSKI (1986c), S. 33ff. Dort werden unterschiedliche "Dämonen" (Agenten) für zwei verschiedenartige Teilaufgaben benutzt: Die einen untersuchen die Aktivierungen von Transitionen (Schaltschritten). Die anderen ermitteln die Schaltwirkungen von aktivierten Transitionen (Schaltschritten).

5.2.3.5.3 Auswertung von Erreichbarkeitsgraphen

Jedem Synthetischen Netz ist ein Erreichbarkeitsgraph zugeordnet. Seine Konstruktion wurde im voranstehenden Kapitel dargelegt. Er expliziert die dynamische Netzstruktur vollständig¹⁾. Mit seiner Hilfe lassen sich alle früher vorgestellten dynamischen Netzeigenschaften untersuchen²⁾. Es wird hier darauf verzichtet, die zugehörigen Analysemethoden im einzelnen vorzustellen³⁾. Es wird lediglich angedeutet, wie sich die Netzeigenschaften in endlichen Erreichbarkeitsgraphen⁴⁾ grundsätzlich erkennen lassen⁵⁾.

- Beschränktheit: Jedes Netz mit einem endlichen Erreichbarkeitsgraphen ist K -beschränkt. K ist der maximale Wert der Zählfunktion "#", die jede Stelle $s_m \in S$ eines Netzes und jeden Markierungsknoten $M_r \in RM(M_0)$ aus seinem Erreichbarkeitsgraphen auf die Anzahl $\#(M_r(s_m))$ derjenigen Markenkopien abbildet, die sich auf der Stelle s_m unter der Markierung M_r befinden.
- Deadlock: Ein Deadlock ist jeder Markierungsknoten des Erreichbarkeitsgraphen, der keine Ausgangskante besitzt⁶⁾.
- Lebendigkeit einer Markierung: Der Markierungsknoten besitzt mindestens eine Ausgangskante.
- Deadlockfreiheit: Jeder Markierungsknoten im Erreichbarkeitsgraphen besitzt mindestens eine Ausgangskante.
- M_r -Lebendigkeit einer Transition t_n : Vom Markierungsknoten M_r geht mindestens ein Schaltweg aus, der mindestens eine Schaltkante umfaßt, in deren Anschrift die Transition t_n enthalten ist.
- M_r -Blockiertheit einer Transition t_n : Alle Schaltwege, die den Markierungsknoten M_r verlassen, enthalten keine Schaltkante, in deren Anschrift die Transition t_n vorkommt.
- Unsterblichkeit einer Transition t_n : Von jedem Markierungsknoten geht mindestens ein Schaltweg aus, der mindestens eine Schaltkante umfaßt, in deren Anschrift die Transition t_n enthalten ist.
- Totheit einer Transition t_n : Der Erreichbarkeitsgraph enthält keine Schaltkante, die mit einem Schaltschritt beschriftet ist, der die Transition t_n umfaßt.
- Transitionsschwache Lebendigkeit: Jeder Markierungsknoten im Erreichbarkeitsgraphen besitzt mindestens eine Ausgangskante⁷⁾.
- Markierungsschwache Lebendigkeit: Für jede Transition aus der Transitionenmenge des Netzes gilt: Vom Knoten der Ausgangsmarkierung geht mindestens ein Schaltweg aus, der mindestens eine Schaltkante umfaßt, in deren Anschrift diese Transition enthalten ist.
- Starke Lebendigkeit⁸⁾: Es werden zunächst alle⁹⁾ maximalen stark zusammenhängenden¹⁰⁾ Teilgraphen - kurz: starken Komponenten - des Erreichbarkeitsgraphen betrachtet. Wenn im Erreichbarkeitsgraph mindestens ein Weg existiert, der von einem Markierungsknoten aus einer starken Komponente (einer Quelle) zu einem Markierungsknoten aus einer anderen starken Komponente (einer Senke) führt, dann wird die Quelle aus dem nachfolgenden Analyseschritt ausgeschlossen. Für jede der verbleibenden¹¹⁾ starken Komponenten muß gelten¹²⁾: Jede Transition aus der Transitionenmenge des Netzes ist in mindestens einem derjenigen Schaltschritte enthalten, mit denen die Kanten aus der starken Komponente beschriftet sind.
- Livelock: Ein Livelock ist jeder Markierungsknoten, durch den ein zyklischer Weg im Erreichbarkeitsgraphen verläuft.

- Livelockfreiheit: Es gibt keinen zyklischen Weg im Erreichbarkeitsgraphen.
- Finalität: Vom Knoten der Ausgangsmarkierung aus führt mindestens ein Weg zu mindestens einem Endmarkierungsknoten, der eine erwünschte Endmarkierung darstellt.
- Optimalität: Es gibt mindestens einen Weg, der vom Knoten der Ausgangsmarkierung zu mindestens einem erwünschten Endmarkierungsknoten führt. Alle derart finalen Wege sind bewertet. Das Netzoptimum bilden jene finalen Wege, deren Bewertung das jeweils zugrundeliegende Formalzielsystem bestmöglich erfüllt¹³).
- Terminieren: Von allen Markierungsknoten aus führt jeweils mindestens ein Weg zu mindestens einem erwünschten Endmarkierungsknoten.
- Echtes Terminieren: Von allen Markierungsknoten aus führt jeweils mindestens ein Weg zu mindestens einem Endmarkierungsknoten. Alle Endmarkierungsknoten besitzen keine Ausgangskanten. Alle Knoten, die keine Endmarkierungsknoten sind, weisen jeweils mindestens eine Ausgangskante auf. Der Erreichbarkeitsgraph enthält keinen zyklischen Weg¹⁴).
- Robustheit: Es werden alle Schaltwege betrachtet, die einer von zwei alternativen Wegeklassen angehören. Einerseits handelt es sich um Schaltwege von intendierten Schaltprozessen, die in einem erwünschten Endmarkierungsknoten enden und dabei das Formalzielsystem bestmöglich erfüllen. Andererseits kommen Schaltwege in Betracht, die nicht-intendierten Schaltprozessen entsprechen. Diese Schaltwege enden entweder jeweils in einem Markierungsknoten ohne Ausgangskante, die keinen Endmarkierungsknoten darstellt (unerwünschter Deadlock), oder aber sie münden in jeweils einen Markierungsknoten ein, durch den ein zyklischer Weg im Erreichbarkeitsgraphen verläuft (Livelock).
- Promptheit: Bezüglich einer Systemmeldung SK_i werden alle Markierungsknoten betrachtet, von denen jeweils mindestens eine Schaltkante ausgeht, in deren Anschrift eine Melde- $t_{SK,i}$ für das Eintreffen einer Systemmeldung enthalten ist (Meldeknoten). Hinzu kommen alle Markierungsknoten, in die mindestens eine Schaltkante eingeht, deren Anschrift die korrespondierende Anweisungstransition $t_{KS,i}$ für das Absenden einer Steuerungsanweisung umfaßt (Anweisungsknoten). Jeder Weg des Erreichbarkeitsgraphen, der sich zwischen einem solchen Melde- und einem solchen Anweisungsknoten erstreckt, wird untersucht. Das Koordinationsnetz verhält sich R_i -prompt, wenn zwei Bedingungen erfüllt sind: Erstens ist R_i mit $R_i \in \mathcal{N}_+$ die größte Anzahl von Schaltkanten, die jeder solche Weg umfaßt. Zweitens darf von keinem dieser Wege vor der letzten Schaltkante ein zyklischer Weg abzweigen. Falls sich die R_i -Promptheit für alle Systemmeldungen SK_i mit $i \in \{1, \dots, I\}$ aufzeigen läßt, dann erweist sich das Netz insgesamt als R -prompt für $R = \max \{R_i; i = 1, \dots, I\}$.
- M_0 -Reversibilität: Durch jeden Markierungsknoten M_r , der nicht mit dem Knoten der Ausgangsmarkierung M_0 identisch ist ($M_r \in (RM(M_0) - M_0)$), verläuft mindestens ein zyklischer Weg, der auch den Knoten der Ausgangsmarkierung M_0 umfaßt¹⁵).

Die Untersuchung dieser Netzeigenschaften mit Hilfe der Erreichbarkeitsanalyse gestattet es, Netzmodelle auf vielfältige Weise auszuwerten¹⁶). Dabei können die jeweils interessierenden Eigenschaften frei miteinander kombiniert werden. Es liegt außerhalb des Erkenntnisinteresses dieser Arbeit, das dadurch aufgespannte Analysespektrum von Netzmodellen systematisch zu entfalten. Statt dessen wird es in exemplarischer Weise anhand von vier typischen Untersuchungskomplexen verdeutlicht¹⁷).

Erstens bietet es sich an zu untersuchen, ob vom Ausgangszustand M_0 eines Netzmodells aus mindestens ein erwünschter Endzustand $M_{E,e}$ durch mindestens einen Schaltprozeß erreicht werden kann. Falls sich kein erwünschter Endzustand erreichen läßt, ist es möglich, eine Ursachenanalyse anzuschließen. Mit ihrer Hilfe kann untersucht werden, ob die fehlende Netzfinalität durch eine mangelhafte Prozeßkoordinierung innerhalb des modellierten Systems, durch einen unzureichenden Ausgangszustand oder durch eine inkonsistente Netzspezifizierung hervorgerufen wird. So kann etwa ein Deadlock als eine *unerwünschte* Endmarkierung aufgedeckt werden,

in der alle Schaltprozesse nach endlicher Zeit einmünden. Die Schaltprozesse sterben dort ab, ohne daß jemals einer der erwünschten Endzustände erreicht wird. Unter Umständen gelingt es jedoch aufzuzeigen, daß sich diese Deadlockmarkierung vermeiden läßt, falls die Ausgangsmarkierung des Netzes modifiziert wird. Dann ist ein unzureichender Ausgangszustand des Netzmodells identifiziert. Zugleich ist bekannt, wie er sich überwinden läßt. Statt dessen könnte aber auch nachgewiesen werden, daß keine zulässige Ausgangsmarkierung existiert, welche die Deadlockmarkierung zu verhindern hilft. Dann steht die Inkonsistenz der Netzspezifizierung fest. Schließlich könnte als Ursache mangelnder Netzfinalität aber auch ein Livelock im Erreichbarkeitsgraphen entdeckt werden, in den abermals alle Schaltprozesse nach endlicher Zeit geraten. Er bedeutet, daß sich die Schaltprozesse stets in unerwünschten Endlosschleifen verfangen. Daher ist die Prozeßkoordinierung mangelhaft. Sie muß im modellierten System so modifiziert werden, daß der zyklische Livelockprozeß nicht mehr zulässig ist.

Zweitens kann einer Frage nachgegangen werden, die sich zur vorangehenden Livelockvermeidung entgegengesetzt verhält. Dabei wird untersucht, ob in einem Netzmodell mindestens ein erwünschter zyklischer Schaltprozeß existiert, der den Ausgangszustand M_0 des Modells immer wieder herzustellen vermag. Dies ist der Fall, wenn sich die M_0 -Reversibilität des Netzmodells nachweisen läßt. Der zugehörige Reproduktionsprozeß erweist sich für die Prozeßkoordinierung in einem System als erstrebenswert, wenn es gilt, das System nach der Realisierung eines Sachziels in seinen Ausgangszustand zurückzusetzen. Dies ist immer dann notwendig, wenn ein zyklisches Systemverhalten realisiert werden soll. Beispielsweise kann die Koordination Flexibler Fertigungssysteme so modelliert werden, daß jedes Durchschleusen eines Fertigungsauftrags einem Schaltprozeß im Netzmodell entspricht. Um das wiederholte Ausführen unterschiedlicher Aufträge zu ermöglichen, wäre es wünschenswert¹⁸⁾, wenn sich das Netzmodell nach jeder Auftragsfertigstellung wieder in seinem Ausgangszustand befände. Dies ist tatsächlich der Fall, wenn es nachzuweisen gelingt, daß die Auftragsbearbeitung durch einen Schaltprozeß abgebildet wird, der die Ausgangsmarkierung des Netzmodells reproduziert. Ein anderes Beispiel für die Bedeutung M_0 -reversibler Netzmodelle wird später anhand der Netztheoreme von MURATA/ZHANG und LAUTENBACH vorgeführt¹⁹⁾.

Drittens ist es möglich, für ein Netzmodell zunächst Finalität, Dead- und Livelockfreiheit zu überprüfen. Diese Netzeigenschaften stellen sicher, daß jeder einmal begonnene Schaltprozeß nach endlich vielen Schaltschrittausführungen eine erwünschte Endmarkierung erreicht und dort auch endet²⁰⁾. Falls dieser Test positiv ausfällt, erfüllt jeder zu Ende geführte Schaltprozeß aus entscheidungstheoretischer Perspektive das Sachzielsystem des Entscheidungsträgers. Hinsichtlich aller derart terminierenden Schaltprozesse läßt sich ermitteln, wie sie sich auf das Formalzielsystem des Entscheidungsträgers auswirken²¹⁾. Dabei brauchen lediglich die Schaltwerte aller Schaltschritte, die in der Schaltfolge eines betrachteten Schaltprozesses enthalten sind, addiert zu werden. Diese Schaltschritte lassen sich z.B. als Kantenanschriften desjenigen Schaltwegs ablesen, der diesen Schaltprozeß im Erreichbarkeitsgraphen des Netzmodells wiedergibt. Die Erreichbarkeitsanalyse terminierender Schaltprozesse kann drei unterschiedliche Formen annehmen²²⁾:

- Bei einer projektiven Analyse wird untersucht, wie sich ein einzelner Schaltprozeß auf das Formalzielsystem auswirkt. Dadurch läßt sich z.B. untersuchen, ob der betrachtete Schaltprozeß alle Satisfizierungsziele aus dem Formalzielsystem zu erfüllen vermag.
- Durch eine komparative Analyse kann der Frage nachgegangen werden, welche Mitglieder aus einer endlichen Gruppe alternativer Schaltprozesse das Formalzielsystem - relativ zu allen anderen Gruppenmitgliedern - am besten erfüllen. Falls ein Schaltprozeß und dessen Erfüllung von Meliorisierungszielen als Bezugspunkt fest vorgegeben ist, ist es ebenso möglich zu untersuchen, ob in der Schaltprozeßgruppe echt besserstellende Mitglieder enthalten sind.

- Mit Hilfe einer optimierenden Analyse läßt sich feststellen, welche Schaltprozesse aus der Gesamtheit aller vorliegenden terminierenden Schaltprozesse das Formalzielsystem bestmöglich erfüllen.

In der Netzliteratur überwiegen bei weitem die projektiven und komparativen Untersuchungsvarianten. Aus entscheidungstheoretischer Perspektive besteht dagegen ein größeres Interesse daran, aus allen zulässigen Verhaltensweisen eines Netzmodells seine optimalen Schaltprozesse herauszufiltern²³⁾. Dieser Optimierungsaspekt wird hier nicht weiter vertieft, sondern im anschließenden Kapitel ausführlicher behandelt.

Viertens wird kann aus handlungstheoretischer Sicht ein Interesse daran bestehen, Handlungsspielräume und -folgen näher zu untersuchen. Auch für diesen Zweck läßt sich die Erreichbarkeitsanalyse von Netzmodellen einsetzen. Zunächst wird jeder Modellzustand mit einer Handlungssituation identifiziert. Der dort jeweils offenstehende Handlungsspielraum kann unmittelbar aufgezeigt werden: Jeder Schaltschritt, der unter der Netzmarkierung des betrachteten Modellzustands aktiviert ist, stellt eine Handlungsoption dar. Ihr entspricht im Erreichbarkeitsgraphen des Netzmodells eine Kante, die vom aktuellen Markierungsknoten ausgeht. Daher liegt der Handlungsspielraum in jedem situationsspezifischen Zustand eines Netzmodells als die Gesamtheit aller aktivierten Schaltschritte bzw. aller Ausgangskanten des zugehörigen Markierungsknotens fest²⁴⁾. Die Erreichbarkeitsanalyse erlaubt somit eine unkomplizierte und transparente Repräsentation des situationsspezifischen Handlungsspielraums. Ebenso lassen sich die Folgen von Handlungsoptionen im Rahmen der Erreichbarkeitsanalyse beurteilen. Dabei wird festgestellt, wie sich die Verwirklichung der Handlungsoptionen hinsichtlich eines vorgegebenen Formalzielsystems auswirken würde. Die Folge einer einzelnen Handlungsoption, die nur aus einem aktivierten Schaltschritt besteht, beschränkt sich auf den Schaltwert dieses Schaltschritts²⁵⁾. Interessanter fällt die Beurteilung von Handlungskomplexen dar, die aus einzelnen Handlungsoptionen so zusammengesetzt sind, daß ihrer gemeinsamen Realisierung im Netzmodell die Ausführung eines Schaltprozesses entspricht²⁶⁾. Die Handlungsfolgen eines solchen Handlungskomplexes werden dann durch den Schaltwert des zugehörigen Schaltprozesses bestimmt. Falls alternative Handlungsoptionen oder -komplexe zur Auswahl stehen, ist es möglich, ihre relative Vorteilhaftigkeit analog zur o.a. komparativen Erreichbarkeitsanalyse zu ermitteln. Dabei werden die Schaltwerte der zugehörigen Schaltschritte bzw. -prozesse miteinander verglichen.

Die vier voranstehend skizzierten typischen Untersuchungskomplexe zeichnen sich gemeinsam dadurch aus, daß sie solche Eigenschaften von Netzmodellen beleuchten, die für die Bearbeitung der jeweils modellierten Realprobleme praktisch interessant sein können. Eine rein theoretisch motivierte Fragestellung betrifft dagegen das Verhältnis zwischen der Analyse von Erreichbarkeitsgraphen einerseits und der Untersuchung von Netzinvarianten andererseits. Dabei wird wegen der inferioren Rolle, die der Analyse von S-Invarianten zukommt, nur auf T-Invarianten Bezug genommen. Gegenüber der Invariantenanalyse zeichnet sich die Erreichbarkeitsanalyse von Netzmodellen durch vier Vorteile aus. Sie unterstreichen noch einmal die Festlegung des Verf., die Invariantenanalyse von Netzen grundsätzlich durch ihre Erreichbarkeitsanalyse zu ersetzen²⁷⁾.

Erstens ist es möglich, genau alle zulässigen semi-positiven T-Invarianten eines Netzes auch durch die Untersuchung seines Erreichbarkeitsgraphen zu erkennen. Denn jeder zyklische Weg $w_{g,r}$, der im Erreichbarkeitsgraphen von einer Referenzmarkierung M_r ausgeht und auch dorthin wieder zurückkehrt, stellt eine zulässige semi-positive T-Invariante \underline{it}_h dar²⁸⁾. Dabei nimmt jede Komponente $it_{h,n}$ der T-Invariante \underline{it}_h mit $n \in \{1, \dots, N\}$ genau dann den Wert $it_{h,n} = k$ an, wenn die Transition t_n in den Schaltschritten SS_a , welche die Schaltkanten des zyklischen Wegs $w_{g,r}$ beschriften, insgesamt k -mal vorkommt. Dabei ist besonders zu beachten, daß von vornherein nur zulässige semi-positive T-Invarianten erkannt werden. Denn jeder Weg im Erreichbarkeitsgraphen erstreckt sich per constructionem ausschließlich auf erreichbare Markierungen und zu-

lässige Schaltfolgen. Daher entfällt der Aufwand, der im Rahmen der Invariantenanalyse erforderlich war, um für semi-positive T-Invarianten festzustellen, ob diese auch zulässig sind²⁹⁾.

Zweitens lassen sich im Erreichbarkeitsgraphen sofort alle erreichbaren Markierungen erkennen, unter denen Schaltfolgen aktiviert sind, deren Schaltvektoren durch die jeweils betrachtete semi-positive T-Invariante dargestellt werden. Es handelt sich um alle Markierungen, die als Markierungsknoten an dem invariantendefinierenden Weg $w_{g,r}$ teilhaben. Jeder dieser Markierungsknoten repräsentiert einen Livelock³⁰⁾. Aus T-Invarianten können diese Markierungen nicht unmittelbar abgelesen werden. Drittens ist es möglich, für jeden dieser Livelocks nicht nur die potentiellen³¹⁾, sondern sogar die tatsächlichen zugehörigen Livelock-Transitionen zu erkennen³²⁾. Viertens kann aus den Beschriftungen der wegzugehörigen Schaltkanten mit Schaltschritten unmittelbar abgeleitet werden, wie die Schaltakte von Transitionen innerhalb einer zulässigen Schaltfolge kausal zusammenhängen. Auf diese Weise wird der Informationsverlust hinsichtlich des Schaltzusammenhangs vermieden, der das vektorielle Defizit der Invariantenanalyse begründete.

Es wurde bereits zu Beginn dieses Kapitels darauf hingewiesen, daß - abgesehen vom Aspekt der Netzoptimierung - nicht beabsichtigt wird, Methoden für die Untersuchung von Erreichbarkeitsgraphen detailliert zu erörtern. Dennoch empfiehlt es sich, die grundlegenden Konzepte zu skizzieren, mit deren Hilfe sich die Erreichbarkeitsgraphen von Netzmodellen studieren lassen. Denn die Erreichbarkeitsanalyse wurde als die wichtigste Untersuchungstechnik für Petrinetze herausgestellt. Daher hängt das analytische Potential des Petrinetz-Konzepts unter anderem auch davon ab, auf welche Unterstützungskonzepte bei der Erreichbarkeitsanalyse zurückgegriffen werden kann. Dies wird bei der späteren Eignungsbeurteilung des Petrinetz-Konzepts eine größere Rolle spielen.

Erreichbarkeitsgraphen lassen sich zunächst durch Methoden der konventionellen Graphentheorie auswerten, die auf die Untersuchung gerichteter Graphen zugeschnitten sind. Für diesen Anwendungsbereich steht ein breites Spektrum graphentheoretischer Analysemethoden bereit. Es übersteigt den Erkenntnisrahmen dieser Arbeit, jene Analysemethoden zusammenzutragen oder gar ihre Anwendung auf die Erreichbarkeitsanalyse von Netzen im einzelnen zu erläutern. Statt dessen wird nur auf drei verdeutlichende Beispiele hingewiesen:

- Die Identifizierung zyklischer Wege³³⁾ läßt sich nutzen, um Livelockfreiheit oder M_0 -Reversibilität zu untersuchen.
- Die Bestimmung aller maximalen stark zusammenhängenden Teilgraphen unterstützt die Überprüfung der starken Netzlebendigkeit.
- Die Ermittlung kürzest möglicher Wege zwischen ausgezeichneten Start- und Zielknoten³⁴⁾ gestattet es, optimale Schaltprozesse in finalen Netzmodellen zu bestimmen.

Daher erweisen sich Erreichbarkeitsgraphen als eine konzeptionelle Schnittstelle, mit deren Hilfe umfangreiches graphentheoretisches Analysewissen in das Petrinetz-Konzept eingebettet werden kann. Durch Rückgriff auf bereits erarbeitete Analysemethoden der Graphentheorie läßt sich der Ressourceneinsatz vermeiden, der sonst zur Entwicklung netzspezifischer Untersuchungskonzepte erforderlich wäre.

Für manche - aber nicht alle - Netzeigenschaften ist es auch möglich, Wissen über eigenschaftsspezifische Charakteristika der zugehörigen Erreichbarkeitsgraphen mit konventionellen Analysemethoden der Graphentheorie zu kombinieren. Dadurch läßt sich vor allem die Effizienz der Methodenanwendung steigern.

Einen bemerkenswerten Ansatz in dieser Richtung stellt die Untersuchung kondensierter Erreichbarkeitsgraphen³⁵⁾ dar. In einem kondensierten Erreichbarkeitsgraphen wird die Aufmerksamkeit auf die maximalen stark zusammenhängenden Teilgraphen - d.h. auf die starken Komponenten - des zugrundeliegenden Erreichbarkeitsgraphen fokussiert: Jeder Knoten des

kondensierten Erreichbarkeitsgraphen ist eine starke Komponente aus dem vorgegebenen Erreichbarkeitsgraphen. Eine Kante ist von einem Knoten des kondensierten Erreichbarkeitsgraphen zu einem anderen Knoten des kondensierten Erreichbarkeitsgraphen genau dann gerichtet, wenn im zugrundeliegenden Erreichbarkeitsgraphen gilt: Mindestens ein Schaltweg führt von mindestens einem Markierungsknoten, der zur starken Komponente des ersten Knotens aus dem kondensierten Erreichbarkeitsgraphen gehört, zu mindestens einem Markierungsknoten, der aus der starken Komponente des zweiten Knotens aus dem kondensierten Erreichbarkeitsgraphen stammt.

Kondensierte Erreichbarkeitsgraphen fallen in der Regel wesentlich kompakter und übersichtlicher aus als ihre zugrundeliegenden Erreichbarkeitsgraphen, weil eine Fokussierung auf die starken Komponenten erfolgt³⁶⁾. Daher lassen sich kondensierte Erreichbarkeitsgraphen mittels konventioneller graphentheoretischer Analysemethoden - im Vergleich zur Analyse konventioneller Erreichbarkeitsgraphen - relativ effizient auswerten³⁷⁾. Kondensierte Erreichbarkeitsgraphen stellen deshalb ein Analyseinstrument dar, das die Aussicht eröffnet, Effizienzschwächen der Erreichbarkeitsanalyse zu lindern. Allerdings bedeuten kondensierte Erreichbarkeitsgraphen wegen ihrer Fokussierung auf starke Komponenten notwendig einen Informationsverlust³⁸⁾. Sie gestatten daher nur die Untersuchung eines kleinen Teils der oben angeführten interessanten dynamischen Netzeigenschaften. Immerhin gehören dazu so aussagekräftige Eigenschaften wie die der Deadlocks, der Deadlockfreiheit, der starken Lebendigkeit und der M_0 -Reversibilität³⁹⁾. Bevor die Untersuchungsmöglichkeiten der drei vorgenannten Netzeigenschaften konkretisiert werden können, bedarf es jedoch einer kurzen Erläuterung einiger weniger Charakteristika von kondensierten und gewöhnlichen Erreichbarkeitsgraphen⁴⁰⁾:

- Eine Senke (Quelle) ist eine starke Komponente des zugrundeliegenden Erreichbarkeitsgraphen, die im kondensierten Erreichbarkeitsgraphen einen Knoten darstellt, zu dem mindestens eine Kante hinführt (von dem mindestens eine Kante ausgeht)⁴¹⁾.
- Eine reine Senke (Quelle) ist ein Knoten des kondensierten Erreichbarkeitsgraphen, der eine Senke (Quelle) und zugleich *keine* Quelle (Senke) darstellt⁴²⁾. Falls der kondensierte Erreichbarkeitsgraph aus genau einem isolierten Knoten besteht⁴³⁾, so wird vereinbart, ihn ebenso als eine reine Senke anzusehen.
- Eine starke Komponente aus einem gewöhnlichen Erreichbarkeitsgraphen heißt genau dann lebendig (tot), wenn für jede (keine) Transition aus der Transitionenmenge des zugrundeliegenden Netzes gilt: Die Transition ist in mindestens einem derjenigen Schaltschritte enthalten, mit denen die Kanten der starken Komponente beschriftet sind⁴⁴⁾.

Mit Hilfe der voranstehenden Festlegungen lassen sich nun die oben angekündigten dynamischen Netzeigenschaften in kondensierten Erreichbarkeitsgraphen wie folgt überprüfen⁴⁵⁾:

- Deadlock: Eine reine Senke des kondensierten Erreichbarkeitsgraphen stellt eine tote starke Komponente des zugrundeliegenden Erreichbarkeitsgraphen dar⁴⁶⁾.
- Deadlockfreiheit: Keine reine Senke des kondensierten Erreichbarkeitsgraphen ist eine tote starke Komponente des zugrundeliegenden Erreichbarkeitsgraphen.
- Starke Lebendigkeit: Jede reine Senke des kondensierten Erreichbarkeitsgraphen ist eine lebendige starke Komponente des zugrundeliegenden Erreichbarkeitsgraphen.
- M_0 -Reversibilität⁴⁷⁾: Der kondensierte Erreichbarkeitsgraph besteht aus genau einem isolierten Knoten. Dieser isolierte Knoten gibt eine starke Komponente wieder, die mit dem zugrundeliegenden Erreichbarkeitsgraphen zusammenfällt.

Des weiteren bieten sich Erreichbarkeitsgraphen an, um auch Erkenntnisse aus der Erforschung Künstlicher Intelligenz zu integrieren. Ausgangspunkt ist die enge konzeptionelle Verwandtschaft von Erreichbarkeitsgraphen auf der einen und Problemgraphen auf der anderen Seite. Erreichbarkeitsgraphen spezifizieren die dynamische Struktur eines Netzmodells, indem ihre

Markierungsknoten alle zulässigen Modellzustände und ihre Schaltkanten alle zulässigen Zustandsveränderungen des Netzmodells repräsentieren. Problemgraphen⁴⁸⁾ werden dagegen in der KI-Forschung genutzt, um die zulässigen Zustände von problembeschreibenden Systemen und die zulässigen Übergänge zwischen solchen Systemzuständen auszudrücken⁴⁹⁾. Die unmittelbare Korrespondenz zwischen Erreichbarkeits- und Problemgraphen ist offensichtlich: Die zulässigen Zustände von Netzmodellen und Problemsystemen entsprechen einander. Gleiches gilt für die zulässigen Transformationen von Modell- bzw. Systemzuständen.

Seitens der KI-Forschung wird eine Fülle von Konzepten angeboten, um in Problemgraphen nach Problemlösungen zu suchen⁵⁰⁾. Wegen der ausgeprägten konzeptionellen Übereinstimmung zwischen Problem- und Erreichbarkeitsgraphen lassen sich diese Suchkonzepte grundsätzlich auf die Erreichbarkeitsanalyse von Netzmodellen übertragen. Es würde abermals den Erkenntnisrahmen dieser Arbeit übersteigen, die Transformmöglichkeiten solcher Konzepte im einzelnen zu erörtern⁵¹⁾. Statt dessen werden wiederum nur einige wenige Beispiele für solche Wissensübertragungen aus dem Bereich der KI-Forschung angedeutet. Diese Beispiele unterstreichen nochmals die Qualität von Erreichbarkeitsgraphen, als eine konzeptionelle Schnittstelle zu wirken. Sie läßt in das Petrinetz-Konzept nicht nur Erkenntnisse aus der konventionellen Graphentheorie, sondern ebenso aus der Erforschung Künstlicher Intelligenz einfließen. Daher weisen Petrinetze eine bemerkenswert große konzeptionelle Integrationsfähigkeit auf.

Erstens bietet sich die bidirektionale Lösungssuche⁵²⁾ für die Ermittlung finaler Schaltprozesse in Netzmodellen an. Dabei wird vorausgesetzt, daß die Netzmodelle über nicht-leere Mengen von erwünschten Endmarkierungen verfügen. Ihre Erreichbarkeitsgraphen besitzen daher jeweils mindestens einen Endmarkierungsknoten. Jedem finalen Schaltprozeß entspricht im Erreichbarkeitsgraphen ein finaler Schaltweg. Ein solcher finaler Schaltweg kann in zwei Richtungen gleichzeitig gesucht werden:

- Einerseits wird in konventioneller Weise vom Knoten der Ausgangsmarkierung M_0 aus gestartet, um sich einem der Endmarkierungsknoten durch sukzessives Ausführen von Schaltschritten zu nähern.
- Andererseits wird in inverser Weise von einem der Endmarkierungsknoten $M_{E/c}$ ausgegangen, um durch sukzessives Rücksetzen von Schaltschritten⁵³⁾ zu jeweils vorangehenden Markierungsknoten zurückzugelangen. Ziel ist es, sich dabei an den Knoten der Ausgangsmarkierung anzunähern.

Die KI-Konzepte der bidirektionalen Lösungssuche sind so ausgelegt, daß die beiden simultan ablaufenden, aber entgegengesetzt gerichteten Suchprozesse im Erreichbarkeitsgraphen nicht aneinander vorbeilaufen, sondern sich im Innern des Graphen gegenseitig treffen sollen. Wenn dieses bidirektionale Zusammentreffen zweier Suchprozesse im Erreichbarkeitsgraphen erfolgreich endet, läßt sich aus den beiden Suchprozessen einer der gesuchten finalen Schaltwege zwischen den Knoten der Ausgangs- und einer Endmarkierung rekonstruieren. In dieser bidirektionalen Vorgehensweise liegt ein dreifacher Effizienzvorteil⁵⁴⁾. Erstens wird die Geschwindigkeit, mit der sich ein finaler Schaltweg auffinden läßt, allein dadurch ungefähr⁵⁵⁾ verdoppelt, daß zwei Suchprozesse dieselbe Suchaufgabe *zeitgleich* erfüllen. Zweitens wird die Suchgeschwindigkeit noch dadurch erhöht, daß der Erreichbarkeitsgraph in zwei Teilgraphen *aufgeteilt* wird. Dadurch sinkt die Komplexität des Suchproblems tendenziell um weitaus mehr als die Hälfte⁵⁶⁾. Denn das Phänomen der kombinatorischen Explosion von Erreichbarkeitsgraphen⁵⁷⁾ wirkt sich hier in umgekehrter Richtung aus⁵⁸⁾. Drittens läßt sich die Suchgeschwindigkeit noch weiter steigern, falls die Endmarkierungsmenge ME mehrere Elemente besitzt. Dann können die gleichen bidirektionalen Suchprozesse für mehrere erwünschte Endmarkierungsknoten simultan abgewickelt werden⁵⁹⁾. Dies führt insbesondere⁶⁰⁾ dann zu einer Beschleunigung der Suchprozesse, wenn ein Interesse daran besteht, mehrere finale Schaltprozesse zu identifizieren. Anstatt sie nacheinander suchen zu müssen, lassen sich auf diese Weise im Erreichbarkeitsgraphen mehrere finale Schaltwege nebenläufig auffinden.

Ein zweiter Beitrag der KI-Forschung zur Unterstützung der Erreichbarkeitsanalyse von Netzmodellen betrifft das weite Feld heuristischer, wissensbasierter Suchmethoden⁶¹⁾. Solche Suchmethoden beruhen grundsätzlich auf heuristischem Wissen⁶²⁾ über erfolgversprechende Strategien, in Problemgraphen nach den jeweils interessierenden Problemlösungen zu suchen. Dieses strategische Wissen soll der kombinatorischen Explosion von Problemgraphen so entgegenwirken, daß sich die Lösungssuche wesentlich rascher erfolgreich abschließen läßt, als wenn die Problemgraphen vollständig erforscht werden müßten⁶³⁾. Dieser Ansatz kann wegen der o.a. engen Verwandtschaft zwischen Erreichbarkeits- und Problemgraphen ohne Schwierigkeiten auf die Untersuchung von Erreichbarkeitsgraphen übertragen werden. Dabei erstreckt sich das heuristische Wissen auf Auswahlregeln, die in jeder Phase der Untersuchung eines Erreichbarkeitsgraphen festlegen, in welcher Untersuchungsrichtung fortgeschritten werden soll⁶⁴⁾. Es bestehen mehrere Freiheitsgrade:

- Falls ein Markierungsknoten des Erreichbarkeitsgraphen noch nicht vollständig untersucht⁶⁵⁾ worden ist, muß entschieden werden, ob als nächstes mit der Analyse dieses Knotens fortgefahren oder aber ein anderer Knoten untersucht werden soll. Wenn im ersten Fall ein Markierungsknoten immer erst vollständig untersucht wird, bevor ein anderer Knoten in Angriff genommen wird, so entspricht dies der Strategie der Breitensuche⁶⁶⁾. Die konträre Strategie der Tiefensuche⁶⁷⁾ wird dagegen im zweiten Fall verfolgt, sofern in jedem Markierungsknoten immer nur genau ein aktivierter Schaltschritt ausgeführt und danach sofort der Knoten jener Folgemarkierung untersucht wird, der aus der Schaltschrittausführung resultiert.
- Wenn für einen Markierungsknoten mindestens zwei Schaltschritte noch nicht untersucht worden sind und die Entscheidung getroffen wurde, diesen Knoten weiter zu analysieren, dann muß der nächste zu untersuchende Schaltschritt ausgewählt werden. Es handelt sich dabei um einen derjenigen Schaltschritte, die unter der Markierung des analysierten Knotens aktiviert sind, aber bisher noch nicht ausgeführt wurden.
- Sofern ein Markierungsknoten entweder vollständig untersucht ist oder aber beschlossen wurde, einen noch nicht vollständig analysierten Markierungsknoten vorerst nicht weiter zu betrachten, dann muß der nächste zu bearbeitende Markierungsknoten ausgewählt werden⁶⁸⁾.

Das heuristische Wissen über erfolgversprechende Strategien, die vorgenannten Freiheitsgrade auszufüllen, hängt von den jeweils betrachteten Problemstellungen ab. Daher müßte für jede der eingangs skizzierten Vorgehensweisen, durch die Auswertung von Erreichbarkeitsgraphen Eigenschaften von Netzmodellen zu untersuchen, eine spezifische Untersuchungsstrategie entwickelt werden. Es wurde jedoch bereits dargelegt, die Analysemethoden für Netzeigenschaften in dieser Arbeit nicht detailliert zu entfalten. A fortiori wird auch darauf verzichtet, eigenschaftsspezifische Untersuchungsstrategien zu diskutieren. Statt dessen hat der Verf. an anderer Stelle⁶⁹⁾ für ein spezielles Problem⁷⁰⁾ ausführlicher dargelegt, wie sich die Auswertung der Erreichbarkeitsgraphen von Netzmodellen durch heuristische Strategien für wissensbasierte Suchmethoden bereichern läßt⁷¹⁾.

Darüber hinaus können die wissensbasierten Suchheuristiken mit Konzepten für die nebenläufige Lösungssuche in Problemgraphen⁷²⁾ kombiniert werden. Auf die Geschwindigkeitsvorteile, die solche nebenläufige Konzepte bei ihrer Implementierung auf parallelen Automatischen Informationsverarbeitungssystemen auszeichnen, wurde bereits eingegangen. Sie werden beim nebenläufigen Erforschen von Erreichbarkeitsgraphen noch verstärkt, wenn die Fortsetzung der Lösungssuche von heuristischem Wissen über erfolgversprechende Suchrichtungen unterstützt wird⁷³⁾. Dies betrifft allerdings nur den - derzeit noch vorherrschenden - Fall, daß die Prozessoren eines parallel arbeitenden Informationsverarbeitungssystems nicht ausreichen, um alle nebenläufig ausführbaren Operationen tatsächlich zeitgleich zu übernehmen. Andernfalls braucht keine Auswahl von Suchrichtungen zu erfolgen, weil ohnehin alle denkmöglichen Suchrichtungen simultan bearbeitet werden. Daher verlieren die o.a. heuristischen Suchmethoden bei massiv parallelen Automatenarchitekturen ihre Bedeutung.

Als drittes - und letztes - Beispiel für die Möglichkeiten, durch Erkenntnisse der KI-Forschung die Erreichbarkeitsanalyse von Netzmodellen zu unterstützen, dient das Konzept der automatischen Begründungsverwaltung. Begründungsverwaltende Systeme⁷⁴⁾ dienen im allgemeinen dazu, vorläufige Problemlösungen zu entwickeln, falls zunächst nur unvollständige Informationen für die Problemlösung bereitstehen. Die Informationslücken werden durch zusätzliche Annahmen⁷⁵⁾ geschlossen, die hypothetischen Charakter besitzen. Verfügbare Informationen und Annahmen werden in der gleichen Weise als prädikatenlogische Formeln dargestellt. Auf diese Formelmengende werden konventionelle Inferenzkonzepte der Künstlichen Intelligenz so lange angewandt, bis eine vorläufige Problemlösung vorliegt. Dabei werden die Formeln, aus denen in einem Inferenzschritt eine andere Formel abgeleitet worden ist, als *Begründung* der abgeleiteten Formel festgehalten⁷⁶⁾. Der Inferenzzusammenhang zwischen allen Begründungen und den daraus abgeleiteten Formeln, der zur Erkenntnis der vorläufigen Problemlösung geführt hat, läßt sich durch einen Abhängigkeitsgraphen repräsentieren⁷⁷⁾.

Es kann sich nachträglich herausstellen, daß eine vorläufige Problemlösung zurückgenommen werden muß. Dies ist dann der Fall, wenn im Zeitablauf neue Informationen über das bearbeitete Problem gewonnen werden, die den hypothetischen Annahmen so weit widersprechen, daß die ursprünglich abgeleitete Problemlösung nicht mehr aufrechterhalten werden kann. Dies wird vom begründungsverwaltenden System anhand des Abhängigkeitsgraphen selbständig erkannt. Anstatt eine völlig neue Problemlösung zu erarbeiten, wird im Abhängigkeitsgraphen auf die inkonsistenzverursachenden, durch die neuen Informationen ungültig gewordenen Annahmen zurückgeschlossen⁷⁸⁾. Aus dieser Kenntnis wird eine neue Problemlösung abgeleitet, die an den erweiterten Informationsstand in "intelligenter Weise"⁷⁹⁾ angepaßt ist. Dabei werden neue Annahmen gesetzt⁸⁰⁾, die sich mit den nunmehr verfügbaren Informationen konsistent vereinbaren lassen⁸¹⁾. Daher besitzt auch die neue Problemlösung weiterhin vorläufigen Charakter. Der Inferenzzusammenhang, der zur Ableitung dieser neuen vorläufigen Problemlösung geführt hat, wird wieder als ein - nunmehr modifizierter - Abhängigkeitsgraph dargestellt. Als Begründungsverwaltung wird der Prozeß bezeichnet, bei dem sich das Ableiten von vorläufigen Problemlösungen aus Begründungen abwechselt mit dem Revidieren der Informationen und Annahmen, die eine Lösungsableitung jeweils begründet haben.

Das Konzept begründungsverwaltender Systeme läßt sich auf die Auswertung der Erreichbarkeitsgraphen von Netzmodellen übertragen. Dabei wird an die oben skizzierte wissensbasierte Suche in Erreichbarkeitsgraphen angeknüpft. Die heuristischen Untersuchungsstrategien besitzen die kognitive Qualität von Annahmen darüber, welche Fortsetzungsrichtung der Analyse eines Erreichbarkeitsgraphen am erfolgversprechendsten ist. Diese Richtung wird durch die Auswahl derjenigen Knoten oder Schaltschritte festgelegt, die als jeweils nächste betrachtet werden sollen. Ob die Auswahl dieser Untersuchungsrichtung tatsächlich zum Erfolg führt, ist im Zeitpunkt des Auswahlakts unbekannt. Insofern liegt ein typischer Fall unvollständiger Information vor. Unter Umständen wird nachträglich erkannt, daß diese Untersuchungsrichtung erfolglos abgebrochen werden muß. Dadurch wird die ursprüngliche Informationslücke hinsichtlich des Erfolgsbeitrags der eingeschlagenen Untersuchungsrichtung - im negativen Sinne - geschlossen. Zugleich wird die ursprüngliche, heuristisch motivierte Annahme über die erfolgversprechendste Untersuchungsrichtung widerlegt. Diese fehlerhafte Annahme muß zurückgenommen und die neu gewonnene Information, daß die untersuchte Region des Erreichbarkeitsgraphen keine erfolgreiche Beendigung der Eigenschaftsuntersuchung gestattet, in den Pool verfügbarer Informationen aufgenommen werden. Auf der Basis des derart erweiterten Informationsbestands und neuer heuristischer Annahmen über erfolgversprechende Knoten- oder Schaltschrittauswahlen wird die Untersuchung des Erreichbarkeitsgraphen fortgesetzt.

Die voranstehend skizzierte Auswertung von Erreichbarkeitsgraphen mit Hilfe heuristischer Annahmen über erfolversprechende Untersuchungsrichtungen läßt sich durch das oben vorgestellte Konzept der automatischen Begründungsverwaltung verwirklichen. Von besonderem Interesse ist dabei die neuere ATMS-Konzeptvariante⁸²⁾. Sie gestattet es, verschiedene heuristische Annahmen, die sich wechselseitig ausschließen, *nebenläufig* zu bearbeiten⁸³⁾. Dies entspricht aus modallogischer Perspektive einer KRIPKE-Semantik, in der multiple Welten koexistieren⁸⁴⁾. Jede dieser Welten entspricht einer anderen der konkurrierenden heuristischen Annahmen. Falls eine solche Annahme nachträglich zurückgenommen wird, weil die zugehörige Untersuchungsrichtung im Erreichbarkeitsgraphen erfolglos abgebrochen werden mußte, so braucht nicht zu jenem früheren Punkt der Erreichbarkeitsanalyse zurückgeschritten werden, an dem diese Annahme getroffen wurde⁸⁵⁾. Denn aufgrund der nebenläufigen Behandlung multipler Welten wurden in der Zwischenzeit auch die Konsequenzen von anderen, konkurrierenden heuristischen Annahmen weiterverfolgt. Diese alternativen Untersuchungsrichtungen werden fortgesetzt, wenn das Scheitern eines Heurismus offenkundig geworden ist.

Das ATMS-Konzept der automatischen Begründungsverwaltung läßt sich vor allem nutzen, um die Auswirkungen von Schaltstrategien zu untersuchen. Eine Schaltstrategie legt für jede aktuelle Markierung eines Netzmodells fest, welcher der dort aktivierten Schaltschritte tatsächlich ausgeführt werden soll. Daher ist mit einer Schaltstrategie bei vorgegebener Ausgangsmarkierung eines Netzmodells ein Schaltprozeß verknüpft, der aus der sukzessiven Anwendung der Schaltstrategie auf die Ausgangsmarkierung und alle daraus erzeugten Folgemarkierungen hervorgeht⁸⁶⁾. Solche Schaltstrategien werden hier als heuristische Annahmen darüber aufgefaßt, welche Schaltschrittauswahlen in Netzmodellen am ehesten erwarten lassen, ein erwünschtes Netzverhalten hervorzubringen. Dieses Netzverhalten kann z.B. darin bestehen, von der Ausgangsmarkierung des Netzmodells auf einem Schaltweg mit möglichst geringem oder möglichst hohem Schaltwert zu einer der vorgegebenen, intendierten Endmarkierungen zu gelangen. Diese Endmarkierungen zeichnen sich dadurch aus, daß bei ihrem Erreichen das Sachzielsystem, das dem untersuchten Netzmodell zugrundeliegt, erfüllt wird. Von einem solchen erwünschten Netzverhalten wird fortan ausgegangen.

Wegen der kombinatorischen Explosion von Erreichbarkeitsgraphen wird oftmals darauf verzichtet, für ein Netzmodell zunächst seinen Erreichbarkeitsgraphen vollständig zu konstruieren und dann im Erreichbarkeitsgraphen die finalen Schaltwege mit jeweils bestmöglichen Schaltwerten zu bestimmen. Statt dessen wird eine der vorgenannten Schaltstrategien direkt auf das Netzmodell angewendet, ohne seinen Erreichbarkeitsgraphen zu betrachten. Diese Vorgehensweise wird im allgemeinen als Netzsimulation bezeichnet⁸⁷⁾. Sie erfolgt in der doppelten Hoffnung, erstens eine der erwünschten Endmarkierungen zu erreichen und dabei zweitens dem Schaltwert eines sw^* -optimalen, aber unbekanntem Schaltprozesses "möglichst"⁸⁸⁾ nahe zu kommen. Falls sich die erste Hoffnung nicht erfüllt, muß die Netzsimulation erfolglos abgebrochen werden⁸⁹⁾. Dann kann eine andere Schaltstrategie ergriffen werden, um durch eine erneute Netzsimulation abermals zu versuchen, eine der erwünschten Endmarkierungen zu erreichen. Dieser Zyklus läßt sich im Prinzip mit alternativen Schaltstrategien so lange wiederholen, bis ein finaler Schaltprozeß verwirklicht wird⁹⁰⁾. Sobald dieses Ergebnis vorliegt, ist das Simulationsziel einer projektiven Netzanalyse erfüllt. Im Rahmen einer komparativen Netzanalyse können darüber hinaus erfolgreiche Netzsimulationen, die jeweils mit der Erkenntnis eines finalen Schaltprozesses enden, mehrfach ausgeführt werden. Aus diesen Schaltprozessen wird am Untersuchungsende einer ausgewählt, dessen Schaltwert von den Schaltwerten aller anderen erzeugten finalen Schaltprozesse nicht dominiert wird. Dies kommt der zweiten o.a. Hoffnung zugute, sich an den Schaltwert eines sw^* -optimalen Schaltprozesses anzunähern.

Sowohl die projektive als auch die komparative Variante der Netzsimulation lassen sich durch das ATMS-Konzept der automatischen Begründungsverwaltung bereichern. Zu diesem Zweck wird die Ausführung einer Netzsimulation durch die schritthaltende Teilkonstruktion des zugehörigen Erreichbarkeitsgraphen begleitet⁹¹⁾. Jeder Konstruktionsschritt erfolgt unter der heu-

ristischen Annahme, daß die schaltstrategiespezifische Auswahl eines Schaltschritts unter der jeweils aktuellen Netzmarkierung schließlich zu einer der erwünschten Endmarkierungen führen wird. Falls diese Annahme nachträglich widerlegt wird, weil die Anwendung der Schaltstrategie erfolglos abgebrochen werden muß, so liegt bereits ein teilweise konstruierter Erreichbarkeitsgraph vor. Durch die Technik der automatischen Begründungsverwaltung läßt sich in diesem Rumpferreichbarkeitsgraphen kennzeichnen, welche Schrittausführungen sich unter welchen Markierungsknoten als verfehlt erwiesen haben. Die zugehörigen heuristischen Auswahlentscheidungen werden als unfruchtbar - als "nogoods" - qualifiziert⁹²⁾. Bei konventioneller Netzsimulation müßte eine andere Schaltstrategie gewählt und der Versuch, eine erwünschte Endmarkierung zu erreichen, erneut unternommen werden.

An dieser Stelle läßt sich das ATMS-Konzept benutzen, um die Netzsimulation zu beschleunigen. Es ermöglicht, auf dasselbe Netzmodell mehrere konkurrierende Schaltstrategien nebenläufig anzuwenden. Falls die Anwendung einer Strategie erfolglos abgebrochen werden muß, braucht die Ausführung der Netzsimulation nicht wieder von vorn begonnen zu werden. Statt dessen lassen sich die nebenläufig bearbeiteten, noch nicht gescheiterten Schaltstrategien fortsetzen. Auf diese Weise kann eine projektive Netzanalyse im allgemeinen wesentlich rascher zum erfolgreichen Nachweis eines finalen Schaltprozesses führen, als es bei mehrfacher, sequentieller Wiederholung der Simulationsausführung zu erwarten wäre. Falls mehrere der nebenläufig angewandten Schaltstrategien jeweils eine erwünschte Endmarkierung hervorbringen, liegen am Ende derselben Simulationsausführung entsprechend viele finale Schaltprozesse vor. Aus ihnen kann im Sinne der komparativen Netzanalyse ein Schaltprozeß mit nicht-dominiertem Schaltwert ausgewählt werden⁹³⁾.

Der Vorzug des ATMS-Konzepts liegt nicht nur darin, innerhalb derselben Simulationsausführung mehrere alternative Schaltstrategien anwenden zu können. Hinzu kommt, daß sich die schritthaltende Konstruktion des Erreichbarkeitsgraphen ausnutzen läßt, um die Informationen über gescheiterte Strategieanwendungen in die Fortsetzung der Alternativstrategien einfließen zu lassen. Beispielsweise können die oben erwähnten Unfruchtbarkeits-Kennzeichnungen verwendet werden, um zu verhindern, daß eine noch nicht gescheiterte Schaltstrategie fehlerhafte Auswahlentscheidungen von bereits erfolglos abgebrochenen Schaltstrategien wiederholt. Dabei wird mindestens eine heuristische Auswahlentscheidung der abgebrochenen Schaltstrategie, unter einer Markierung einen bestimmten Schaltschritt auszuführen, im Rumpferreichbarkeitsgraphen als unfruchtbare Schaltkante ausgewiesen⁹⁴⁾. Es handelt sich um diejenige Schaltkante, die vom Knoten der vorgenannten Markierung ausgeht und mit dem dort ausgewählten Schaltschritt beschriftet ist. Falls die Anwendung einer anderen, bis dahin noch nicht abgebrochenen Schaltstrategie zum selben Markierungsknoten führt, liegt bereits die Information vor, daß der Schaltschritt der unfruchtbaren Schaltkante gemäß der alten - jedoch abgebrochenen - Schaltstrategie zu keinem der gesuchten finalen Schaltprozesse führen kann. Falls die alternative Schaltstrategie in diesem Knoten zufällig den gleichen Schaltschritt auswählt und danach genau so wie die gescheiterte Schaltstrategie fortfahren würde, so läßt sich ihre Anwendung ohne weitere Untersuchung abbrechen. Denn sie vermag mit Sicherheit keinen finalen Schaltprozeß mehr hervorzubringen. Daraus kann wiederum die Erkenntnis über eine unfruchtbare Schaltkante abgeleitet werden⁹⁵⁾ usw.

Auf diese Weise wächst mit zunehmender Ausführungsdauer der Netzsimulation nicht nur der schritthaltend mitkonstruierte Rumpferreichbarkeitsgraph an. Vielmehr nimmt auch die Anzahl der darin ausgewiesenen unfruchtbaren Schaltkanten zu. Dies wirkt der kombinatorischen Explosion von Erreichbarkeitsgraphen tendenziell entgegen. Denn diejenigen Bezirke im anwachsenden Rumpferreichbarkeitsgraphen, die aufgrund ihrer Unfruchtbarkeits-Kennzeichnungen nicht weiter untersucht zu werden brauchen, weiten sich ebenfalls aus. Daher vereinigt das ATMS-Konzept bei der Untersuchung finaler Schaltprozesse durch projektive oder komparative Netzsimulationen zwei interessante Aspekte. Einerseits beschleunigt es die Suche nach finalen Schaltprozessen durch *nebenläufiges* Anwenden mehrerer konkurrierender Schaltstrate-

gien. Hierdurch unterstreicht es die inhärente Nebenläufigkeit des Petrinetz-Konzepts. Andererseits bereichert es die Netzsimulation um die partielle Konstruktion eines Erreichbarkeitsgraphen. Die darin erfolgende, mit der Simulationsdauer anwachsende Ausgrenzung von unfruchtbaren Bezirken, die mit Sicherheit zu keinen finalen Schaltprozessen führen, trägt abermals dazu bei, die Geschwindigkeit der Suche finaler Schaltprozesse zu erhöhen⁹⁶⁾.

Bisher wurden nur Möglichkeiten aufgezeigt, wie sich Konstruktion und Auswertung von Erreichbarkeitsgraphen seitens der Automatischen Informationsverarbeitung wirksam unterstützen lassen. Daher wendet sich ein abschließender Ausblick den Möglichkeiten zu, die Erreichbarkeitsanalyse von Netzmodellen durch Interaktionen zwischen einem Modellbenutzer und einem Automatischen Informationsverarbeitungssystem zu bereichern. Ausgangspunkt ist die graphische Visualisierung desjenigen Netzmodells, das dem auszuwertenden Erreichbarkeitsgraphen zugrundeliegt. Auf die kognitive Adäquanz von graphisch visualisierten Netzen wurde schon an früherer Stelle hingewiesen. Diese Adäquanz veranlaßt zu hoffen, daß sich die Intuition und Kreativität⁹⁷⁾ eines Modellbenutzers einsetzen lassen, um bei der Exploration eines Erreichbarkeitsgraphen seiner kombinatorischen Explosion auf "intelligente" Weise entgegenzuwirken⁹⁸⁾.

Fortan wird unterstellt, daß ein Automatisches Informationsverarbeitungssystem benutzt wird, um ein Netzmodell zusammen mit seinem Erreichbarkeitsgraphen informationstechnisch zu verwalten. Zugleich dient ein hochauflösender Graphikbildschirm dazu, das graphisch visualisierte Netzmodell benutzerfreundlich zu präsentieren. Schließlich wird davon ausgegangen, daß eine Netzsimulation mit schritthaltender Teilkonstruktion des Erreichbarkeitsgraphen erfolgt⁹⁹⁾. Jeder Simulationsschritt besteht in der Ausführung eines zulässigen Schaltschritts. Die sukzessive Abfolge der Simulationsschritte stellt einen Schaltprozeß dar, der einen Ausschnitt aus dem Erreichbarkeitsgraphen des untersuchten Netzmodells erzeugt. In Abhängigkeit vom jeweils verfolgten Analyseziel kann dieser Schaltprozeß ausgewertet, gegebenenfalls auch - teilweise oder vollständig - zurückgenommen und in anderer Richtung fortgesetzt werden. Wie sich eine derart verschränkte Konstruktion und Auswertung von Erreichbarkeitsgraphen zur projektiven oder komparativen Analyse von Netzmodellen einsetzen läßt, wurde schon oben aufgezeigt. Daher wird auf diesen Aspekt nicht weiter eingegangen.

Vielmehr interessiert jetzt nur noch die Ausführung eines einzelnen Simulationsschritts. Dafür gilt es einen auszuführenden Schaltschritt zu bestimmen. Dieses Ermittlungsproblem unterliegt der kombinatorischen Explosion von Erreichbarkeitsgraphen: Unter der aktuellen Markierung des Netzmodells sind in der Regel mehrere Transitionen aktiviert. Hinzu kommt noch, daß des öfteren dieselbe aktivierte Transition infolge von Abundanzkonflikten mit mehreren Schaltfarben geschaltet werden kann. Sowohl die potentielle Vielfalt der aktivierten Transitionen als auch die potentielle Vielfalt der transitionsspezifisch aktivierenden Schaltfarben eröffnen das kombinatorische Potential aller Schaltschritte, die unter der aktuellen Netzmarkierung zu untersuchen sind. Dabei kann es sich herausstellen, daß potentielle Schaltschritte unzulässig sind, weil ihre zugehörigen Transitionen mit den jeweils zugeordneten Schaltfarben konfliktionär aktiviert sind. Aber dies kann erst festgestellt werden, *nachdem* ein solcher potentieller Schaltschritt durch das Zusammenfassen von aktivierten Transitionen und aktivierenden Schaltfarben gebildet worden ist.

Aufgrund der vorausgesetzten Netzsimulation wird die kombinatorische Explosion des Erreichbarkeitsgraphen dadurch "bewältigt", daß unter jeder Netzmarkierung genau ein zulässiger Schaltschritt ausgeführt wird¹⁰⁰⁾. Die Ermittlung dieses Schaltschritts soll geschehen, indem ein Modellbenutzer mit dem modellverwaltenden Automatischen Informationsverarbeitungssystem interagiert. Eine solche Interaktion kann z.B. wie folgt ausgestaltet werden¹⁰¹⁾:

- Zunächst stellt das Automatische Informationsverarbeitungssystem fest, welche Transitionen unter der aktuellen Markierung des Netzmodells mit welchen Schaltfarben aktiviert sind. Dazu reicht es aus, die Schaltregel für Synthetische Netze nacheinander auf alle Transitionen des Netzmodells anzuwenden. Falls eine Transition mit mehreren Schaltfarben aktiviert ist, sorgt die angeregte PROLOG-Implementierung des Netzmodells dafür, daß alle aktivierenden Schaltfarben automatisch erkannt werden.
- Das Automatische Informationsverarbeitungssystem zeigt das Ergebnis seiner Aktivierungsanalyse am Graphikbildschirm an. Im graphisch visualisierten Netzmodell werden die aktivierten Transitionen besonders hervorgehoben. Beispielsweise lassen sich die aktivierten Transitionen mit einer aggressiven Signalfarbe - etwa Rot - oder durch ein periodisches Blinken ihres Namens kenntlich machen¹⁰²⁾. In oder neben dem Graphiksymbol jeder aktivierten Transition befindet sich ein kleines Informationsfeld, das die Anzahl der aktivierenden Schaltfarben wiedergibt. Durch "Anklicken" des Informationsfelds mit einer "Maus" kann der Benutzer ein "Fenster" öffnen, in dem er Informationen über die Markenkopien aller involvierten Schaltfarben vorfindet.
- Der Modellbenutzer legt ohne jede weitere Automatenhilfe einen potentiellen Schaltschritt fest. Z.B. kann er alle aktivierten Transitionen "anklicken", die zum potentiellen Schaltschritt gehören sollen. Für alle ausgesuchten Transitionen, die durch mehrere Schaltfarben aktiviert sind, muß der Benutzer zusätzlich die gewünschte Schaltfarbe angeben¹⁰³⁾. Bei dieser Schaltschrittfestlegung wird unterstellt, daß die anschauliche Visualisierung des Ergebnisses der automatischen Aktivierungsanalyse die intuitiven und kreativen Fähigkeiten des Benutzers stimuliert¹⁰⁴⁾: In unmittelbarer Anschauung des "vernetzten" Zusammenhangs der Transitionen soll der Modellbenutzer in der Lage sein, einen potentiellen Schaltschritt herauszufiltern, der für die Verwirklichung des jeweils verfolgten Analyseziels geeignet ist¹⁰⁵⁾.
- Es ist nicht auszuschließen, daß der Benutzer von seiner Intuition verleitet worden ist, einen Schaltschritt zusammenzustellen, dessen Transitionen mit den zugehörigen Schaltfarben konfliktionär aktiviert sind. Der festgelegte potentielle Schaltschritt ist dann unzulässig. Daher wird vom Automatischen Informationsverarbeitungssystem abschließend die Zulässigkeit des Schaltschritts überprüft. Wenn der Zulässigkeitstest positiv ausfällt, wird der festgelegte Schaltschritt tatsächlich ausgeführt. Der betrachtete Simulationsschritt ist beendet, und eine neue Netzmarkierung liegt vor. Andernfalls wird der Schaltschritt als unzulässig zurückgewiesen. Der Modellbenutzer wird aufgefordert, einen anderen potentiellen Schaltschritt festzulegen¹⁰⁶⁾.

Die voranstehend skizzierte Interaktion bei der Ausführung eines Simulationsschritts läßt sich in mehrfacher Hinsicht ausbauen. Von den vielfältigen Optionen wird hier nur eine kleine Auswahl angeschnitten. Zunächst wird die Einschränkung gelockert, daß der Modellbenutzer bei der Festlegung eines potentiellen Schaltschritts ausschließlich auf seine Intuition und Kreativität vertraut. Statt dessen läßt der Benutzer jetzt genau einen potentiellen Schaltschritt automatisch erzeugen und am Graphikbildschirm anzeigen. Es bietet sich dann an, die automatische Zulässigkeitsprüfung in die Schaltschrittgenerierung zu integrieren. Das Automatische Informationsverarbeitungssystem bietet daher seinem Benutzer nur einen solchen Schaltschritt an, von dem bereits feststeht, daß seine Transitionen mit den zugeordneten Schaltfarben konfliktfrei aktiviert sind. Der Modellbenutzer beurteilt dann intuitiv, ob er den vorgeschlagenen zulässigen Schaltschritt für geeignet hält. Wenn sein Urteil positiv ausfällt, wird der Schaltschrittorschlag des Automaten vom Benutzer als Schaltschrittfestlegung akzeptiert. Andernfalls veranlaßt der Modellbenutzer das Automatische Informationsverarbeitungssystem, einen anderen zulässigen Schaltschritt vorzuschlagen. Der Zyklus von Schaltschrittorschlag und Schaltschrittbeurteilung wird so oft wiederholt, bis der Benutzer einen vorgeschlagenen Schaltschritt für geeignet hält¹⁰⁷⁾. Falls alle zulässigen Schaltschritte generiert worden sind¹⁰⁸⁾, jedoch keiner vom Benutzer akzeptiert wurde, stehen drei Optionen offen:

- Der Modellbenutzer akzeptiert nachträglich doch noch einen derjenigen Vorschläge, die er früher verworfen hat, als "kleinstes Übel"¹⁰⁹⁾.
- Er unterbricht die Netzsimulation an der aktuellen Netzmarkierung und setzt den Schaltprozeß zurück, der bis dahin ausgeführt worden ist. Das Ausmaß, in dem die Prozeßausführung zurückgenommen wird, kann der Benutzer bestimmen. Danach befindet sich das Netzmodell in einem Markierungszustand, der schon einmal früher erreicht worden war. Von dort aus wird die Netzsimulation so wiederaufgenommen, daß derjenige Schaltschritt, der an dieser Stelle vormals ausgeführt worden war, ausgeschlossen wird. Die Exploration des Erreichbarkeitsgraphen wird deshalb in einer anderen Richtung fortgesetzt, als es zuvor geschehen war.
- Der Benutzer bricht die Netzsimulation in der aktuellen Markierung endgültig ab. Er beurteilt die Erkenntnisse, die aus Schaltprozeßausführungen bei der Exploration des Erreichbarkeitsgraphen gewonnen wurden, angesichts des jeweils verfolgten Analyseziels.

Der Modellbenutzer kann die Unterstützung des Automatischen Informationsverarbeitungssystems in noch größerem Umfang in Anspruch nehmen: Er läßt sich nicht nur einen zulässigen Schaltschritt vorschlagen, sondern ein Menü aus *mehreren* zulässigen Schaltschritten. Dabei besitzt der Benutzer den Freiheitsgrad, die Anzahl K der Schaltschritte vorzugeben, die das Menü umfassen soll (mit $K \in \mathcal{N}_+$ und $K \geq 2$)¹¹⁰⁾. Diese quantitative Ausweitung der Automatenunterstützung kann eine qualitative Verbesserung der interaktiv betriebenen Erreichbarkeitsanalyse nach sich ziehen. Dabei liegt die Hypothese zugrunde, daß der Modellbenutzer die Eignung von zulässigen Schaltschritten für das verfolgte Analyseziel besser einzuschätzen vermag, wenn er eine begrenzte Anzahl von alternativen Schaltschritten gegeneinander abwägt¹¹¹⁾. Die kognitive Reichweite einer solchen Abwägung legt der Benutzer durch den Menüumfang K selbst fest.

Die Interaktion kann noch dadurch bereichert werden, daß im Automatischen Informationsverarbeitungssystem Heurismen für die Generierung von Schaltschrittorschlägen implementiert werden. Diese Heurismen erschließen Erfahrungswissen, das andere Personen bei früheren Erreichbarkeitsanalysen gesammelt haben¹¹²⁾. Diese Erfahrungen werden jetzt dem aktuellen Modellbenutzer in der "kondensierten" Form von Automatenorschlägen zugänglich gemacht. Hinzu kommen Generierungsregeln, die nicht aus früheren Erfahrungen gewonnen zu werden brauchen, weil sie sich aus den charakteristischen Eigenarten von Netzmodellen unmittelbar ableiten lassen. Als solche Heurismen kommen z.B. in Betracht¹¹³⁾:

- Zunächst werden nur solche aktivierten Transitionen in einem Schaltschrittorschlag zusammengefaßt, die weder gemeinsame Eingangs- noch gemeinsame Ausgangsstellen besitzen. Solche Schaltschritte sind per constructionem zulässig, weil die zugehörigen Transitionen niemals konfliktionär aktiviert sein können. Daher wird der Ressourceneinsatz für die Zulässigkeitsprüfung eingespart¹¹⁴⁾. Zugleich geschieht aber auch eine heuristische Perspektivenverengung, weil alle Schaltschritte, deren Transitionen konfliktionär aktiviert sein können - aber keineswegs müssen - unbeachtet bleiben.
- Der voranstehende Heurismus, konfliktionäre Aktivierungen von schaltschrittzugehörigen Transitionen von vornherein auszuschließen, wird radikalisiert: Es werden zunächst nur unäre Schaltschritte vorgeschlagen. Da sie jeweils aus genau einer aktivierten Transition bestehen, müssen sie auf jeden Fall zulässig sein. Allerdings wird mit diesem Radikalheurismus auf den ebenso interessanten Aspekt des Petrinetz-Konzepts verzichtet, nebenläufiges Schaltverhalten zuzulassen.

Es kann der Fall eintreten, daß alle Schaltschrittorschläge, die mittels eines der beiden voranstehenden Heurismen erzeugt wurden, vom Modellbenutzer insgesamt als ungeeignet verworfen worden sind¹¹⁵⁾. Dann müssen auf einer nächsten Vorschlagsstufe auch andere Schaltschritte berücksichtigt werden, deren zugehörige Transitionen grundsätzlich konfliktionär aktiviert sein können. Daher besteht das unvermeidliche Risiko, potentielle Schaltschritte vorläufig zu erzeugen

gen, die aufgrund eines anschließenden Zulässigkeitstests wieder verworfen werden müssen¹¹⁶). Solche gescheiterten Generierungsversuche verursachen einen unproduktiven Ressourcenverzehr. Abermals lassen sich Heurismen einsetzen, um das Generierungsrisiko von unzulässigen Schaltschritten gegenüber einer rein zufälligen Schaltschritterzeugung zu verringern:

- Die zunächst vorgeschlagenen zulässigen, aber vom Benutzer verworfenen Schaltschritte werden auf kleinstmögliche Weise inkrementell erweitert¹¹⁷: Ihnen wird zunächst jeweils genau eine aktivierte Transition hinzugefügt. Falls hierdurch keine konfliktionäre Aktivierung eintritt, wird der erweiterte zulässige Schaltschritt dem Modellbenutzer vorgeschlagen¹¹⁸). Erst wenn alle Schaltschritterweiterungen um jeweils eine Transition ausgeschöpft sind, aber keiner der Schaltschrittorschläge vom Benutzer akzeptiert wurden, wird zu Schaltschritterweiterungen um jeweils zwei aktivierte Transitionen übergegangen. Das Vorschlagsverfahren wird mit sukzessiv wachsender Kardinalität der Schaltschritterweiterungen analog wiederholt, bis der Modellbenutzer entweder einen Schaltschrittorschlag als geeignet akzeptiert oder aber alle aktivierten Transitionen für Schaltschritterweiterungen herangezogen worden sind.
- Bei jeder Exploration desselben Netzmodells kann registriert werden, welche aktivierten Transitionen bei dem Versuch, sie zu einem Schaltschritt zusammenzufassen, zu einem Schaltkonflikt geführt haben. Solche Kombinationen aus konfliktionär aktivierten Transitionen lassen sich als "gefährliche" Transitionenkombinationen speichern. Bei jedem späteren Schaltschrittorschlag kann dieses Erfahrungswissen im Zusammenhang mit dem voranstehenden Heurismus genutzt werden¹¹⁹): Bei der Schaltschritterweiterung werden zunächst alle aktivierten Transitionen gemieden, deren Vereinigung mit dem jeweils vorliegenden Schaltschritt zu einer gefährlichen Transitionenkombination führen würde. Erst wenn alle ungefährlichen Schaltschritterweiterungen erzeugt und entweder wegen Unzulässigkeit zurückgezogen oder aber vom Benutzer als ungeeignet verworfen worden sind, werden auch die gefährlichen Transitionenkombinationen für inkrementelle Schaltschritterweiterungen in Betracht gezogen¹²⁰).
- Wenn Schaltprioritäten für Transitionen zugelassen werden¹²¹), bietet sich ein weiterer Heurismus an, um die o.a. inkrementellen Schaltschritterweiterungen "intelligent" generieren zu lassen: Für eine Schaltschritterweiterung um eine aktivierte Transition werden stets zwei Bedingungen geprüft¹²²). Erstens wird untersucht, ob es sich um eine Transition handelt, für die eine Schaltpriorität definiert ist¹²³). Zweitens wird festgestellt, ob zum betrachteten Schaltschritt bereits eine aktivierte Transition mit einer höheren Schaltpriorität gehört. Falls beide Bedingungen zutreffen, wäre eine Schaltschritterweiterung um die Transition mit der niedrigeren Priorität unzulässig¹²⁴). Folglich unterbleibt die Schaltschritterweiterung um die aktivierte Transition von vornherein.

Alle voranstehend skizzierten Heurismen beruhen auf der gleichen Basisstrategie, von Schaltschritten kleinen Umfangs sukzessiv zu umfangreicheren Schaltschritten überzugehen. Dies wurde damit gerechtfertigt, daß sich auf diese Weise unproduktiver Ressourceneinsatz tendenziell gering halten läßt: Es wird das Risiko gesenkt, Schaltschritte vorläufig generieren zu lassen, die beim anschließenden Zulässigkeitstest wegen konfliktionärer Aktivierung ihrer Transitionen zurückgenommen werden müssen.

Allerdings führt diese ressourcenorientierte Basisstrategie dazu, das Nebenläufigkeitspotential des untersuchten Netzmodells zunächst nur gering auszuschöpfen. Zwar kann der Modellbenutzer die anfangs vorgeschlagenen Schaltschritte als ungeeignet zurückweisen, weil er an Schaltschritten interessiert ist, die möglichst viele nebenläufig aktivierte Transitionen umfassen. Doch dann wäre die heuristische Basisstrategie verfehlt. Denn die knappen Informationsverarbeitungsressourcen würden zum Beginn der interaktiven Untersuchung eines Simulationsschritts für solche Schaltschrittorschläge "vergeudet", an denen der Modellbenutzer wegen ihrer geringen Anzahl nebenläufig aktivierter Transitionen ohnehin kein Interesse hegt. Daher liegt es nahe,

alternative Heuristiken vorzuhalten, die zunächst möglichst umfangreiche Schaltschritte erzeugen¹²⁵).

Am einfachsten ist es, einen maximalen Schaltschritt zu erzeugen, der alle Transitionen umfaßt, die unter der aktuellen Netzmarkierung aktiviert sind. Dabei brauchen nur noch die aktivierenden Schaltfarben bestimmt zu werden, falls mindestens eine Transition mit mehreren Schaltfarben aktiviert ist¹²⁶). Ein derart maximaler Heurismus bedeutet jedoch ein überaus großes Risiko, daß die beteiligten Transitionen konfliktionär aktiviert sind. Für eine subtilere Ausgestaltung des Heurismus bietet es sich an, abermals auf das Konzept gefährlicher Transitionenkombinationen zurückzugreifen: Zwar wird zuerst auch noch ein maximaler Schaltschritt erzeugt. Unmittelbar anschließend erfolgt aber ein Abgleich mit allen bis dahin bekannten gefährlichen Transitionenkombinationen. Falls keine gefährliche Transitionenkombination eine Teilmenge der Transitionenmenge des maximalen Schaltschritts ist, wird abschließend die Zulässigkeit des Schaltschritts überprüft. Andernfalls werden aus dem maximalen Schaltschritt so lange einzelne aktivierte Transitionen herausgenommen¹²⁷), bis keine der gefährlichen Transitionenkombinationen mehr eine Teilmenge aus der Transitionenmenge des reduzierten Schaltschritts ist. Erst dann erfolgt die Zulässigkeitsprüfung des reduzierten Schaltschritts. In beiden Fällen wird der Schaltschritt - analog zum Vorhergesagten - nur dann dem Benutzer vorgeschlagen, wenn das Prüfungsergebnis seiner Zulässigkeit positiv ausgefallen ist.

Schließlich können für den speziellen Fall, daß sich der Benutzer ein Menü aus mehreren Schaltschrittorschlägen anzeigen läßt, Empfehlungen hinsichtlich des Menüumfangs K vorgehalten werden. Beispielsweise läßt sich eine Voreinstellung wählen, die aus kognitionspsychologischen Erwägungen über die "optimale" Reichweite relativer Alternativenvergleiche abgeleitet ist. Ebenso ist es möglich, daß ein Automatisches Informationsverarbeitungssystem bei allen seinen interaktiven Benutzungen durch verschiedene Personen den jeweils gewählten Menüumfang protokolliert. Der Durchschnitt daraus, der eventuell um "Ausreißer" bereinigt ist, kann als "bewährter" Menüumfang empfohlen werden.

Anmerkungen zum Kapitel:

- 1) Dies wurde schon früher für die einfachere Klasse der Stelle/Transition-Netze ausführlicher dargelegt.
- 2) Darüber hinaus lassen sich andere Aspekte von Netzen untersuchen, die in dieser Arbeit entweder nicht näher interessierten oder aber nicht als eigenständige Netzeigenschaften hervorgehoben wurden. Beispielsweise führt ABEL,D. (1990), S. 64f., die Erreichbarkeit einer Markierung explizit als eine Netzeigenschaft an, die sich mit Hilfe der Erreichbarkeitsanalyse überprüfen läßt. Eine Markierung läßt sich genau dann erreichen, wenn sie als Knoten im Erreichbarkeitsgraphen enthalten ist. Allerdings hält der Verf. die Erreichbarkeit einer Markierung für so trivial, daß er sie nicht als eigenständige Netzeigenschaft thematisiert hat.
- 3) Eine Ausnahme bildet die Netzeigenschaft der sw^* -Optimalität. Für sie wird im anschließenden Kapitel eine entsprechende Auswertungsmethode eingehender dargestellt.
- 4) Vgl. dazu die Finitheitsprämisse.
- 5) Entsprechende Hinweise auf die Fülle von Netzeigenschaften, die sich durch die Auswertung von Erreichbarkeitsgraphen überprüfen lassen, finden sich auch bei BAUMGARTEN (1978), S. 37; RAZOUK (1985c), S. 19f.; ABEL,D. (1990), S. 64ff.; PAGNONI (1990), S. 23, 143 u. 161.
- 6) Anschauliche Beispiele dafür finden sich bei ABEL,D. (1990), Abb. 3.4 b) auf S. 25 u. Abb. 5.9 a) auf S. 74; vgl. auch in dieser Arbeit das Beispiel zweier Fertigungslinien.
Einen bemerkenswerten Algorithmus zur Aufdeckung von Deadlocks hat VALMARI (1988a), S. 106ff. i.V.m. S. 97ff., vorgelegt. Um alle Deadlockmarkierungen eines Netzmodells aufzudecken, braucht nicht der gesamte Erreichbarkeitsgraph des Netzmodells konstruiert zu werden, sondern nur ein algorithmusspezifischer Ausschnitt. Dadurch wird die worst case-Komplexität des Auswertungsalgorithmus nicht mehr durch eine exponentielle, sondern nur noch durch eine quadratische Funktion der Problemgröße bestimmt; vgl. VALMARI (1988a), S. 96, 101f. u. 111.
- 7) Wegen der vorausgesetzten Endlichkeit des Erreichbarkeitsgraphen folgt aus dem Umstand, daß jeder Markierungsknoten mindestens eine Ausgangskante besitzt: Der Erreichbarkeitsgraph muß mindestens einen zyklischen Weg enthalten. Andernfalls würde die Bedingung, daß alle Markierungsknoten jeweils mindestens eine Ausgangskante besitzen müssen, notwendig zu einem unendlichen Erreichbarkeitsgraphen führen.
- 8) Wenn ein Netz stark lebendig ist, dann ist es ebenso sowohl transitionsschwach als auch markierungsschwach lebendig. Daher werden folgende zwei Bedingungen zusätzlich erfüllt: a) Jeder Markierungsknoten im Erreichbarkeitsgraphen besitzt mindestens eine Ausgangskante. Der Erreichbarkeitsgraph ist deshalb zyklisch (vgl. die voranstehende Anmerkung zur transitionsschwachen Netzlebendigkeit). b) Für jede Transition aus der Transitionenmenge des Netzes gilt: Vom Knoten der Ausgangsmarkierung geht mindestens ein Schaltweg aus, der mindestens eine Schaltkante umfaßt, in deren Anschrift diese Transition enthalten ist. Die beiden zusätzlichen Bedingungen erweisen sich aber als redundant. Denn sie werden auf jeden Fall erfüllt, wenn die nachfolgend ausgeführte - komplexere - Lebendigkeitsbedingung zutrifft.
- 9) Der Erreichbarkeitsgraph besitzt immer mindestens einen maximalen stark zusammenhängenden Teilgraphen: Da der Erreichbarkeitsgraph endlich ist und jeder seiner Knoten mindestens eine Ausgangskante aufweist, muß der Erreichbarkeitsgraph mindestens einen zyklischen Schaltweg enthalten. Zumindest dieser Schaltweg konstituiert einen stark zusammenhängenden Teilgraphen. Denn auf einem zyklischen Weg kann jeder wegzugehörige Knoten von jedem anderen wegzugehörigen Knoten erreicht werden. Entweder ist ein stark zusammenhängender Teilgraph selbst maximal. Oder er ist in einen größeren stark zusammenhängenden Teilgraphen eingebettet. Da der zugrundeliegende Erreichbarkeitsgraph endlich ist, läßt sich die Einbettung nicht unendlich fortsetzen. Daher ist spätestens der Erreichbarkeitsgraph selbst ein maximaler stark zusammenhängender Teilgraph. Dabei ist zu beachten, daß ein Teilgraph im Grenzfall mit demjenigen Graphen zusammenfallen kann, bezüglich dessen Knoten- und Kantenmenge der Teilgraph definiert ist.
- 10) Ein gerichteter Graph heißt genau dann stark zusammenhängend, wenn in ihm jeder seiner Knoten von jedem anderen Knoten aus über mindestens einen Weg erreicht werden kann. Dabei wird entsprechend zur früheren Definition von Wegen in gerichteten Graphen vorausgesetzt, daß alle Kanten desselben Weges gleichsinnig ausgerichtet sind.
Vgl. zu stark zusammenhängenden Graphen HÄSSIG (1979), S. 20; STARKE (1980), S. 121; ABEL,D. (1990), S. 67; SEICHE (1990), S. 19.
Vgl. zu Algorithmen, mit deren Hilfe sich der starke Zusammenhang eines gerichteten Graphen feststellen läßt, HÄSSIG (1979), S. 20 i.V.m. S. 13; FREEDMAN (1988b), S. 337.
Ein stark zusammenhängender Teilgraph kann in einen - noch größeren - stark zusammenhängenden Teilgraphen eingebettet sein. Es erweist sich im allgemeinen als ineffizient, alle stark zusammenhängenden Teilgraphen zu untersuchen. Das trifft auch auf die hier betrachtete Lebendigkeitsanalyse zu. Vielmehr reicht es oftmals aus, sich auf jene stark zusammenhängenden Teilgraphen zu konzentrieren, die bei der Erweiterung um einen beliebigen Knoten ihre starke Zusammenhangseigenschaft verlieren würden. Es wird dann von maximalen stark zusammen-

hängenden Teilgraphen - oder kurz: maximalen Komponenten - gesprochen. Vgl. zu solchen maximalen stark zusammenhängenden Teilgraphen (starken Komponenten) ABEL,D. (1990), S. 67ff.

Algorithmen für die Identifizierung aller maximalen stark zusammenhängenden Teilgraphen (starken Komponenten) eines gerichteten Graphen finden sich bei ABEL,D. (1990), S. 68f.

11) Bei den verbleibenden starken Komponenten handelt es sich entweder um den stark zusammenhängenden Erreichbarkeitsgraphen selbst oder aber um die Gesamtheit seiner starken Komponenten, die reine Senken darstellen. Denn es bestehen nur zwei Möglichkeiten:

Einerseits kann der Erreichbarkeitsgraph stark zusammenhängend sein. Dann stellt der Erreichbarkeitsgraph selbst genau eine starke Komponente dar. Da keine weitere starke Komponente definiert ist, handelt es sich beim Erreichbarkeitsgraphen insgesamt weder um eine Quelle noch um eine Senke. Es ist keine Senke definiert, die sich als starke Komponente ausschließen ließe. Folglich wird der Erreichbarkeitsgraph unverändert als genau eine verbleibende starke Komponente in den anschließenden Analyseschritt übernommen.

Andererseits ist es möglich, daß der Erreichbarkeitsgraph selbst nicht stark zusammenhängend ist. Dann werden alle starken Komponenten des Erreichbarkeitsgraphen betrachtet. Er besitzt immer mindestens eine starke Komponente. Das wurde kurz zuvor aufgezeigt. Aus allen starken Komponenten des Erreichbarkeitsgraphen werden alle Quellen aussortiert. Für jede aussortierte Quelle muß es per definitionem mindestens eine starke Komponente geben, die - bezogen auf die zugehörige Quelle - eine Senke darstellt. Diese Senke kann aber ihrerseits die Quelle einer anderen starken Komponente sein, die wiederum eine Senke darstellt. Es bestünde nur eine Möglichkeit, daß jede Senke zugleich auch die Quelle einer anderen starken Komponente ist. Dieser Fall träte genau dann ein, wenn alle starken Komponenten zyklisch miteinander verknüpft sind. Ein solcher Zyklus bedeutete aber, daß jeder Markierungsknoten des Erreichbarkeitsgraphen von jedem anderen Markierungsknoten aus über diesen Zyklus erreicht werden kann. Folglich müßte der Erreichbarkeitsgraph als Ganzes stark zusammenhängend sein. Dies widerspräche aber der Voraussetzung, daß der Erreichbarkeitsgraph nicht stark zusammenhängend ist. Folglich muß mindestens eine starke Komponente existieren, die eine Senke darstellt und zugleich keine Quelle ist. Mindestens diese eine starke Komponente verbleibt für den anschließenden Analyseschritt.

12) Vgl. ULLRICH (1976), S. 5/7f.; BERTHELOT (1977), S. 219; STARKE (1980), S. 119; SEICHE (1990), S. 20 i.V.m. S. 19; ABEL,D. (1990), S. 72 (mit Beweis auf S. 72f. und vorbereitenden Erläuterungen auf S. 70ff.).

Die Ausführungen von ABEL und SEICHE beziehen sich zwar auf kondensierte Erreichbarkeitsgraphen. Diese werden aber erst später vorgestellt. Daher werden die Beiträge der beiden vorgenannten Autoren hier ohne Bezug auf kondensierte Erreichbarkeitsgraphen ausgelegt.

13) Dies wird im nächsten Kapitel ausführlich erörtert.

14) Für das echte Terminieren eines Netzes müssen potentiell unendliche Schaltprozesse ausgeschlossen werden. Hierbei kann es sich einerseits um einen azyklischen potentiell unendlichen Schaltprozeß handeln. Dann ist aber auch der zugehörige Erreichbarkeitsgraph unendlich. Dies wird in dieser Arbeit jedoch ausgeschlossen. Andererseits kann ein zyklischer Schaltprozeß vorliegen, der sich in einem endlichen Erreichbarkeitsgraphen unbegrenzt oft wiederholen läßt. Dieser zweite Fall wird durch die o.a. Bedingung der Zyklensfreiheit verboten.

15) Weitere, inhaltlich äquivalente Kriterien für die M_0 -Reversibilität eines Netzes sind:

- Der Erreichbarkeitsgraph ist als Ganzes stark zusammenhängend; vgl. ABEL,D. (1990), S. 70; SEICHE (1990), S. 20.
- Der Erreichbarkeitsgraph stellt genau eine starke Komponente dar; vgl. SEICHE (1990), S. 20.

ABEL und SEICHE beziehen sich zwar nicht auf M_0 -, sondern auf M -Reversibilität. Es wurde jedoch schon gezeigt, daß beide Netzeigenschaften äquivalent sind. Darüber hinaus läßt sich die Äquivalenz der beiden voranstehenden Reversibilitätskriterien mit dem oben angeführten leicht nachvollziehen: a) Wenn durch jeden Markierungsknoten M_i mit $M_i \in (RM(M_0) - M_0)$ mindestens ein zyklischer Weg verläuft, der auch den Markierungsknoten M_0 umfaßt, dann gehört *jeder* Markierungsknoten M_i des Erreichbarkeitsgraphen zu mindestens einem zyklischen Weg. Jeder zyklische Weg stellt einen stark zusammenhängenden Teilgraphen dar. Alle stark zusammenhängenden Teilgraphen hängen zumindest über die Ausgangsmarkierung M_0 miteinander zusammen. Folglich ist es möglich, von jedem stark zusammenhängenden Teilgraphen zumindest über die Ausgangsmarkierung in jeden anderen stark zusammenhängenden Teilgraphen zu gelangen. Folglich ist der Erreichbarkeitsgraph als Ganzes stark zusammenhängend. In einem Graphen, der als Ganzes stark zusammenhängend ist, gibt es keine Knoten mehr, deren Hinzufügen die starke Zusammenhangseigenschaft zerstören können. Deshalb ist der Erreichbarkeitsgraph zugleich ein maximaler stark zusammenhängender Graph, d.h. eine starke Komponente. b) Wenn ein Erreichbarkeitsgraph als Ganzes stark zusammenhängend ist - und somit eine starke Komponente darstellt -, dann gilt für den Knoten seiner Ausgangsmarkierung M_0 und jeden anderen Knoten für eine beliebige erreichbare Markierung M_i mit $M_i \in (RM(M_0) - M_0)$: Es ist immer möglich, vom Knoten der Ausgangsmarkierung M_0 zunächst zum Markierungsknoten M_i zu gelangen und danach vom Markierungsknoten M_i zum Knoten der Ausgangsmarkierung M_0 zurückzugelangen. (Andernfalls wäre der Erreichbarkeitsgraph nicht als Ganzes stark zusammenhängend.) Folglich liegt jeder Markierungsknoten M_i mit $M_i \in (RM(M_0) - M_0)$ auf mindestens einem zyklischen Weg, der auch durch den Knoten der Ausgangsmarkierung verläuft. (q.e.d.)

Darüber hinaus wertet SEICHE nicht die Erreichbarkeitsgraphen selbst, sondern eine kondensierte Spielart von Erreichbarkeitsgraphen aus. Auf ihre Konstruktion und Anwendung wird in Kürze zurückgekommen.

16) Damit wird zugleich das Vorurteil widerlegt, beim Übergang zu Höheren Netzen ließen sich interessante Eigenschaften von Netzmodellen nicht mehr untersuchen. Diese Ansicht wird z.B. von ESCHENBACHER (1991), S. 223, vertreten: "... wurden die ursprünglichen Petri-Netze stark verallgemeinert. Dadurch eignet sich ein solches Modell nicht mehr für theoretische Untersuchungen wie Deadlock-Freiheit." ESCHENBACHER scheint dabei nur an Deadlockfreiheits- oder Lebendigkeitsuntersuchungen mit Hilfe der Invariantenanalyse zu denken. Diese Analysevariante läßt sich in der Tat auf Höhere Netze nur noch unter erheblichen Schwierigkeiten anwenden. ESCHENBACHER übersieht jedoch vollkommen, daß sein Interesse an der Deadlockfreiheit von Netzmodellen weiterhin mittels der Erreichbarkeitsanalyse erfüllt werden kann (s.o.). Dabei ist entscheidend, daß die Erreichbarkeitsanalyse sich auf (nahezu) alle Netzklassen unabhängig davon anwenden läßt, in welchem Ausmaß sie "verallgemeinert" wurden.

17) Die Fragen werden jeweils bearbeitet, indem die Erreichbarkeitsgraphen der zugrundeliegenden Netzmodelle ausgewertet werden. Dabei stellen die Markierungsknoten der Erreichbarkeitsgraphen die zulässigen Zustände der Netzmodelle dar. Zulässige Verhaltensweisen - oder synonym: Schaltprozesse - der Netzmodelle äußern sich in ihren Erreichbarkeitsgraphen als Schaltwege.

18) Dafür spricht, daß jeder Bearbeitungszyklus unter den gleichen Ausgangsbedingungen gestartet werden kann. Hierin liegt im allgemeinen eine Vereinfachung von Koordinierungskonzepten. Denn es brauchen nicht mehrere Koordinierungsalternativen für unterschiedliche Ausgangsbedingungen untersucht zu werden.

19) Vgl. die Erläuterungen deklarativer Netzmodelle, in denen es um die Reproduktion einer ausgezeichneten Ausgangsmarkierung M_0 (Nullmarkierung) ging. Vgl. des weiteren die Ausführungen zu der Möglichkeit, die Netzmodule, die für die Pseudocode-Notation von Algorithmen eingeführt worden waren, so zu präzisieren, daß sie die Gestalt M_0 -reversibler Synthetischer Netze annehmen.

20) Ein äquivalenter Test bestünde darin, das echte Terminieren des Netzmodells zu untersuchen.

21) Die Untersuchung der Formalzielwirkungen wird hier auf terminierende Schaltprozesse nur in *exemplarischer* Weise bezogen. Grundsätzlich lassen sich die Formalzielwirkungen aller Schaltprozesse beleuchten, sobald den Transitionen eines Netzmodells entsprechende Schaltwerte zugeordnet worden sind. Beispielsweise könnten ebenso finale Netzmodelle betrachtet werden, die nicht echt terminierend, sondern M_0 -reversibel sind. Auch dort böte es sich an, die Formalzielwirkungen von finalen, nicht terminierenden, aber auch nicht zyklisch wiederholten Schaltprozessen in den drei anschließend vorgestellten Kategorien zu untersuchen.

22) Vgl. dazu die Ausdifferenzierung der pragmatischen Analysedimension von Netzmodellen in projektive, komparative und optimierende Netzuntersuchungen.

23) Allerdings könnte der Einwand erhoben werden, daß im Sinne der Handlungstheorie rationaler Akteure projektive und komparative Modelluntersuchungen für die praktische Problembewältigung eine weitaus bedeutsamere Rolle spielen als alle Versuche, optimale Modellverhaltensweisen zu bestimmen. Einen solchen Vorbehalt möchte der Verf. keineswegs von der Hand weisen. Es bereitet aber auch keine Schwierigkeiten, ihm gerecht zu werden. Denn projektive und komparative Netzanalysen werden bereits durch vielfältige Beiträge aus der Netzliteratur abgedeckt. Optimierende Netzanalysen finden dagegen weitaus weniger Beachtung. Dies gilt zumindest dann, wenn es sich um Netzmodelle handelt, die keinen rigiden Einschränkungen ihrer topologischen Strukturen unterliegen. Daher versteht der Verf. sein Konzept für optimierende Netzanalysen, das im anschließenden Kapitel vorgestellt wird, als ein bedingtes Angebot: *Falls* ein betriebswirtschaftliches Interesse an optimalen Netzverhaltensweisen besteht, dann lassen sie sich durch die angesprochenen Netzoptimierungen ermitteln.

24) Da in Synthetischen Netzen die Aktivierungsmöglichkeiten von schaltschrittzugehörigen Transitionen durch die Schaltfarben (Variablenbelegungen) festgelegt sind, wurde die Menge aller Schaltfarben, die für eine Transition unter einer Netzmarkierung zulässig sind, früher schon als ein "Spielraum" bezeichnet.

25) Der Schaltprozeß schrumpft dann auf einen einfachen Schaltprozeß zusammen. Die projektive "Analyse" der Formalzielwirkungen dieses einfachen Schaltprozesses fällt mit der Kenntnisnahme des schaltschrittzugehörigen Schaltwerts zusammen.

26) Der Handlungskomplex stellt im allgemeinen Fall eine halbgeordnete Menge von Handlungsoptionen dar, der im Netzmodell ein nebenläufiger Schaltprozeß zugeordnet ist. Als Grenzfall wird hiervon eine vollständig geordnete Optionenmenge umschlossen, die im Netzmodell durch einen sequentiellen Schaltprozeß abgebildet wird.

27) Zwar könnte der Einwand vorgetragen werden, daß die o.a. Vorteile nur die Untersuchung zulässiger semi-positiver T-Invarianten betreffen. Alle anderen T-Invarianten und sämtliche S-Invarianten würden dagegen von der Erreichbarkeitsanalyse nicht erfaßt. Diese Vorhaltung ist korrekt, aber unerheblich. Denn die früheren Ausführungen zur Invariantenanalyse zeigten, daß auch dort die Untersuchung zulässiger semi-positiver T-Invarianten dominiert. Semi-positive, aber unzulässige T-Invarianten sind im allgemeinen unerwünscht, weil sie keine Schaltvektoren von

zulässigen Schaltfolgen darstellen. T-Invarianten, die nicht semi-positiv sind, interessieren noch viel weniger, weil sie sich überhaupt nicht als Schaltvektoren von Schaltfolgen interpretieren lassen (sofern vom Sonderfall des Rückwärtsschaltens abgesehen wird). Andere fruchtbare Interpretationen für T-Invarianten sind bisher nicht bekannt geworden. Schließlich spielen S-Invarianten in der Invariantenanalyse nur eine periphere Rolle, da ihrer formalen Interpretation nur selten eine beachtenswerte materielle Bedeutung zukommt.

28) Dies korrespondiert mit der früher erwähnten Eigenschaft von zulässigen semi-positiven T-Invarianten, daß sie die Schaltvektoren von Schaltfolgen darstellen, die durch ihre Ausführung jeweils eine erreichbare Markierung M_r reproduzieren.

29) Semi-positive T-Invarianten konnten innerhalb der Invariantenanalyse auch unzulässig sein. Dies ist genau dann der Fall, wenn sie die Schaltvektoren nur solcher Schaltfolgen darstellen, die unter überhaupt keiner erreichbaren Markierung aktiviert sind.

30) Von jeder erreichbaren Markierung aus, die auf dem zyklischen Weg $w_{g,r}$ liegt, kann eine Schaltfolge gestartet und potentiell unendlich oft wiederholt werden, deren Schaltvektor durch die betrachtete semi-positive T-Invariante dargestellt wird. Dies entspricht genau der Definition von Livelock-Markierungen.

31) Dies war für die Invariantenanalyse der Fall.

32) Eine Transition t_n ist tatsächlich eine Livelock-Transition, falls sie zwei Bedingungen erfüllt. Erstens darf sie zu keinem derjenigen Schaltschritte gehören, mit denen die Schaltkanten aus dem Weg $w_{g,r}$ beschriftet sind. (Dies entspricht der Identifizierung potentieller Livelock-Transitionen seitens der Invariantenanalyse.) Zweitens muß die Transition t_n unter mindestens einer Markierung M_p deren Markierungsknoten zum betrachteten zyklischen Weg $w_{g,r}$ gehört, aktiviert sein. Dann ist diese Transition unendlich oft aktiviert, falls die Schaltfolge, die durch den zyklischen Weg $w_{g,r}$ repräsentiert wird, unendlich oft wiederholt ausgeführt wird. Wegen der ersten Bedingung wird diese unendlich oft aktivierte Transition niemals geschaltet. Folglich handelt es sich um eine Livelock-Transition. Um dies zu erkennen, muß im Erreichbarkeitsgraphen lediglich ermittelt werden, ob die Transition t_n unter mindestens einer Markierung M_p deren Markierungsknoten zum betrachteten zyklischen Weg $w_{g,r}$ gehört, aktiviert ist. Das ist genau dann der Fall, wenn die Transition t_n zu mindestens einem Schaltschritt gehört, mit dem mindestens eine Ausgangskante von mindestens einem Markierungsknoten M_i so beschriftet ist, daß gilt: Der Markierungsknoten M_i gehört zwar zum zyklischen Weg $w_{g,r}$. Aber die Ausgangskante dieses Knotens, die mit dem transitionsenthaltenden Schaltschritt beschriftet ist, gehört nicht zum Weg $w_{g,r}$. Ob die beiden voranstehenden Bedingungen für eine Livelock-Transition tatsächlich erfüllt sind, läßt sich für jede Transition anhand des Erreichbarkeitsgraphen eines Netzes feststellen.

33) Vgl. TIERNAN (1970), S. 722ff.; TARJAN (1973), S. 211ff., insbesondere S. 213f.; DÖRFLER (1973), S. 97f.; TSUKIYAMA (1975), S. 92ff.; NEUMANN, K. (1975b), S. 64f.; SZWARCFITER (1976), S. 192ff., insbesondere S. 196ff.; Hässig (1979), S. 33f. (i.V.m. S. 29ff., allerdings mit einer netzplanspezifischen Besonderheit, die hier für Zyklen in beliebigen gerichteten Graphen nicht erforderlich ist); NEUMANN, K. (1989a), S. 62f.; WYSK (1992), S. 226f.

34) Vgl. KNÖDEL (1969), S. 26ff.; KÜPPER, W. (1975), S. 28ff.; EISELT (1977), S. 151ff.

Von besonderem Interesse sind dabei Methoden, die eine nebenläufige (parallele) Ermittlung optimaler Wege in Graphen zulassen. Sie erlauben eine besonders schnelle Auswertung von Erreichbarkeitsgraphen durch das zeitgleiche Abarbeiten mehrerer Schaltwege. Vgl. zu solchen parallelen Algorithmen für die Wegoptimierung in Graphen IMAI (1979), S. 416ff.; NAKAGAWA (1979), S. 628ff.; KANAL (1981), S. 570f.

35) Vgl. zur Konstruktion und Anwendung solcher kondensierten Erreichbarkeitsgraphen ABEL, D. (1990), S. 66ff. u. 109f.; SEICHE (1990), S. 16ff., insbesondere S. 19f.

36) Anschauliche Beispiele für den kompaktifizierenden Effekt der Erreichbarkeitsgraphkondensierung finden sich bei ABEL, D. (1990), S. 71: Abb. 5.8 b) versus c), S. 74: Abb. 5.9 a) versus b), S. 109: Abb. 7.3 u. 7.4.

37) Allerdings wird der Effizienzvorteil durch den Ressourceneinsatz für die vorangehende Kondensierung eines Erreichbarkeitsgraphen partiell kompensiert. Zusätzliche Ressourcen werden benötigt, um für jede starke Komponente festzustellen, ob es sich um eine lebendige oder tote starke Komponente handelt. Die Definitionen dieser beiden konträren Ausprägungen von starken Komponenten werden in Kürze vorgelegt.

Praktische Anwendungen von kondensierten Erreichbarkeitsgraphen deuten jedoch darauf hin, daß der Effizienzvorteil ihrer einfachen Auswertung den retardierenden Kondensierungseinfluß insgesamt deutlich überwiegt. Dafür liegen dem Verf. zwar keine gesicherten empirischen Erkenntnisse vor. Doch die Beispiele von ABEL, D. (1990), S. 71, 74 u. 109f., weisen in diese Richtung. Darüber hinaus spricht auch ein theoretisches Tendenzargument zugunsten der Effizienzüberlegenheit von kondensierten Erreichbarkeitsgraphen. Denn wegen der kombinatorischen Explosion von Erreichbarkeitsgraphen bereitet es tendenziell weniger Aufwand, die starken Komponenten eines gewöhnlichen Erreichbarkeitsgraphen jeweils einzeln auf ihre Lebendigkeit oder Totheit hin zu untersuchen, als den Erreichbarkeitsgraphen insgesamt zu analysieren. Denn bei kombinatorischer Explosion von Untersuchungsobjekten schöpft jede Verkleinerung des jeweils analysierten Objektausschnitts das explosive Größenwachstum des Objekts

in umgekehrter Richtung aus: Die Ausschnittverkleinerungen bewirken tendenziell eine drastische Reduzierung des Untersuchungsaufwands.

Selbst wenn sich der mutmaßliche Effizienzvorteil nicht einstellen sollte, so bleibt zumindest der Vorzug, daß die kondensierten Erreichbarkeitsgraphen aufgrund ihrer Kompaktheit wesentlich anschaulicher ausfallen. Vgl. dazu die Beispiele für kondensierte Erreichbarkeitsgraphen. Daher unterstützen kondensierte Erreichbarkeitsgraphen kraft ihrer Transparenz zumindest die Benutzerfreundlichkeit von Erreichbarkeitsanalysen.

38) Vgl. ABEL,D. (1990), S. 68 (ansatzweise).

39) Strenggenommen bezieht sich SEICHE (1990), S. 19, nicht auf die M_{σ} , sondern auf die M-Reversibilität. Da beide Netzeigenschaften jedoch äquivalent sind, kann von dieser Nuancierung hier abgesehen werden.

40) Vgl. ABEL,D. (1990), S. 72; SEICHE (1990), S. 19.

41) Die Begriffe der Senke und Quelle wurden schon bei der Erörterung der starken Netzlebensdauer verwendet. Dort wurden sie jedoch auf die starken Komponenten eines gewöhnlichen Erreichbarkeitsgraphen bezogen. Hier gelten sie dagegen für die Knoten eines kondensierten Erreichbarkeitsgraphen.

42) Ein Knoten des kondensierten Erreichbarkeitsgraphen kann auch eine Quelle und eine Senke zugleich sein. Dies ist genau dann der Fall, wenn im kondensierten Erreichbarkeitsgraphen ein Weg durch den Knoten hindurchführt.

43) Jeder stark zusammenhängende Erreichbarkeitsgraph ist zugleich ein maximal stark zusammenhängender "Teilgraph". Ein stark zusammenhängender Erreichbarkeitsgraph stellt somit genau eine starke Komponente dar. In diesem Fall degeneriert der kondensierte Erreichbarkeitsgraph per constructionem zu dem genau einen Knoten, der die eine starke Komponente des gesamten zugrundeliegenden Erreichbarkeitsgraphen wiedergibt. Ein einfaches Beispiel dafür findet sich bei ABEL,D. (1990), S. 67f. (der Fall D_1).

44) Eine tote starke Komponente kann immer nur aus genau einem Markierungsknoten des Erreichbarkeitsgraphen bestehen (vgl. ABEL,D. (1990), S. 73). Denn per definitionem darf keine Transition in einem derjenigen Schaltschritte enthalten sein, mit denen die Kanten in der starken Komponente beschriftet sind. Jeder Schaltschritt besteht aber aus mindestens einer Transition. Folglich kann keine einzige mit einem Schaltschritt beschriftete Kante in der toten starken Komponente vorkommen. Daher muß die tote starke Komponente aus genau einem - in bezug auf die starke Komponente isolierten - Markierungsknoten des zugrundeliegenden Erreichbarkeitsgraphen bestehen.

45) Vgl. SEICHE (1990), S. 20; ABEL,D. (1990), S. 72f.

46) Eine tote starke Komponente besteht aus genau einem Markierungsknoten des zugrundeliegenden Erreichbarkeitsgraphen. Da die tote starke Komponente eine Senke des kondensierten Erreichbarkeitsgraphen darstellt, existiert einerseits im zugrundeliegenden Erreichbarkeitsgraphen mindestens ein Weg, auf dem der genau eine Markierungsknoten aus der toten starken Komponente erreicht werden kann. Andererseits besitzt der genau eine Markierungsknoten der toten starken Komponente keine Ausgangskante, weil diese Komponente eine *reine* Senke darstellt. Folglich liegt im zugrundeliegenden Erreichbarkeitsgraphen ein erreichbarer Markierungsknoten vor, von dem keine einzige Schaltkante ausgeht. Genau dadurch wird eine Deadlockmarkierung bestimmt.

47) Strenggenommen bezieht sich SEICHE (1990), S. 20, - wie schon in Anmk. 39 festgestellt wurde - nicht auf die M_{σ} , sondern auf die M-Reversibilität. Da beide Netzeigenschaften jedoch äquivalent sind, kann von dieser Nuancierung hier abgesehen werden. ABEL,D. (1990) spricht den Reversibilitätsaspekt im Zusammenhang mit kondensierten Erreichbarkeitsgraphen überhaupt nicht explizit an. Aber aus seinem Beispiel auf S. 70f., wird deutlich, daß er die oben getroffene Aussage teilt.

48) Problemgraphen wurden bereits anlässlich der problemtheoretischen Strukturierung von Prozeßkoordinierungen eingeführt.

49) Die problembeschreibenden Systeme werden auch kurz als Problemsysteme bezeichnet.

50) Vgl. zu solchen Konzepten für die Lösungssuche in Problemgraphen, Zustandsräumen und ähnlichen Konstrukten REITMAN (1964), S. 305ff.; RAPHAEL (1976), S. 142ff.; WINSTON (1977), S. 130ff.; BODEN (1977), S. 354ff.; KIRSCH (1977b), S. 173ff.; MEIBNER (1978), S. 35ff. u. 67ff.; BRUDERER (1978), S. 19ff.; SACERDOTI (1979), S. 1077ff.; NILSSON,N. (1980a), S. 32ff. u. 281ff.; BARR,A. (1981), S. 23f. u. 46ff.; WEIZENBAUM (1982), S. 229ff.; BUCHANAN (1982), S. 20ff.; MERTENS (1983), S. 692ff.; SIMON,H. (1983a), S. 14ff.; HUBER,O. (1983), S. 65ff.; BUNGERS (1984), S. 7ff.; LAURENT (1984), S. 150ff.; RETTI (1984), S. 81f.; ZELEWSKI (1986a), S. 241ff., 293 u. 295ff., insbesondere S. 249ff.; MILLER,D.P. (1988), S. 174ff.

Die vorgenannten Quellen erstrecken sich nicht nur auf die Lösungssuche in Problemgraphen. Statt dessen liegen die vielfach variierenden Formulierungen zugrunde, die bereits in einer früheren Anmerkung angesprochen wurden. So wurde beispielsweise schon auf das Konzept problembeschreibender Systeme hingewiesen.

51) Dagegen befaßt sich MILLER, D.P. (1988), S. 169ff., insbesondere S. 173ff., ausführlicher mit den Möglichkeiten, die Suchkonzepte der KI-Forschung auf produktionswirtschaftliche Maschinenbelegungsprobleme zu übertragen.

52) Vgl. POHL (1971), S. 130ff.; DE CHAMPEAUX (1975), S. 309ff.; DÖRNER, D. (1979), S. 69; NILSSON, N. (1980a), S. 34, 88ff. u. 253ff.; BARR, A. (1981), S. 51ff. u. 72f.; KOWALSKI (1983a), S. 99ff.; RETTI (1984), S. 82; ZELEWSKI (1986a), S. 253f.

53) Das Rücksetzen von Schaltschritten entspricht nicht dem früher angesprochenen - und ausgegrenzten - Rückwärtsschalten von Transitionen. Denn die Schaltschritte werden im zugrundeliegenden Netz immer nur so ausgeführt, wie es der Markenflußrichtung der Ein- und Ausgangskanten aller schaltschrittzugehörigen Transitionen entspricht. Daraus wird der Erreichbarkeitsgraph des Netzes abgeleitet. Danach werden Schaltschritte im Netz nicht mehr ausgeführt - und daher ihre Transitionen auch nicht rückwärts geschaltet. Das "Rücksetzen" von Schaltschritten bezieht sich nur auf den Erreichbarkeitsgraphen. Dabei wird von einer Folgemarkierung M_p , die über eine Schaltkante $ka_{r,a,f}=(M_r, M_p)$ mit der Referenzmarkierung M_r verknüpft ist, zur Referenzmarkierung M_r zurückgeschritten. Da die betroffene Schaltkante $ka_{r,a,f}$ mit dem Schaltschritt SS_a beschriftet ist, entspricht diese Rückwendung zur Referenzmarkierung M_r einem Widerruf ("Rücksetzen") des Schaltschritts SS_a . Dadurch wird der Schaltschritt SS_a aber nicht im zugrundeliegenden Netz tatsächlich - rückwärts - ausgeführt. Statt dessen wird lediglich im bereits vorliegenden Erreichbarkeitsgraphen zu einem - in Schalttrichtung gesehen - früheren Markierungsknoten zurückgegangen. Hierbei erfolgt kein Rückwärtsschalten von Transitionen. Denn es werden - wie an anderer Stelle näher erläutert wird - zusätzliche komplementäre Transitionen mit *entgegengesetzt* gerichteten Kanten eingeführt, um das Rücksetzen von Schaltschritten in Erreichbarkeitsnetzen durch Schaltakte von Transitionen tatsächlich ausführen zu können.

54) Als Effizienzmaßstab wird hier nur die Zeitdauer betrachtet, die bis zum Auffinden eines finalen Schaltprozesses verstreicht. Damit wird von Effizienzminderungen abgesehen, die durch zusätzlichen Ressourceneinsatz für die Ausführung der bidirektionalen Lösungssuche entstehen können. Vor allem ist an die Vervielfachung der Verarbeitungsressourcen zu denken, die zur zeitgleichen Abwicklung mehrerer Suchprozesse erforderlich ist. Daneben kann es auch zu Ressourcenverlusten durch erfolglose Suchprozesse kommen. Dies ist möglich, wenn zwei entgegengesetzt gerichtete Suchprozesse einander verfehlen. Dann durchläuft jeder von ihnen den *gesamten* Erreichbarkeitsgraphen. Dabei werden zwei finale Schaltwege erzeugt. Darin liegt eine Ressourcenvergeudung, sofern nur nach genau einem finalen Schaltweg gesucht worden war. Ein solches gegenseitiges Verfehlen schadet dagegen nicht, wenn ohnehin ein Interesse daran bestand, mehrere - unter Umständen sogar alle - finalen Schaltwege zu ermitteln. Dies kann z.B. bei Optimierungsanalysen der Fall sein, in denen nach *allen* sw^* -optimalen Schaltprozessen eines Netzmodells gesucht wird. Allerdings kann das gegenseitige Verfehlen zweier Schaltwege eine Effizienzeinbuße bedeuten, weil die nachfolgend vorgestellte, komplexitätsreduzierende Separation des Erreichbarkeitsgraphen in zwei Teilgraphen gescheitert ist.

55) Eine strenge Verdopplung der Suchgeschwindigkeit kann nicht garantiert werden. Beispielsweise läßt sich vorstellen, daß zwei entgegengesetzt gerichtete Suchprozesse erst nach einem kleinen Umweg aufeinanderstoßen, der bei einem einzelnen, unidirektionalen Suchprozeß nicht angefallen wäre.

56) MÄDLER (1990), S. 85, merkt allgemein für die Zerlegung von Problemen in Teilprobleme an: "Die Komplexität eines zerlegten Problems ist proportional zur *Summe* der Einzelkomplexitäten, während die Komplexität des unzerlegten Problems eher dem *Produkt* dieser Einzelkomplexitäten entspricht, das im allgemeinen ihre Summen weit übersteigt - und schneller zur kombinatorischen Explosion tendiert als die Teilprobleme." Vgl. dazu auch die formale Konkretisierung bei MÄDLER (1990), S. 87.

57) Sowohl Erreichbarkeitsgraphen im besonderen als auch Problemgraphen im allgemeinen können mit der Größe der jeweils zugrundeliegenden Netzmodelle bzw. Problemsysteme überaus rasch anwachsen ("explodieren"). Aufgrund kombinatorischer Zusammenhänge läßt sich das Wachstum der Graphen, das jeweils in Abhängigkeit von der Größe der zugrundeliegenden Netzmodelle und Problemsysteme ausgedrückt wird, in der Regel nur durch eine exponentielle Funktion nach oben beschränken. Daher wird in plastischer Diktion von einer "kombinatorischen Explosion" der Graphengröße gesprochen.

Vgl. zur kombinatorischen Explosion von Problemgraphen, Zustandsräumen von problembeschreibenden Systemen u.ä. NEWELL (1965), S. 199; NEWELL (1969), S. 386; POHL (1977), S. 69; STALLMAN (1977), S. 135; LATOMBE (1979), S. 508; STALLMAN (1979), S. 37; BARR, A. (1981), S. 27, 58 u. 154; NEWELL (1982b), S. 57; GEVARTER (1982), S. 157; GRAHAM, N. (1983), S. 17 u. 87; BUNGERS (1984), S. 2; LAURENT (1984), S. 150; NIEVERGELT (1984), S. 240; ZELEWSKI (1986a), S. 246 u. 253; VALMARI (1988a), S. 105; DRESSLER (1989b), S. 13f. Im entscheidungstheoretischen Kontext wurde die kombinatorische Explosion von NP-harten und lösungsdefekten Optimierungsproblemen erörtert. Der spezielle Aspekt der kombinatorischen Explosion von Erreichbarkeitsgraphen wurde bereits erläutert.

58) Vgl. DÖRNER, D. (1979), S. 69; ZELEWSKI (1986a), S. 253; MÄDLER (1990), S. 85 u. 87.

Für die hier betrachtete Suche in Erreichbarkeitsgraphen gilt: Die beiden entgegengesetzt gerichteten Suchprozesse erforschen jeweils einen Teilgraphen des gemeinsam zugrundeliegenden Erreichbarkeitsgraphen. Der eine Teilgraph hat den Knoten der Ausgangsmarkierung als Wurzel. Die Wurzel des anderen Teilgraphen ist der Knoten derjenigen erwünschten Endmarkierung, von der aus der zweite Teilprozeß startet. Die Umfänge der beiden Teilgraphen sind zwar a priori nicht genau bekannt, weil ihre Ausdehnung erst durch den Ablauf der beiden Suchprozesse konkret festgelegt wird. Doch werden die beiden Teilgraphen - in einer groben Annäherung - jeweils deutlich weniger als die Hälfte des Erreichbarkeitsgraphen umfassen.

Dies läßt sich am leichtesten veranschaulichen, indem der Erreichbarkeitsgraph (1. Ordnung) zunächst in ein äquivalentes Erreichbarkeitsnetz aus der Klasse der Stelle/Transition-Netze transformiert wird. In diesem Erreichbarkeitsnetz wird der erste Suchprozeß dargestellt durch eine Marke, die sich zunächst auf derjenigen Stelle befindet, die den Ausgangsmarkierungsknoten aus dem Erreichbarkeitsgraphen repräsentiert. Der zweite Suchprozeß wird durch eine komplementäre Marke vertreten, die zunächst diejenige Stelle belegt, die den Endmarkierungsknoten aus dem Erreichbarkeitsgraphen abbildet. Damit der zweite Suchprozeß durch Rücksetzen von Schaltschritten ausgeführt werden kann, wird für jede ursprüngliche Transition des Erreichbarkeitsnetzes eine komplementäre Transition eingeführt. Die komplementäre Transition stimmt mit ihrer Referenztransition vollkommen überein bis auf die eine Ausnahme, daß alle ihre Kanten genau entgegengesetzt gerichtet sind. (Zusätzlich wird die Markenzahl aller Stellen des Erreichbarkeitsnetzes auf zwei Marken heraufgesetzt.) Dann werden der erste und der zweite Suchprozeß im Erreichbarkeitsnetz dadurch ausgeführt, daß nur die ursprünglichen bzw. nur die komplementären Transitionen geschaltet werden. Die beiden Suchprozesse treffen sich in der intendierten Weise genau dann, wenn ihre beiden prozeßrepräsentierenden Marken unter derselben Markierung erstmals dieselbe Stelle im Erreichbarkeitsnetz belegen. Diese Markierung wird als Zielmarkierung des Erreichbarkeitsnetzes bezeichnet.

Nun können für das Erreichbarkeitsnetz zwei Erreichbarkeitsgraphen (2. Ordnung) konstruiert werden. Im ersten Erreichbarkeitsgraphen werden nur die Schaltakte der ursprünglichen Transitionen, im zweiten nur die Schaltakte der komplementären Transitionen berücksichtigt. Der erste Erreichbarkeitsgraph spezifiziert den abstrakten Suchraum des ersten Suchprozesses, der zweite Erreichbarkeitsgraph den analogen Suchraum des zweiten Suchprozesses. Beide Erreichbarkeitsgraphen besitzen die gleiche erwünschte Endmarkierung: die Zielmarkierung des Erreichbarkeitsnetzes. Da diese Zielmarkierung beim Starten der beiden Suchprozesse noch nicht bekannt ist, liegen auch die beiden Erreichbarkeitsgraphen 2. Ordnung noch nicht fest. Statt dessen werden sie schritthaltend oder mit geringem zeitlichem Vorlauf zur Ausführung der beiden Suchprozesse nebenläufig konstruiert. Falls eine Zielmarkierung des Erreichbarkeitsnetzes gefunden wird, liegen beide Erreichbarkeitsgraphen höherer Ordnung fest. Zugleich haben sich die beiden entgegengesetzt gerichteten Suchprozesse getroffen. Ebenso wurde einer der gesuchten finalen Schaltwege im zugrundeliegenden Erreichbarkeitsgraphen 1. Ordnung gefunden, der sich aus den beiden zusammengetroffenen Suchprozessen rekonstruieren läßt.

Der wesentliche Effekt dieser Vorgehensweise liegt darin, daß die beiden komplementären Probleme, von den Wurzelknoten der beiden Erreichbarkeitsgraphen 2. Ordnung zu ihrer gemeinsamen erwünschten Endmarkierung zu gelangen, jeweils nur ungefähr die Hälfte des gemeinsam zugrundeliegenden Erreichbarkeitsnetzes betreffen. (Wenn die Einführung der komplementären Transitionen als eine Netzverdopplung betrachtet wird, involvieren die beiden Erreichbarkeitsgraphen 2. Ordnung jeweils nur etwa ein Viertel des Erreichbarkeitsnetzes. Beide Aspekte zusammen ergeben wiederum eine grobe Halbierung des Problemumfangs.) Eine Verdopplung des Umfangs auszuwertender Netze bedeutet wegen der oben angesprochenen kombinatorischen Explosion, daß die Größe der zugehörigen Erreichbarkeitsgraphen um weitaus mehr als den Faktor "2" ansteigt. Daher gilt für die ungefähre Halbierung des Erreichbarkeitsnetzes in der umgekehrten Richtung: Die Größe seiner Erreichbarkeitsgraphen 2. Ordnung schrumpft auf deutlich weniger als die Hälfte der Größe des ursprünglichen Erreichbarkeitsgraphen 1. Ordnung. Daher fällt die Aufgabe, in zwei Erreichbarkeitsgraphen 2. Ordnung je einen Suchprozeß auszuführen, aus denen sich ein finaler Schaltweg im zugrundeliegenden Erreichbarkeitsgraphen 1. Ordnung rekonstruieren läßt, weit weniger komplex aus als die alternative Aufgabe, diesen finalen Schaltweg im Erreichbarkeitsgraphen 1. Ordnung direkt zu finden. (Dabei wird von der Komplexität der Zusatzaufgabe, die beiden Suchprozesse im Erreichbarkeitsnetz so zu koordinieren, daß sie sich dort tatsächlich treffen, abgesehen.) Genau diese Komplexitätsreduktion wurde oben dadurch umschrieben, daß die kombinatorische Explosion von Erreichbarkeitsgraphen in umgekehrter Richtung ausgenutzt wurde.

59) Ein analoger Parallelisierungsgedanke bei der Zerlegung von (linearen) Plänen findet sich bei MÄDLER (1990), S. 88.

60) Daneben ist es möglich, selbst dann noch einen beschleunigenden Effekt zu erzielen, wenn nur genau ein finaler Schaltprozeß gesucht ist. Denn bei mehreren erwünschten Endmarkierungen kann die Zeitdauer, die bis zum Auffinden eines finalen Schaltprozesses verstreicht (Lösungsdauer), unter anderem auch davon abhängen, welche Endmarkierung für die bidirektionale Lösungssuche ausgewählt wird. Im Zeitpunkt der Endmarkierungsauswahl kann aber noch nicht bekannt sein, welche der alternativen Endmarkierungen am schnellsten zu einem finalen Schaltprozeß führen würde. Die Endmarkierungsauswahl muß daher zufallsgesteuert erfolgen. Wenn mehrere bidirektionale Suchprozesse nebenläufig ausgeführt werden können, lassen sich jedoch entsprechend viele Endmarkierungen auswählen. Die Wahrscheinlichkeit, daß sich unter den zufällig ausgewählten Endmarkierungen mindestens eine Endmarkierung befindet, für die eine jeweils vorgegebene - nicht notwendig minimale - Lösungsdauer nicht über-

schritten wird, fällt aufgrund einfacher kombinatorischer Überlegungen um so höher aus, je mehr Endmarkierungen ausgewählt werden. Daher wird die durchschnittliche Geschwindigkeit für das Auffinden von finalen Schaltprozessen ansteigen, wenn diese Suchaufgabe oftmals erfüllt wird und dabei für das Auffinden je eines finalen Schaltprozesses mehrere bidirektionale Suchprozesse nebenläufig abgewickelt werden.

61) Es würde hier zu weit führen, diese Suchmethoden und die zugrundeliegenden Konzepte der Künstlichen Intelligenz näher zu erläutern. Vgl. statt dessen die Ausführungen dazu bei POHL (1970), S. 194; HALL, P. (1971), S. 641ff.; MARTELLI (1975), S. 345ff.; SIMON, H. (1975), S. 248ff.; RAPHAEL (1976), S. 75ff.; ITZINGER (1976), S. 109ff.; WINSTON (1977), S. 99ff. u. 115ff.; BODEN (1977), S. 348ff.; HARALICK (1979), S. 356ff.; NILSSON, N. (1980a), S. 112ff.; EISINGER (1980), S. 400 u. 404ff.; BARR, A. (1981), S. 58ff., 74ff. u. 94ff.; BROWN, C.A. (1981), S. 588ff.; KANAL (1981), S. 569ff.; LENAT (1982), S. 192ff.; POSPELOV, G. (1984), S. 1; NIEMANN (1984), S. 18ff.; ZELEWSKI (1986a), S. 279, 285ff., 299 u. 300ff., insbesondere S. 292f. i.V.m. S. 305ff.

Als *pars pro toto* wird hier nur ein Ansatz grob skizziert, der in der KI-Forschung eine herausragende Rolle spielt. Er beruht darauf, den Knoten oder Kanten eines Problemgraphen Schätzwerte ihres Beitrags zur Problemlösung zuzuordnen. Diese Werte sollen Wissen über die Struktur des jeweils bearbeiteten Problems ausdrücken. Es ist beabsichtigt, daß die Schätzwerte an jedem Knoten des Problemgraphen die Lösungssuche in eine Richtung weiterlenken, die möglichst schnell eine Problemlösung auffinden läßt. Das bekannteste Beispiel für eine solche heuristische Suchmethode, die auf heuristischem Wissen über "problemangemessene" Schätzwerte beruht, stellt der A*-Algorithmus dar. Nähere Ausführungen zu Schätzwert-Suchmethoden, insbesondere zum A*-Algorithmus, finden sich bei HART, P. (1968), S. 100ff.; POHL (1970), S. 195ff.; KOHLER, W. (1976), S. 231ff.; POHL (1977), S. 59ff.; GASCHNIG (1977), S. 434ff.; MEIBNER (1978), S. 33ff. u. 39ff.; BANERJI (1980), S. 25ff.; NILSSON, N. (1980a), S. 72ff.; BARR, A. (1981), S. 60ff., insbesondere S. 64ff.; PEARL (1981), S. 554ff.; MERO (1981), S. 572ff.; DAVIS, R. (1982a), S. 8f. u. 85ff.; SIEKMANN (1982b), S. 27f. u. 35ff.; JOHNSON, D. (1982), S. 965ff.; VALTORTA (1983), S. 777ff.; PEARL (1983), S. 1ff.; GRAHAM, N. (1983), S. 58ff.; NIEMANN (1983), S. 40ff.; BUNGERS (1984), S. 12f.; MARCUS (1984), S. 1046f.; HORN, W. (1984), S. 33f.; ZELEWSKI (1986a), S. 307ff.; PASSINO (1988a), S. 628ff.

Allerdings bleiben die vorgenannten Quellen in der Regel Auskunft darüber schuldig, wie sie an problembezogene Schätzwerte gelangen wollen. Die Ermittlung dieser Werte setzt zumeist eine intensive Auseinandersetzung mit den speziellen Charakteristika der jeweils zu bearbeitenden Probleme voraus. Dies unterbleibt aber in den allgemeinen Beschreibungen der Schätzwert-Suchmethoden zumeist. Vgl. zu Vorbehalten dieser Art ZELEWSKI (1986a), S. 308f.; vgl. auch die kritische Auseinandersetzung mit den Anwendungsvoraussetzungen des A*-Algorithmus und ihren Folgen bei HERTZBERG (1985), S. 19ff.

Ein typisches und zugleich besonders interessantes Beispiel für die Schätzwertproblematik des A*-Algorithmus liefert der Beitrag von PASSINO und ANTSAKLIS in PASSINO (1988a), S. 626ff. Seine Bedeutung für diese Arbeit liegt darin, daß der A*-Algorithmus mit dem Petrinetz-Konzept kombiniert wird. Zunächst werden Probleme aus dem Bereich der KI-Forschung durch Netze zu modelliert, die eine spezielle Gestalt annehmen. Die Netzmodelle werden mit dem formalen Apparat der Theorie abstrakter Automaten formuliert (S. 627), auf dessen Eigenarten hier nicht weiter eingegangen wird. Eine "Kostenfunktion" bildet - von Feinheiten der automatentheoretischen Netzformulierung abgesehen - die Transitionen auf Schaltwerte ab. Für diese Netzmodelle wird ein Graph konstruiert, der mit den Erreichbarkeitsgraphen von Netzen übereinstimmt. Dabei wird jeweils immer nur eine Transition geschaltet, um von einer Referenz- eine Folgemarkierung zu erreichen. Die (Schalt-)Kanten des (Erreichbarkeits-)Graphen werden so mit Schätzwerten gewichtet, daß sich mit Ihrer Hilfe der A*-Algorithmus anwenden läßt. Als typisch erweist sich jedoch, daß keine konkrete Ermittlungsvorschrift für diese Schätzwerte angegeben wird. Vielmehr werden alle nicht-negativen Schätzwerte zugelassen, sofern sie für alle Schaltkanten des Erreichbarkeitsgraphen folgende Restriktion erfüllen: Der Schätzwert einer Schaltkante, die das Schalten einer Transition ausdrückt, darf höchstens so groß sein, wie der Schaltwert dieser Transition ist. Diese Eingrenzung läßt aber einen breiten Spielraum für zulässige Schätzwerte offen. Der Beitrag von PASSINO und ANTSAKLIS gewährt *keine allgemeine* Auskunft darüber, wie sich aus dem Wissen über das jeweils untersuchte Problem Schätzwerte ableiten lassen. Er präsentiert zwar auf S. 630f. drei Beispiele, in denen jeweils konkret bestimmte Schätzwerte angewendet werden. Diese Schätzwerte werden aber ad hoc eingeführt. Lediglich Plausibilitätsargumente werden angeboten, um die Schätzungen nachträglich zu untermauern. Diese Plausibilitätsargumente sind jedoch nicht in der Lage, alternative Schätzwertfestlegungen schlüssig auszugrenzen. Darüber hinaus bleibt die Schätzwertfestlegung in mindestens einem Fall - dem dritten Beispiel - äußerst unbefriedigend. Denn dort wird *jeder* Schaltkante im Erreichbarkeitsgraphen *derselbe* Schätzwert "1" zugewiesen. Diese Gleichgewichtung bedeutet aber, daß überhaupt kein heuristisches Wissen vorliegt, um in einem Knoten des Erreichbarkeitsgraphen entscheiden zu können, welche Richtung am erfolgversprechendsten ist, um die Suche nach einem erwünschten Endknoten im Erreichbarkeitsgraphen fortzusetzen. Durch solche Einheitsschätzwerte werden zwar die formalen Anwendungsvoraussetzungen des A*-Algorithmus erfüllt. Doch sie zeigen zugleich auch, daß ad hoc-Festlegungen der Schätzwerte - zumindest in dem vorgelegten Beispiel - den intendierten Problembezug vermissen lassen. Sie drücken kein heuristisches Wissen über erfolgversprechende Suchrichtungen in Erreichbarkeitsgraphen aus. Damit wird aber der originäre Anspruch von wissensbasierten heuristischen Suchmethoden im Kern verfehlt. In ihren beiden ersten Beispielen deuten PASSINO und ANTSAKLIS zwar an, wie das erforderliche Wissen über erfolgverheißende Suchheuristiken in die Schätzwerte einfließen könnte. Doch bleiben ihre Ausführungen diesbezüglich leider unvollständig - die konkreten Schätzwerte werden nicht angegeben - und unklar; vgl.

PASSINO (1988a), S. 630. Daher vermag auch dieser Beitrag die Aufgabe, für den A*-Algorithmus problemspezifische Schätzwerte zu ermitteln, nicht befriedigend zu erfüllen.

62) Heuristisches Wissen wird hier in einer groben Annäherung verstanden als Erfahrungswissen darüber, welche Vorgehensweisen oder Mittel sich in der Vergangenheit zur Erreichung bestimmter Ziele in bestimmten Situationen bewährt haben. Einzelne Komponenten werden auch als Heuristiken oder - sofern sie in Regelform ausgedrückt sind - als heuristische Regeln bezeichnet. Nähere Erläuterungen der Eigenarten von heuristischem Wissen finden sich bei POLYA (1949), S. 118f. u. 155ff.; STREIM (1975), S. 143ff.; MEIBNER (1978), S. 2 u. 9ff.; BARR, A. (1981), S. 28ff.; ZELEWSKI (1986a), S. 228f. u. 285ff.

Der Verf. vermeidet hier bewußt den Heuristikbegriff, um ihn für die Heuristik als umfassende Lehre über Konstruktion und Anwendung von Heuristiken vorzubehalten. Der Begriff heuristischer Regeln wird nur dann benutzt, wenn die regelartige Darstellung eines Heurismus angesprochen werden soll.

63) Vgl. zu dieser Zielsetzung, die Effizienz heuristischer Suchmethoden durch Wissen über die jeweils zu bearbeitenden Probleme zu steigern, FEIGENBAUM (1963), S. 6; POHL (1970), S. 194; SACERDOTI (1974), S. 116; ITZINGER (1976), S. 107; KOHLER, W. (1976), S. 229f.; DILGER (1979), S. 267; SIEKMANN (1982b), S. 23; GRAHAM, N. (1983), S. 17f.; LANGLEY (1984), S. 310f.; ZELEWSKI (1986a), S. 299 u. 305.

64) Dabei wird vorausgesetzt, daß jeweils mehr alternative Untersuchungsrichtungen offenstehen, als mit den vorhandenen Informationsverarbeitungsressourcen zugleich bewältigt werden können. Dies stellt wegen der kombinatorischen Explosion von Erreichbarkeitsgraphen den Normalfall dar. Andernfalls braucht keine Auswahl zwischen konkurrierenden Untersuchungsrichtungen zu erfolgen. Vgl. zum letzten Fall die Anmerkungen zu Automatischen Informationsverarbeitungssystemen mit massiv parallelen Architekturen.

65) Ein Markierungsknoten gilt als vollständig untersucht, wenn dort alle potentiellen Schaltschritte hinsichtlich ihrer Aktivierung geprüft, alle aktivierten Schaltschritte ausgeführt und die Knoten aller dadurch bewirkten Folgemarkierungen ermittelt worden sind.

66) Vgl. zum Heurismus der Breitensuche WINSTON (1977), S. 91f.; MEIBNER (1978), S. 36; NILSSON, N. (1980a), S. 69 u. 71; BARR, A. (1981), S. 46ff.; GRAHAM, N. (1983), S. 49ff.; HORN, W. (1984), S. 29ff.; BUNGERS (1984), S. 7f.; ZELEWSKI (1986a), S. 297f.; ZELEWSKI (1988b), S. 358 u. 360.

67) Vgl. zum Heurismus der Tiefensuche WINSTON (1977), S. 90f.; MEIBNER (1978), S. 36; NILSSON, N. (1980a), S. 68ff.; BARR, A. (1981), S. 49f.; GRAHAM, N. (1983), S. 53ff.; HORN, W. (1984), S. 31f.; BUNGERS (1984), S. 8ff.; ZELEWSKI (1986a), S. 298; ZELEWSKI (1988b), S. 358 u. 360.

68) Eine solche Auswahl entfällt, wenn entweder nur noch genau ein Knoten existiert, der bisher nicht vollständig untersucht wurde, oder wenn bereits alle Knoten vollständig abgearbeitet sind.

69) Vgl. die wissensbasierten "Metastrategien" bei ZELEWSKI (1986c), S. 29ff.; ZELEWSKI (1988b), S. 359f.

70) Es betrifft die Lösung von linear-ganzzahligen Ungleichungssystemen, die als äquivalente Netzmodelle repräsentiert werden.

71) Ein weiterer Ansatz, Wissen über ein Analyseproblem für Petrinetze in eine heuristische Suchmethode für das Auffinden von Problemlösungen zu integrieren, findet sich bei VALMARI (1988a), S. 106.

Darüber hinaus kann auf die Methode der satisfizierenden Suche verwiesen werden, die SIMON und KADANE vorgestellt haben (vgl. SIMON, H. (1975), S. 249ff., insbesondere S. 259f.). Sie setzt gerichtete Graphen voraus, in denen eine Teilmenge derjenigen Knoten, die keine Nachfolger besitzen, als Zielknoten ausgezeichnet sind. Alle Zielknoten werden als gleich wünschenswert behandelt. Es wird ein Heurismus vorgestellt, der den Erwartungswert des Suchaufwands für das Erreichen eines ersten (beliebigen) Zielknotens minimiert. Die Zielknoten lassen sich mit den Knoten eines Erreichbarkeitsgraphen identifizieren, die erwünschte Endmarkierungen eines zugrundeliegenden Netzmodells repräsentieren. Daher kann die satisfizierende Suchmethode verwendet werden, um die Finalität eines Netzmodells zu überprüfen. Zu diesem Zweck braucht lediglich beobachtet zu werden, ob sich die Methodenanwendung mit dem Erreichen eines ersten Zielknotens erfolgreich beenden läßt. Zugleich handelt es sich um eine konstruktive Überprüfungsmethode: Falls ihre Anwendung erfolgreich endet, liegt zugleich ein finaler Schaltprozeß vor, der von der Ausgangsmarkierung des Netzmodells zu einer seiner erwünschten Endmarkierungen führt. Daher läßt sich die satisfizierende Suchmethode ebenso einsetzen, um für ein Netzmodell finale Schaltprozesse konkret zu bestimmen.

Allerdings wird in SIMON, H. (1975), S. 237f. u. 259f., die Kenntnis von Wahrscheinlichkeitsverteilungen vorausgesetzt, welche die Suche im o.a. gerichteten Graphen leiten. Die involvierten Wahrscheinlichkeiten sind aber bei der Suche in Erreichbarkeitsgraphen im allgemeinen nicht bekannt. Dies räumt SIMON, H. (1975), S. 260, sogar ausdrücklich ein. SIMON und KADANE vertrösten in diesem Fall lediglich auf eine "heuristische Schätzung". Wie diese Schätzung verwirklicht werden soll, lassen die beiden Autoren jedoch vollkommen unbestimmt. Daher gelten hier im Prinzip die gleichen Vorbehalte, die schon gegenüber dem A*-Algorithmus geäußert wurden. Dies unterstreichen die knotenbezogenen Schätzwerte der satisfizierenden Suchmethode, die bei SIMON, H. (1975), S. 241 u. 259f., ana-

log zu den Schätzwerten des A*-Algorithmus verwendet werden. Deshalb steht der Verf. der o.a. "Minimierung" des erwarteten Suchaufwands skeptisch gegenüber. Dies beeinträchtigt leider erheblich den zuvor herausgestellten konstruktiven Aspekt, finale Schaltprozesse auf heuristische Weise zu bestimmen. Denn es wird die Verfügbarkeit von Schätzwerten schlicht vorausgesetzt. Die entscheidende Frage, wie sich diese Informationsprämisse erfüllen läßt, wird jedoch nicht mehr konstruktiv beantwortet.

72) Konzepte für die nebenläufige Suche in (Problem-)Graphen finden sich z.B. bei IMAI (1979), S. 416ff.; NAKAGAWA (1979), S. 628ff.; KANAL (1981), S. 570f.; GEORGEFF (1981), S. 566.

73) Einen Ansatz in dieser Richtung hat KORNFELD (1981a), S. 578, vorgestellt. Er betrachtet einen nondeterministischen Algorithmus für die nebenläufige Lösungssuche in Problemgraphen, für dessen Implementierung nur eine kleine Anzahl von unabhängig arbeitenden Prozessoren zur Verfügung steht. Alternative Suchrichtungen im Problemgraphen werden auf der Grundlage von heuristischem Erfahrungswissen Wahrscheinlichkeiten für das erfolgreiche Auffinden einer gesuchten Problemlösung zugeordnet. Statt dessen können die Wahrscheinlichkeiten auch ein Maß dafür darstellen, daß die jeweils zugehörigen Suchrichtungen besonders rasch als unfruchtbar erkannt und entsprechend abgebrochen werden können. Sobald die Anzahl möglicher Richtungen, in denen die Lösungssuche aufgrund der inhärenten Nebenläufigkeit des untersuchten Problemgraphen fortgesetzt werden könnte, die beschränkte Prozessoranzahl übersteigt, erfolgt eine wissensbasierte Richtungsauswahl. Die Prozessoren werden auf jene Suchrichtungen angesetzt, welche die höchsten der o.a. Wahrscheinlichkeiten aufweisen.

74) Vgl. zu begründungsverwaltenden Systemen (truth/reason maintenance systems) DOYLE (1977a); DOYLE (1977b), S. 247; DOYLE (1979a), S. 232ff.; DOYLE (1979b), S. 117ff.; MCDERMOTT (1979), S. 564; MCDERMOTT (1980), S. 47 u. 65ff.; GOODWIN (1982), S. 2ff.; DOYLE (1983), S. 349ff.; DE KLEER (1984), S. 80f.; REINFRANK (1985b), S. 59ff., insbesondere S. 63ff.; DOYLE (1985), S. 87ff.; REINFRANK (1986), S. 11ff.; DE KLEER (1986a), S. 129ff.; DE KLEER (1986c), S. 216ff.; HUBER, A. (1986), S. 97; WINSLETT (1986), S. 421; ZELEWSKI (1986a), S. 362ff. (dort als nondeduktives Schließen); VAN MARCKE (1987), S. 263ff.; NARDI (1987), S. 563ff.; BROWN, A. (1987), S. 973ff.; PUPPE (1987), S. 100ff.; REINFRANK (1988a), S. 2 u. 7ff., insbesondere S. 9ff.; BECKSTEIN (1988a), S. 153ff.; MORRIS, P. (1988), S. 384 u. 388ff.; DRESSLER (1988a), S. 64ff.; FREITAG (1988a), S. 2ff.; FIDELAK (1988b), S. 28ff.; ZELEWSKI (1988c), S. 55f.; PETRIE (1989), S. 54ff.; DRESSLER (1989b), S. 13ff., insbesondere S. 17ff.; REINFRANK (1989b), S. 20; SCHNUPP (1989), S. 68; vgl. auch die weiteren Beiträge in dem Sammelwerk STOYAN (1988a).

Der Begriff der begründungsverwaltenden Systeme lehnt sich an die nachfolgend erläuterten Begründungen (justifications) an, die in den vorgenannten "truth maintenance systems" eine herausragende Rolle spielen. Von "Begründungsverwaltung" spricht auch explizit STOYAN (1988a), Titel und Vorwort. Des weiteren erinnern Formulierungen daran, wie sie sich bei DOYLE (1979a), S. 232 ("The TMS ... maintains 'proofs' of ... beliefs.", und DOYLE (1979b), S. 121 ("... a truth maintenance system ... maintains ... deductions."; Unterstreichungen im Original hier unterlassen), finden. Der sonst dominierende Begriff "truth" erscheint dem Verf. nicht angemessen, weil logische Wahrheitswerte nicht im Zentrum des Interesses stehen. Gleicher Ansicht ist z.B. GOODWIN (1982), S. 2. Inhaltlich treffender mutet die Diktion "reason maintenance" an; vgl. REINFRANK (1985b), S. 59 u. 61; STOYAN (1988a), Vorwort; REINFRANK (1988a), S. 1f.; DRESSLER (1988a), S. 64; FREITAG (1988a), S. 2; BECKSTEIN (1988a), S. 153; PETRIE (1989), S. 54f. Weitere Begriffsvarianten finden sich bei GOODWIN (1982), S. 2 ("Tenebility Maintenance System" und "dependency net update"); FIDELAK (1988b), S. 28 (Abhängigkeitsverwaltung).

75) Bei solchen Annahmen (assumptions) kann es sich vor allem um Voreinstellungen (default values) handeln, die dem Hintergrundwissen des Modellierungsträgers entstammen. Sie stellen Plausibilitätsannahmen über die Ausprägung eines Sachverhalts dar. Sie werden so lange aufrechterhalten, wie keine widersprechende Informationen über die tatsächliche Sachverhaltsausprägung zur Verfügung stehen. Die Beschäftigung mit solchen Voreinstellungen nimmt in der KI-Forschung einen breiten Raum ein; vgl. zu diesem "default reasoning" z.B. REITER (1978a), S. 210ff.; DOYLE (1979a), S. 234f.; DOYLE (1979b), S. 124ff.; REITER (1980), S. 82ff.; MCDERMOTT (1980), S. 44f. (normic statements); REITER (1981), S. 270ff.; CHRISTALLER (1982), S. 63f.; COHEN, P. (1982), S. 115ff.; HABEL (1983), S. 132ff.; WEBBER (1983), S. 43f.; MOORE, R. (1985), S. 77ff.; REINFRANK (1985b), S. 29, 33 u. 38ff.; LUKASZEWICZ (1985), S. 403ff.; DOYLE (1985), S. 87; DE KLEER (1986b), S. 166ff. u. 188f.; ZELEWSKI (1986a), S. 380f.; POOLE, D. (1988), S. 30ff.; MORRIS, P. (1988), S. 384ff.; KONOLIGE (1988), S. 343ff.; DELGRANDE (1988), S. 63ff.; BECKSTEIN (1988a), S. 160 u. 169f.; BREWKA (1989), S. 88ff., insbesondere S. 90f., 93 u. 95; DRESSLER (1989b), S. 17f.

Vgl. auch - außerhalb der KI-Forschung - die analogen Ausführungen von FRESE (1991), S. 71, über "vereinfachende Annahmen", die zum Ausfüllen von Wissensdefiziten getroffen werden.

76) Der Begründungsbegriff wird terminologisch inkohärent verwendet. Einmal werden nur die Antezedensformeln, aus denen eine Konklusionsformel abgeleitet wurde, als eine Begründung (justification) jener Konklusionsformel behandelt; vgl. DE KLEER (1986a), S. 146 u. 148; NARDI (1987), S. 566. Ein anderes Mal wird die Schlußfolgerung (Inferenz), in der eine Konklusions- aus mindestens einer Antezedensformel abgeleitet wird, in ihrer Gesamtheit als Begründung angesprochen; vgl. DE KLEER (1986b), S. 164 u. 167f.; HAAG (1988), S. 115ff.; BECKSTEIN (1988a), S.

154. Der Verf. hält sich hier an die erste Begriffsvariante, die vor allem der Knotenstruktur des unten angeführten ATMS-Konzepts zugrundeliegt.

77) Dies wird besonders ausführlich von LONDON (1978), S. 6ff., 78ff. u. 179ff.; GOODWIN (1982), S. 1ff.; REINFRANK (1985b), S. 63ff.; REINFRANK (1988a), S. 8ff., dargestellt. Zumeist wird der Abhängigkeitsgraph als ein "dependency net(work)" oder ein "Begründungsnetz" bezeichnet. Diese Begriffe werden hier jedoch vermieden, da der Abhängigkeitsgraph keine Beziehungen zum Petrinetz-Konzept aufweist. Der Abhängigkeitsgraph ist auch keine obligatorische Komponente von begründungsverwaltenden Systemen. Es wird aber hier auf seiner Grundlage argumentiert, weil er besonders anschaulich ist.

78) Das Kernkonzept dieses retrograden Erschließens von Annahmen, die zurückgenommen werden müssen, bildet das problemsensitive Backtracking (dependency-directed backtracking); vgl. STALLMAN (1977), S. 135ff.; STALLMAN (1979), S. 37ff. u. 58ff.; DOYLE (1979a), S. 233ff., insbesondere S. 235ff.; DOYLE (1979b), S. 124f. u. 127ff.; SACERDOTI (1979), S. 1080; MCDERMOTT (1980), S. 47 u. 67f.; DOYLE (1980), S. 9f.; DE KLEER (1984), S. 80; REINFRANK (1985b), S. 60f., 71ff., 90 u. 92f.; REINFRANK (1986), S. 14ff.; DE KLEER (1986a), S. 136f.; DE KLEER (1986c), S. 205; DE KLEER (1986d), S. 913ff.; ZELEWSKI (1986a), S. 362; STEEL (1987), S. 207ff., insbesondere S. 208f.; DRESSLER (1988a), S. 66f.; BECKSTEIN (1988a), S. 159; MORRIS, P. (1988), S. 397; REINFRANK (1988a), S. 16ff.; DRESSLER (1989b), S. 16; PETRIE (1989), S. 56.

Das Konzept des problemsensitiven Backtracking unterscheidet sich deutlich von dem konventionellen Backtracking-Konzept, das in einer früheren Anmerkung als chronologisches Backtracking angesprochen wurde. Beide Backtrackingvarianten setzen voraus, daß in einem Problemgraphen nach Lösungen für ein Planungsproblem gesucht wird. Bei einer solchen Lösungssuche kann der Fall eintreten, daß der Suchprozeß in einem Knoten - z.B. infolge Inkonsistenzerkennnis - abgebrochen werden muß. Dann wird seitens des chronologischen Backtrackings zu demjenigen Knoten zurückgeschritten, der dem Abbruchknoten am nächsten liegt und noch nicht vollständig untersucht worden ist. Dieses Zurückschreiten verhält sich in dem Sinne "blind", daß die Ursache für den Abbruch der Lösungssuche keine Berücksichtigung findet. Das problemsensitive Backtracking kehrt dagegen nicht notwendig zu diesem nächstliegenden Knoten zurück. Statt dessen schreitet es zu jenem Knoten zurück, der den inkonsistenzbedingten Abbruch der Lösungssuche verursacht hat. Dieser Knoten wird mit der Hilfe eines Abhängigkeitsgraphen ermittelt, in dem alle Ableitungsschritte für die Lösungssuche im Problemgraphen festgehalten sind. Dem Rückwärtsschreiten im Problemgraphen liegt daher eine analoge Rückwärtsbewegung im Abhängigkeitsgraphen zugrunde. Allerdings erfolgt das "intelligente" Rückwärtsschreiten in Problem- und Abhängigkeitsgraphen nur bei "einfachen" begründungsverwaltenden Systemen. Verfeinerte Varianten - wie das nachfolgend vorgestellte ATMS-Konzept - kommen ohne eine solche Retroduktion aus.

Ähnliche Backtracking-Varianten, die gegenüber dem "blinden" chronologischen Backtracking ebenso versuchen, beim Rückwärtsschreiten in "intelligenter" Weise Wissen über das jeweils bearbeitete Problem anzuwenden, finden sich bei FAHLMAN (1974), S. 28ff., insbesondere S. 31f.; DILGER (1983), S. 189ff.; ZELEWSKI (1986a), S. 355ff.

79) Die alte vorläufige Problemlösung wird nur so weit modifiziert, wie es nötig ist, damit sich die neue vorläufige Problemlösung mit allen aktuell verfügbaren Informationen - und den nicht zurückgenommenen Annahmen - konsistent vereinbaren läßt. Es wird darauf verzichtet, mit "roher Gewalt" (brute force) vorzugehen. Dies wäre bei konventioneller Problemlösungstechnik der Fall, die erfordert, einen Problemlösungsprozeß vollständig zu wiederholen, sobald sich die problemdefinierende Informationslage so weit verändert hat, daß die ursprüngliche Problemlösung nicht mehr zutrifft. Begründungsverwaltende Systeme erlauben daher eine intelligente Lösungsanpassung anstelle einer rohen Wiederholung der Lösungsermittlung. Dies wird besonders klar von REINFRANK (1985b), S. 58f. (in umfassenderem Kontext), dargelegt. Allerdings muß auch der Ressourceneinsatz in Rechnung gestellt werden, der für die Begründungsverwaltung selbst erforderlich ist und beträchtlichen Umfang annehmen kann; vgl. DE KLEER (1984), S. 79 u. 81; REINFRANK (1985b), S. 73 u. 86.

Am Rande sei vermerkt, daß begründungsverwaltende Systeme einen interessanten Ansatz darstellen, um betriebswirtschaftliche Planungstechniken zu bereichern. Denn sie stellen konkrete Instrumente zur Verfügung, mit deren Hilfe sich bei Veränderung der Informationsgrundlagen - z.B. über die jeweils vorliegende Produktionssituation - echte Anpassungsplanungen vollziehen lassen. Insbesondere erlaubt die Begründungsverwaltung, nur genau jene früheren Planungsschritte zu überprüfen (und gegebenenfalls zu modifizieren), deren ursprüngliche Begründungen wegen der zwischenzeitlich eingetretenen Informationsveränderung nachträglich ungültig geworden sind. Daher erfolgt eine zielgerichtete Beschränkung der Anpassungsplanung auf die jeweils relevanten, von Situationsveränderungen potentiell betroffenen Planungsaspekte. Aufgrund ihrer derart "ökonomischen" Vorgehensweise werden begründungsverwaltende Systeme insbesondere für Anpassungsplanungen *unter Realzeitbedingungen* in Erwägung gezogen; vgl. FIDELAK (1988b), S. 28. Dies ist insofern bedeutsam, als Konzepte für *systematische* Plananpassungen ansonsten zumeist nur in rudimentären, nicht operational ausformulierten Ansätzen vorliegen. Darauf wurde schon in der einleitenden Rahmenlegung hingewiesen. Auf Anpassungsplanungen mit Hilfe begründungsverwaltender Systeme wird hier jedoch nicht detaillierter eingegangen. Denn einerseits liegt dies außerhalb des Erkenntnisinteresses der hier vorgelegten Ausführungen, die nicht auf konkrete Problemlösungen abzielen. Andererseits handelt es sich um ein spezifisches Planungskonzept aus der Erforschung Künstlicher Intelligenz, das nicht mehr zum Umkreis des hier interessierenden Petrinetz-Konzepts gerechnet werden kann. Vgl. statt dessen die Hinweise, die im Zusam-

menhang mit Robustheitsanalysen skizzierten, wie sich im Rahmen des Petrinetz-Konzepts Anpassungsplanungen verwirklichen lassen. An anderer Stelle hat sich der Verf. ausführlicher mit Anpassungsplanungen befaßt, die speziell auf das Konzept begründungsverwaltender Systeme zurückgreifen; vgl. Zelewski (1991d), S. 1ff., insbesondere S. 8ff.

80) Dies gilt zwar nur so lange, wie immer noch Informationslücken bestehen und durch entsprechende Annahmen geschlossen werden müssen. Doch stellt dies den Normalfall dar. Von begründungsverwaltenden Systemen wird im allgemeinen unterstellt, ein Zustand vollständiger Information werde niemals erreicht.

81) Daher leisten begründungsverwaltende Systeme nicht nur eine Anpassung vorläufiger Problemlösungen, sondern ebenso eine Anpassung der Annahmen an den jeweils veränderten Informationsstand. Daher gehören begründungsverwaltende Systeme zu einem umfassenderen Konzept der KI-Forschung, das sich mit dem informationsbedingten Revidieren von Annahmen oder Überzeugungen befaßt (belief revision). Vgl. zu dieser Einordnung REINFRANK (1985b), S. 57ff.; REINFRANK (1988a), S. 2(ff.); FREITAG (1988a), S. 2.

82) ATMS ist ein Akronym für: Automated Truth Maintenance System. Das ATMS-Konzept wurde maßgeblich von DE KLEER entwickelt. Es wird näher beschrieben bei DE KLEER (1984), S. 81ff.; REINFRANK (1985b), S. 86ff.; DE KLEER (1986a), S. 128ff., insbesondere S. 141ff.; DE KLEER (1986b), S. 163ff.; DE KLEER (1986c), S. 197ff.; DE KLEER (1986d), S. 910ff.; MORRIS,P. (1986), S. 13ff.; STRUB (1986), S. 19ff.; DRESSLER (1987), S. 185ff.; NARDI (1987), S. 564ff.; PUPPE (1987), S. 100f.; HAAG (1988), S. 114ff.; BECKSTEIN (1988a), S. 162ff.; REINFRANK (1988a), S. 15 u. 17f.; DRESSLER (1988a), S. 68ff.; ARLABOSSE (1988), S. 8ff.; FIDELAK (1988b), S. 28ff.; PETRIE (1989), S. 54ff.; DRESSLER (1989a), S. 17ff.; DRESSLER (1989b), S. 17; WEDEKIND (1989c), S. 25f.; ZELEWSKI (1990a), S. 62.

83) Vgl. zur inhärenten Nebenläufigkeit des ATMS-Konzepts DE KLEER (1986c), S. 222; BECKSTEIN (1988a), S. 163.

84) Auf die modallogische KRIPKE-Semantik multipler Welten wurde bereits hingewiesen. Vgl. zur Charakteristik des ATMS-Konzepts, auf das Instrument multipler Welten zurückzugreifen, MORRIS,P. (1986), S. 13ff.; NARDI (1987), S. 563ff.; ARLABOSSE (1988), S. 9ff.; BECKSTEIN (1988a), S. 161.

Bemerkenswert ist, daß für die Programmiersprache PROLOG, die in dieser Arbeit der Implementierung Synthetischer Netze zugrundegelegt wurde, bereits ein Dialekt (MULTILOG) vorliegt, der die Verwaltung multipler Welten ermöglicht; vgl. KAUFFMANN (1987), S. 235ff.

85) Ein derartiges Zurückschreiten würde dem Backtracking-Konzept entsprechen. Da dies beim ATMS-Konzept entfällt, wird auch von einer Backtracking-freien Untersuchungsstrategie gesprochen; vgl. DE KLEER (1984), S. 79 u. 82; REINFRANK (1985b), S. 87; PETRIE (1989), S. 57.

Allerdings wurde das ATMS-Konzept so fortentwickelt, daß ihm nachträglich doch noch eine Backtracking-Komponente überlagert werden kann; vgl. DE KLEER (1986c), S. 205 u. 215; DE KLEER (1986d), S. 914ff.; DRESSLER (1989a), S. 19. Dies empfiehlt sich immer dann, wenn kein Interesse besteht, alle Problemlösungen zu erkennen. Denn die nebenläufige Bearbeitung multipler Welten durch das ATMS-Konzept entspricht der Breitensuche, bei der alle Problemlösungen erzeugt werden (sofern die Verarbeitungsressourcen ausreichen). Falls jedoch nur eine Problemlösung gesucht wird, würde dieses Gewinnen aller Problemlösungen unnötig Ressourcen verbrauchen. Daher wird die Option vorgehalten, die ursprünglich konzipierte Breitensuche so durch eine Backtracking-Komponente zu überlagern, daß sich die Problembearbeitung dem Charakter einer Tiefensuche annähert.

86) Daher können Schaltstrategien und zugehörige Schaltprozesse in einer ersten, groben Annäherung gleichgesetzt werden. Strenggenommen kann dieselbe Schaltstrategie aber durchaus unterschiedliche Schaltprozesse hervorbringen. Die ersten beiden Differenzierungsmöglichkeiten bestehen darin, dieselbe Schaltstrategie entweder auf verschiedene Ausgangsmarkierungen desselben Netzmodells oder aber auf verschiedene Netzmodelle anzuwenden. Darüber hinaus kann eine Schaltstrategie auch einen Zufallsgenerator umfassen, der dazu führt, daß unter mindestens einer Netzmarkierung aus den dort aktivierten Schaltschritten einer zufällig ausgewählt wird. Dann wird die mehrfache Anwendung derselben Schaltstrategie auf dieselbe Ausgangsmarkierung desselben Netzmodells in der Regel unterschiedliche, zufällig variierende Schaltprozesse bewirken.

87) Vgl. RAZOUK (1985c), S. 4 (indirekt, da nicht explizit auf Netze bezogen).

Es könnte über die Angemessenheit dieser Bezeichnungsweise gestritten werden. Denn es wird nicht - wie der Begriff der Netzsimulation nahezu legen scheint - das Netzmodell selbst simuliert. Statt dessen wird das Netzmodell benutzt, um das Verhalten des jeweils modellierten Objekts zu simulieren.

88) Da der Erreichbarkeitsgraph nicht untersucht wird, bleibt das Netzoptimum sw^* unbekannt. Daher ist es - innerhalb einer Netzsimulation - unmöglich festzustellen, wie nahe sie diesem Netzoptimum tatsächlich gekommen ist. Daher handelt es sich bei der Hoffnung, diesem Optimum möglichst nahe zu kommen, um eine Erwartungshaltung, deren Erfüllungsgrad aus der Netzsimulation selbst nicht ersichtlich ist.

89) Abbruchsgründe sind vor allem das Einmünden der Simulationsausführung in eine Endlosschleife (Livelock), in der die erwünschten Endmarkierungen nicht vorkommen, und das Absterben der Simulationsausführung in einer unerwünschten Endmarkierung (Deadlock). In beiden läßt sich keine der erwünschten Endmarkierungen erreichen.

90) Dabei wird vorausgesetzt, daß ein finales Netzmodell vorliegt, so daß in ihm mindestens ein finaler Schaltprozeß ausgeführt werden kann. Andernfalls würde keine von allen denkmöglichen Schaltstrategien zu einer der erwünschten Endmarkierungen führen.

91) Auf diese Weise wird von der konventionellen zwei- zu einer einstufigen Lösungssuche übergegangen. Vgl. zur Differenzierung zwischen diesen beiden Suchkonzepten ZELEWSKI (1986a), S. 241.

Bei einer zweistufigen Lösungssuche wird zunächst der Erreichbarkeitsgraph konstruiert. Erst danach wird im Lösungsraum, der durch den vollständig vorliegenden Erreichbarkeitsgraphen konstituiert wird, nach einer Problemlösung gesucht. Die einstufige Lösungssuche zeichnet sich dagegen dadurch aus, darauf zu verzichten, den Erreichbarkeitsgraphen von vornherein vollständig zu erzeugen. Statt dessen werden aus diesem potentiellen oder "virtuellen" Erreichbarkeitsgraphen nur Teilgraphen so weit realisiert, wie es schritthaltend zum aktuellen Fortschritt der Lösungssuche erforderlich ist. Die dabei sukzessiv anwachsenden Teilgraphen stellen jeweils Ausschnitte aus dem zu erforschenden Lösungsraum dar. Sie lassen sich auch als Suchbäume auffassen, die das Voranschreiten der aneinander gekoppelten Lösungsraumkonstruktion und Lösungssuche in graphischer Weise beschreiben (vgl. GRAHAM,N. (1983), S. 46ff.). Falls es gelingt, die gesuchte(n) Problemlösung(en) zu finden, bevor der gesamte Lösungsraum erzeugt worden ist, so lassen sich dadurch gegenüber der vollständigen Konstruktion von Erreichbarkeitsgraphen Ressourcen einsparen. Allerdings bereitet es im allgemeinen auch größere Schwierigkeiten, einen Lösungsraum zu konstruieren und zugleich auch nach Problemlösungen zu durchforschen.

Einstufige Konzepte, die vorsehen, einen Lösungsraum schritthaltend zur darin erfolgenden Lösungssuche zu konstruieren, finden sich z.B. bei POHL (1970), S. 193ff.; ITZINGER (1976), S. 99ff.; RAPHAEL (1976), S. 73ff.; WINSTON (1977), S. 89ff.; SIEKMANN (1982b), S. 22ff.; STEDE (1983), S. 79ff.; ZELEWSKI (1986a), S. 247f.

Vgl. zu entsprechenden Vorbehalten gegenüber der zweistufigen Methode, allgemeine Problemgraphen zunächst vollständig zu konstruieren, SACERDOTI (1974), S. 115; GRAHAM,N. (1983), S. 71; ZELEWSKI (1986a), S. 245ff.; mittelbar auch WITTE,T. (1979a), S. 39.

92) Vgl. zum Konstrukt der "nogoods" DOYLE (1979a), S. 236f.; DOYLE (1979b), S. 129f.; REINFRANK (1985b), S. 71f., 83, 88 u. 93; MORRIS,P. (1986), S. 14f.; REINFRANK (1986), S. 15; DE KLEER (1986a), S. 137 u. 147f.; DE KLEER (1986b), S. 168; HAAG (1988), S. 116 u. 123; BECKSTEIN (1988a), S. 158f.; REINFRANK (1988a), S. 17; DRESSLER (1988a), S. 72ff.; DRESSLER (1989b), S. 14ff.

93) Dies klingt auch bei DRESSLER (1989b), S. 17, an (allerdings ohne Bezug auf Netzanalysen).

94) Das wäre z.B. in folgender Konstellation der Fall: Eine Schaltstrategie wurde abgebrochen, weil sie in eine unerwünschte Endmarkierung - einen Deadlock - geführt hat. Dabei wurde im Rumpferreichbarkeitsgraphen irgendwann unter einem Markierungsknoten ein Schaltschritt so ausgeführt, daß danach keine Möglichkeit mehr besteht, den Knoten der Deadlockmarkierung zu entkommen. Im einfachsten Fall handelt es sich um einen Markierungsknoten, der im Rumpferreichbarkeitsgraphen dem Knoten der Deadlockmarkierung unmittelbar vorangeht, und um den Schaltschritt, der in diesem Markierungsknoten durch die abgebrochene Schaltstrategie ausgewählt wurde. Dann wird die Schaltkante, die mit diesem schaltstrategiespezifischen Schaltschritt beschriftet ist und im Knoten der Deadlockmarkierung endet, als unfruchtbare Schaltkante gekennzeichnet.

95) In der Konstellation, die in der voranstehenden Anmerkung skizziert wurde, handelt es sich - wiederum im einfachsten Fall - um eine Schaltkante, die in demjenigen Markierungsknoten endet, von dem die bereits eingeführte unfruchtbare Schaltkante ausgeht. Dann stellt auch die erstgenannte Schaltkante eine unfruchtbare Schaltkante dar.

96) Es werden diejenigen Untersuchungsrichtungen ausgeschlossen, die zwar keinen der gesuchten finalen Schaltprozesse auffinden lassen, aber durch ihren Ressourcenbedarf die Suche solcher Schaltprozesse - ceteris paribus - verlangsamen. Vgl. dazu - allerdings ohne Bezug auf das Petrinetz-Konzept - auch NARDI (1987), S. 564.

Allerdings muß auch beim ATMS-Konzept der zusätzliche Ressourcenaufwand für die Technik der Begründungsverwaltung in Rechnung gestellt werden. Sie kann durchaus erheblichen Aufwand bereiten. Vgl. dazu die Hinweise auf Effizienzprobleme bei DE KLEER (1986b), S. 190, 193 u. 195; DE KLEER (1986c), S. 215; DE KLEER (1986d), S. 912; HAAG (1988), S. 113, 118ff. (mit pessimistisch stimmenden Überschlagsbetrachtungen) u. 131; DRESSLER (1988a), S. 81.

Effizienz Nachteile können sich vor allem dann einstellen, wenn Probleme betrachtet werden, die viele potentielle Lösungen besitzen, von denen aber nur jeweils eine gesucht wird. Darauf wurde bereits eingegangen; vgl. auch DE KLEER (1986c), S. 205, 215 u. 222f.; DE KLEER (1986d), S. 910f. u. 913; DRESSLER (1989a), S. 17.

97) Es ließe sich darüber streiten, ob sich Intuition und Kreativität inhaltlich scharf voneinander abgrenzen lassen. Der Verf. verzichtet darauf, sich auf eine solche Debatte einzulassen. Im Zweifelsfall ist die Formulierung "Intuition und Kreativität" als eine pleonastische Verbeugung vor den menschlichen Informationsverarbeitungsfähigkeiten zu verstehen.

98) Da hier nur Optionen für die Erreichbarkeitsanalyse von Netzmodellen aufgezeigt werden, erfolgt keine nähere Überprüfung, ob sich die erweckten Hoffnungen tatsächlich erfüllen lassen. Zu diesem Zweck müßte die skizzierte interaktive Erreichbarkeitsanalyse zunächst implementiert und dann anhand von Testfällen praktisch erprobt werden. Solche Implementierungen und Testserien liegen jedoch außerhalb des Erkenntnisinteresses der hier vorgelegten Untersuchungen; vgl. dazu die einleitende Rahmenlegung. Statt dessen wird auf die Quellen verwiesen, die an späterer Stelle zugunsten interaktiver Informationsverarbeitungskonzepte angeführt werden.

99) Die voranstehend skizzierten Beiträge zur Netzsimulation können daher integriert werden. Der Übersichtlichkeit halber werden sie jedoch im folgenden nicht weiter berücksichtigt.

100) Dabei wird unterstellt, daß unter den aktuellen Netzmarkierungen jeweils mindestens eine aktivierte Transition existiert und vom Automatischen Informationsverarbeitungssystem auch als solche erkannt wird. Unter dieser Voraussetzung kann immer zumindest derjenige Schaltschritt ausgewählt werden, welcher die mindestens eine aktivierte Transition umfaßt. Andernfalls wird angenommen, daß das Automatische Informationsverarbeitungssystem das Vorliegen einer toten Netzmarkierung korrekt erkennt und den Modellbenutzer darüber informiert. Die Netzsimulation endet dann. Oder sie wird durch ein Zurücksetzen des bisher simulierten Schaltprozesses in anderer Richtung fortgesetzt. Um die Ausführungen übersichtlich zu halten, wird im folgenden auf das Erreichen einer toten Netzmarkierung nicht weiter eingegangen.

101) Grundkenntnisse im Umgang mit interaktiven graphischen Benutzeroberflächen von Automatischen Informationsverarbeitungssystemen werden im folgenden vorausgesetzt. Vgl. auch die späteren Ausführungen zur Benutzerfreundlichkeit des Petrinetz-Konzepts. Dort wird auch auf Übersichtsliteratur zur graphischen Informationsverarbeitung verwiesen.

102) In der Vergangenheit sind auch einige automatengestützte Entwicklungsumgebungen für Netzmodelle vorgestellt worden, bei denen die Aktivierung einer Transition durch das Graphiksymbol einer Kanone visualisiert wurde. Das Graphiksymbol leitet sich von der weit verbreiteten Bezeichnungsvariante ab, nicht vom Schalten, sondern vom "Feuern" einer aktivierten Transition zu sprechen. Angesichts dieser Redeweise ist die kognitive Adäquanz des Kanonensymbols offensichtlich. Allerdings sind sowohl das Reden vom "Feuern" einer Transition als auch die Verwendung der Kanonensymbole auf starke Kritik gestoßen. Ihnen wurde ein inakzeptabler militaristischer Einschlag vorgehalten. Daher werden heute Kanonen als Graphiksymbole für aktivierte Transitionen kaum noch benutzt.

103) Zu diesem Zweck kann er z.B. im oben erwähnten Informationsfenster die präferierte Schaltfarbe "ankreuzen".

104) Es wird auf frühere Vorbehalte verwiesen: Bei praktischen Tests kann sich durchaus herausstellen, daß die erhofften intuitiven und kreativen Leistungen des Modellbenutzers ausbleiben. Die hier skizzierte interaktive Erreichbarkeitsanalyse verfolgt lediglich die Absicht aufzuzeigen, wie ein Zugang zu - behaupteten oder tatsächlichen - Leistungspotentialen "typisch menschlicher" Informationsverarbeitung erschlossen werden kann. Eine Aussage darüber, ob diese Leistungspotentiale tatsächlich existieren und auch genutzt werden können, erfolgt dadurch jedoch nicht.

105) Eine ähnliche Filterfunktion des Benutzers interaktiver Modellierungskonzepte liegt den Ausführungen von STADTLER (1983a), S. 2f., 73ff. u. 180f., zugrunde. Dort zeigt STADTLER auf, daß es plausibel ist, bei schlechtstrukturierten Problemen auf die Filterfähigkeiten eines Modellbenutzers zu vertrauen. Im Kern laufen seine Erläuterungen auf ein "faute de mieux"-Argument hinaus. Unterstellt wird ein Problem, das so schlecht strukturiert ist, daß darauf verzichtet wurde, das Problem vollständig durch ein formales Modell zu repräsentieren. Statt dessen werden nur wohlstrukturierte Teilaspekte des Problems durch ein formales Modell repräsentiert und automaten-gestützt implementiert. Unter *diesen* Prämissen kann die vollautomatische Bearbeitung des unvollständigen Modells allenfalls zufällig zu korrekten Problemlösungen führen. Daher liegt es nahe, die automatisch generierten Lösungsvorschläge durch einen Modellbenutzer beurteilen zu lassen. Dabei wird zusätzlich angenommen, daß der Benutzer ein vertieftes Problemverständnis besitzt, das über das unvollständige formale Modell hinausreicht. Dann bestehen nur zwei Möglichkeiten: Entweder ist der Modellbenutzer in der Lage, aufgrund seines vertieften Problemverständnisses "gute" von "schlechten" Lösungsvorschlägen zu trennen. Dann besitzt er die oben unterstellte Filterfähigkeit tatsächlich. Oder der Modellbenutzer kann die Lösungsunterscheidung nicht vornehmen. In diesem Fall erfüllt er die Filterfunktion nicht. Aber es existiert auch kein Drittes, das diese Filterfunktion erfolgreich übernehmen könnte. Denn einerseits scheidet ein Automatisches Informationsverarbeitungssystem für diese Aufgabe wegen der vorausgesetzten Schlechtstrukturiertheit des Problems aus. Andererseits wäre es selbstwidersprüchlich, einen anderen Modellbenutzer heranziehen zu wollen. Auf ihn träfe die voranstehende Fallunterscheidung in derselben Weise zu. Daher wäre nicht einzusehen, warum der zweite Modellbenutzer die Filterfunktion erfüllen sollte, wenn eben dies für den ersten Modellbenutzer ausgeschlossen wurde. Daher bleibt nur das eingangs angekündigte "faute de mieux"-Argument: Mangels einer überzeugenden Alternative kann nur darauf gehofft werden, daß die Intuition des Modellbenutzers ausreicht, um "gute" von "schlechten" Lösungsvorschlägen zu unterscheiden. Falls die Hoffnung trügen sollte, lassen die vereinbarten Voraussetzungen keinen "heilenden" Ausweg erkennen.

Abschließend bleibt darauf hinzuweisen, daß die voranstehend skizzierte Argumentation auch auf die hier interessierende Erreichbarkeitsanalyse von Netzmodellen zutrifft: Die Schlechtstrukturiertheit der Analyseprobleme äußert sich im Lösungsdefekt der kombinatorisch explodierenden Erreichbarkeitsgraphen. Um diesen Lösungsdefekt zu umgehen, wird auf eine vollständige Konstruktion von Erreichbarkeitsgraphen verzichtet. Durch die Netzsimulation erfolgt also nur eine unvollständige Exploration der dynamischen Struktur des jeweils untersuchten Netzmodells. Dabei bearbeitet das interaktiv genutzte Automatische Informationsverarbeitungssystem zunächst die wohlstrukturierten Teilprobleme, unter jeder betrachteten Netzmarkierung die aktivierten Transitionen und die aktivierenden Schaltfarben festzustellen. Alle potentiellen Schaltschritte, die sich daraus kombinieren lassen, kommen als Lösungsvorschläge für die Ermittlung der nächsten Schaltschrittausführung in Betracht. Ob diese Lösungsvorschläge vom Automatischen Informationsverarbeitungssystem generiert oder vom Modellbenutzer selbst erzeugt werden, spielt hier keine Rolle. Darauf wird in Kürze näher eingegangen. An dieser Stelle ist lediglich entscheidend, daß der Modellbenutzer genau einen potentiellen Schaltschritt für die Fortsetzung der Erreichbarkeitsanalyse festlegt. Falls mehrere potentielle Schaltschritte existieren, setzt dies eine - zumindest implizite - Selektion durch den Modellbenutzer voraus. Genau an diesem Punkt greift das o.a. "faute de mieux"-Argument: Entweder ist der Modellbenutzer in der Lage, einen "guten" potentiellen Schaltschritt herauszufiltern. Oder er vermag dies nicht. Dann existieren aber auch keine glaubwürdigen Alternativen, um die Filterfunktion zu erfüllen.

106) Es wurde oben vorausgesetzt, daß nur solche Netzmarkierungen betrachtet werden, unter denen mindestens eine Transition aktiviert ist und als solche vom Automatischen Informationsverarbeitungssystem auch angezeigt wird. Daher ist der Benutzer immer in der Lage, zumindest einen Schaltschritt festzulegen, der aus genau einer aktivierten Transition - zusammen mit ihrer aktivierenden Schaltfarbe - besteht. Dieser unäre Schaltschritt ist stets zulässig. Falls der Modellbenutzer diese Option nicht sehen sollte, kann er vom Automatischen Informationsverarbeitungssystem darauf hingewiesen werden.

107) Als Interaktionsvariante kann dem Modellbenutzer die Option zugestanden werden, jederzeit die automatische Generierung von Schaltschrittorschlägen abzubrechen, um ohne weitere Automatenunterstützung einen Schaltschritt selbst festzulegen. Es wird dann zu derjenigen Interaktionsweise zurückgekehrt, die zuerst vorgestellt wurde. Es ist dann allerdings darauf zu achten, daß der festgelegte Schaltschritt nachträglich auf seine Zulässigkeit überprüft wird. Diese zusätzliche Option bietet sich insbesondere dann an, wenn der Modellbenutzer bereits eine größere Anzahl von Schaltschrittorschlägen als ungeeignet verworfen hat und an der Generatorqualität des Automatischen Informationsverarbeitungssystems zu zweifeln beginnt. In diesem Fall kann dem aufkeimenden Benutzerunmut durch das "Ventil" begegnet werden, einen potentiellen Schaltschritt automatenunabhängig festzulegen. Hierdurch wird dem Aspekt der Benutzerautonomie Rechnung getragen.

108) Die Finitheitsbedingungen, die im Verlaufe dieser Arbeit für Petrinetze - insbesondere für Synthetische Netze - eingeführt wurden, stellen sicher, daß unter jeder Netzmarkierung immer nur eine *endliche* Anzahl zulässiger Schaltschritte existieren kann. Daher ist es unmöglich, daß der Zyklus von Schaltschrittorschlägen durch den Automaten und korrespondierenden Schaltschritt ablehnungen durch den Benutzer unendlich lange andauert.

109) Zu diesem Zweck empfiehlt es sich, die generierten Schaltschrittorschläge für die aktuelle Netzmarkierung zwischenspeichern, auch wenn sie vom Benutzer zunächst verworfen werden. Sie werden erst dann gelöscht, wenn sich der Benutzer für eine der drei hier vorgestellten Optionen entschieden hat.

110) Diese Anforderung kann vom Automatischen Informationsverarbeitungssystem nur so lange erfüllt werden, wie sich aus den aktivierten Transitionen und ihren aktivierenden Schaltfarben mindestens K verschiedene zulässige Schaltschritte erzeugen lassen. Andernfalls wird dem Benutzer mitgeteilt, daß nur weniger als K verschiedene Schaltschritte zulässig sind. Sie werden dann in ihrer Gesamtheit angezeigt.

111) Die Angemessenheit dieser Hypothese wäre wieder durch entsprechende Testreihen empirisch zu überprüfen.

112) Es bieten sich abermals Beiträge aus der Erforschung Künstlicher Intelligenz an, um die Akquisition dieses Erfahrungswissens in automatisch auswertbarer Form zu unterstützen. Insbesondere läßt sich vorstellen, daß Benutzer von interaktiv ausgewerteten Netzmodellen von einem entsprechend ausgelegten Expertensystem beobachtet werden. Das Expertensystem analysiert das Benutzerverhalten nach Verhaltensregularitäten bei der eigenständigen Festlegung von Schaltschritten und bei der Auswahl von geeignet erscheinenden Schaltschritten aus automatisch generierten Schaltschrittorschlägen. Wenn solche Verhaltensregularitäten erkannt werden, stellen sie heuristisches Metawissen für die Ablaufsteuerung von Erreichbarkeitsanalysen dar. Dieses Metawissen läßt sich z.B. in der Gestalt von heuristischen Kompositionsregeln explizieren, die später gestatten, die Generierung von Schaltschrittorschlägen zu steuern. Es würde hier zu weit führen, die zugrundeliegenden KI-Konzepte im Detail zu erläutern. Der Verf. hat dies an anderer Stelle aus der Perspektive lernfähiger Automaten dargelegt; vgl. ZELEWSKI (1986a), S. 481ff., insbesondere S. 489ff. (Erlernen heuristischer Regeln).

113) Die nachfolgend aufgelisteten Aspekte gelten unabhängig davon, ob das Automatische Informationsverarbeitungssystem entweder nur genau einen oder aber ein Menü aus mehreren zulässigen Schaltschritten vorschlägt.

114) Zwar müssen statt dessen Analyseressourcen eingesetzt werden, um das Fehlen gemeinsamer Eingangs- und gemeinsamer Ausgangsstellen zu überprüfen. Aber diese alternative Untersuchung fällt im allgemeinen wesentlich einfacher aus als der Test auf konfliktionär aktivierte Transitionen. Darüber hinaus hängt die Überprüfung gemeinsamer Eingangs- und gemeinsamer Ausgangsstellen ausschließlich von der topologischen Netzstruktur ab. Da diese Netztopologie eine Invariante des Netzmodells darstellt, ist es möglich, für das Netzmodell genau einmal die Kombinationen aller Transitionen zu bestimmen, die keine Eingangs- und keine Ausgangsstellen teilen. Die Liste dieser Kombinationen wird dann dauerhaft vorgehalten. Der o.a. Heurismus kann sich darauf beschränken, diese Liste sukzessiv nach solchen Kombinationen zu durchsuchen, deren zugehörige Transitionen unter der aktuellen Netzmarkierung gemeinsam aktiviert sind. Die Liste umfaßt auch die "degenerierten" Kombinationen, die jeweils aus genau einer Transition bestehen. Es wurde vorausgesetzt, daß unter jeder betrachteten Netzmarkierung mindestens eine Transition aktiviert ist. Daher bringt der Heurismus zumindest einen zulässigen unären Schaltschritt hervor, der genau eine aktivierte Transition umfaßt.

115) Ebenso ist es möglich, die beiden voranstehenden Heurismen von vornherein nicht zu beachten. Dennoch können die nachfolgenden Heurismen auch dann angewendet werden. Zu diesem Zweck reicht es aus, anstelle der vorgeschlagenen - aber vom Benutzer nicht akzeptierten - Schaltschritte jeweils einen unären Schaltschritt zu setzen, dessen eine Transition unter der aktuellen Netzmarkierung aktiviert ist.

116) Nur bei positivem Prüfungsergebnis werden die Schaltschritte dem Modellbenutzer tatsächlich vorgeschlagen.

117) Heuristische Grundlage dieser minimalen Inkremente ist die Plausibilitätsüberlegung, daß die Gefahr von konfliktionär aktivierten Transitionen tendenziell um so größer ausfällt, je mehr Transitionen in einem Schaltschritt zusammengefaßt sind.

118) Andernfalls wird die erwogene Schaltschritterweiterung vom Automaten wegen Unzulässigkeit sofort wieder zurückgenommen und dem Benutzer überhaupt nicht vorgeschlagen. Darüber hinaus kann der Automat den erweiterten unzulässigen Schaltschritt speichern. Dabei muß er sowohl konfliktionär aktivierte Transitionen als auch deren aktivierende Schaltfarben erfassen. Dann ist es möglich, jede spätere - noch weiter reichende - Schaltschritterweiterung zu unterlassen, welche die konfliktionär aktivierten Transitionen aus dem unzulässigen alten Schaltschritt mit ihren aktivierenden Schaltfarben umfassen würde. Denn die konfliktionäre Aktivierung von Transitionen verhält sich in Petrinetzen monoton: Zu konfliktionär aktivierten Transitionen, deren aktivierende Schaltfarben nicht verändert werden, können beliebige andere Transitionen hinzugefügt werden. Dadurch wird die konfliktionäre Aktivierung der erstgenannten Transitionen niemals aufgehoben. Dies folgt unmittelbar aus der Definition der Schaltregel für die Transitionen eines Petrinetzes.

119) Es liegt die heuristische Annahme zugrunde, daß die Gefahr von Schaltkonflikten tendenziell um so größer ist, je häufiger Kombinationen aus Transitionen unter früheren Netzmarkierungen konfliktionär aktiviert waren. Dabei handelt es sich aber aus zwei Gründen um keine strenge Schlußfolgerung. Denn einerseits kann die aktuelle Netzmarkierung von den früheren Netzmarkierungen so stark abweichen, daß die Transitionen aus einer gefährlichen Transitionenkombination unter der aktuellen Markierung nebenläufig aktiviert sind. Andererseits werden von den gefährlichen Transitionenkombinationen die Schaltfarben der zugehörigen Transitionen nicht berücksichtigt. Daher kann es durchaus der Fall sein, daß sich ein Schaltkonflikt der betroffenen Transitionen durch eine Variation der aktivierenden Schaltfarben auflösen läßt. Dies ist selbst dann möglich, wenn die aktuelle Netzmarkierung mit einer derjenigen Markierungen zusammenfällt, unter denen früher - bei anderen Schaltfarben - ein Schaltkonflikt festgestellt wurde.

Die beiden Vorbehalte zeigen zugleich auf, wie der o.a. Heurismus verfeinert werden kann. Er läßt sich erstens um die Schaltfarben der konfliktionär aktivierten Transitionen erweitern. Zweitens können auch die Markierungen charakterisiert werden, unter denen die Schaltkonflikte früher registriert worden sind. Dabei ist es wegen der Lokalität des Petrinetz-Konzepts nicht erforderlich, die Markierungen insgesamt zu speichern. Vielmehr reicht es aus, für die konfliktionär aktivierten Transitionen jeweils die Markierungen ihrer inzidenten Stellen festzuhalten. Die beiden Erweiterungen des Heurismus bereiten allerdings erheblichen Aufwand für die Ermittlung und Speicherung des zusätzlich relevanten Erfahrungswissens. Ob sich dies durch die größere "heuristische Kraft" des verfeinerten Heurismus rechtfertigen läßt, kann letztlich nur durch entsprechende Tests untersucht werden.

120) Aus den Erläuterungen der voranstehenden Anmerkung folgt unmittelbar, daß ein Schaltschritt, der aus den Transitionen einer gefährlichen Transitionenkombination zusammengesetzt ist, durchaus zulässig sein kann.

121) Diese Erweiterung von Synthetischen Netzen wird später vorgenommen.

122) Hinzu kommt die triviale dritte Bedingung, daß die Transition noch nicht zum jeweils betrachteten Schaltschritt gehören darf.

123) An dieser Stelle kann der Erweiterungsheurismus verschärft werden: Die aktivierte Transition wird vorläufig hintenangestellt, sobald die erste Bedingung erfüllt ist. Daher werden zunächst nur solche aktivierten Transitionen für eine Schaltschritterweiterung in Betracht gezogen, für die keine Schaltprioritäten definiert sind. Dies erspart den

Aufwand für die Überprüfung der zweiten Bedingung, falls sich auf diese Weise eine Schaltschritterweiterung finden läßt, für die gilt: Der erweiterte Schaltschritt ist zulässig und wird vom Modellbenutzer als geeignet akzeptiert. Andernfalls müssen die aktivierten Transitionen, die wegen Verletzung der ersten Bedingung vorerst zurückgestellt worden waren, nachträglich doch noch hinsichtlich der zweiten Bedingung überprüft werden.

124) Dies folgt aus der späteren Vereinbarung für Schaltprioritäten, daß eine aktivierte Transition mit einer definierten Schaltpriorität nur dann schalten darf, wenn unter derselben Markierung keine andere Transition mit einer höheren Schaltpriorität (aus der gleichen Prioritätsgruppe) aktiviert ist.

125) Darüber hinaus ist es empfehlenswert, die interaktive Exploration des Erreichbarkeitsgraphen so auszugestalten, daß der Modellbenutzer selbst bestimmen kann, welche heuristische Basisstrategie er bevorzugt: Er kann entweder ressourcenorientiert vorgehen und sich entsprechend kleine Schaltschritte vorschlagen lassen, die sukzessiv vergrößert werden. Oder er kann nebenläufigkeitsbezogen arbeiten, indem er Vorschläge für möglichst umfassende Schaltschritte erbittet. Die Benutzervorliebe für eine dieser beiden Basisstrategien kann in einem Benutzermodell vom Automatischen Informationsverarbeitungssystem vorgehalten werden. Das jeweils relevante Benutzermodell läßt sich bei der Initialisierung eines Analyselaufs durch die Selbstidentifikation des Benutzers (sein "Einloggen") abrufen. Darüber hinaus ist dem Modellbenutzer der Freiheitsgrad einzuräumen, während des Ablaufs einer interaktiven Erreichbarkeitsanalyse seine Vorliebe für eine der beiden Basisstrategien zu verändern. Dazu können ihn z.B. unbefriedigende Erfahrungen veranlassen, die er mit der vorangehenden Anwendung der jeweils anderen Basisstrategie gesammelt hat.

126) Z.B. können die aktivierenden Schaltfarben zufällig ausgewählt werden.

127) Oftmals wird es möglich sein, zwischen mehreren herauszunehmenden Transitionen auszuwählen. Auch diese Transitionsauswahl kann von einem Heurismus geleitet werden. Beispielsweise lassen sich bevorzugt jene Transitionen aus dem Schaltschritt entfernen, die gleichzeitig zu mehreren gefährlichen Transitionenkombinationen gehören. Dann fällt die Chance, den betrachteten Schaltschritt mit nur wenigen Transitionenstreichungen von allen gefährlichen Transitionenkombinationen zu befreien, größer aus, als wenn die Transitionen zufällig eliminiert worden wären.

5.2.3.5.4 Optimierung von Netzmodellen

Eine Methode zur Erreichbarkeitsanalyse von Netzmodellen wird in exemplarischer Weise näher vorgestellt. Sie veranschaulicht, wie sich dynamische Netzeigenschaften durch die systematische Auswertung von Erreichbarkeitsgraphen untersuchen lassen. Die Auswertungsmethode gestattet es, den optimalen Schaltwert für ein finales Netzmodell¹⁾ zu ermitteln²⁾.

Als Anwendungsbereich der Optimierungsmethode werden Netzmodelle vorausgesetzt, deren monodimensionale Formalzielsysteme aus jeweils einem Extremierungsziel bestehen³⁾. Die Zielerreichung dieses Extremierungsziels wird durch eine Zielfunktion gemessen, deren Funktionswert - je nach gewünschter Extremierungsrichtung - entweder minimiert oder aber maximiert werden soll. Ferner wird unterstellt, daß allen Transitionen eines Netzmodells und ihren Schaltfarben die Schaltwerte $sw(t_n, c_n)$ zugeordnet sind⁴⁾. Sie sind jeweils so bemessen, daß jeder Schaltwert $sw(t_n, c_n)$ diejenige Veränderung des Zielfunktionswerts angibt, die durch das Schalten der Transition t_n mit der Schaltfarbe c_n verursacht wird. Darüber hinaus muß für das Netzmodell eine nicht-leere Menge ME aus erwünschten Endmarkierungen $M_{E/c}$ mit $e \in \{1, \dots, E\}$ und $E \in \mathcal{N}_+$ definiert sein⁵⁾. Unter diesen Voraussetzungen wird nach dem optimalen Schaltwert aller finalen Schaltprozesse - dem Netzoptimum $sw^{*6)}$ - gesucht.

Die netzoptimierende Methode wird hier aus drei Gründen eingehender erläutert. Erstens unterstellt der Verf., daß von allen früher entfaltenen dynamischen Netzeigenschaften die sw^* -Optimalität aus betriebswirtschaftlicher Perspektive am interessantesten erscheint. Dafür spricht, daß die Ermittlung optimaler Verhaltensweisen (Schaltprozesse) von dynamischen Systemen (Netzmodellen) ein zentrales Anliegen der betriebswirtschaftlichen Entscheidungstheorie darstellt⁷⁾. Zweitens bietet sich gerade bei der Untersuchung der sw^* -Optimalität von Netzen ein Ansatzpunkt, um das Petrinetz-Konzept durch ein Analysekonzept zu bereichern, das sonst vornehmlich im betriebswirtschaftlichen Bereich thematisiert wird. Es handelt sich um das Roll back-Konzept. Es spielt bei der Entscheidungsbaumtechnik⁸⁾ und bei der Dynamischen Optimierungsrechnung⁹⁾ eine größere Rolle¹⁰⁾. Drittens gehört die Optimierung von Netzmodellen prima facie zu den zentralen Schwächen des Petrinetz-Konzepts. Denn es besteht *scheinbar* das Dilemma, zwischen zwei Alternativen wählen zu müssen:

- Entweder werden Netzklassen mit einfacher topologischer Struktur - wie etwa Synchronisationsnetze - verwendet. Für sie sind mehrere Optimierungsansätze aus der Netzliteratur bekannt. Aber aufgrund ihrer restriktiven Topologie schränken sie das hohe Ausdrucksvermögen des Petrinetz-Konzepts so weit ein, daß ihr Wert für die Objektmodellierung erheblich leidet.
- Oder die Netzmodelle dürfen das gesamte Ausdruckspotential des Petrinetz-Konzepts ausschöpfen. Dann versagen aber die vorgenannten Optimierungsansätze. Verallgemeinerte oder neuartige Konzepte für die Optimierung von Netzmodellen, deren topologische Struktur keinen Einschränkungen unterliegt, wurden bisher nicht angeboten.

Dieses scheinbare Dilemma wird anschließend durch die Konstruktion und Auswertung von Optimierungsnetzen überwunden. Hierdurch wird eine neuartige Optimierungsmethode für Netzmodelle eingeführt. Sie läßt sich auf *alle* Netzklassen anwenden, die vom Ausdrucksvermögen der Synthetische Netze umschlossen werden. Dazu gehören vor allem auch die schon mehrfach angesprochenen Stelle/Transition- und die Prädikat/Transition-Netze.

Ausgangspunkt für die Ermittlung des optimalen Schaltwerts sw^* eines finalen Netzmodells ist die früher vorgestellte Möglichkeit, den Erreichbarkeitsgraphen eines Netzes selbst wieder als ein Netz darzustellen. Allerdings werden die dort konstruierten Erreichbarkeitsnetze hier zu Optimierungsnetzen abgewandelt¹¹⁾. Hervorzuheben ist, daß nicht vom auszuwertenden finalen Netzmodell selbst, sondern von dessen Erreichbarkeitsgraphen ausgegangen wird¹²⁾. Daher läßt

sich die vorgelegte Optimierungsmethode auf Netzmodelle unabhängig davon anwenden, zu welcher Netzklasse sie gehören. Es müssen lediglich drei Voraussetzungen erfüllt sein:

- Die Netzmodelle besitzen wohldefinierte dynamische Strukturen, so daß ihre Erreichbarkeitsgraphen definiert sind.
- Die Erreichbarkeitsgraphen der Netzmodelle sind endlich.
- Die Erreichbarkeitsgraphen liegen als bewertete Erreichbarkeitsgraphen vor, in denen bereits die Schaltwerte der Transitionen berücksichtigt sind.

Ein bewerteter Erreichbarkeitsgraph $RG_b = (KN_{RG}, KA_{RG}; bk_b)$ stimmt mit einem gewöhnlichen Erreichbarkeitsgraphen $RG = (KN_{RG}, KA_{RG}; bk)$ bis auf die Beschriftungsfunktion für die Schaltkanten überein. Die Beschriftungsfunktion, die früher mit SSM als Menge aller denkmöglichen Schaltschritte durch $bk: KA_{RG} \rightarrow SSM$ definiert wurde, wird in bewerteten Erreichbarkeitsgraphen durch eine Beschriftungsfunktion bk_b mit $bk_b: KA_{RG} \rightarrow (SSM \times \mathcal{R})$ ersetzt¹³⁾. Dabei erstreckt sich die zweite Komponente des Nachbereichs der Beschriftungsfunktion bk_b auf die Menge \mathcal{R} aller reellen Zahlen. Die Schaltwerte von Schaltschritten können daher beliebige reellzahlige Beträge annehmen. Eine Schaltkante $ka_{r,a,f} = (M_r, M_f)$ wird in einem bewerteten Erreichbarkeitsgraphen durch $bk_b(ka_{r,a,f}) = (SS_a, sw_a)$ beschriftet. Diese Kantenanschrift ordnet der Schaltkante $ka_{r,a,f}$ das Paar aus ihrem Schaltschritt SS_a und dessen Schaltwert $sw(SS_a) = sw_a$ zu. Der Schaltwert sw_a eines Schaltschritts SS_a ist die Summe der Schaltwerte $sw(t_n, c_n)$ aller Transitionen t_n und Schaltfarben c_n , die zum Schaltschritt SS_a gehören¹⁴⁾:

$$SS_a = \{(t_{n(w)}, c_{n(w)}): w = 1, \dots, W_a\}$$

$$\Rightarrow sw(SS_a) = \sum(w \in \{1, \dots, W_a\}): sw(t_{n(w)}, c_{n(w)}) = sw_a$$

Um die Schaltwerte erfassen zu können, müssen die Optimierungsnetze zur Klasse der Höheren Netze gehören. Andernfalls wäre es nicht möglich, in der unten entfalteten Weise die Schaltwerte von Schaltschritten als Attributausprägungen von Kopien der Optimierungsmarke auszudrücken¹⁵⁾. Die Optimierungsnetze lassen sich sowohl durch Prädikat/Transition- als auch durch Synthetische Netze realisieren. Die vollständige Definition eines solchen Höheren Netzes wäre jedoch sehr aufwendig¹⁶⁾. Es geht dem Verf. aber hier nur darum, die Konstruktionsidee für Optimierungsnetze zu verdeutlichen. Daher beschränkt er sich darauf, Optimierungsnetze als Kanal/Instanz-Netze zu behandeln. Ihre semi-formale Festlegung reicht aus, um das grundlegende Konstruktionskonzept zu veranschaulichen. Gleiches gilt für die später präsentierte Methode zur Auswertung von Optimierungsnetzen. Unter den vorgenannten Voraussetzungen legen folgende Konstruktionsregeln fest, wie der bewertete Erreichbarkeitsgraph RG_b für ein beliebiges finales Netzmodell in ein Optimierungsnetz $OP(RG_b) = OP_{RG}$ transformiert wird¹⁷⁾.

Konstruktion eines Optimierungsnetzes

a) Jeder Markierungsknoten M_r aus dem Erreichbarkeitsgraphen wird zunächst auf genau eine Stelle des Optimierungsnetzes abgebildet. Knoten- und Stellenindizes werden der Übersichtlichkeit halber miteinander identifiziert, so daß jedem Knoten einer erreichbaren Markierung M_r eine Stelle s_r zugeordnet ist (und umgekehrt)¹⁸⁾. Die Stellenmenge S_{OP} des Optimierungsnetzes nimmt daher die Gestalt $S_{OP} = \{s_r: r = 0, 1, \dots, R\}$ mit $R = \#(KN_{RG}) - 1$ an¹⁹⁾. Ausgezeichnete Stellen des Optimierungsnetzes sind die Ausgangsstelle s_0 und die Endstellen $s_{E/e}$ mit $e \in \{1, \dots, E\}$ und $E \in \mathcal{N}_+$ ²⁰⁾. Die Ausgangsstelle s_0 repräsentiert den Markierungsknoten M_0 des bewerteten Erreichbarkeitsgraphen, also die Ausgangsmarkierung des zugrundeliegenden Netzmodells. Die Endstellen $s_{E/e}$

stellen jeweils einen Markierungsknoten $M_{E/e}$ dar, also eine erwünschte Endmarkierung aus der Endmarkierungsmenge ME des Netzmodells²¹⁾.

b) Für die Markierung der Stellen des Optimierungsnetzes wird genau eine Marke definiert. Es handelt sich um eine Attributmarke der Sorte "marke" mit zwei Attributen. Das erste Attribut ist die Sorte "schaltwert". Dieses Schaltwert-Attribut besitzt als Definitionsbereich zulässiger Attributausprägungen "wert"²²⁾ die Menge "REAL" aller reellen Zahlen. Das zweite Attribut ist die Sorte "schaltfolge". Es wird als Schaltfolgen-Attribut bezeichnet. Sein Definitionsbereich zulässiger Attributausprägungen "folge"²³⁾ ist die Menge "LISTE" aller listenförmigen Ausdrücke $\langle \text{liste} \rangle$ ²⁴⁾. Listen werden hier - in Anlehnung an die Programmiersprache PROLOG - dadurch gekennzeichnet, daß sie in die reservierten Klammern "[" und "]" eingeschlossen werden²⁵⁾. Der Sorte "marke" wird die Objektmenge "SYMBOL" aller symbolischen Ausdrücke zugewiesen. Die Attributmarke wird durch das Operationssymbol "Optimierungsmarke" konstituiert und ebenso benannt. Wenn die interne Attributstruktur von Kopien der Optimierungsmarke nicht expliziert zu werden braucht, lassen sie sich kompakt als Markenkopien " m_{opt} " notieren. Daher gilt für die Optimierungsmarke und alle ihre Kopien:

Optimierungsmarke: schaltwert schaltfolge \rightarrow marke

optimierungsmarke: REAL \times LISTE \rightarrow SYMBOL
 (wert,folge) \rightarrow optimierungsmarke(wert,folge) = m_{opt}

c) Die Attributausprägung "wert", die eine Kopie der Optimierungsmarke unter einer Markierung ausweist, wird auch als deren aktueller Schaltwert bezeichnet. Falls es sich um eine erwünschte Endmarkierung des Optimierungsnetzes handelt, heißt die Attributausprägung "wert" der finale Schaltwert der Markenkopie.

d) In Kantengewichten und Markierungen wird für die Kopien der Optimierungsmarke die Kurznotation " $\langle \dots \rangle$ " zugelassen, die früher für Attributmarken von Synthetischen Netzen eingeführt wurde: Der Ausdruck " $\langle \text{wert,folge} \rangle$ " kürzt die Schreibweise "optimierungsmarke(wert, folge)" ab²⁶⁾.

e) Jede Stelle s_r aus der Stellenmenge S_{OP} besitzt die Markenkapazität $KAP_r=1$. Für die Kapazitätsfunktion K_{OP} des Optimierungsnetzes gilt daher $K_{\text{OP}}: S_{\text{OP}} \rightarrow \{1\}$.

f) Unter der Ausgangsmarkierung $M_{0,\text{OP}}$ ²⁷⁾ des Optimierungsnetzes wird jede Endstelle $s_{E/e}$ durch genau eine Kopie der Optimierungsmarke belegt. Alle anderen Stellen des Optimierungsnetzes bleiben unter seiner Ausgangsmarkierung unmarkiert. Die Kopien der Optimierungsmarke werden für die Ausgangsmarkierung durch die Attributausprägungen "0"²⁸⁾ für das Schaltwert- und "[]"²⁹⁾ für das Schaltfolgen-Attribut initialisiert. In der Kurznotation nehmen sie daher die Gestalt " $\langle 0,[] \rangle$ " an. Folglich gilt für die Ausgangsmarkierung $M_{0,\text{OP}}$ des Optimierungsnetzes bei formaler Summennotation³⁰⁾:

$$\begin{aligned} M_{0,\text{OP}}: \quad s_{E/e} &\rightarrow M_{0,\text{OP}}(s_{E/e}) = \langle 0,[] \rangle; && \text{für alle } e \in \{1, \dots, E\} \\ s_r &\rightarrow M_{0,\text{OP}}(s_r) = 0; && \text{für alle } s_r \in (S_{\text{OP}} - \{s_{E/e} : e=1, \dots, E\}) \end{aligned}$$

g) Jede Endmarkierung $M_{E/g,\text{OP}}$ des Optimierungsnetzes mit $g \in \{1, \dots, G\}$ und $G \in \mathcal{N}_r$ ³¹⁾ wird so festgelegt, daß unter ihr die Ausgangsstelle s_0 mit genau einer Kopie $\langle \text{wert}_e, \text{folge}_e \rangle$ der Optimierungsmarke belegt ist. Verschiedene Endmarkierungen unterscheiden sich durch die Ausprägungen "wert_e" oder "folge_e" der Attribute ihrer Kopie der Optimierungsmarke. Alle anderen Stellen weisen unter den Endmarkierungen des Optimierungsnetzes keine Marke auf. Folglich

gilt für die Endmarkierungsmenge $ME_{OP} = \{M_{E/g,OP} : g = 1, \dots, G\}$ des Optimierungsnetzes bei formaler Summennotation³²⁾ für alle Endmarkierungen $M_{E/g,OP}$ mit $g \in \{1, \dots, G\}$:

$$\begin{aligned} M_{E/g,OP} : s_0 &\rightarrow M_{E/g,OP}(s_0) = \langle \text{wert_e, folge_e} \rangle \\ s_r &\rightarrow M_{E/g,OP}(s_r) = 0; \quad \text{für alle } s_r \in (S_{OP} - \{s_0\}) \end{aligned}$$

h) Jede Schaltkante $ka_{r,a,f} = (M_r, M_f)$, die im Erreichbarkeitsgraphen wegen $bk(ka_{r,a,f}) = (SS_a, sw_a)$ mit einem Schaltschritt SS_a und dessen Schaltwert sw_a beschriftet ist, wird im Optimierungsnetz durch genau eine Transition t_n abgebildet (vice versa). Der Übersichtlichkeit halber werden die Indizes von Transitionen t_n und zugehörigen Schaltkanten $ka_{r,a,f}$ miteinander identifiziert. Dann ist jeder Schaltkante $ka_{r,a,f}$ des Erreichbarkeitsgraphen genau eine Transition $t_{r,a,f}$ im Optimierungsnetz zugeordnet (und umgekehrt). Diese Transitionen werden als Schaltschritt-Transitionen bezeichnet. Denn sie dienen dazu, im Optimierungsnetz die Schaltschritte SS_a zu repräsentieren, mit denen die abgebildeten Schaltkanten beschriftet sind.

i) Jede Schaltschritt-Transition $t_{r,a,f}$ wird mit zwei Bestimmungsgleichungen beschriftet³³⁾. Sie legen fest, wie die Attributausprägungen aller Kopien der Optimierungsmarke, die von einem Schalttakt der Transition betroffen sind, modifiziert werden. Dabei wird jeweils auf den Schaltschritt SS_a Bezug genommen, der von der Schaltschritt-Transition $t_{r,a,f}$ repräsentiert wird. Die erste Bestimmungsgleichung gibt an, daß beim Schalten der Transition die alte Ausprägung "Wert_alt"³⁴⁾ des Attributs "schaltwert" um den Schaltwert sw_a des Schaltschritts SS_a erhöht³⁵⁾ wird, um die neue Attributausprägung "Wert_neu" zu erhalten³⁶⁾:

$$\text{Wert_neu} := \text{Wert_alt} + sw_a$$

Die zweite Bestimmungsgleichung legt fest, daß die neue Ausprägung "Folge_neu" des Attributs "schaltfolge" gewonnen wird, indem die Bezeichnung " SS_a " des Schaltschritts der Schaltschritt-Transition $t_{r,a,f}$ an die Liste "Folge_alt" der alten Attributausprägung vorne angehängt wird³⁷⁾:

$$\text{Folge_neu} := [SS_a] \oplus \text{Folge_alt}$$

j) Jede Schaltschritt-Transition $t_{r,a,f}$, deren Schalten eine Referenzmarkierung M_r in eine Folgemarkierung M_f transformiert, wird im Optimierungsnetz zunächst durch genau eine Eingangskante $(s_r, t_{r,a,f})$ und durch genau eine Ausgangskante $(t_{r,a,f}, s_f)$ mit den beiden Stellen s_r bzw. s_f verknüpft, welche die Markierungen M_r bzw. M_f repräsentieren.

k) Die Eingangskante $(s_r, t_{r,a,f})$ einer Schaltschritt-Transition $t_{r,a,f}$ erhält das Kantengewicht $\langle \text{Wert_alt}, \text{Folge_alt} \rangle$ und ihre Ausgangskante das Kantengewicht $\langle \text{Wert_neu}, \text{Folge_neu} \rangle$. Das Schalten einer Schaltschritt-Transition bewirkt daher, von ihrer einen Eingangsstelle s_r genau eine Kopie der Optimierungsmarke mit den alten Attributausprägungen "Wert_alt" und "Folge_alt" abzuziehen. Zugleich wird eine Kopie der Optimierungsmarke mit den neuen Attributausprägungen "Wert_neu" und "Folge_neu" auf der einen Ausgangsstelle s_f der Schaltschritt-Transition abgelegt.

l) Jede Stelle s_r , deren Vor- oder³⁸⁾ Nachbereich im vorläufigen Optimierungsnetz aus mehreren Schaltschritt-Transitionen besteht, wird durch ein stellenberandetes Teilnetz substituiert³⁹⁾. Das Teilnetz besteht aus genau einer Transition t_r . Sie wird auch als Markierungs-Transition bezeichnet. Für jede Schaltschritt-Transition, die zum Vor- oder Nachbereich der Stelle s_r gehört, enthält das Teilnetz genau eine transitionsspezifische Stelle $s_{r,h}$ mit $h \in \{1, \dots, H_{V,r}, H_{V,r}+1, \dots, H_{V,r}+H_{N,r}\}$, $H_{V,r} = \#(VB(s_r))$ und $H_{N,r} = \#(NB(s_r))$. Jede Stelle $s_{r,h}$, die mit $h \in \{1, \dots, H_{V,r}\}$ für eine

Schaltschritt-Transition aus dem Vorbereich der Stelle s_r eingeführt wurde, wird im Teilnetz mit der Markierungs-Transition t_r durch genau eine Kante ($s_{r,h}, t_r$) verbunden. Ebenso wird jede Stelle $s_{r,h}$, die mit $h \in \{H_{V,r}+1, \dots, H_{V,r}+H_{N,r}\}$ für eine Schaltschritt-Transition aus dem Nachbereich der Stelle s_r eingeführt wurde, im Teilnetz mit der Markierungs-Transition t_r durch genau eine Kante ($t_r, s_{r,h}$) verknüpft. Im vorläufigen Optimierungsnetz wird die Stelle s_r zusammen mit allen adjazenten Kanten ($t_{g,a,r}, s_r$) und ($s_r, t_{r,a,f}$) eliminiert. Statt dessen wird das zuvor definierte stellenberandete Teilnetz für die Stelle s_r eingesetzt. Jede Stelle $s_{r,h}$, die für eine Schaltschritt-Transition $t_{g,a,r}$ aus dem Vorbereich der Stelle s_r eingeführt wurde, wird mit einer Eingangskante ($t_{g,a,r}, s_{r,h}$) versehen. In gleicher Weise bekommt jede Stelle $s_{r,h}$, die für eine Schaltschritt-Transition $t_{r,a,f}$ aus dem Nachbereich der Stelle s_r eingeführt wurde, eine Ausgangskante ($s_{r,h}, t_{r,a,f}$). Die Ein- und Ausgangskanten der Stellen $s_{r,h}$ erhalten jeweils die gleichen Gewichte wie die Kanten ($t_{g,a,r}, s_r$) bzw. ($s_r, t_{r,a,f}$) aus dem vorläufigen Optimierungsnetz, deren Platz sie eingenommen haben.

m) Eine Markierungs-Transition t_r heißt eine Multiplikator-Transition, wenn sie nur eine Eingangsstelle besitzt. Aufgrund der voranstehenden Teilnetzkonstruktion weist eine solche Multiplikator-Transition mehrere Ausgangsstellen auf. Ihre Ein- und Ausgangskanten erhalten jeweils dasselbe Kantengewicht $\langle m_{opt} \rangle$ mit m_{opt} als einer beliebigen Kopie der Optimierungsmarke. Die Funktion einer Multiplikator-Transition besteht lediglich darin, eine Kopie der Optimierungsmarke identisch zu vervielfachen.

n) Eine Markierungs-Transition t_r wird hingegen als eine Auswahl-Transition bezeichnet, wenn sie mehrere Eingangsstellen $s_{r,h}$ mit $h \in \{1, \dots, H_{V,r}\}$, $H_{V,r} = \#(VB(t_r))$ und $H_{V,r} \geq 2$ aufweist⁴⁰⁾. Ihr Schalten bewirkt die Auswahl von genau einer Kopie der Optimierungsmarke. Diese Markenkopie zeichnet sich dadurch aus, daß ihre Ausprägung des Attributs "schaltwert" bezüglich aller anderen Markenkopien, die sich unter derselben Markierung auf den Eingangsstellen der Auswahl-Transition befinden, einen Extremwert⁴¹⁾ darstellt. Falls mehrere Kopien der Optimierungsmarke denselben extremalen - minimalen oder maximalen - Schaltwert besitzen, wird eine beliebige von ihnen ausgewählt⁴²⁾. Die Selektion einer Markenkopie mit mini- oder maximalem Schaltwert leistet folgende Beschriftung⁴³⁾:

$$\begin{aligned} \text{Wert} &:= \text{ext}(\{\text{Wert}_h; h=1, \dots, H_{V,r}\}) && \text{mit: } \text{ext} \in \{\min, \max\} \\ \text{Wert} &= \text{Wert}_1 && \rightarrow \text{Folge} := \text{Folge}_1 \\ &\dots && \\ \text{Wert} &= \text{Wert}_{H_{V,r}} && \rightarrow \text{Folge} := \text{Folge}_{H_{V,r}} \end{aligned}$$

Die Schaltakte von Auswahl-Transitionen bilden den zentralen Ansatzpunkt, auf den später das Roll back-Konzept der Dynamischen Programmierung angewendet wird, um das gesuchte Netzoptimum sw^* zu ermitteln.

o) Jede Eingangskante ($s_{r,h}, t_r$) einer Auswahltransition t_r mit $h \in \{1, \dots, H_{V,r}\}$ erhält das Kantengewicht $\langle \text{Wert}_h, \text{Folge}_h \rangle$. Jede ihrer Ausgangskanten ($t_r, s_{r,h}$) mit $h \in \{H_{V,r}+1, \dots, H_{V,r}+H_{N,r}\}$ wird mit dem Gewicht $\langle \text{Wert}, \text{Folge} \rangle$ versehen.

p) Andere Stellen, Transitionen oder Kanten als die voranstehend konstruierten besitzt das vorläufige Optimierungsnetz nicht.

q) Von besonderem Interesse sind die finalen Schaltprozesse $PRO_{0,E/g}$ des Optimierungsnetzes OP_{RG} . Jeder finale Schaltprozeß des Optimierungsnetzes startet unter der Ausgangsmarkierung $M_{0,OP}$ des Optimierungsnetzes und endet in einer seiner erwünschten Endmarkierungen $M_{E/g,OP}$ mit $g \in \{1, \dots, G\}$. Nach der Ausführung des finalen Schaltprozesses liegt unter seiner Endmarkierung $M_{E/g,OP}$ auf der Ausgangsstelle s_0 des Optimierungsnetzes OP_{RG} genau eine Kopie m_{opt} der Optimierungsmarke mit $m_{opt} = \langle \text{wert}_e, \text{folge}_e \rangle$.

r) Jedem finalen Schaltprozeß $PRO_{0,E/g}$ des Optimierungsnetzes OP_{RG} entspricht im zugrundeliegenden bewerteten Erreichbarkeitsgraphen RG_b genau ein Schaltweg $wg_{0,E/e,p}$ (und umgekehrt)⁴⁴). Ein solcher Schaltweg des Erreichbarkeitsgraphen wird als ein finaler Schaltweg bezeichnet, weil an seinem Ende eine der intendierten Endmarkierungen des zugrundeliegenden Petrinetzmodells erreicht wird. Er ist vom Knoten für die Ausgangsmarkierung M_0 des Netzmodells zu einem Knoten für eine der erwünschten Endmarkierungen $M_{E/e}$ des Netzmodells hin gerichtet: $wg_{0,E/e,p} = (M_0, \dots, M_{E/e})$. Die differenzierende Subskriptkomponente "p" mit $p \in \{1, \dots, P_{0,E/e}\}$ berücksichtigt, daß mehrere Schaltwege $wg_{0,E/e,p}$ existieren können, die im bewerteten Erreichbarkeitsgraphen von der Ausgangsmarkierung M_0 zu einer der erwünschten Endmarkierungen $M_{E/e}$ führen. In der Bezeichnung $PRO_{0,E/g}$ eines Schaltprozesses kann dies dadurch berücksichtigt werden, daß die Subskriptkomponente "g" auf den jeweils korrespondierenden Schaltweg $wg_{0,E/e,p}$ bezogen wird: $PRO_{0,E/g(e,p)}$. Andernfalls - wenn der zugehörige Schaltweg nicht näher interessiert - wird die kürzere Prozeßbezeichnung $PRO_{0,E/g}$ verwendet.

s) Allerdings wird der finale Schaltweg $wg_{0,E/e,p}$ aus dem Erreichbarkeitsgraphen im zugehörigen Schaltprozeß $PRO_{0,E/g(e,p)}$ des Optimierungsnetzes per constructionem *rückwärts* durchlaufen. Denn die Ausgangsmarkierung $M_{0,OP}$ des Optimierungsnetzes OP_{RG} korrespondiert wegen ihrer Markierung aller Endstellen $s_{E/e}$ mit der Menge aller erwünschten Endmarkierungen $M_{E/e}$ des zugrundeliegenden Netzmodells. Eine erwünschte Endmarkierung $M_{E/g,OP}$ des Optimierungsnetzes OP_{RG} entspricht hingegen wegen ihrer Markierung der einen Ausgangsstelle s_0 der Ausgangsmarkierung M_0 des zugrundeliegenden Netzmodells. Zugleich haben die Schaltakte der Transitionen im Optimierungsnetz während der Ausführung des finalen Schaltprozesses $PRO_{0,E/g}$ dafür gesorgt, daß aus der Menge aller erwünschten Endmarkierungen $M_{E/e}$ des zugrundeliegenden Netzmodells durch das Ausführen einer Schaltwegvariante "p" genau eine erwünschte Endmarkierung $M_{E/e}$ selektiert worden ist⁴⁵). Schaltwegvariante und Endmarkierung werden beide durch die Ausprägung des Schaltfolgen-Attributs "folge_e" derjenigen Kopie der Optimierungsmarke m_{opt} angezeigt, die mit der Ausprägung $m_{opt} = \langle \text{wert}_e, \text{folge}_e \rangle$ auf der Ausgangsstelle s_0 im Optimierungsnetz abgelegt worden ist. Daher kann in der Tat davon gesprochen werden, daß jeder finale Schaltprozeß $PRO_{0,E/g}$ im Optimierungsnetz OP_{RG} genau einem *rückwärts* durchlaufenen finalen Schaltweg $wg_{0,E/e,p}$ entspricht, der zum bewerteten Erreichbarkeitsgraphen RG_b des zugrundeliegenden Netzmodells gehört.

t) Gesucht wird nach mindestens einem optimalen finalen Schaltprozeß $PRO_{0,E/g}^*$. Jeder optimale finale Schaltprozeß zeichnet sich dadurch aus, daß er auf der Ausgangsstelle s_0 des Optimierungsnetzes OP_{RG} eine Kopie m_{opt}^* der Optimierungsmarke mit $m_{opt}^* = \langle \text{wert}_e^*, \text{folge}_e^* \rangle$ ablegt, deren Ausprägung "wert_e*" des Schaltwert-Attributs nach Maßgabe des vorausgesetzten Extremalziels bestmöglich ist. Die Attributausprägung "wert_e*" ist genau dann bestmöglich, wenn kein anderer finaler Schaltprozeß im Optimierungsnetz zulässig ist, der an seinem Ausführungsende auf der Ausgangsstelle s_0 eine Kopie m_{opt} der Optimierungsmarke mit $m_{opt} = \langle \text{wert}_e, \text{folge}_e \rangle$ ablegt, die zu einer - im Sinne des Extremalziels - besseren Ausprägung "wert_e" des Schaltwert-Attributs der Optimierungsmarkenkopie führen würde.

u) Das bisher konstruierte Optimierungsnetz besitzt im allgemeinen allerdings immer noch einen vorläufigen Charakter. Denn es kann durchaus noch Livelock- oder (unerwünschte) Deadlock-Komponenten⁴⁶) enthalten. Diese Netzkomponenten sind aber überflüssig: Einerseits können Schaltprozesse, die im Optimierungsnetz von Livelock- oder Deadlock-Komponenten betroffen sind⁴⁷), niemals finale Schaltprozesse darstellen. Andererseits gehören die gesuchten optimalen Schaltprozesse per definitionem zur Menge der finalen Schaltprozesse. Daher ist es ausgeschlossen, unter den Schaltprozessen, die von Livelock- oder Deadlock-Komponenten berührt werden, jemals einen der gesuchten optimalen Schaltprozesse zu finden. Deshalb läßt sich das vorläufig konstruierte Optimierungsnetz noch so weit vereinfachen⁴⁸), daß es von allen Livelock- und Deadlock-Komponenten frei ist⁴⁹):

- Alle Livelock-Komponenten eines Optimierungsnetzes werden eliminiert. Zu diesem Zweck werden alle zyklischen Wege aus der graphischen Repräsentation eines Optimierungsnetzes betrachtet⁵⁰⁾. Jeder Knoten und jede Kante, der bzw. die *ausschließlich* zu dem jeweils betroffenen Zyklus⁵¹⁾ gehört, wird aus dem vorläufigen Optimierungsnetz eliminiert. Danach ist das Optimierungsnetz von allen Livelock-Komponenten befreit⁵²⁾, ohne daß finale Schaltprozesse untergegangen wären⁵³⁾.
 - Ebenso werden alle Deadlock-Komponenten getilgt. Dabei handelt es sich um Stellen des Optimierungsnetzes, die jeweils einen Knoten des Erreichbarkeitsgraphen RG_b repräsentieren, der eine unerwünschte Endmarkierung des zugrundeliegenden Netzmodells darstellt. Solche deadlockartigen Stellen werden einschließlich aller ihrer Eingangskanten aus dem vorläufigen Optimierungsnetz entfernt.
- v) Es resultiert schließlich ein Optimierungsnetz OP_{RG} , das zur Klasse der Synchronisationsnetze gehört⁵⁴⁾. Denn alle seine Stellen, die zunächst noch mehrere Ein- oder Ausgangstransitionen besaßen, wurden so durch stellenberandete Subnetze verfeinert, daß danach alle Stellen höchstens eine Ein- und höchstens eine Ausgangstransition besitzen. Genau diese Stellencharakteristik definiert ein Synchronisationsnetz.

Ein Beispiel verdeutlicht die Konstruktion eines solchen Optimierungsnetzes. Aus Gründen der Übersichtlichkeit wird von einem sehr einfachen finalen Netz ausgegangen. Es handelt sich um das Stelle/Transition-Netz, das bereits in Abb. 18 zur Verdeutlichung gewöhnlicher Erreichbarkeitsgraphen eingeführt wurde. Da in Stelle/Transition-Netzen keine Schaltfarben c_n definiert sind, vereinfachen sich die Schaltwerte der Transitionen t_n zu $sw(t_n)$. Das betroffene Stelle/Transition-Netz wird in Abb. 92 auf der nächsten Seite noch einmal in Erinnerung gerufen und zugleich mit den Schaltwerten seiner Transitionen versehen.

Die sieben Schaltschritte SS_a , die im Stelle/Transition-Netz der Abb. 92 ausgeführt werden können, und ihre Schaltwerte $sw_a = sw(SS_a)$ werden mit $a \in \{1, \dots, 7\}$ in der nachfolgenden Auflistung wiedergegeben.

$SS_1 = \{t_1\}$	$sw_1 = sw(SS_1) = sw(t_1)$
$SS_2 = \{t_2\}$	$sw_2 = sw(SS_2) = sw(t_2)$
$SS_3 = \{t_3\}$	$sw_3 = sw(SS_3) = sw(t_3)$
$SS_4 = \{t_4\}$	$sw_4 = sw(SS_4) = sw(t_4)$
$SS_5 = \{t_1, t_4\}$	$sw_5 = sw(SS_5) = sw(t_1) + sw(t_4)$
$SS_6 = \{t_3, t_4\}$	$sw_6 = sw(SS_6) = sw(t_3) + sw(t_4)$
$SS_7 = \{t_1, t_3\}$	$sw_7 = sw(SS_7) = sw(t_1) + sw(t_3)$

Für die Markierungen, die durch das Ausführen der zuvor aufgelisteten Schaltschritte im Stelle/Transition-Netzes der Abb. 92 erreicht werden können, gilt:

$\underline{M}_0^{tr} = (1, 2, 0)$	
$\underline{M}_1^{tr} = (0, 3, 0)$	$\underline{M}_2^{tr} = (1, 0, 2)$
$\underline{M}_3^{tr} = (1, 1, 1)$	$\underline{M}_4^{tr} = (0, 0, 3)$
$\underline{M}_5^{tr} = (0, 1, 2)$	$\underline{M}_6^{tr} = (0, 2, 1)$

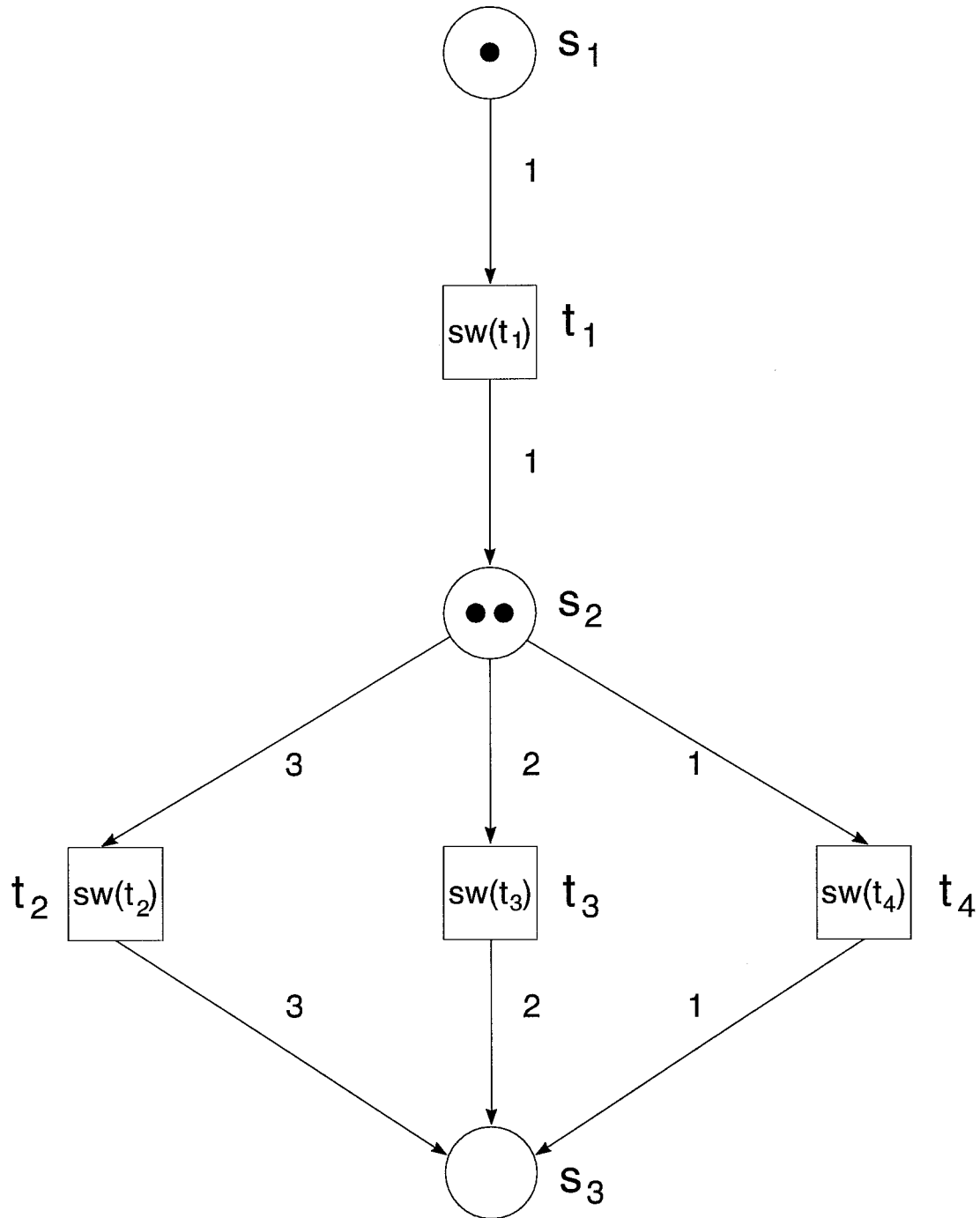


Abb. 92: Stelle/Transition-Netz zur Verdeutlichung der Konstruktion von Optimierungsnetzen

Abb. 93 u. 94 zeigen den zugehörigen bewerteten Erreichbarkeitsgraphen⁵⁵⁾ und das daraus abgeleitete Erreichbarkeitsnetz⁵⁶⁾. Abb. 95 bis 99 stellen schließlich das daraus abgeleitete Optimierungsnetz dar. Aufgrund seines Umfangs mußte das Optimierungsnetz in mehrere Teilnetze zerlegt werden⁵⁷⁾. Um einen Überblick zu ermöglichen, werden diese Teilnetze im Gesamtnetz der Abb. 100 zusammengefaßt. Dabei wird die Netztopologie vollständig visualisiert, aber von den umfangreichen Transitionen- und Kantenbeschriftungen abstrahiert. Da für das Netz nur genau eine erwünschte Endmarkierung $M_{E,1} = M_4 = M_E$ definiert wurde, enthält auch das zugehörige Optimierungsnetz nur genau eine korrespondierende Endstelle $s_{E,1} = s_E$.

Ein Vergleich des ursprünglichen Netzes aus Abb. 92 mit seinem Optimierungsnetz in Abb. 100 verdeutlicht, wie die oben erläuterte zweistufige Transformation von Netzmodellen über ihre bewerteten Erreichbarkeitsgraphen in Optimierungsnetze das Netzvolumen ansteigen läßt. Daraus den Schluß zu ziehen, von vornherein auf solche Netztransformationen zu verzichten, wäre jedoch aus zwei Gründen verfehlt. Erstens würde damit die eingangs herausgestellte Möglichkeit untergehen, optimale Schaltprozesse für Netzmodelle zu ermitteln, die das Ausdrucksvermögen des Petrinetz-Konzepts vollständig - insbesondere ohne Einschränkungen der topologischen Netzstruktur - ausschöpfen dürfen⁵⁸⁾. Zweitens braucht für ein Netzmodell nur einmal sein Optimierungsnetz konstruiert werden. Liegt das Optimierungsnetz erst einmal vor, so kann es für unterschiedlichste Schaltwerte der Transitionen des Netzmodells stets wiederverwendet werden. Dazu reicht es aus, lediglich die Schaltwerte der Schaltschritte SS_a aus den transitionsbezogenen Schaltwerten neu zu ermitteln. Diese Schaltwertanpassung stellt wegen des simplen additiven Schaltwertzusammenhangs eine triviale Aufgabe dar.

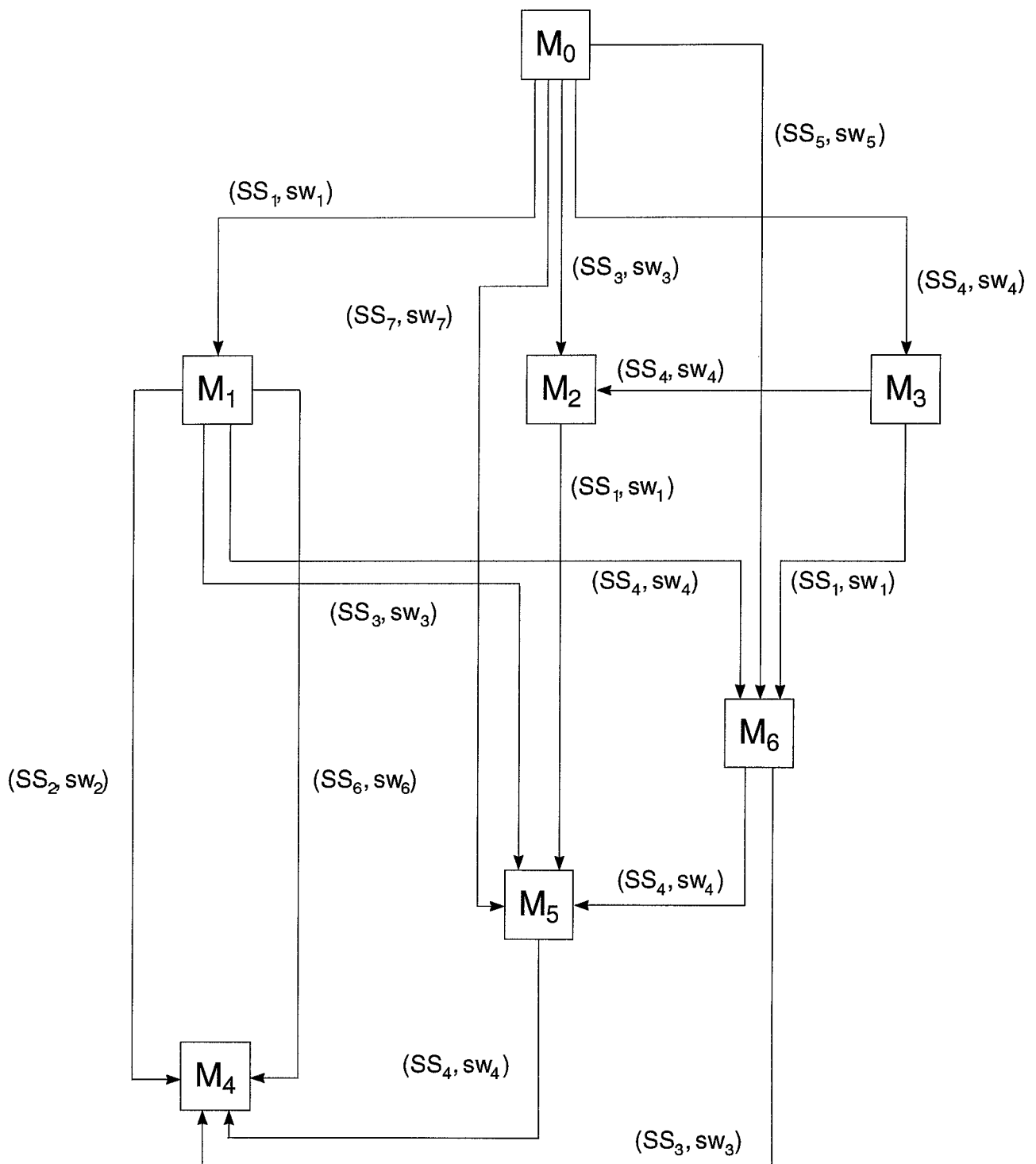


Abb. 93: Erreichbarkeitsgraph für das Stelle/Transition-Netz aus Abb. 92

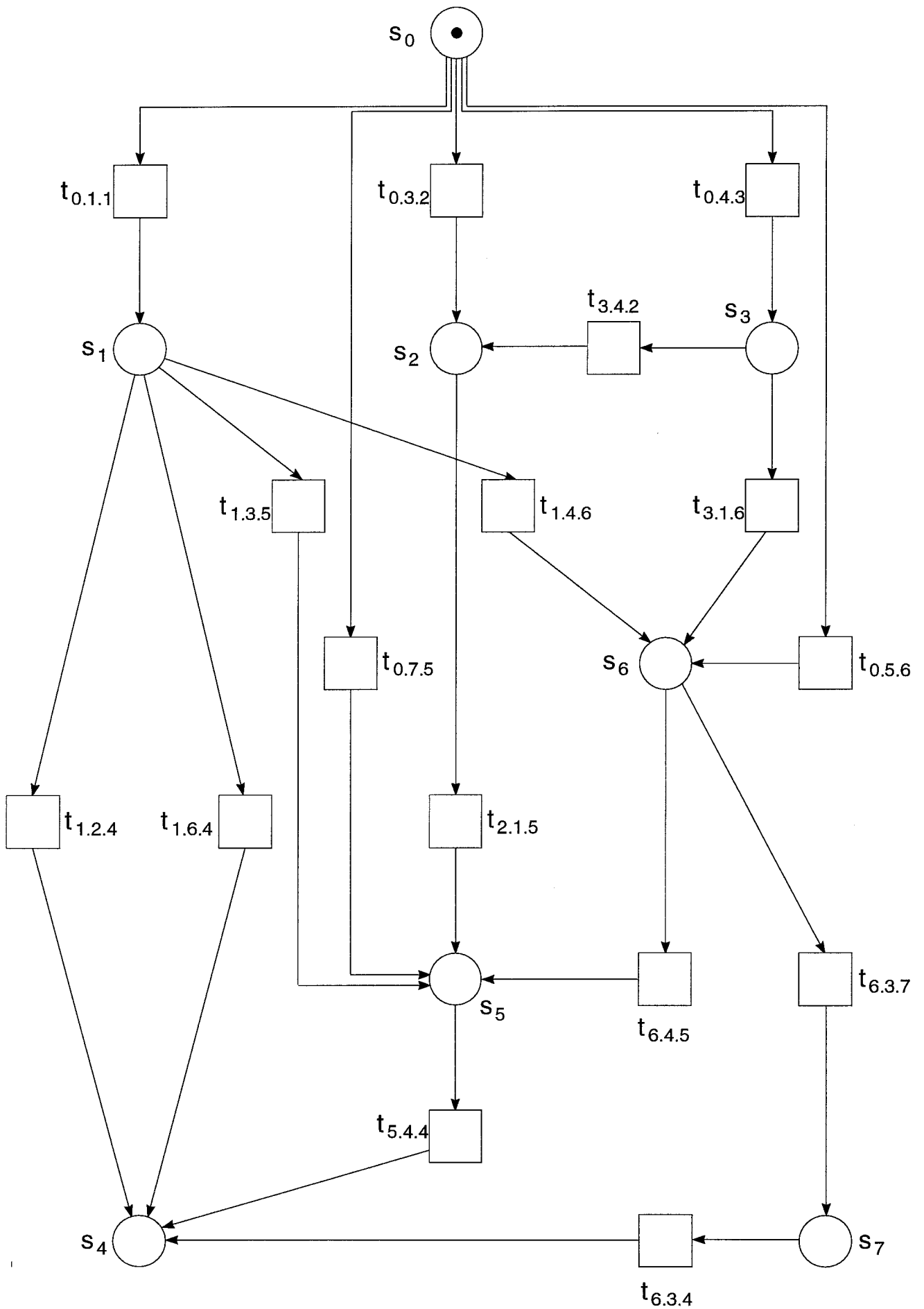


Abb. 94: Erreichbarkeitsnetz für den Erreichbarkeitsgraphen aus Abb. 93

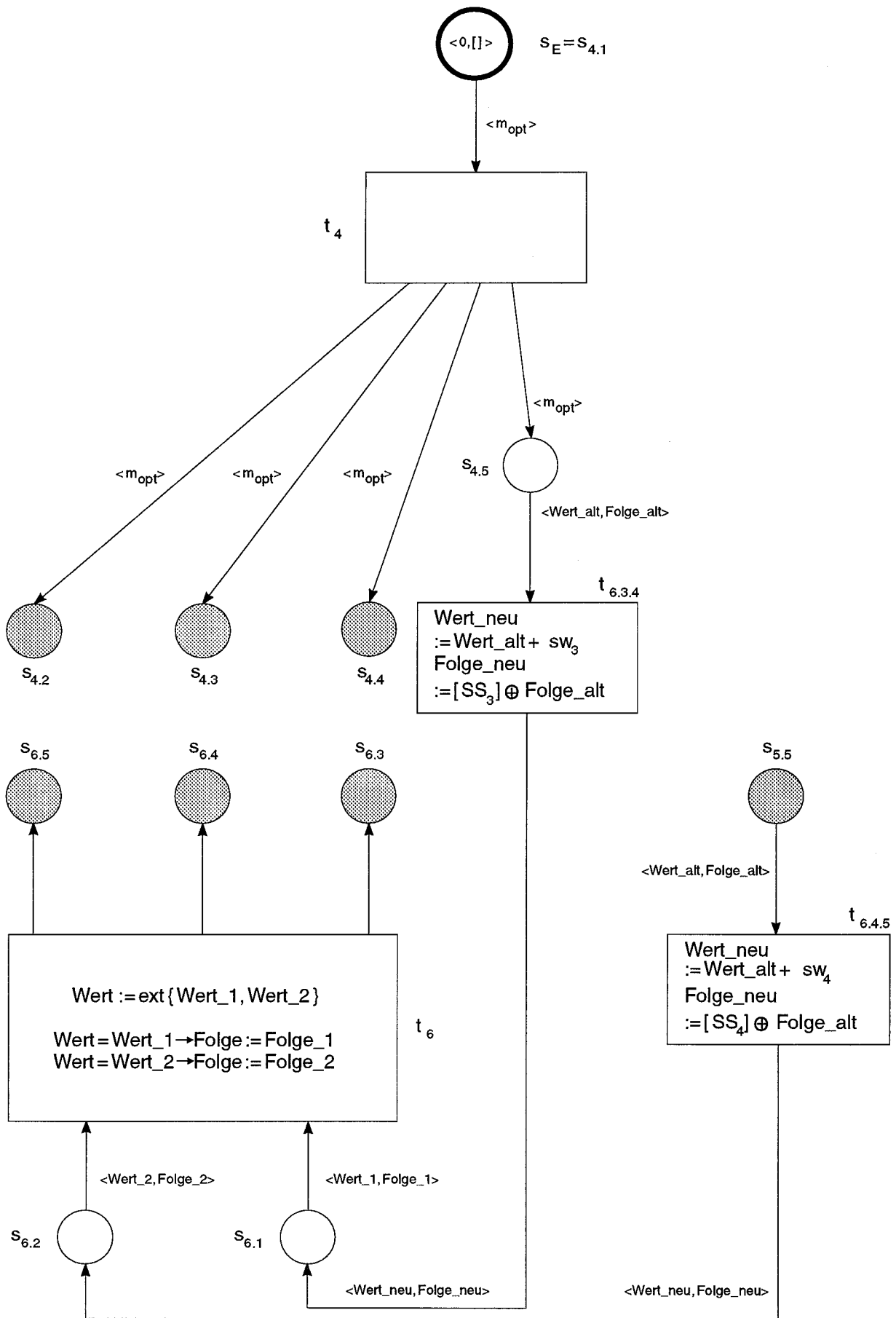


Abb. 95: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (erster Teil)

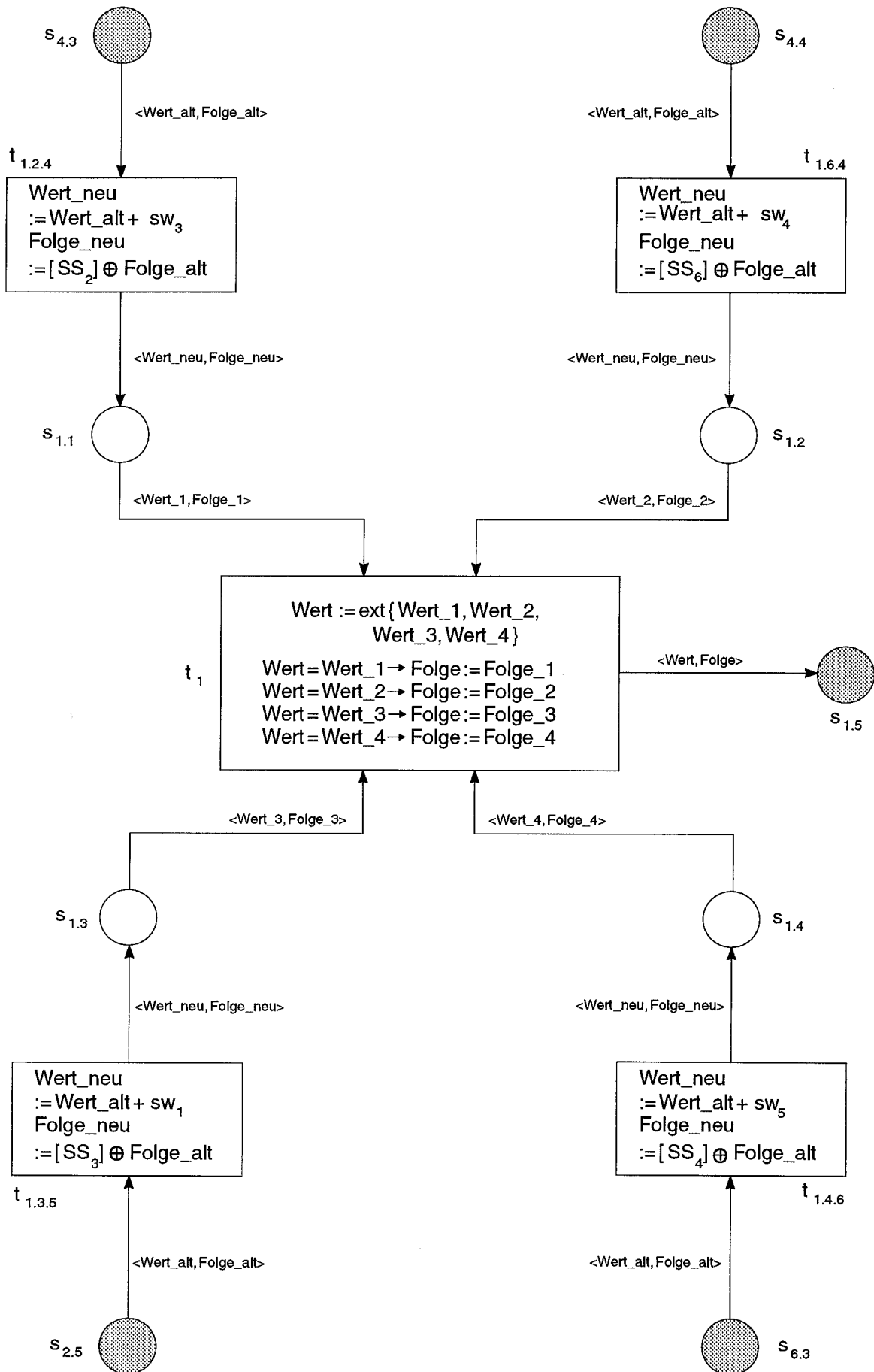


Abb. 96: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (zweiter Teil)

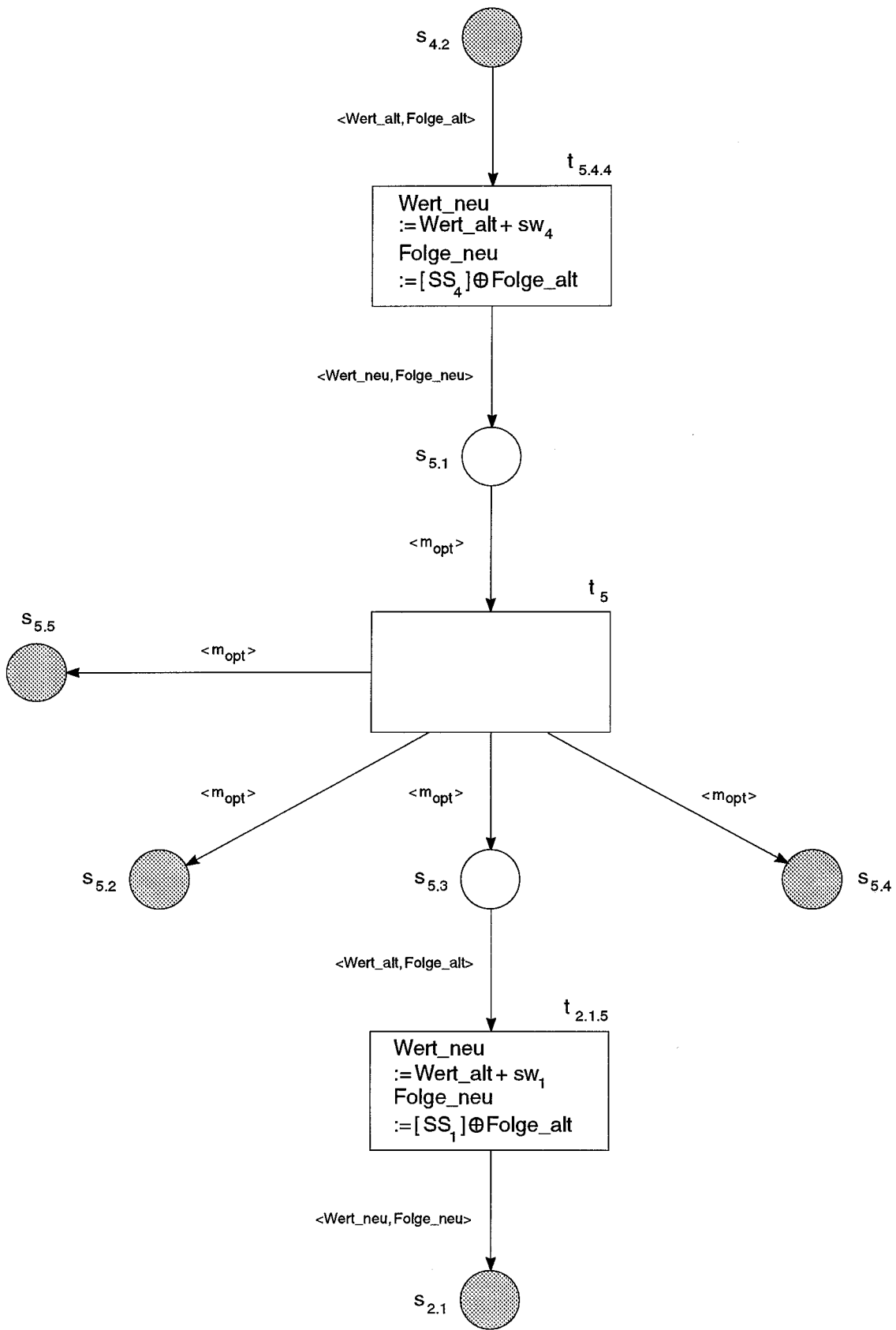


Abb. 97: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (dritter Teil)

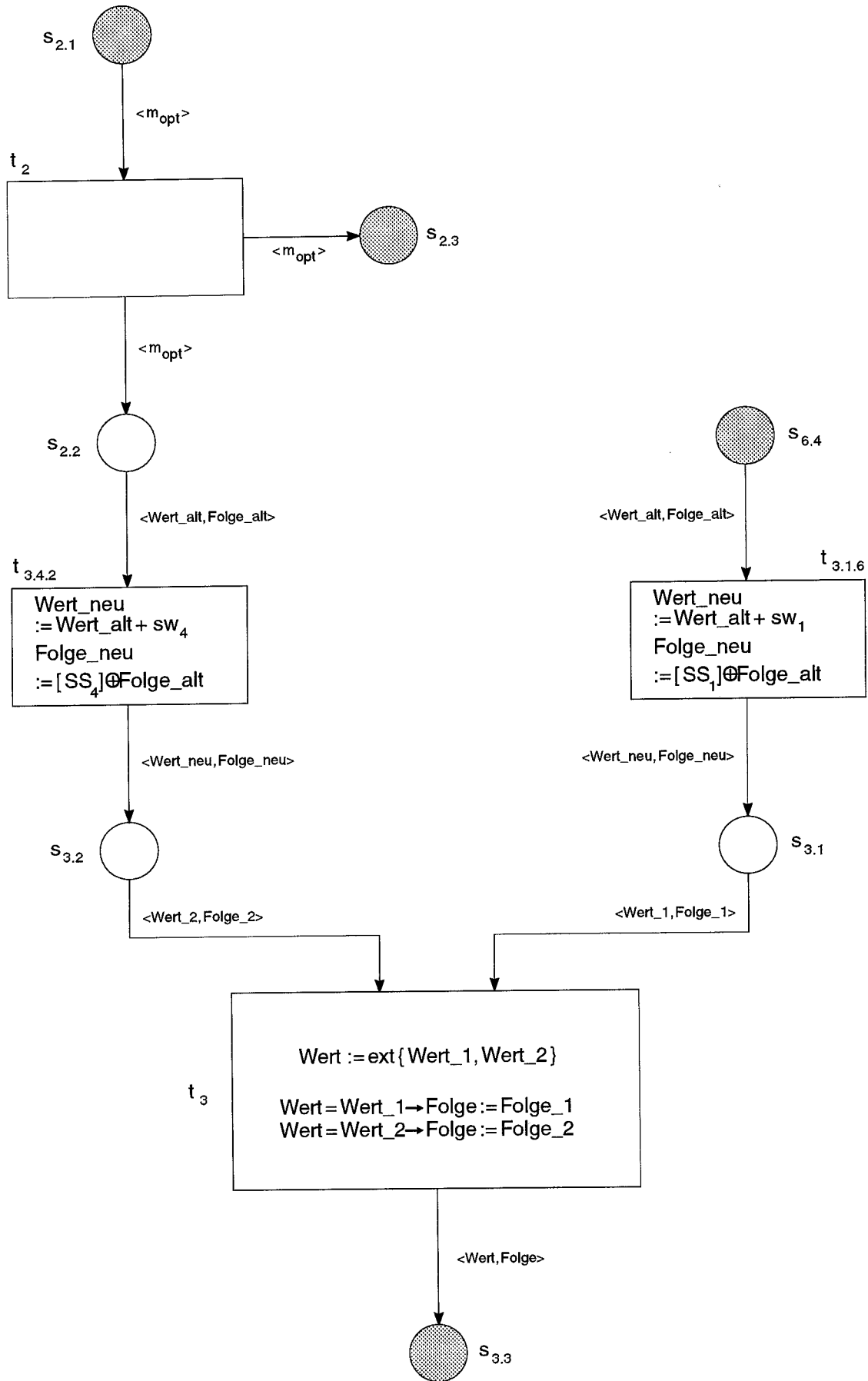


Abb. 98: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (vierter Teil)

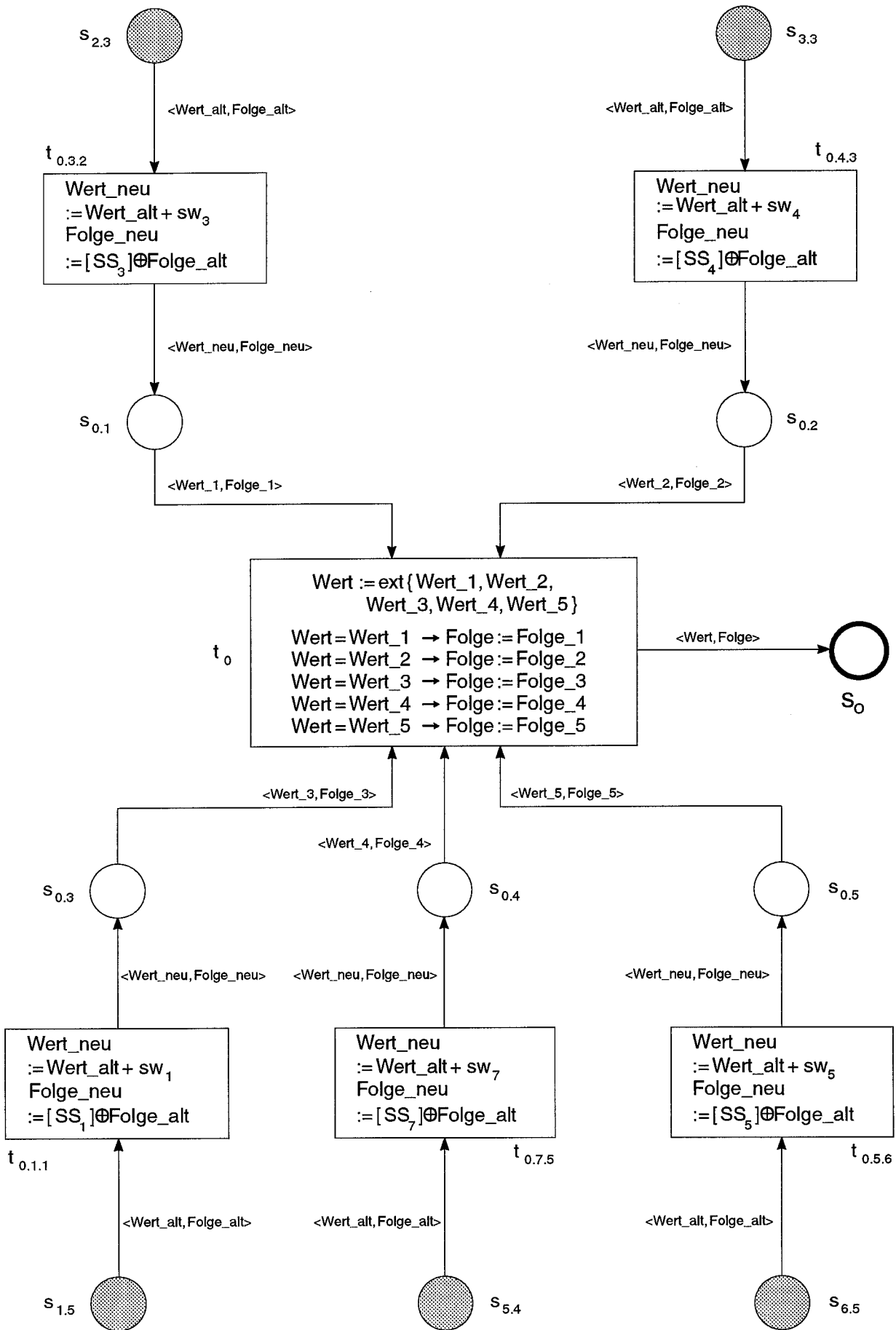


Abb. 99: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (fünfter Teil)

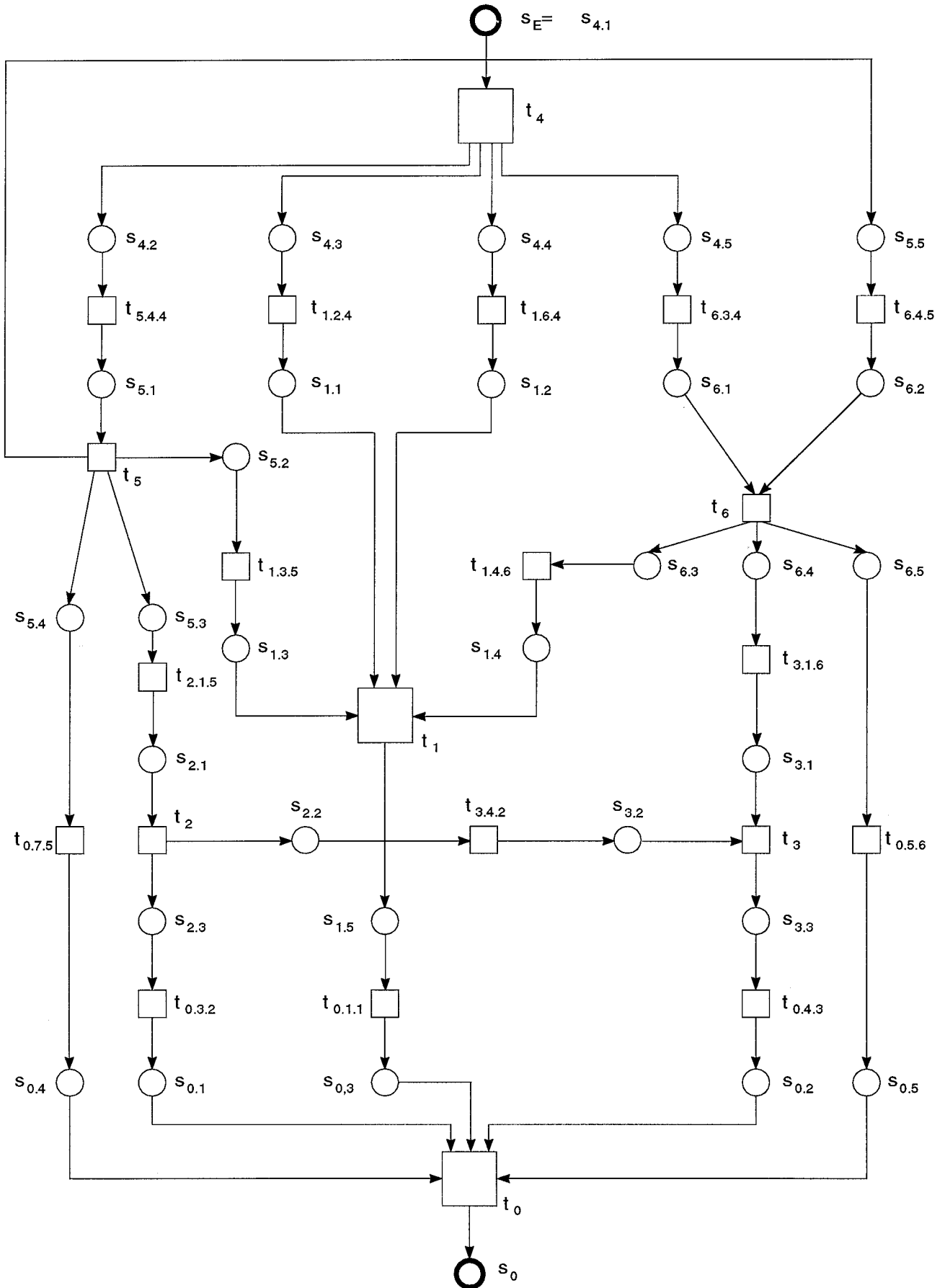


Abb. 100: Optimierungsnetz für das Erreichbarkeitsnetz aus Abb. 93 (Gesamtnetz im Überblick)

Bisher wurde nur die erste - die *konstruktive* - Phase der Optimierungsmethode für Netzmodelle beschrieben. Sie endet mit der Vorlage eines Optimierungsnetzes OP_{RG} für ein zugrundeliegendes Netzmodell. Es steht noch die zweite - die *explorative* - Methodenphase aus. Sie erstreckt sich auf die Aufgabe, im Optimierungsnetz nach einem optimalen Schaltprozeß $PRO_{0,E,g}^*$ zu forschen, der das gesuchte Netzoptimum sw^* als eine Attributausprägung der Optimierungsmarke hervorbringt. Dieser explorativen Phase der Optimierungsmethode widmen sich die nachfolgenden Überlegungen.

Zunächst wird eine deutliche Unterscheidung getroffen: Es gilt zwischen finalen Schaltprozessen zu differenzieren, die einerseits im zugrundeliegenden Netzmodell und andererseits im zugehörigen Optimierungsnetz ausgeführt werden. Die finalen Schaltprozesse des Optimierungsnetzes wurden bereits als Schaltprozesse $PRO_{0,E/g(e,p)}$ eingeführt, die von der Ausgangsmarkierung $M_{0,OP}$ des Optimierungsnetzes zu einer seiner erwünschten Endmarkierungen $M_{E,g,OP}$ führen. Sie werden im folgenden als finale Schaltprozesse $PRO_{0,E/g(e,p),OP}$ - oder kurz: $PRO_{0,E/g,OP}$ - aufgeführt, um zu verdeutlichen, daß sie innerhalb des Optimierungsnetzes OP_{RG} ausgeführt werden. Es wurde auch schon dargelegt, daß mit jedem dieser finalen Schaltprozesse genau ein finaler Schaltweg $wg_{0,E/e,p}$ aus dem bewerteten Erreichbarkeitsgraphen RG_b korrespondiert. Dieser finale Schaltweg repräsentiert seinerseits genau einen Schaltprozeß, der im zugrundeliegenden Netzmodell von dessen Ausgangsmarkierung M_0 genau so zu einer seiner erwünschten Endmarkierungen $M_{E,f}$ führt, wie es im Erreichbarkeitsgraphen RG_b durch die Schaltwegvariante $wg_{0,E/e,p}$ beschrieben wird. Daher handelt es sich auch bei dem Schaltprozeß im zugrundeliegenden Netzmodell um einen finalen Schaltprozeß. Bei seiner Ausführung wird im Netzmodell eine Schaltfolge SF_L der Länge L abgewickelt, die mit einer Markierungsfolge MF_L alternierend verschränkt ist. Der finale Schaltprozeß aus dem Netzmodell wird daher fortan der Deutlichkeit halber als Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ notiert⁵⁹). Da der finale Schaltprozeß $PRO_{0,E/g(e,p),OP}$ des Optimierungsnetzes OP_{RG} einerseits genau einem finalen Schaltweg $wg_{0,E/e,p}$ aus dem bewerteten Erreichbarkeitsgraphen RG_b entspricht und andererseits mit diesem finalen Schaltweg $wg_{0,E/e,p}$ genau ein finaler Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ im zugrundeliegenden Netzmodell korrespondiert, gilt ebenso: Jedem finalen Schaltprozeß $PRO_{0,E/g(e,p),OP}$ des Optimierungsnetzes OP_{RG} läßt sich auf eineindeutige Weise ein zugehöriger finaler Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ aus dem zugrundeliegenden Netzmodell zuordnen⁶⁰).

Die Konstruktion von Optimierungsnetzen ist so ausgelegt, daß es sich um Petrinetze besonderer Art handelt: Jeder finale Schaltprozeß $PRO_{0,E/g,OP}$ endet damit, daß genau eine Kopie der Optimierungsmarke auf der Ausgangsstelle s_0 des Optimierungsnetzes abgelegt ist. Unter dieser Endmarkierung $M_{E,g,OP}$ geben die beiden Attribute "Schaltwert" und "Schaltfolge" der Optimierungsmarkenkopie mit ihren aktuellen Ausprägungen "wert_e" bzw. "folge_e" das gesuchte Netzoptimum an:

- ☞ Die Ausprägung "wert_e" des Schaltwert-Attributs ist mit dem gesuchten Netzoptimum "sw*" identisch: $wert_e = sw^*$.
- ☞ Die Ausprägung "folge_e" des Schaltfolgen-Attributs weist eine Schaltfolge SF_L^* aus⁶¹). Es handelt sich um eine optimale Schaltfolge aus dem zugrundeliegenden Netzmodell. Diese Schaltfolge SF_L^* wird bei der Ausführung eines Schaltprozesses $PRO_{0,E/e,p}(SF_L^*, MF_L^*)$ im Netzmodell durchlaufen. Der Schaltprozeß $PRO_{0,E/e,p}(SF_L^*, MF_L^*)$ führt von der Ausgangsmarkierung M_0 des Netzmodells zu einer seiner erwünschten Endmarkierungen $M_{E,f}$. Dem Schaltprozeß $PRO_{0,E/e,p}(SF_L^*, MF_L^*)$ entspricht im bewerteten Erreichbarkeitsgraphen RG_b des Netzmodells ein optimaler Schaltweg $wg_{0,E/e,p}^*$, der an seinem Ende das Netzoptimum sw^* erzeugt hat.

Der konstruktive "Clou" von Optimierungsnetzen besteht also darin daß *jeder* finale Schaltprozeß $PRO_{0,E/g,OP}$ zugleich auch einer der gesuchten *optimalen* Schaltprozesse $PRO_{0,E/g,OP}^*$ ⁶²) ist. Daher braucht innerhalb der finalen Schaltprozesse, die schon bei der Konstruktion von Optimierungsnetzen hervorgehoben wurden, überhaupt nicht mehr gesondert nach einem optimalen

Schaltprozeß gesucht zu werden: Sobald ein finaler Schaltprozeß $PRO_{0.E/g.OP}$ des Optimierungsnetzes bekannt ist, steht per constructionem fest, daß es sich auch um einen optimalen Schaltprozeß $PRO_{0.E/g.OP}^*$ handeln muß: $PRO_{0.E/g.OP} = PRO_{0.E/g.OP}^*$.

Dieses verblüffende Resultat bedarf allerdings einer näheren Erläuterung. Denn auf den ersten Blick mag es überhaupt nicht zwingend erscheinen, daß die Konstruktion der Optimierungsnetze das Zusammenfallen von finalen und optimalen Schaltprozessen tatsächlich erzwingt. Erst bei genauerem Hinsehen zeigt sich, wie die Beschriftungen von Schaltschritt- und von Auswahl-Transitionen mit der Topologie des Optimierungsnetzes so zusammenwirken, daß jeder finale Schaltprozeß in der Tat das gesuchte Netzoptimum sw^* hervorbringt. Dies wird im folgenden näher erläutert.

Der Kerngedanke der Netzkonstruktion liegt darin, den Fluß einer Kopie der Optimierungsmarke durch das Optimierungsnetz so zu gestalten, daß durch den Markenfluß der Lösungsraum des zugrundeliegenden Optimierungsproblems implizit erforscht wird. Der Lösungsraum des Optimierungsproblems, im Optimierungsnetz einen optimalen Schaltprozeß $PRO_{0.E/g.OP}^*$ zu finden, besteht grundsätzlich aus der Gesamtheit aller Schaltprozesse, die im Optimierungsnetz OP_{RG} unter dessen Ausgangsmarkierung $M_{0.OP}$ gestartet werden können und sich - vielleicht - so fortsetzen lassen, daß sie zu einer der erwünschten Endmarkierungen $M_{E/g.OP}$ führen. Es wäre jedoch sehr aufwendig, diesen Lösungsraum explizit und vollständig zu erzeugen, um alsdann in der Gesamtheit aller gewonnenen Schaltprozesse einen optimalen Schaltprozeß zu identifizieren.

Statt dessen wird der Lösungsraum nur in einem sehr kleinen Ausschnitt expliziert: Es wird nur genau ein Schaltprozeß erzeugt. Seine Ausführung beginnt unter der Ausgangsmarkierung $M_{0.OP}$ des Optimierungsnetzes. Danach zwingt die spezielle Konstruktion des Optimierungsnetzes dazu, daß der Markenfluß des einmal gestarteten Schaltprozesses zwei wesentliche Eigenschaften erfüllt: Erstens führt der Schaltprozeß *notwendig* zu einer der gewünschten Endmarkierungen $M_{E/g.OP}$, sofern die aktivierten Transitionen auch tatsächlich geschaltet werden. Zweitens stellt sich am Ende der Schaltprozeßausführung, wenn im Optimierungsnetz keine Transition mehr aktiviert ist, das gesuchte Netzoptimum sw^* quasi "automatisch" ein.

Zur Ermittlung des gesuchten Netzoptimums sw^* ist es also im wesentlichen nur erforderlich, die Ausführung irgendeines Schaltprozesses unter der Ausgangsmarkierung $M_{0.OP}$ des Optimierungsnetzes zu starten⁶³). Zu diesem Zweck reicht es aus, irgendeinen derjenigen Schaltschritte auszuführen, die unter der Ausgangsmarkierung $M_{0.OP}$ des Optimierungsnetzes aktiviert sind⁶⁴). Im Beispiel, das oben angeführt wurde, handelt es sich um den genau einen Schaltschritt $SS_{a(0)} = \{t_4\}$. Danach werden die Kopien der Optimierungsmarke durch die Schaltakte der später aktivierten Transitionen so durch das Optimierungsnetz gesteuert, daß am Ende auf der Ausgangsstelle s_0 unter der erreichten Endmarkierung $M_{E/g.OP}$ genau eine Kopie der Optimierungsmarke liegt, die das gesuchte Netzoptimum sw^* anzeigt. Es kann daher in metaphorischer Rede-weise durchaus davon gesprochen werden, daß das Optimierungsnetz "aus sich selbst heraus" einen Fluß von Kopien der Optimierungsmarke bewirkt, der am Ende einen optimalen Schaltprozeß $PRO_{0.E/g.OP}^*$ darstellt. Obwohl nur ein Schaltprozeß explizit erzeugt wurde, ist dennoch das gesuchte Netzoptimum gefunden worden⁶⁵). Daher läßt sich die Vorstellung vertreten, der Markenfluß des Schaltprozesses $PRO_{0.E/g.OP}^*$ habe den Lösungsraum des zugrundeliegenden Optimierungsproblems auf implizite Weise erforscht. Diese spezielle Perspektive wird hier eingenommen, wenn davon die Rede ist, die explorative Phase der Optimierungsmethode für Netzmodelle erstreckte sich darauf, einen finalen - und zwangsläufig auch optimalen - Schaltprozeß im bereits konstruierten Optimierungsnetz OP_{RG} zu auszuführen.

Während der Ausführung des Schaltprozesses $PRO_{0.E/g.OP}^*$ wirken die Attributausprägungen der Kopien der Optimierungsmarke mit den Beschriftungen der Schaltschritt- und der Auswahl-Transitionen so zusammen, daß durch die schaltbedingten, fortlaufenden Aktualisierungen der Attributausprägungen das Netzoptimum sw^* und die optimale Schaltfolge SF_L^* innerhalb des Optimierungsnetzes "von selbst" ermittelt werden. Das Optimierungsnetz kann daher als ein

informationsverarbeitender Automat aufgefaßt werden, der darauf spezialisiert ist, Netzoptima zu berechnen. Seine Besonderheit liegt darin, daß dieser optimierende Netzautomat keiner expliziten Programmierung bedarf. Vielmehr ist sein Optimierungsprogramm bereits implizit in der Konstruktion des Optimierungsnetzes enthalten.

Die konstruktiv bedingte Informationsverarbeitungsfähigkeit von Optimierungsnetzen läßt sich auch noch aus zwei weiter entfernten - aber nichtsdestoweniger bemerkenswerten - Blickwinkeln würdigen. Einerseits verweist die Berechnung eines Netzoptimums durch die Realisierung eines Markenflusses auf das Konzept der Datenflußautomaten⁶⁶): Ein Optimierungsnetz kann unmittelbar als ein Datenflußautomat betrachtet werden. Seine "Daten" erstrecken sich auf die Ausprägungen der Attribute "schaltwert" und "schaltfolge" der Optimierungsmarkenkopien, die durch das Optimierungsnetz hindurchfließen. Andererseits verwirklicht die Selbststeuerung eines Markenflusses durch ein Optimierungsnetz die visionäre Idee des "rechnenden Raums", die vor längerer Zeit von KONRAD ZUSE zur Diskussion gestellt wurde⁶⁷). Die Transitionen eines Optimierungsnetzes, die mit ihren Schaltakten die Attributausprägungen der Optimierungsmarkenkopien aktualisieren, stellen dabei die "Rechenwerke" dar⁶⁸). Ihr räumlicher Zusammenhang wird auf topologische Weise durch die lokalen Nachbarschaftsverhältnisse konstituiert, die zwischen den vernetzten Stellen und Transitionen eines Optimierungsnetzes bestehen.

Zuvor wurde nur in allgemeiner Weise festgestellt, daß der "explorative" Markenfluß eines finalen Schaltprozesses zwangsläufig zur Erkenntnis des gesuchten Netzoptimums sw^* führt. Wie dies durch die Beschriftungen der Transitionen des Optimierungsnetzes im einzelnen sichergestellt ist, wurde aber noch nicht erläutert. Diese Detaillierungslücke wird nun geschlossen.

Es wird eine beliebige Schaltschritt-Transition aus dem Optimierungsnetz betrachtet, die einem Schaltschritt SS_a aus dem bewerteten Erreichbarkeitsgraphen des zugrundeliegenden des Netzmodells zugeordnet ist. Sie erhöht durch ihr Schalten die Ausprägung "wert" des Schaltwert-Attributs der Kopie einer Optimierungsmarke um den Schaltwert sw_a des Schaltschritts SS_a . Zugleich erweitert sie die Ausprägung "folge" des Schaltfolgen-Attributs der Optimierungsmarkenkopie um den Namen " SS_a " des Schaltschritts. Dabei wird die Schaltschrittbezeichnung stets an den Anfang der bis dahin erzeugten Schaltfolgen-Attributausprägung gestellt, weil der Markenfluß durch das Optimierungsnetz invers zur Schaltkantenrichtung des bewerteten Erreichbarkeitsgraphen erfolgt.

Die Kopien der Optimierungsmarke werden in jedem finalen Schaltprozeß des Optimierungsnetzes so fortgeschaltet, daß sie finale Schaltwege $wg_{0,E/e,p}$ aus dem bewerteten Erreichbarkeitsgraphen *rückwärts* durchlaufen: Die Markenkopien befinden sich zunächst unter der Ausgangsmarkierung $M_{0,OP}$ des Optimierungsnetzes auf Endstellen $s_{E/e}$, die jeweils einen *Endmarkierungsknoten* $M_{E/e}$ des Erreichbarkeitsgraphen darstellen. Danach können sie nur noch so durch das Optimierungsnetz fließen, daß sie nach endlich vielen Schaltakten auf der Ausgangsstelle s_0 ankommen müssen, die dem *Ausgangsmarkierungsknoten* M_0 des Erreichbarkeitsgraphen entspricht. Jede ausschließliche Markierung der Ausgangsstelle s_0 ist zugleich als eine *Endmarkierung* $M_{E/g,OP}$ des Optimierungsnetzes definiert. Aufgrund dieses inversen Zusammenhangs repräsentiert der Markenfluß jedes finalen Schaltprozesses $PRO_{0,E/g(e,p),OP}$ aus dem Optimierungsnetz genau einen - jedoch entgegengesetzt gerichteten - finalen Schaltweg $wg_{0,E/e,p}$ im zugrundeliegenden bewerteten Erreichbarkeitsgraphen (vice versa). Da jeder finale Schaltweg $wg_{0,E/e,p}$ des Erreichbarkeitsgraphen seinerseits genau einen finalen Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ des zugrundeliegenden Netzmodells repräsentiert, gilt ebenso: Es besteht eine bijektive Zuordnung zwischen den finalen Schaltprozessen aus dem Optimierungsnetz einerseits und den entgegengesetzt verlaufenden finalen Schaltprozessen im Netzmodell andererseits⁶⁹). Das wurde schon bei der Konstruktion des Optimierungsnetzes dargelegt.

Während eines finalen Schaltprozesses im Optimierungsnetz sorgen die oben angesprochenen Schaltschritt-Transitionen dafür, daß die Ausprägung des Attributs "schaltfolge" der Optimierungsmarkenkopie die Sequenz aller Schaltschritte wiedergibt, die den rückwärts durchlaufenen Schaltweg im Erreichbarkeitsgraphen konstituieren. Da die neu hinzukommenden Schaltschritte zur bereits erzeugten Schaltfolge jeweils *vorne* hinzugefügt werden, listet das Schaltfolgen-Attribut der Kopie der Optimierungsmarke die Schaltschritte stets in der Reihenfolge auf, wie sie auf dem finalen Schaltweg $wg_{0,E/e,p}$ des Erreichbarkeitsgraphen ausgeführt würden. Zusätzlich wird durch jedes Schalten einer Schaltschritt-Transition im Optimierungsnetz die Ausprägung "wert" des Attributs "schaltwert" der Kopie der Optimierungsmarke um den Schaltwert sw_a desjenigen Schaltschritts erhöht, den die Transition repräsentiert.

Daher weist jede Kopie der Optimierungsmarke, die am Ende eines finalen Schaltprozesses im Optimierungsnetz unter einer seiner Endmarkierungen auf der Ausgangsstelle s_0 liegt, zwei charakteristische Eigenschaften auf. Erstens gibt die Ausprägung "wert_e" ihres Schaltwert-Attributs den Schaltwert $sw(wg_{0,E/e,p}) = sw_{0,E/e,p}$ eines finalen Schaltweges $wg_{0,E/e,p}$ aus dem bewerteten Erreichbarkeitsgraphen wieder: $wert_e = sw_{0,E/e,p}$. Dieser finale Schaltwert gibt den Wert der Zielfunktion an, der durch das Ausführen des finalen Schaltweges $wg_{0,E/e,p}$ im Erreichbarkeitsgraphen realisiert wird⁷⁰). Zweitens repräsentiert der finale Schaltweg $wg_{0,E/e,p}$ des Erreichbarkeitsgraphen genau einen finalen Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ im zugrundeliegenden Netzmodell⁷¹). Die Schaltfolge SF_L dieses Schaltprozesses zeigt die Kopie der Optimierungsmarke unter der Endmarkierung des Optimierungsnetzes auf der Ausgangsstelle s_0 durch die Ausprägung "folge_e" ihres Schaltfolgen-Attributs an: $folge_e = SF_L$. Der Schaltwert $sw_{0,E/e,p}$ des finalen Schaltweges $wg_{0,E/e,p}$ ist zugleich der Schaltwert des korrespondierenden finalen Schaltprozesses $PRO_{0,E/e,p}(SF_L, MF_L)$ aus dem Netzmodell: $sw_{0,E/e,p} = sw(PRO_{0,E/e,p}(SF_L, MF_L))$. Ebenso läßt sich davon sprechen, daß der Schaltwert $sw_{0,E/e,p}$ den Wert der Zielfunktion angibt, der durch das Ausführen des finalen Schaltprozesses $PRO_{0,E/e,p}(SF_L, MF_L)$ im Netzmodell verwirklicht wird.

Auf diese Weise lassen sich jeder finale Schaltprozeß aus dem zugrundeliegenden Netzmodell und sein zugehöriger Zielfunktionswert durch die Optimierungsmarkenkopien im Optimierungsnetz rekonstruieren: Ihre Kopie, die in einem ebenso finalen Schaltprozeß durch das Optimierungsnetz geflossen ist, gibt auf der Ausgangsstelle s_0 durch ihre beiden Attributausprägungen "wert_e" und "folge_e" den Zielfunktionswert $wert_e = sw_{0,E/e,p}$ und die Schaltfolge $folge_e = SF_L$ des finalen Schaltprozesses $PRO_{0,E/e,p}(SF_L, MF_L)$ an. Daher repräsentiert jede Kopie der Optimierungsmarke genau einen finalen Schaltprozeß des Netzmodells, sobald sie am Ende eines komplementären finalen Schaltprozesses im Optimierungsnetz auf der Ausgangsstelle s_0 angelangt ist. Der Fluß von Kopien der Optimierungsmarke durch das Optimierungsnetz realisiert also die Schaltwerte und -folgen von finalen Schaltprozessen im zugrundeliegenden Netzmodell.

Aus allen denkmöglichen finalen Schaltprozessen $PRO_{0,E/e,p}(SF_L, MF_L)$ des Netzmodells gilt es nun noch die optimalen Schaltprozesse auszuwählen. Sie zeichnen sich dadurch aus, daß ihre Schaltwerte $sw_{0,E/e,p}$ den gesuchten mini- oder maximalen Zielfunktionswert sw^* darstellen. Es können mehrere optimale Schaltprozesse existieren, die jeweils denselben extremalen Schaltwert sw^* aufweisen. Die hier vorgelegte Optimierungsmethode erlaubt es, je nach Interessenlage des Modellierungsträgers die Menge aller optimalen Schaltprozesse oder auch jede ihrer nicht-leeren Teilmengen zu erzeugen.

Die Auswahl-Transitionen t_r bilden den konzeptionellen Kern der Optimierungsnetze⁷²). Sie dienen dazu, aus allen finalen Schaltprozessen eines zugrundeliegenden Netzmodells die optimalen Schaltprozesse herauszufiltern. Jede Auswahl-Transition wird dadurch aktiviert, daß auf jeder ihrer Eingangsstellen eine Kopie der Optimierungsmarke liegt. Die unterschiedlichen Ausprägungen ihres Schaltfolgen-Attributs geben die Auswahlmöglichkeiten zwischen alternativen Schaltwegen im bewerteten Erreichbarkeitsgraphen wieder. Die erste Bestimmungsgleichung einer Auswahl-Transition sorgt durch ihren Extremierungsoperator "ext(...)" zusammen mit der

Schaltregel dafür, daß von allen Kopien $\langle \text{Wert}_h, \text{Folge}_h \rangle$ der Optimierungsmarke, die sich auf ihren Eingangsstellen $s_{r,h}$ befinden, nur diejenigen fortgeschaltet werden können, deren aktuellen Schaltwerte "Wert_h" einen *lokalen* Extremwert der Zielfunktion darstellen⁷³). Mit der Lokalität des Extremums ist gemeint, daß nur die Schaltwerte "Wert_h" derjenigen Markenkopien miteinander verglichen werden, die vor dem Schalten der Auswahl-Transition auf ihren Eingangsstellen liegen. Dennoch sorgt die Konstruktion des Optimierungsnetzes dafür, daß das gesuchte *globale* Extremum sw^* der zugrundeliegenden Zielfunktion aufgefunden wird. Dabei werden die einzelnen lokalen Zielfunktionsextrema, die jeweils durch Schaltakte der Auswahl-Transitionen realisiert werden⁷⁴, durch jeden finalen Schaltprozeß zu einem globalen Extremum - dem Netzoptimum sw^* - zusammengeführt.

Diese globale Optimierung beruht auf dem eingangs erwähnten Roll back-Konzept⁷⁵). Um dieses Konzept auf bewertete Erreichbarkeitsgraphen anwenden zu können, wurden die daraus abgeleiteten Optimierungsnetze so konstruiert, daß ihre Markenflüsse entgegengesetzt zur Richtung der Schaltwege in den Erreichbarkeitsgraphen verlaufen. Daher stellen die aktuellen Ausprägungen des Schaltfolgen-Attributs der Kopien von Optimierungsmarken jeweils *Endsequenzen* von Schaltfolgen dar, die im zugrundeliegenden Netzmodell ausgeführt werden können und im Erreichbarkeitsgraphen durch entsprechende Schaltwege repräsentiert werden. Die Kopien der Optimierungsmarke fließen so durch das Optimierungsnetz, daß durch die Schaltakte der Auswahl-Transitionen sukzessiv Entscheidungen darüber getroffen werden, wie die Schaltfolgen-Endsequenzen *nach vorne*⁷⁶) verlängert werden.

Auf diese Konstruktion läßt sich BELLMAN's Optimalprinzip⁷⁷) für die Dynamische Optimierung anwenden. Es betrifft dynamische Entscheidungsprobleme, in denen mehrere Teilentscheidungen nacheinander getroffen werden müssen und jede vorausgehende die jeweils nachfolgende Teilentscheidung beeinflusst. Eine Entscheidungssequenz liegt vor, wenn Teilentscheidungen in der korrekten Reihenfolge getroffen wurden. Sie wird genau dann als vollständig bezeichnet, wenn in ihr alle Teilentscheidungen getroffen sind, die zur Lösung des betrachteten Entscheidungsproblems erforderlich sind. Falls ein Entscheidungspunkt derart existiert, daß bis zu diesem Entscheidungspunkt bereits alle Teilentscheidungen getroffen sind, aber nach diesem Entscheidungspunkt noch alle Teilentscheidungen offenstehen, so wird die Teilsequenz der noch offenen Teilentscheidungen eine Restsequenz genannt. Dabei wird vorausgesetzt, daß die Restsequenz mindestens eine noch offene Teilentscheidung umfaßt. Für jede Restsequenz besteht das Restproblem, die noch offenen Teilentscheidungen so zu treffen, als ob ihre Gesamtheit ein eigenständiges dynamisches Entscheidungsproblem darstellte.

Aufgrund des Optimalprinzips von BELLMAN gilt nun: Wenn eine vollständige Entscheidungssequenz optimal ist, dann ist auch jede ihrer Restsequenzen optimal für das jeweils zugehörige Restproblem. Die Kontraposition dieser Implikation lautet: Falls eine Restsequenz für ihr Restproblem nicht optimal ist, dann kann sie auch nicht eine Teilsequenz aus einer vollständigen optimalen Entscheidungssequenz sein. Folglich lassen sich alle nicht-optimalen Restsequenzen als Lösungen von Restproblemen eliminieren. Sie können niemals zu einer optimalen Lösung des Gesamtproblems führen. Genau diesen Sachverhalt nutzt das Roll back-Konzept aus⁷⁸). Es werden zunächst die kleinstmöglichen Restprobleme betrachtet, in denen eine jeweils *letzte* Teilentscheidung aus einer - noch unbekanntem - Entscheidungssequenz getroffen werden muß. Für diese jeweils kleinstmöglichen Restprobleme werden alle nicht-optimalen Lösungen eliminiert. Die verbleibenden Lösungen könnten zu einer vollständigen optimalen Entscheidungssequenz gehören, müssen es aber nicht. Diese potentiell-optimalen Lösungen werden in die zweitkleinsten Restprobleme eingesetzt, in denen jeweils zwei letzte Teilentscheidungen aus einer - noch unbekanntem - Entscheidungssequenz getroffen werden müssen. Da die letzte Teilentscheidung bereits durch die potentiell-optimale Lösung der vorangehend betrachteten kleinstmöglichen Restprobleme festliegt, gilt es nur noch, die zweitletzte Teilentscheidung zu treffen. Abermals werden nach Maßgabe des Optimalprinzips alle nicht-optimalen Lösungen der zweitkleinsten Restprobleme eliminiert. Dieses Entscheidungsverfahren wird so lange auf alle Restprobleme

angewandt, die sukzessiv um jeweils eine vorangehende Teilentscheidung erweitert werden, bis schließlich das letzte "Restproblem" mit dem ursprünglichen Entscheidungsproblem zusammenfällt. Nachdem auch für dieses letzte "Restproblem" alle nicht-optimalen Lösungen eliminiert worden sind, müssen die verbleibenden Lösungen notwendig die gesuchten optimalen Lösungen des ursprünglichen Entscheidungsproblems darstellen.

Genau diese retrograde Aussonderung nicht-optimaler Lösungen von Restproblemen wird im oben konstruierten Optimierungsnetz geleistet. Dabei löst jeder Schaltakt einer Auswahl-Transition ein Restproblem desjenigen dynamischen Entscheidungsproblems, die optimalen finalen Schaltprozesse des Netzmodells zu ermitteln. Es handelt sich jeweils um ein *Restproblem*, weil zwischen Kopien der Optimierungsmarke ausgewählt wird, die in ihrem Schaltfolgen-Attribut verschiedene *Endsequenzen* von - noch unbekanntem - finalen Schaltfolgen aus dem zugrundeliegenden Netzmodell ausdrücken. Jede dieser finalen Schaltfolgen entspricht einer vollständigen Entscheidungssequenz. Denn sie legt für alle Markierungen, die während eines finalen Schaltprozesses im Netzmodell erreicht werden, den dort jeweils auszuführenden Schaltschritt als Teilentscheidung fest. Da der Markenfluß im Optimierungsnetz genau *umgekehrt* zur Reihenfolge der Schaltschrittausführungen eines finalen Schaltprozesses im Netzmodell erfolgt, werden von den Auswahl-Transitionen die Restprobleme genau in derjenigen *retrograden* Abfolge bearbeitet, die oben für BELLMAN's Optimalprinzip der Dynamischen Optimierung herausgestellt wurde. Schließlich werden durch jeden Schaltakt einer Auswahl-Transition jeweils diejenigen Kopien der Optimierungsmarke nicht fortgeschaltet, deren Schaltwerte "wert" bezüglich aller anderen Markenkopien, die sich auf den Eingangsstellen derselben Auswahl-Transition befinden, nicht in der geforderten Weise minimal oder maximal sind. Dies entspricht genau dem Aussondern aller Lösungen eines Restproblems, die mit Sicherheit nicht optimal sind.

Folglich wird das Roll back-Konzept der Dynamischen Optimierung durch das sukzessive Schalten der Auswahl-Transitionen eines Optimierungsnetzes realisiert. Zugleich wird durch jeden Schaltakt einer Auswahl-Transition BELLMAN's Optimalprinzip angewendet. Darüber hinaus zeichnen sich Optimierungsnetze dadurch aus, daß sie Roll back-Konzept und BELLMAN's Optimalprinzip mit größtmöglicher Nebenläufigkeit verwirklichen. Denn die Multiplikator-Transitionen spalten den Markenfluß in einem Optimierungsnetz so auf, daß sich die identisch vervielfachten Kopien der Optimierungsmarke zeitgleich durch das Netz bewegen. Diese Teilflüsse werden lediglich durch die Auswahl-Transitionen koordiniert, die alle Markenkopien tilgen, die mit Sicherheit zu keinem Netzoptimum mehr führen können. Es unterbleiben also alle artifiziellen Sequentialisierungen. Daher stellen Optimierungsnetze einen zweiten⁷⁹⁾ Ansatzpunkt dar, um die inhärente Nebenläufigkeit des Petrinetz-Konzepts zur Geltung zu bringen. Sie bieten sich für die Implementierung auf Automatischen Informationsverarbeitungssystemen mit parallelen Architekturen an⁸⁰⁾. Auf der Basis solcher nebenläufigen Systemarchitekturen ließe sich jeder finale Schaltprozeß in einem Optimierungsnetz wesentlich effizienter⁸¹⁾ ermitteln als auf der Grundlage von sequentiellen Systemarchitekturen⁸²⁾. Denn die Teilflüsse seiner vervielfachten Kopien der Optimierungsmarke könnten durch mehrere Prozessoren zeitgleich abgearbeitet werden.

Wenn eine Kopie der Optimierungsmarke durch einen finalen Schaltprozeß im Optimierungsnetz bis zur Ausgangsstelle s_0 durchgeschaltet worden ist, dann gilt wegen der Erfüllung von BELLMAN's Optimalprinzip: Jeder finale Schaltprozeß $PRO_{0,E/e,p}(SF_L, MF_L)$ aus dem zugrundeliegenden Netzmodell, der durch den komplementären finalen Schaltprozeß des Optimierungsnetzes repräsentiert wird, stellt zugleich einen optimalen Schaltprozeß $PRO_{0,E/e,p}(SF_L^*, MF_L^*)$ dar. Daher ist die Ausprägung "wert_e" des Schaltwert-Attributs jeder Kopie der Optimierungsmarke, die auf der Ausgangsstelle s_0 angelangt ist, mit $wert_e = sw_{0,E/e,p}$ das globale Optimum sw^* der vorausgesetzten Zielfunktion. Der finale Schaltwert dieser Markenkopie gibt also immer das gesuchte Netzoptimum sw^* an. Die Ausprägung "folge_e" des Schaltfolgen-Attributs derselben Markenkopie kennzeichnet zugleich die Schaltfolge SF_L^* , die zum sw^* -optimalen Schaltprozeß $PRO_{0,E/e,p}(SF_L^*, MF_L^*)$ des Netzmodells gehört.

Allerdings repräsentiert die Kopie der Optimierungsmarke, die am Ende eines finalen Schaltprozesses die Ausgangsstelle s_0 des Optimierungsnetzes belegt, nur genau einen optimalen Schaltprozeß des Netzmodells. Es wurde jedoch eingangs darauf hingewiesen, daß in einem Netzmodell durchaus mehrere verschiedene Schaltprozesse möglich sein können, die alle denselben optimalen Schaltwert sw^* besitzen. Daher bleibt noch zu untersuchen, ob aus dem Optimierungsnetz auch die Gesamtheit aller sw^* -optimalen Schaltprozesse des Netzmodells gewonnen werden können⁸³). Dies bereitet grundsätzlich keine Schwierigkeiten. Dazu braucht lediglich der Erreichbarkeitsgraph des Optimierungsnetzes untersucht zu werden⁸⁴). In ihm können - wie in jedem Erreichbarkeitsgraphen für finale Netze - *alle* finalen Schaltwege ermittelt werden, die im Knoten der Ausgangsmarkierung des Optimierungsnetzes starten und im Knoten einer der erwünschten Endmarkierungen des Optimierungsnetzes enden. Jeder dieser finalen Schaltwege aus dem Erreichbarkeitsgraphen des Optimierungsnetzes legt genau einen finalen Schaltprozeß im Optimierungsnetz fest. Jeder finale Schaltprozeß repräsentiert seinerseits durch die Kopie der Optimierungsmarke, die an seinem Ende auf der Ausgangsstelle s_0 des Optimierungsnetzes abgelegt wird, einen sw^* -optimalen Schaltprozeß aus dem zugrundeliegenden Netzmodell. Folglich gilt auch: Die Gesamtheit aller finalen Schaltwege, die im Erreichbarkeitsgraphen des Optimierungsnetzes existieren, determiniert eindeutig die Gesamtheit aller sw^* -optimalen Schaltprozesse des Netzmodells. Daher kann durch die Erreichbarkeitsanalyse des Optimierungsnetzes die Menge aller sw^* -optimalen Schaltprozesse eines Netzmodells vollständig bestimmt werden.

Zum Abschluß erfolgt noch ein Ausblick auf befruchtende Impulse, die vom Konzept der Optimierungsnetze auf andere Probleme ausstrahlen könnten, die selbst nicht mehr zum Bereich der Petrinetze gehören. Ausgangspunkt ist die oben skizzierte Möglichkeit, die Ermittlung von Netzoptima sw^* auf Automatischen Informationsverarbeitungssystemen mit parallelen Architekturen zu implementieren. Aus dieser Perspektive bietet das Petrinetz-Konzept einen interessanten Ansatz, um konzeptfremde Optimierungsprobleme in nebenläufiger Weise zu bewältigen⁸⁵). Dazu brauchen die Optimierungsprobleme lediglich in der früher spezifizierten Art als Netzmodelle formuliert zu werden. Sofern diese netzartigen Optimierungsmodelle die oben angeführten drei Voraussetzungen erfüllen, können daraus die zugehörigen Optimierungsnetze konstruiert werden. Die nebenläufig berechneten Netzoptima stellen dann per constructionem zugleich die gesuchten Optima der ursprünglichen Optimierungsprobleme dar. Daher lassen sich Netzmodelle und die daraus abgeleiteten Optimierungsnetze einsetzen, um die inhärente Nebenläufigkeit von Optimierungsproblemen zur möglichst effizienten Ermittlung optimaler Problemlösungen auszunutzen⁸⁶). Der Anwendungsbereich dieses nebenläufigen Optimierungskonzepts ist recht weit ausgedehnt, weil in den Netzmodellen das Ausdrucksvermögen Synthetischer Netze vollständig ausgeschöpft werden kann. Es umfaßt sowohl die gesamte Prädikatenlogik als auch algebraische Signaturen.

Beispielsweise können Optimierungsnetze benutzt werden, um die Gesamtheit aller optimalen Lösungen von Entscheidungsbäumen⁸⁷) zu ermitteln. Dazu brauchen Entscheidungsbäume lediglich als Optimierungsnetze reformuliert zu werden. Abb. 101 bis 103 auf den nächsten Seiten skizzieren, wie eine solche Transformation grundsätzlich geleistet werden kann⁸⁸). Jeder finale Schaltprozeß im Optimierungsnetz liefert dann eine optimale Entscheidungssequenz. Durch die Untersuchung des Erreichbarkeitsgraphen des Optimierungsnetzes lassen sich wiederum alle finalen Schaltwege bestimmen. Sie drücken alle optimalen Entscheidungssequenzen aus, die für den analysierten Entscheidungsbaum definiert sind.

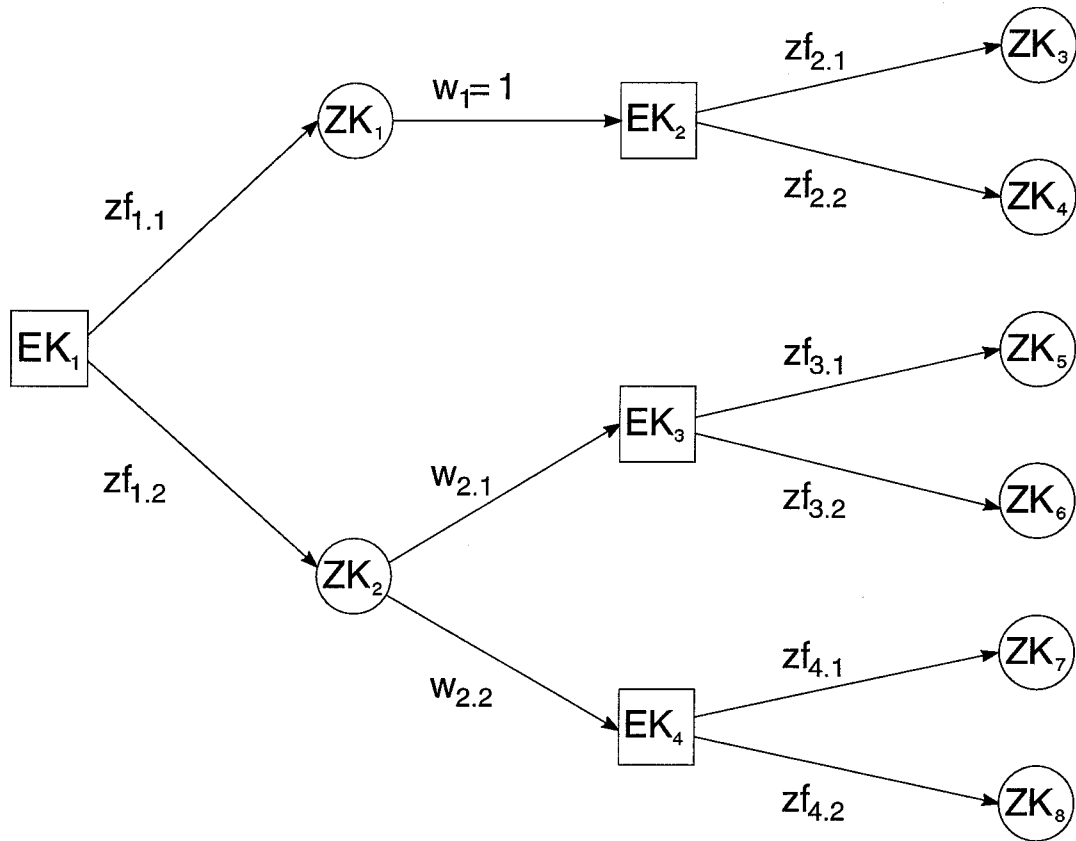


Abb. 101: ursprünglicher Entscheidungsbaum

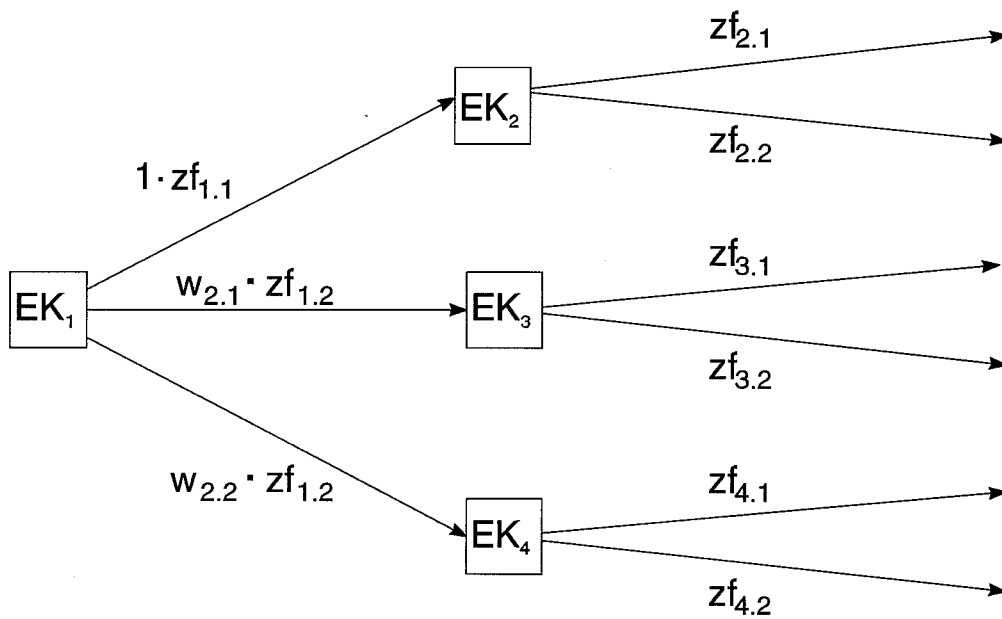


Abb. 102: Derivat des Entscheidungsbaums

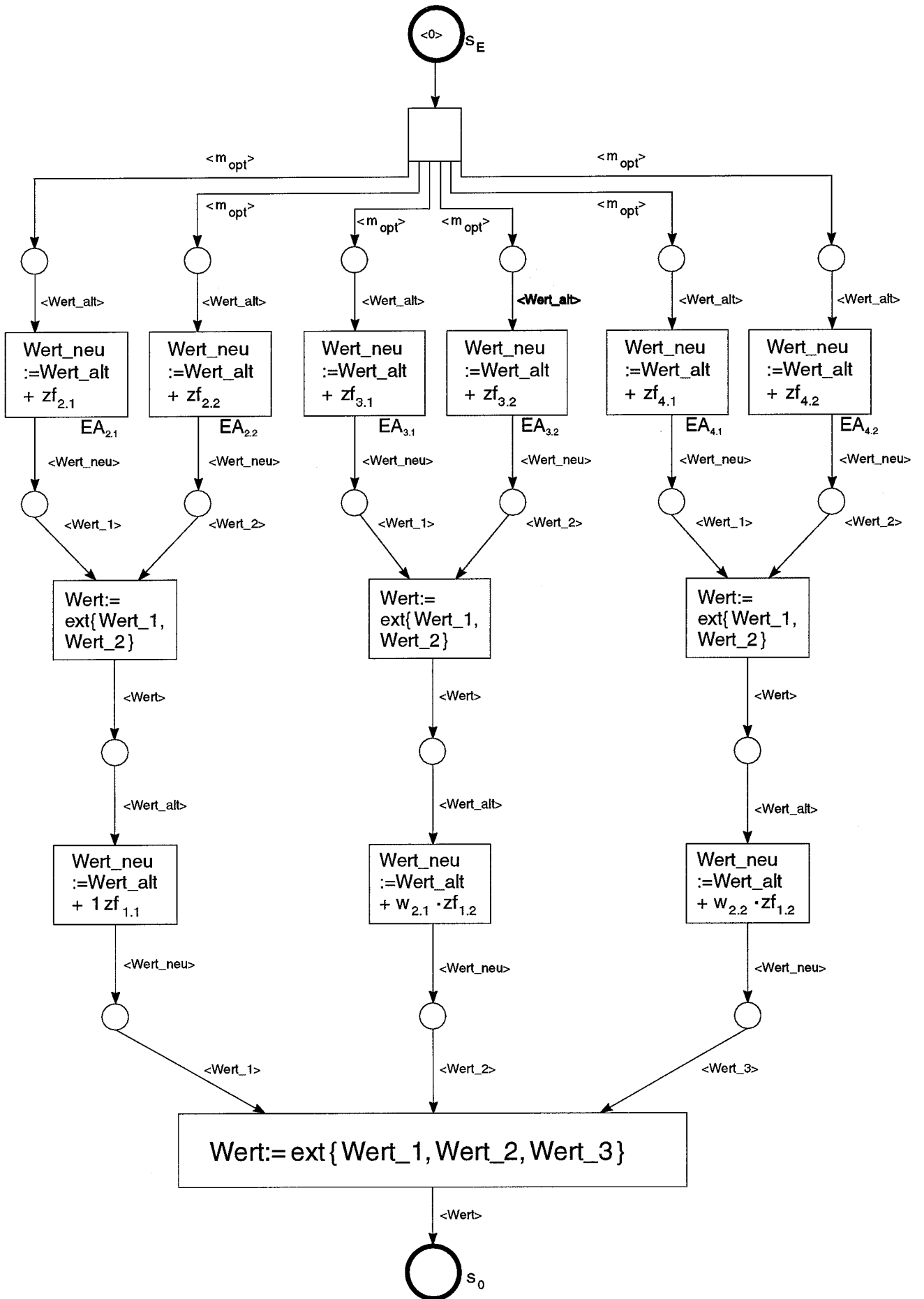


Abb. 103: Optimierungsnetz für den Entscheidungsbaum aus Abb. 101

Allerdings wäre zu klären, ob sich dieser Ansatz gegenüber alternativen, bereits etablierten Auswertungskonzepten für Entscheidungsbäume als vorteilhaft erweist. Dagegen spricht der Aufwand für die Reformulierung von Entscheidungsbäumen als Optimierungsnetze. Zugunsten der Optimierungsnetze könnte die Möglichkeit angeführt werden, mit ihrer Hilfe die optimalen Lösungen nebenläufig und vollständig zu erzeugen. Insbesondere wäre es interessant zu untersuchen, ob das nebenläufige Erzeugen optimaler Lösungen die Effizienzvorteile paralleler Informationsverarbeitung so stark zur Geltung zu bringen vermag, daß die Effizienzeinbußen infolge der zusätzlichen Netzformulierung überkompensiert werden. Hinzu käme das didaktische Argument, daß der Markenfluß durch ein Optimierungsnetz imstande ist, das Roll back-Konzept der retrograden Eliminierung nicht-optimaler Lösungen von Restproblemen besonders anschaulich vor Augen zu führen⁸⁹⁾. Eine detailliertere Klärung dieser Aspekte liegt jedoch außerhalb des Erkenntnisinteresses dieser Arbeit⁹⁰⁾.

Eine weitere konzeptionelle Verwandtschaft besteht zwischen Optimierungs- und Kontrakt-netzen. Das Konzept der Kontraktnetze wurde bereits im Zusammenhang mit der opportunistischen Prozeßkoordinierung vorgestellt⁹¹⁾. Sie wurden dort als ein Beitrag skizziert, die Idee des verteilten Problemlösens durch teilautonome Agenten zu konkretisieren. Es handelt sich um Netze, deren Knoten teilautonom operierende Agenten darstellen. Diese Netzknoten verhalten sich insofern "intelligent", als sie Teilaufgaben ausführen und ihre Aktivitäten mit Verrechnungspreisen bewerten können. Darüber hinaus sind die Agenten in der Lage, die Ausführungen von Teilaufgaben anderen Agenten anzubieten oder bei anderen Agenten nachzufragen. Die Agenten verhandeln miteinander über die Ausführungen von Teilaufgaben, indem sie über die Kanten des Netzwerks entsprechende Nachrichten über Ausführungsangebote bzw. -nachfragen austauschen. Verhandlungsprotokolle sollen den Nachrichtenaustausch so koordinieren, daß das arbeitsteilige Zusammenwirken der Agenten insgesamt die jeweils zugrundeliegende Gesamtaufgabe erfüllt.

Ein Optimierungsnetz läßt sich als ein einfacher, degenerierter, aber leistungsfähiger⁹²⁾ Sonderfall des Kontraktnetzkonzepts auffassen. Dabei stellen die Transitionen die teilautonomen Agenten dar. Sie erfüllen simple Teilaufgaben, die sich darauf beschränken, die Ausprägungen der Schaltwert- und Schaltfolgenattribute von Kopien der Optimierungsmarke zu berechnen. Die Auswahl-Transitionen des Optimierungsnetzes wählen wie in einem Kontraktnetz aus mehreren Angeboten, die sich als Kopien der Optimierungsmarke mit - in der Regel unterschiedlichen - Attributausprägungen auf den Eingangsstellen der Auswahl-Transitionen befinden. Ein jeweils lokal bestes Angebot wird durch den Extremierungsoperator aus der Transitionsbeschriftung ausgewählt⁹³⁾. Dabei verhalten sich die Kopien der Optimierungsmarke, die durch das Optimierungsnetz fließen, wie die Nachrichten, die in einem Kontraktnetz zwischen dessen Netzknoten ausgetauscht werden. Netztopologie und Schaltregel des Optimierungsnetzes koordinieren den Markenfluß in der Art eines Verhandlungsprotokolls für Kontraktnetze. Sie sorgen gemeinsam dafür, daß die zugrundeliegende Optimierungsaufgabe durch das arbeitsteilige Zusammenwirken aller Transitionen des Netzes erfüllt wird.

Allerdings handelt es sich bei Optimierungsnetzen nur um degenerierte Kontraktnetze. Denn der kontraktnetztypische Verhandlungsaspekt kommt nicht zur Geltung⁹⁴⁾. Statt dessen werden die Transitionen zur Ausführung ihrer Teilaufgaben nach dem Prinzip eines Datenflußautomaten angestoßen⁹⁵⁾. Ein echtes Kontraktnetz läge dagegen vor, wenn die Transitionen nicht unmittelbar mit den Agenten eines Kontraktnetzes identifiziert würden. Statt dessen könnten die Agenten als teilautonome Informationsverarbeitungssysteme realisiert werden, welche das Schalten beliebiger Transitionen auszuführen vermögen. Diese Schaltagenten könnten dann darüber verhandeln, die jeweils aktivierten Transitionen eines Optimierungsnetzes zu schalten. Auf eine weitere Konkretisierung dieses Ansatzes wird hier jedoch verzichtet, weil es sich dabei "nur" noch um eine Ausarbeitung seiner informationstechnischen Implementierung handelt⁹⁶⁾.

Anmerkungen zum Kapitel:

- 1) Es wird also vorausgesetzt, daß die untersuchten Netze jeweils eine nicht-leere Menge ME erwünschter Endmarkierungen besitzen. Darüber hinaus wird unterstellt, daß aus einer früheren Analysephase die Finalität des Netzes bereits bekannt ist. Es muß also mindestens ein Schaltprozeß zulässig sein, der in der Ausgangsmarkierung des Netzes startet und nach dem Ausführen endlich vieler Schaltschritte eine der erwünschten Endmarkierungen erreicht. Wäre mindestens eine der beiden vorgenannten Prämissen verletzt, läge kein finales Netz vor. Dann wäre es in sich widersprüchlich, dennoch den optimalen Schaltwert sw^* seiner - tatsächlich nicht existenten - finalen Schaltprozesse ermitteln zu wollen.
- 2) Daher wird diese Methode auch als Methode zur Netzoptimierung, als netzoptimierende Methode oder kurz als Optimierungsmethode angesprochen.
- 3) Die Fokussierung auf monodimensionale Formalzielsysteme erfolgt hier aus zwei Gründen. Erstens wird die nachfolgende Argumentation von der aufwendigeren, vektorbezogenen Notation zugunsten einer rein skalaren Darstellungsweise entlastet. Hierin liegt keine grundsätzliche Einschränkung des Geltungsbereichs der Optimierungsmethode, sondern lediglich eine Vereinfachung der Methodendarstellung. Zweitens läßt sich die Auswertungsmethode für die Eigenschaft der sw^* -Optimalität mit den bisher erarbeiteten Netzkonstrukten nur dann universell anwenden, falls keine mehrdimensionalen Formalzielsysteme zugelassen werden. Dies liegt einerseits daran, daß das Dominanzkonzept der Vektoroptimierung im allgemeinen mehrdeutige Netzoptima gestattet. Dies wurde bereits an früherer Stelle erläutert. Dort wurde auch gezeigt, daß die Eindeutigkeit des optimalen Schaltwerts sw^* für ein Netz gewährleistet ist, falls nur ein Formalziel beachtet zu werden braucht. Andererseits kann die nachfolgend vorgestellte Auswertungsmethode nur dann angewendet werden, wenn das gesuchte Netzoptimum sw^* eindeutig ist. Denn dieses Optimum wird sukzessiv dadurch ermittelt, daß jeweils *genau eine* Markenkopie mit dem vorläufig bestmöglichen Zielfunktionswert über die Ausgangskante einer Stelle fortgeschaltet wird. Bei mehrdimensionalen Formalzielsystemen müßte hingegen berücksichtigt werden, daß aufgrund des Dominanzkonzepts der Vektoroptimierung unter Umständen mehrere vorläufig bestmögliche Markenkopien fortzuschalten sind. Ob dies tatsächlich der Fall ist und um wie viele Markenkopien es sich dann jeweils handelt, läßt sich jedoch nicht allgemein festlegen. Daher müßten spezielle Netzkonstrukte eingeführt werden, die gestatten, eine *variable Anzahl* von Markenkopien über die Ausgangskante einer Stelle abfließen zu lassen. Dies ist aber mit den bisher entfaltenen Ausdrucksmitteln des Petrinetz-Konzepts noch nicht möglich. Denn die Koeffizienten in den Multimengen, mit deren Hilfe die Kantengewichte von Synthetischen Netzen festgelegt werden, stellen immer *Konstanten* dar. Auch bei den Kantengewichten von Stelle/Transition-Netzen handelt es sich um Konstanten. Allerdings wird später eine Erweiterung von Synthetischen Netzen eingeführt. Sie ermöglicht es, einer Kante eine Multimenge mit variabler Extension zuzuordnen, deren Koeffizienten Variablen darstellen können. Dadurch könnte über eine Kante genau diejenige variable Anzahl von Markenkopien fließen, die notwendig wäre, um die nachfolgend erläuterte Auswertungsmethode auch bei multidimensionalen Formalzielsystemen anwenden zu können. Daher wird diese Auswertungsmethode durch die oben erfolgte Voraussetzung eindimensionaler Formalzielsysteme zwar erheblich vereinfacht, jedoch in ihrem Geltungsbereich nicht grundsätzlich eingeschränkt.
- 4) Schaltwerte wurden im Zusammenhang mit den Eigenschaften der Netzfinalität und -optimalität eingeführt. Falls in einem finalen Netzmodell einzelne Transitionen t_n keine expliziten Schaltwerte besitzen, wird ihnen nachträglich jeweils der neutrale Schaltwert $sw(t_n, c_n) = 0$ zugeordnet.
- 5) Andernfalls könnte überhaupt kein finales Netzmodell vorliegen.
- 6) Da ein monodimensionales Formalzielsystem vorausgesetzt wird, degeneriert der früher definierte Schaltwertvektor $\underline{sw}(t_n, c_n)$ für jede Transition t_n eines Netzes und deren Schaltfarbe c_n zum Einkomponentenvektor $\underline{sw}(t_n, c_n) = (sw_1(t_n, c_n))$. Er läßt sich vereinfacht als Schaltwertskalar $sw(t_n, c_n)$ notieren. Davon wird fortan ausgegangen. Daher werden auch die Schaltwerte $\underline{sw}(SS_a)$ von Schaltschritten SS_a und die Schaltwerte \underline{sw} von Netzen vereinfacht als Skalare $sw(SS_a)$ bzw. sw notiert. Folglich läßt sich auch die früher in vektorieller Weise definierte Eigenschaft der sw^* -Optimalität eines Netzes bei monodimensionalem Formalzielsystem als vereinfachte skalare sw^* -Optimalität behandeln.
- 7) Darüber hinaus lassen sich alle anderen dynamischen Netzeigenschaften in konventionellen betriebswirtschaftlichen Entscheidungsmodellen nicht wiederfinden. Sie erscheinen daher aus betriebswirtschaftlicher Perspektive - zumindest prima facie - als weniger interessant.
- 8) Vgl. zur betriebswirtschaftlichen Relevanz der Entscheidungsbaumtechnik WITTMANN (1959), S. 188f. u. 201ff. (in ersten Ansätzen); MAGEE (1964a), S. 127ff.; MAGEE (1964b), S. 79ff., insbesondere S. 84ff.; HESPOS (1965), S. B-248ff.; ACKOFF (1970), S. 244ff.; LAUX (1971), S. 39ff. u. 121ff.; HAX, H. (1972), S. 327f.; RAIFFA (1973), S. 25ff. u. 290ff.; KERN, W. (1974), S. 354ff.; HAX, H. (1974), S. 83ff.; KERN, W. (1977), S. 193ff.; EISENFÜHR (1978), S. 442; RIVETT (1980), S. 81ff.; SCHIEMENZ (1982), S. 90ff.; HAX, H. (1985), S. 170f. u. 178ff.; ZELEWSKI (1986a), S. 674ff.; FRANKE, G. (1988), S. 221ff.; BLOHM (1988), S. 250ff.; BAMBERG (1989), S. 215ff.; SIEBEN (1990), S.

38ff., 83ff. u. 89ff.; MARTIN,R.K. (1990), S. 128; LAUX (1991), S. 256ff. u. 265ff.; PERRIDON (1991), S. 122ff.; GÖTZE,U. (1991), S. 328ff.

9) Vgl. zu Überblicken über die Dynamische Optimierungsrechnung - oder kurz: Dynamische Optimierung - NEUMANN,K. (1977), S. 21ff.; KERN,W. (1987), S. 66ff.; STEPAN (1988), S. 175ff.; FAHRION (1989), S. 511ff.; ELLINGER (1990a), S. 241ff.; DOMSCHKE (1990), S. 126ff.; MARTIN,R.K. (1990), S. 128ff.

10) Näheres zum Roll back-Konzept und seiner Verankerung im Konzept der Dynamischen Optimierung findet sich in den Quellen, die in Anmk. 75 angeführt werden.

11) Erstens wurden Erreichbarkeitsnetze als Stelle/Transition-Netze eingeführt. Dagegen werden jetzt aus den untersuchten Erreichbarkeitsgraphen Optimierungsnetze abgeleitet, die Höhere Netze - z.B. Synthetische Netze - darstellen. Denn das Ausdruckspotential Höherer Netze ist erforderlich, um ein quantitatives Markenattribut zu formulieren, das hinsichtlich seiner Ausprägungen verändert und verglichen werden kann. Zweitens entspricht die Kantenrichtung in den Optimierungsnetzen nicht mehr der Richtung der Schaltkanten der untersuchten Erreichbarkeitsgraphen, sondern weist in die entgegengesetzte Richtung. Die Kantenrichtung der Erreichbarkeitsnetze stimmt hingegen per constructionem mit der Kantenrichtung der Erreichbarkeitsgraphen überein. Drittens besaßen die Erreichbarkeitsnetze genau so viele Stellen und Transitionen, wie ihre zugrundeliegenden Erreichbarkeitsgraphen Markierungsknoten bzw. Schaltkanten umfassen. Für die Optimierungsnetze gilt dies nur noch in dem besonderen Fall, daß im zugrundeliegenden Erreichbarkeitsgraphen jeder Markierungsknoten höchstens eine Ein- und höchstens eine Ausgangskante besitzt. Andernfalls weisen Optimierungsnetze per constructionem eine größere Stellen- und Transitionenanzahl als Erreichbarkeitsnetze auf.

12) Der Erreichbarkeitsgraph eines Netzmodells leistet den entscheidenden konzeptionellen Beitrag zur Optimierung des zugrundeliegenden Netzmodells. Denn nur er gestattet es, eine Brücke zwischen zwei scheinbar antagonistischen Aspekten zu schlagen: Einerseits zeichnen sich alle Netze durch ihre charakteristische *Lokalität* aus. Andererseits stellt die Ermittlung optimaler Problemlösungen eine *global* definierte Aufgabe dar. Denn ein Optimum ist nur in bezug auf die Menge aller zulässigen Problemlösungen definiert. Auf den ersten Blick scheint es unmöglich zu sein, in Netzen mit lokal definierter Schaltregel dennoch global definierte Netzoptima zu bestimmen. Hinzu kommt der scheinbare Widerspruch zwischen der *kausalen* Schaltcharakteristik von Netzen und der *finalen* Eigenart aller Optimierungsansprüche. Trotz dieser auffälligen Diskrepanzen ist es keineswegs ausgeschlossen, in Netzmodellen optimale Problemlösungen zu ermitteln. Dabei dient der Erreichbarkeitsgraph des Netzmodells als Vermittler zwischen lokal-kausaler Netzcharakteristik und global-finaler Optimalität. Er nimmt in sich die Lokalität der Schaltregel von Netzen auf, indem jeder Schaltschritt durch eine Schaltkante repräsentiert wird. Zugleich wird der globale Zusammenhang zwischen allen zulässigen Schaltschritten durch die topologische Struktur des Erreichbarkeitsgraphen wiedergegeben. Diese topologische Struktur ist der Zusammenhang aller Schaltkanten und Markierungsknoten im Erreichbarkeitsgraphen. Darüber hinaus wird die kausale Schaltcharakteristik des zugrundeliegenden Netzmodells durch die Richtung der Schaltkanten im Erreichbarkeitsgraphen erfaßt. Sie entspricht der kausalen Folgebeziehung zwischen der Referenzmarkierung, unter der ein Schaltschritt aktiviert ist, und der Folgemarkierung, die durch das Ausführen des Schaltschritts hervorgebracht wird. Die Finalität der Ermittlung optimaler Problemlösungen wird dadurch eingebunden, daß sich der Erreichbarkeitsgraph - entgegengesetzt zur Richtung seiner Schaltkanten - rückwärts durchschreiten läßt. Dabei wird in einer seiner erwünschten Endmarkierungsknoten gestartet und bis hin zum Ausgangsmarkierungsknoten zurückgeschritten. Dies wird später ausgenutzt, um Kopien einer Optimierungsmarke finale Schaltwege rückwärts durchlaufen zu lassen. Daher vereinigt der Erreichbarkeitsgraph eines Netzmodells alle vier vorgenannten Aspekte in sich: Er umgreift sowohl lokal definierte Schaltakte als auch den global definierten Zusammenhang aller zulässigen Schaltprozesse. Er drückt einerseits kausale Folgebeziehungen von Markierungsübergängen aus und läßt andererseits finale Auswertungen durch inverse Markenflüsse zu.

Der Erreichbarkeitsgraph eines Netzmodells reicht also grundsätzlich aus, um globale Netzoptimierungen vorzunehmen. Die Optimierungen selbst können entweder im konventionellen graphentheoretischen Rahmen oder aber wiederum in der Gestalt eines Netzes erfolgen. Im ersten Fall läßt sich das breite Spektrum graphentheoretischer Analysealgorithmen nutzen, um einen optimalen Schaltweg im Erreichbarkeitsgraphen zu ermitteln. Für diesen Zweck kommen vor allem Algorithmen in Betracht, mit deren Hilfe sich maximale oder minimale Flüsse in bewerteten Graphen berechnen lassen. Im zweiten Fall wird der Erreichbarkeitsgraph des zugrundeliegenden Netzmodells in ein eigenständiges Netz transformiert, in dem ein optimaler Schaltprozeß erzeugt wird. Diese letztgenannte Option wird im folgenden durch die Konstruktion eines Optimierungsnetzes detailliert ausgeführt. Die erstgenannte Option, auf graphentheoretische Analysealgorithmen zurückzugreifen, wird dagegen nicht weiter behandelt. Sie würde keine neuartigen Erkenntnisse vermitteln. Denn seitens der konventionellen Graphentheorie wird für die Berechnung extremaler Flüsse (z.B. kürzester Wege) in bewerteten Graphen eine Reihe von ausgereiften, präzise dokumentierten Algorithmen angeboten. Vgl. z.B. NEUMANN,K. (1975b), S. 76ff. u. 112ff.; CUNNINGHAM,W. (1976), S. 113ff.; BRADLEY,S. (1977), S. 345ff.; HASTINGS (1978), S. 5ff.; HÄSSIG (1979), S. 23ff., 58ff., 64ff. u. 95ff.; HAMACHER,H. (1980a), S. 159ff.; HAMACHER,H. (1980b), S. 31ff.; HANSEN,P. (1980), S. 151ff.; HURA (1982d), S. 1033f.; WHITEHOUSE (1982), S. 14.5.5ff. i.V.m. S. 14.5.1ff.; LOUI (1983), S. 673ff.; MEHLHORN (1984), S. 38ff. u. 59ff.; GLOVER (1985), S. 1106ff., insbesondere S. 1112ff.; ARONSON (1986), S. 448ff., insbesondere S.

454ff.; TARDOS (1986), S. 365ff.; NEUMANN, K. (1989a), S. 63ff. u. 95ff.; GOLDFARB (1990), S. 624ff.; SCHANNATH (1991), S. 10ff., insbesondere S. 63ff.

13) Dieser Ansatz ließe sich auch ohne Schwierigkeiten auf mehrdimensionale Formalzielsysteme übertragen, in denen es jeweils G verschiedene Extremalziele mit $G \in \mathcal{N}_+$ und $G \geq 2$ zu berücksichtigen gilt. Dann müßten die bewertenden Beschriftungsfunktionen bk_b zu $bk_b: KA_{RG} \rightarrow (SSM \times \mathcal{R}^G)$ verallgemeinert werden. Entsprechend würden die Kantenanschriften $bk_b(ka_{r,a,t}) = (SS_a, \underline{sw}_a)$ jeder Schaltkante nunmehr das Paar aus ihrem Schaltschritt SS_a und dessen Schaltwertvektor $\underline{sw}_a = (sw_1(SS_a), \dots, sw_G(SS_a))$ zuordnen. Da hier jedoch nur monodimensionale Formalzielsysteme mit genau einem Extremalziel interessieren, kann auf diese Verallgemeinerung verzichtet werden.

14) Die Berechnung der Schaltwerte $\underline{sw}(SS_a)$ von Schaltschritten SS_a wurde an früherer Stelle festgelegt. Der dort definierte Schaltwertvektor vereinfacht sich hier zum Schaltwertskealar $sw(SS_a)$, weil nur monodimensionale Formalzielsysteme betrachtet werden. Die nachfolgende Definition gilt analog für Stelle/Transition-Netze. Dazu braucht lediglich die Implikationsprämisse durch den Ausdruck $SS_a = \{t_{n(w)}; w = 1, \dots, W_a\}$ ersetzt zu werden.

15) Gleiches gilt für die nachfolgend beschriebene Anreicherung des Informationsgehalts dieser Markenkopien um das Wissen über Schaltfolgen.

16) Dies träfe insbesondere zu, wenn Synthetische Netze gewählt würden. Vgl. dazu die formale Definition dieser Netzklasse. Vgl. auch die umfangreiche Netzdefinition, die schon das einfache Beispiel einer Autovermietung bereitete.

17) Die Konstruktionsregeln sind in sequentieller Weise anzuwenden, um eine korrekte Transformation des Erreichbarkeitsgraphen RG_b in ein Optimierungsnetz OP_{RG} zu gewährleisten.

18) Daher repräsentiert jede Stelle s_r des Optimierungsnetzes genau einen Markierungsknoten M_r aus dem transformierten Erreichbarkeitsgraphen und zugleich die erreichbare Markierung M_r des zugrundeliegenden Netzmodells.

19) Da ein endlicher Erreichbarkeitsgraph RG_b vorausgesetzt wurde, gelten $\#(KN_{RG}) \in \mathcal{N}_+$ und $(\#(KN_{RG}) - 1) \in \mathcal{N}_0$.

20) Falls wegen $E=1$ nur genau eine Endstelle $s_{E/e}$ vorgesehen ist, läßt sie sich auch vereinfacht als s_E notieren.

21) Die Endstellen $s_{E/e}$ für erwünschte Endmarkierungen $M_{E/e}$ lassen sich in der Stellenmenge S_{OP} als Stellen s_r mit $r=E/e$ erfassen.

22) Der Ausdruck "wert" stellt einen Basisterm dar. Falls der Basisterm als Variable konkretisiert werden soll, so wird dies wieder - in Anlehnung an die Notation der Programmiersprache PROLOG - durch die Schreibweise "Wert" angezeigt.

23) Beim Ausdruck "folge" handelt es sich wiederum um einen Basisterm. Wenn er eine Variable darstellen soll, so wird er als "Folge" notiert.

24) Die Objektmenge "LISTE" ist in der Programmiersprache Turbo-PROLOG, die für die Implementierung Synthetischer Netze vorgesehen wurde, nicht als Objektmenge originär definiert. Vielmehr müßte sie dort als Iteration derjenigen Objektmenge gebildet werden, die für alle Listenelemente vorgesehen ist. Da hier aber nur Kanal/Instanz-Netze betrachtet werden, läßt sich von solchen implementierungstechnischen Details absehen.

25) Diese Notation wird bevorzugt, um eine Implementierung von Optimierungsnetzen mit Hilfe der Programmiersprache PROLOG zu unterstützen. Denn listenartige Informationsobjekte der Gestalt "[...]" lassen sich in dieser Sprache besonders einfach handhaben. Auf Darstellungen der PROLOG-Listenverarbeitung wurde bereits anlässlich der Listennotation von Multimengen hingewiesen. Auf weiterführende Besonderheiten der PROLOG-orientierten Listenverarbeitung wird nur noch zurückgekommen, sofern es zum Verständnis von Optimierungsnetzen hilfreich erscheint.

26) In einem Synthetischen Netz würde sogar "<wert,folge>" für eine Kantengewichtskomponente "(1,prä_u(optimierungsmarke(wert,folge)))" oder für ein Fakt "fakt₁(1,prä_u(optimierungsmarke(wert,folge)))" stehen. In den hier konstruierten Kanal/Instanz-Netzen wird aber auf die formale Netzpräzisierung durch Prädikatssymbole, welche die Stellen beschriften, und Faktenmengen, die zur Definition von Netzmarkierungen dienen, verzichtet. Dies geschieht im oben angedeuteten Interesse einer übersichtlicheren Darstellung derjenigen Aspekte, welche die Konstruktionsidee der Optimierungsnetze im wesentlichen prägen.

27) Die Markierungen des Optimierungsnetzes OP_{RG} werden durch die zusätzlichen Subskript-Komponenten "OP" gekennzeichnet. Damit werden sie deutlich unterschieden von den Markierungen des jeweils zugrundeliegenden Netzmodells. Denn die Markierungen des Netzmodells finden sich im Optimierungsnetz - vermittelt durch den bewerteten Erreichbarkeitsgraphen RG_b - als Stellen wieder. Bei den Markierungen dieser Stellen, die sich ausschließlich auf Kopien der einen Optimierungsmarke erstrecken, handelt es sich um eine Markierungsklasse sui generis. Sie steht mit den Markierungen des zugrundeliegenden Netzmodells in keinem unmittelbaren Zusammenhang.

28) Das Schaltwert-Attribut wird mit dem Startwert der Zielfunktion des vorausgesetzten Extremierungsziels initialisiert. Dabei kann es sich grundsätzlich um jeden fixen Beitrag zur Zielfunktion handeln, der nicht vom Schalten der Transitionen im zugrundeliegenden Netzmodell abhängt. Beispielsweise lassen sich auf diese Weise Fixkosten berücksichtigen, falls ein Kostenminimierungsziel betrachtet wird. Solche fixen Zielfunktionsbeiträge sind aber für Netzmodelle niemals entscheidungsrelevant. Denn in Netzmodellen werden alle Entscheidungen als Auswahlkonflikte abgebildet, die das Schalten von Transitionen betreffen. Da die fixen Zielfunktionsbeiträge per definitionem vom Transitionsschalten unabhängig sind, können sie sich in keiner Weise auf eine modellierte Entscheidungsoption auswirken. Daher besteht die einzige Funktion fixer, aber entscheidungsirrelevanter Zielfunktionsbeiträge darin, das absolute Niveau der Zielfunktion in ihrem Optimum zu beeinflussen. Dieser Aspekt kann in der Initialisierung des Schaltwert-Attributs einer Kopie der Optimierungsmarke durch einen entsprechenden fixen Zielfunktionsbeitrag berücksichtigt werden. Statt dessen läßt er sich aufgrund seiner Entscheidungsirrelevanz aber auch ausklammern. Der Verf. beschreitet den letztgenannten Weg und initialisiert daher das Schaltwert-Attribut mit dem Startwert "0".

29) Der Ausdruck "[]" ist die leere Liste.

30) Bei der Verwendung von Multimengen in binärer Mengennotation müßten dagegen die Werte der Markierungsfunktion als $M_{0,OP}(s_{E/e}) = \{ \text{fakt}_0(1, \text{prä}_{E/e}(\text{optimierungsmarke}(0, []))) \}$ und $M_{0,OP}(s_e) = \emptyset$ dargestellt werden. Dabei ist $\text{Prä}_{E/e}$ der Name desjenigen Prädikatssymbols, das der Endstelle $s_{E/e}$ in einem Synthetischen Netz zugeordnet wäre. Aber es wurde bereits dargelegt, daß Optimierungsnetze hier nicht als präzise Synthetische Netze, sondern als vereinfachte Kanal/Instanz-Netze präsentiert werden. Folglich wird auch auf die o.a. präzise Multimengen-Notation verzichtet.

31) Es wurde schon in einer der voranstehenden Anmerkungen verdeutlicht, daß die Markierungen des Optimierungsnetzes in keinem unmittelbaren Zusammenhang mit den Markierungen des jeweils zugrundeliegenden Netzmodells stehen. Daher stimmen auch die erwünschten Endmarkierungen $M_{E,g,OP}$ des Optimierungsnetzes nicht mit den erwünschten Endmarkierungen $M_{E/e}$ des Netzmodells überein. Folglich mußte vom differenzierenden Index "e" der letztgenannten Endmarkierungen zum neuen differenzierenden Index "g" für die Endmarkierungen des Optimierungsnetzes übergegangen werden.

32) Bei Verwendung von Multimengen in binärer Mengennotation wären die Werte der Markierungsfunktionen als $M_{E/g,OP}(s_0) = \{ \text{fakt}_{E/g}(1, \text{prä}_0(\text{optimierungsmarke}(\text{wert}_e, \text{folge}_e))) \}$ und $M_{E/g,OP}(s_e) = \emptyset$ darzustellen.

33) Die beiden Bestimmungsgleichungen werden als konjunktiv verknüpft vorausgesetzt.

34) Da die Attributausprägungen in den Bestimmungsgleichungen Variablen darstellen, werden sie - wie in der Programmiersprache PROLOG - jeweils mit einem einleitenden Großbuchstaben notiert.

35) In der additiven Erhöhung der Attributausprägung liegt keine bedeutsame Einschränkung. Wenn aus der Perspektive des verfolgten Formalziels das Ausführen eines Schaltschritts mit einer Verringerung des zugehörigen Zielfunktionswerts verknüpft ist, so wird der Schaltwert sw_a des Schaltschritts SS_a mit $sw_a < 0$ als eine negative reelle Zahl festgelegt. Ihre Addition in der nachfolgenden Bestimmungsgleichung führt zu der erforderlichen Erniedrigung der Attributausprägung "wert".

36) Die nachfolgenden Bestimmungsgleichungen müßten bei der Zugrundelegung von Synthetischen Netzen mit der Hilfe von Variablenbindungsfunktionen vb_e formuliert werden. Auf diese formale Präzisierung kann hier jedoch verzichtet werden, weil nur Kanal/Instanz-Netze vorgelegt werden. Darüber hinaus wird der informationstechnisch geprägte Zuweisungsoperator "==" verwendet, um deutlich werden zu lassen, daß es sich nicht um Bestimmungsgleichungen im strengen Sinne des Konzepts Synthetischer Netze handelt.

37) Die Verknüpfung zweier Listen durch den oben benutzten Operator " \oplus " läßt sich mit Hilfe des Prädikats "append" aus der Programmiersprache PROLOG exakt definieren und implementieren. Die einzelnen Schaltschrittbezeichnungen, aus denen die Listen aufgebaut sind, müssen als Ausdrücke " SS_a " der globalen Sorte "string" behandelt werden. Nur so kann ihre Großschreibung mit ihrem Konstantencharakter in der PROLOG-Notation vereinbart werden. (Andernfalls würde der symbolische Ausdruck SS_a wegen seiner Großschreibung als eine Variable interpretiert.) Dafür gilt:

$$\begin{aligned} \text{Liste_neu} &:= [SS_a] \oplus \text{Liste_alt} \\ &\Leftrightarrow \text{append}(["SS_a"], \text{Liste_alt}, \text{Liste_neu}) \end{aligned}$$

Vgl. zum append-Prädikat PROLOG (o.J.), S. 48ff.; CORDES (1988), S. 107f.; CLOCKSIN (1990), S. 172f.

38) Dieses "oder" besitzt - wie auch sonst in dieser Arbeit - im Sinne des logischen "oder" einen inklusiven Charakter.

39) Vgl. dazu die analogen Substitutionen von stellenberandeten Teilnetzen, die anlässlich einer modularen Teilnetzverfeinerung vorgestellt wurden.

40) Es ist dann unerheblich, ob sie nur eine oder aber mehrere Ausgangsstellen besitzt.

41) Je nachdem, ob das vorgegebene Extremalziel als Extremierungsrichtung entweder eine Minimierung oder aber eine Maximierung der zugehörigen Zielfunktion verlangt, nimmt der relevante Extremwert die Gestalt einer minimalen bzw. maximalen Attributausprägung an.

42) Werden die hier konstruierten Kanal/Instanz-Netze als Synthetische Netze präzisiert, so wird diese Selektion von der jeweils ausgewählten Schaltfarbe "c" - der Variablenbindungsfunktion vb_c - bestimmt, mit der die Auswahl-Transition tatsächlich geschaltet wird.

43) Der Extremierungsoperator $ext(\dots)$ aus der ersten Bestimmungsgleichung ist lediglich eine abstrakte Zusammenfassung der beiden früher definierten Minimierungs- und Maximierungsoperatoren $\min(\dots)$ bzw. $\max(\dots)$. Die Subjugate in der Transitionsbeschriftung besitzen rein technischen Charakter. Sie sorgen dafür, daß die ausgewählte Kopie der Optimierungsmarke nicht nur den extremalen Schaltwert erhält, sondern auch die zugehörige Ausprägung des Schaltfolgen-Attributs. Dies ist insbesondere dann wichtig, wenn mehrere Markenkopien mit demselben extremalen Schaltwert auf den Eingangsstellen der Auswahl-Transition liegen. In diesem Fall kann die auszuwählende Markenkopie mit unterschiedlichen Ausprägungen des Schaltfolgen-Attributs versehen werden.

44) Da die Entsprechung eineindeutig ist, kann jedem finalen Schaltprozeß $PRO_{0,E/g}$ ein rückwärts durchlaufener Schaltweg $w_{g,0,E/e,p}$ zugeordnet werden und umgekehrt. Daher existiert eine bijektive Abbildung zwischen den Indizes "g" der finalen Schaltprozesse im Optimierungsnetz einerseits und den komplexen Indizes "e.p" der finalen Schaltwege im Erreichbarkeitsgraphen andererseits. Die Indexausprägungen "g" und "e.p", die sich wechselseitig entsprechen, können durch eine bijektive Funktion "h" mit $h(e.p)=g$ aufeinander abgebildet werden. Darauf wird nachfolgend in verkürzter Weise zurückgegriffen, indem der Index "g" in Abhängigkeit vom Komplexindex "e.p" dargestellt wird: $g(e.p)$.

45) Wie dies im einzelnen geschieht, wird in Kürze näher erläutert.

46) Ein Livelock ist eine Markierung des Netzmodells, für die mindestens ein endlicher Schaltprozeß existiert, der diese Markierung als Start- und Zielmarkierung besitzt. Vgl. dazu die Festlegung von Livelocks, die bereits in einem früheren Kapitel erfolgte. Dort wurde auch schon dargelegt, daß durchaus Petrinetze für die Modellierung von Prozeßkoordinierungen existieren, deren Livelocks aus der Sicht der Prozeßkoordinierung erwünscht sind. Im bewerteten Erreichbarkeitsgraphen des zugrundeliegenden Netzmodells äußert sich ein Livelock durch einen zyklischen Schaltweg, der durch den Knoten der Livelock-Markierung verläuft. Aufgrund der oben vorgestellten Konstruktionsregeln schlägt sich ein solcher Zyklus des bewerteten Erreichbarkeitsgraphen auch in der graphischen Repräsentation des Optimierungsnetzes OP_{RG} als ein zyklischer Weg nieder. Lediglich ist die Kantenrichtung im Zyklus des Optimierungsnetzes gegenüber der im Erreichbarkeitsgraphen entgegengesetzt. Auf einem solchen zyklischen Weg des Optimierungsnetzes liegen im Regelfall mehrere Stellen, die jeweils eine Livelock-Markierung aus dem Erreichbarkeitsgraphen repräsentieren. Der Einfachheit halber wird der gesamte zyklische Weg, der in der graphischen Repräsentation des Optimierungsnetzes zu einer Livelock-Markierung gehört, als eine Livelock-Komponente des Optimierungsnetzes bezeichnet.

Eine Deadlock-Komponente des Optimierungsnetzes besteht dagegen aus genau einer Stelle. Dabei handelt es sich um eine Stelle, die einer *unerwünschten* Deadlock-Markierung aus dem zugrundeliegenden Netzmodell entspricht. Im Kontext von Optimierungsnetzen wird der Klarheit halber der Deadlockbegriff ausschließlich auf unerwünschte Deadlock-Markierungen bezogen. Alle erwünschten Deadlock-Markierungen, die zugleich Endmarkierungen darstellen, werden dagegen kurz als Endmarkierungen angesprochen.

47) Ein Schaltprozeß des Optimierungsnetzes ist von einer Livelock-Komponente genau dann betroffen, wenn mindestens eine Transition, die in diesem Prozeß geschaltet wird, zugleich einen Knoten desjenigen zyklischen Weges darstellt, den die Livelock-Komponente in der graphischen Repräsentation des Optimierungsnetzes bildet (vgl. die voranstehende Anmerkung).

Ein Schaltprozeß des Optimierungsnetzes wäre dagegen von einer Deadlock-Komponente nur dann berührt, wenn in ihm eine Transition geschaltet würde, deren Aktivierung u.a. auch von der Markierung derjenigen Stelle abhängt, welche die Deadlock-Komponente im Optimierungsnetz darstellt (vgl. abermals die voranstehende Anmerkung). Es wird jedoch in einer der späteren Anmerkungen gezeigt, daß dies tatsächlich niemals der Fall sein kann. Daher wird oben die Beeinflussungsmöglichkeit von Schaltprozessen durch Deadlock-Markierungen nur hypothetisch angenommen.

48) Falls ein Optimierungsnetz weder Livelock- noch Deadlock-Komponenten enthält, bleiben die nachfolgenden Vereinfachungsregeln weiterhin gültig. Aber sie werden nicht angewendet, weil ihre Anwendungsvoraussetzungen nicht erfüllt sind. Dann fällt das vorläufige Optimierungsnetz, das mit Hilfe der voranstehend spezifizierten Konstruktionsregeln gewonnen wurde, unmittelbar mit dem endgültigen Optimierungsnetz zusammen.

49) Die Netzvereinfachung besitzt unterschiedliche Qualität je nachdem, ob die Eliminierung von Livelock- oder von Deadlock-Komponenten betroffen ist. Livelocks *müssen* eliminiert werden, damit später das Roll back-Konzept der Dynamischen Programmierung auf Optimierungsnetze angewendet werden kann. Denn es läßt sich nur auf zyklensfreie Graphen anwenden. Die graphische Repräsentation eines Optimierungsnetzes, das noch mindestens eine Livelock-Komponente enthielte, wäre aber ein zyklischer Graph.

Darüber hinaus könnte jeder Zyklus in der graphischen Repräsentation eines Optimierungsnetzes dazu führen, daß in diesem Netz überhaupt kein finaler Schaltprozeß existiert. Denn ein solcher Zyklus wäre in der Lage zu verhindern, daß die erwünschte Endmarkierung des Optimierungsnetzes, unter der nur noch die Ausgangsstelle s_0 markiert ist, jemals erreicht würde. Ein solcher Zyklus kann aufgrund der oben vorgestellten Konstruktionsregeln nur dadurch entstehen, daß in einem Optimierungsnetz mindestens eine Auswahl-Transition mit mehreren Eingangsstellen existiert, für die gilt: Mindestens eine ihrer Eingangsstellen kann nur dann markiert sein, falls die Auswahl-Transition bereits geschaltet worden ist. Die Transition kann aber erst dann schalten, wenn alle ihre Eingangsstellen markiert sind. Folglich wird die Auswahl-Transition niemals schalten. Darüber hinaus muß die Auswahl-Transition mindestens eine weitere Eingangsstelle besitzen, die durch einen Markenfluß markiert wird, der von einer derjenigen Stellen ausgeht, die erwünschte Endmarkierungen des zugrundeliegenden Netzmodells vertreten. (Wäre dies nicht der Fall, so wäre die Ausgangsstelle der Auswahl-Transition keine Stelle, die einen Knoten aus dem Erreichbarkeitsgraphen repräsentierte. Dies widerspräche aber der konstruktiven Definition von Optimierungsnetzen.) Also wird mindestens eine Eingangsstelle der Auswahl-Transition markiert. Da diese Transition niemals schalten kann, bleibt diese Eingangsstelle fortwährend markiert. Als Eingangsstelle einer Transition kann diese Stelle nicht die Ausgangsstelle s_0 des Optimierungsnetzes darstellen. Unter der intendierten Endmarkierung des Optimierungsnetzes soll aber nur diese eine Ausgangsstelle s_0 markiert sein. Folglich ist es unmöglich, diese erwünschte Endmarkierung jemals zu erreichen (q.e.d.).

Deadlock-Komponenten *können* dagegen getilgt werden, brauchen es aber keineswegs. Bei Deadlocks handelt es sich um Markierungen, die in einem Optimierungsnetz durch Stellen ohne Eingangskanten repräsentiert werden. Aufgrund der o.a. Konstruktionsregeln erhalten diese Stellen unter der Ausgangsmarkierung des Optimierungsnetzes keine Kopie der Optimierungsmarke. Da sie keine Eingangskanten besitzen, können sie auch durch keinen Markenfluß im Optimierungsnetz jemals markiert werden. Auf die Bestimmung optimaler Schaltprozesse des zugrundeliegenden Netzmodells wirken aber *nur* diejenigen Kopien der Optimierungsmarke ein, die durch das Optimierungsnetz fließen. (Dies wird in Kürze näher beschrieben.) Daher können Stellen eines Optimierungsnetzes, die Deadlock-Komponenten repräsentieren, das Analyseergebnis prinzipiell nicht beeinflussen. Sie verhalten sich abundant. Folglich spielt es überhaupt keine Rolle, ob die Deadlock-Komponenten aus einem Optimierungsnetz eliminiert oder beibehalten werden. Hier wird der erstgenannte Weg beschritten, um die Optimierungsnetze möglichst kompakt und übersichtlich zu gestalten.

50) Jede Livelock-Komponente äußert sich in der graphischen Repräsentation eines Optimierungsnetzes als ein zyklischer Weg. Umgekehrt muß aber auch jeder Zyklus aus dem Graphen des Optimierungsnetzes mindestens eine Livelock-Markierung aus dem zugrundeliegenden Netzmodell enthalten. Denn aus der Konstruktion von Optimierungsnetzen folgt, daß jede Stelle, die in der graphischen Netzrepräsentation zu einem zyklischen Weg gehört, eine Markierung des Netzmodells darstellt, die durch einen endlichen Schaltprozeß von sich selbst aus erreicht werden kann. Jede solche Markierung stellt per definitionem eine Livelock-Markierung dar.

51) Zyklische Wege aus graphischen Netzrepräsentationen und aus Erreichbarkeitsgraphen werden hier synonym als Zyklen bezeichnet. Sie unterscheiden sich nicht grundsätzlich von denjenigen zyklischen Wegen, die früher als 1-Schleifen thematisiert wurden. Insofern werden auch Zyklus- und Schleifenbegriff synonym verwendet. Um zu verdeutlichen, daß hier die Schwierigkeiten von 1-Schleifen keine Rolle spielen, wird der Klarheit zuliebe der Schleifenbegriff vermieden.

52) Durch das Entfernen mindestens einer Kante aus einem zyklischen Weg verliert dieser seinen in sich geschlossenen Zusammenhang. Der Restweg kann also keinen Zyklus mehr bilden. Wenn auf diese Weise *alle* Zyklen aus der graphischen Repräsentation eines Optimierungsnetzes getilgt sind, kann es auch keine Stelle mehr besitzen, die eine Livelock-Markierung repräsentiert. Denn solche Livelock-Markierungen werden in Optimierungsnetzen immer als Stellen repräsentiert, durch die in der graphischen Netzrepräsentation ein zyklischer Weg verläuft.

53) Es werden lediglich Knoten und Kanten entfernt, die *ausschließlich* zum jeweils betrachteten Zyklus gehören. Alle Knoten und Kanten, über die Marken eines finalen Schaltprozesses fließen, können jedoch nicht ausschließlich zum zyklischen Weg gehören. Andernfalls würde die erwünschte Endmarkierung des Optimierungsnetzes niemals erreicht, so daß auch kein finaler Schaltprozeß vorliegen könnte. Folglich werden weder Knoten noch Kanten, über die Marken eines finalen Schaltprozesses fließen, von der o.a. Eliminierungsoperation betroffen.

54) Die früher vorgetragene Kritik gegenüber Synchronisationsnetzen trifft hier nicht zu. Denn die Einwände bezogen sich insbesondere darauf, daß die Verwendung von Synchronisationsnetzen wegen ihres Mangels an verzweigten Stellen im allgemeinen dazu führt, auf die Modellierung von Knappheitskonflikten zu verzichten. Genau diese Vorhaltung läßt sich aber auf Optimierungsnetze nicht übertragen. Denn Optimierungsnetze können für (endliche)

Erreichbarkeitsgraphen von *beliebigen* Netzmodellen konstruiert werden. Daher werden auch keine Netzmodelle ausgeschlossen, in denen mit der Hilfe von Stellen, die über mehrere Ein- oder Ausgangstransitionen verfügen, Knappheitskonflikte modelliert werden. Auf der Ebene der Objektabbildung durch Netzmodelle erfolgt also keine Einschränkung der Ausdrucksmächtigkeit von Netzen (bis auf die Voraussetzung endlicher Erreichbarkeitsgraphen). Diese Netzmodelle werden zunächst in Erreichbarkeitsgraphen und dann in Optimierungsnetze so transformiert, daß erst die Optimierungsnetze Synchronisationsnetze darstellen. Bei dieser zweistufigen Modelltransformation wird der Informationsgehalt, den diese Modelle durch optimale Schaltprozesse über ihre jeweils abgebildeten Objekte ausdrücken, in keiner Weise beeinträchtigt. Daher bleibt das optimierungsrelevante Ausdruckspotential der ursprünglichen Netzmodelle uneingeschränkt bewahrt, obwohl ihre Optimierungsnetze nur noch zur Klasse der Synchronisationsnetze gehören.

Es mag auf den ersten Blick überraschen, daß es möglich sein soll, einerseits das vollständige Ausdruckspotential des Petrinetz-Konzepts zu nutzen - sofern die Prämisse endlicher Erreichbarkeitsgraphen eingehalten wird - und andererseits bei der Formulierung von Optimierungsnetzen mit der Klasse der Synchronisationsnetze auszukommen. Dennoch widerspricht sich beides nicht. Denn erstens wird nicht behauptet, daß der gesamte Informationsgehalt der ursprünglichen Netzmodelle erhalten bleibt. Statt dessen wurde die Invarianz des Informationsgehalts gegenüber der o.a. zweistufigen Modelltransformation nur in bezug auf die optimalen Schaltprozesse der betroffenen Netzmodelle konstatiert. Zweitens ist zu beachten, daß die Bewahrung der optimierungsrelevanten Modellinformation durch eine beachtenswerte Netzvergrößerung erkaufte wird. Die Optimierungsnetze fallen im allgemeinen wesentlich umfangreicher aus als die jeweils zugrundeliegenden Netzmodelle. Dies wird beispielsweise deutlich, wenn das simple Netz aus Abb. 94 mit dem daraus abgeleiteten Optimierungsnetz in Abb. 95 bis 99 u. 100 verglichen wird. Das läßt erkennen, daß die Transformation von Netzmodellen in Optimierungsnetze zwar durch den Übergang zu Synchronisationsnetzen eine Vereinfachung der Netztopologie bedeuten kann. Aber die Modellinformation, die in der topologischen Netzstruktur enthalten war, geht nicht verloren. Statt dessen finden sich diejenigen Informationsgehalte, die im Netzmodell u.a. durch verzweigte Stellen ausgedrückt wurden, im Optimierungsnetz in topologisch einfacher strukturierten, aber entsprechend umfangreicheren Netzkomponenten wieder.

55) Der zugrundeliegende gewöhnliche Erreichbarkeitsgraph wurde schon an früherer Stelle konstruiert. Er wird hier um die Bewertung der Schaltschritte SS_a durch ihre Schaltwerte $sw(SS_a) = sw_a$ ergänzt, die aus den - hier vereinfachten - Schaltwerten $sw(t_n)$ der schaltschrittzugehörigen Transitionen abgeleitet sind.

56) Es wurde bereits erläutert. Es wird hier nochmals angeführt, um die sukzessive Transformation von Netzmodellen in ihre zugehörigen Optimierungsnetze durchgehend zu veranschaulichen.

57) Teilnetze werden in dieser Arbeit grundsätzlich mit der Hilfe von "Schnittstellen" gebildet. Dabei werden nur zusammenhängende Gesamtnetze betrachtet. Ein Netz muß zwar nicht zusammenhängend sein. Falls es jedoch unzusammenhängend ist, besteht es schon von vornherein aus Teilnetzen, die voneinander isoliert sind. Dann braucht die nachfolgend skizzierte Teilnetzkonstruktion überhaupt nicht angewendet zu werden. Ein ursprünglich vorliegendes Gesamtnetz wird zunächst in genau zwei Teilnetze aufgespalten. Zu diesem Zweck wird eine nicht-leere und echte Teilmenge seiner Stellenmenge S als Schnittstellenmenge festgelegt. Jede Stelle s_m aus dieser Schnittstellenmenge wird dupliziert. Beide Kopien der Schnittstelle s_m erhalten denselben Stellennamen s_m . In der graphischen Netzrepräsentation werden sie durch ihre graue Schraffierung hervorgehoben. Die eine Schnittstellenkopie besitzt alle Eingangskanten der Schnittstelle s_m und deren Vorbereich (aber keine Ausgangskanten), die andere Schnittstellenkopie dagegen alle Ausgangskanten der Schnittstelle s_m und deren Nachbereich (aber keine Eingangskanten). Die Schnittstelle des Gesamtnetzes wird durch ihre beiden Kopien ersetzt. Dann liegen per constructionem zwei in sich zusammenhängende, aber nicht mehr miteinander zusammenhängende Teilnetze vor. Jedes der Teilnetze läßt sich seinerseits als ein Gesamtnetz betrachten, das auf die gleiche Weise in zwei Teilnetze zerlegt werden kann. Diese Teilnetzkonstruktion läßt sich beliebig fortsetzen. Umgekehrt ist es möglich, aus vorliegenden Teilnetzen stets wieder das zugrundeliegende Gesamtnetz zu rekonstruieren. Dazu brauchen lediglich alle Kopien derselben Schnittstelle s_m zu genau einer Stelle s_m zusammengefaßt zu werden. Die Stelle s_m besitzt dann die Gesamtheit der Ein- und Ausgangskanten aller ihrer Kopien. Dadurch verknüpft sie wieder die zuvor unzusammenhängenden Teilnetze.

58) Dies gilt zumindest so lange, wie die Optimierung der Netzmodelle *ausschließlich* mit Techniken des Petrinetz-Konzepts versucht werden soll. Davon unberührt bleibt die Alternative, die bewerteten Erreichbarkeitsgraphen von Netzmodellen durch allgemeine Techniken aus dem Bereich der Graphentheorie auszuwerten. Auch auf diese Weise ließen sich optimale Schaltprozesse für Netzmodelle gewinnen. Dies wurde schon im Zusammenhang mit Aspekten der Künstlichen Intelligenz angedeutet.

59) Die Schaltfolge SF_L wird auch als finale Schaltfolge bezeichnet, weil sie die Schaltfolge eines finalen Schaltprozesses darstellt. Die zugehörige Markierungsfolge MF_L ist durch die Ausgangsmarkierung M_0 des Netzmodells und durch die Schaltfolge SF_L des ausgeführten Schaltprozesses eindeutig festgelegt.

60) Daher werden die beiden finalen Schaltprozesse von Optimierungsnetz und Netzmodell, die jeweils einander eineindeutig zugeordnet sind, als komplementäre Schaltprozesse bezeichnet.

61) Strenggenommen kann die Attributausprägung "folge_e" nicht unmittelbar mit einer optimalen Schaltfolge SF_L^* gleichgesetzt werden. Denn die Ausprägungen des Attributs "schaltfolge" wurden oben als Listen "[...]" definiert, die sich in der Programmiersprache PROLOG leicht handhaben lassen. Schaltfolgen SF_L stellen dagegen L-Tupel $SF_L = (SS_{a(1)}, \dots, SS_{a(L)})$ dar. Daher müßten zunächst die begrenzenden Klammern "[" und "]" einer PROLOG-Liste "[SS_{a(1)}, ..., SS_{a(L)}]" durch die Tupelklammern "(" bzw. ")" ersetzt werden, bevor in formal korrekter Weise die Schaltfolge $SF_L = (SS_{a(1)}, \dots, SS_{a(L)})$ vorliegt. Dabei würde jedoch der materielle Gehalt der Attributausprägung "[SS_{a(1)}, ..., SS_{a(L)}]" nicht verändert. Daher wird hier von dieser formalen Präzisierung abgesehen.

62) In einem Optimierungsnetz können durchaus mehrere optimale Schaltprozesse existieren. Sie führen zwar alle zum selben gesuchten Netzoptimum sw^* . Das Netzoptimum ist also eindeutig definiert (wegen des vorausgesetzten eindimensionalen Formalzielsystems). Aber die optimalen Schaltprozesse unterscheiden sich - sofern mehrere existieren - hinsichtlich derjenigen Schaltfolgen (und somit auch bezüglich der alternierenden Markierungsfolgen), die beim Ausführen der optimalen Schaltprozesse im Optimierungsnetz ausgeführt werden. Diese Divergenz äußert sich in verschiedenen Ausprägungen "folge_e" des Schaltfolgen-Attributs derjenigen Optimierungsmarkenkopie, die von jedem der optimalen Schaltprozesse auf der Ausgangsstelle s_0 schließlich abgelegt wird. Allerdings gilt es dabei zu beachten, daß die verschiedenen Ausprägungen "folge_e" des Schaltfolgen-Attributs nicht die Schaltfolgen der unterschiedlichen Schaltprozesse $PRO_{0,E/g(e,p).OP}$ im *Optimierungsnetz* selbst anzeigen. Vielmehr geben die Attributausprägungen diejenigen Schaltfolgen SF_L wieder, die zu unterschiedlichen Schaltprozessen $PRO_{0,E/e,p}(SF_L, MF_L)$ im zugrundeliegenden *Netzmodell* gehören und jeweils mit einem der unterschiedlichen Schaltprozesse $PRO_{0,E/g(e,p).OP}$ aus dem *Optimierungsnetz* korrespondieren. Darauf wird später noch zurückgekommen, wenn die Markenflüsse des Optimierungsnetzes detaillierter erläutert werden.

63) Die Einschränkung "im wesentlichen" berücksichtigt, daß während der nachfolgenden Fortsetzung der bereits eingeleiteten Schaltprozeßausführung durchaus Optionen bestehen können, den Schaltprozeß auf unterschiedliche Weise weiterzuführen. Das ist immer dann der Fall, wenn unter derselben Markierung des Optimierungsnetzes mehrere Schaltschritte aktiviert sind. Die besondere konstruktive Auslegung des Optimierungsnetzes sorgt dann dafür, daß stets das gesuchte Netzoptimum sw^* gefunden wird - unabhängig davon, welche der alternativen Ausführungsfortsetzungen tatsächlich gewählt wird. Aber die Auswahl entscheidet darüber, welcher von den mehreren zulässigen optimalen Schaltfolgen SF_L^* zu dem einen Netzoptimum sw^* führen wird.

64) Ein Schaltschritt heißt unter einer Markierung aktiviert, wenn alle seine zugehörigen Transitionen unter derselben Markierung nebenläufig aktiviert sind.

65) Es wird also eine ausgesprochen "intelligente" Lösungssuche betrieben: Es braucht nur genau *ein* Lösungsversuch zu Ende gebracht zu werden - und schon steht mit Sicherheit die gesuchte optimale Problemlösung zur Verfügung.

66) Das Konzept der Datenflußautomaten wird näher behandelt bei DENNIS, J. (1974b), S. 402ff.; DENNIS, J. (1975), S. 126ff.; DENNIS, J. (1980), S. 48ff.; GILOI (1981), S. 279ff., insbesondere S. 288ff.; TRELEAVEN (1982), S. 93ff.; TANAKA, H. (1982), S. 209ff.; AMAMIYA (1984), S. 105ff.; YAMAGUCHI, Y. (1984), S. 524ff.; FRÜCHTENICHT (1984), S. 52ff.; ZELEWSKI (1986a), S. 320 u. 685.

67) Vgl. zum Konzept des "rechnenden Raums" ZUSE, K. (1969); ZUSE, K. (1972), S. 129ff. In ZUSE's Konzept des "rechnenden Raums" werden zelluläre Automaten betrachtet. Ihre Zellen sind in einem zweidimensionalen "Raum" so miteinander verknüpft, daß das Zusammenwirken der lokal definierten Zellenoperationen insgesamt die Erfüllung von Berechnungsaufgaben leistet. In ähnlicher Weise erlaubt der topologische Zusammenhang von Stellen und Transitionen in einem Optimierungsnetz, die Schaltwerte und -folgen finaler Schaltprozesse durch die rein lokal definierten Schaltoperationen seiner Transitionen zu ermitteln.

68) Petrinetze lassen sich über den o.a. Spezialfall der Optimierungsnetze hinaus allgemein dazu einsetzen, "Petrinetz-Computer" zu gestalten. Sie sind in der Lage, die Werte von arithmetischen Funktionen für beliebige ganzzahlige Argumente zu berechnen.

69) Daher werden die beiden finalen Schaltprozesse von Optimierungsnetz und Netzmodell, die jeweils einander eineindeutig zugeordnet sind, als komplementäre Schaltprozesse bezeichnet.

70) Der Schaltwert $sw_{0,E/e,p}$ eines finalen Schaltwegs $wg_{0,E/e,p}$ ist per constructionem die Summe aller Veränderungen des Zielfunktionswerts, die durch Schaltakte von Transitionen aus denjenigen Schaltschritten bewirkt werden, mit denen die Schaltkanten des finalen Schaltwegs im Erreichbarkeitsgraphen beschriftet sind. Das Schaltwert-Attribut der Optimierungsmarkenkopie wurde entweder mit dem fixen Startwert der Zielfunktion initialisiert. Oder es wurde zu Beginn auf den Wert "0" gesetzt, falls solche entscheidungsirrelevanten fixen Startwerte ausgeklammert bleiben. Daher muß am Ende eines finalen Schaltwegs die Ausprägung des Schaltwert-Attributs den Wert der Zielfunktion für diesen Schaltweg darstellen.

71) Vgl. die allgemeine Erläuterung des Zusammenhangs zwischen Schaltwegen in Erreichbarkeitsgraphen und Schaltprozessen in Netzen.

72) Multiplikator-Transitionen mit mehreren Ausgangsstellen spielen dagegen nur eine untergeordnete Rolle. Sie entsprechen dem Zusammentreffen von alternativen Schaltprozessen in einer gemeinsamen intermediären Netzmarkierung oder in einer der Endmarkierungen. Dieses Prozeßzusammentreffen ist nur eine mögliche Folge von Auswahloptionen, die in einem früheren Zustand des zugrundeliegenden Netzmodells bestanden. Es muß daher in einem vollständigen Optimierungsnetz zwar berücksichtigt werden, trägt aber zur optimierenden Auswahl zwischen Schaltprozeßalternativen nicht bei. Diese untergeordnete Bedeutung der Transitionen mit mehreren Ausgangsstellen schlägt sich in ihrer Kantenbeschriftung nieder. Ihr zufolge werden die Kopien der Optimierungsmarke nur identisch vervielfältigt, ohne ihre Attributausprägungen zu verändern.

73) Falls sich auf den Eingangsstellen einer Auswahl-Transition mehrere Kopien der Optimierungsmarke mit demselben lokal-extremalen Schaltwert befinden, wird genau eine beliebige von ihnen ausgewählt, um das Schalten der Transition auszuführen.

74) Jeder Schalttakt, bei dem eine Auswahl-Transition mit einer Kopie der Optimierungsmarke geschaltet wird, bedeutet die Auswahl einer rein lokal definierten Entscheidungsalternative. Hierdurch erfolgt keine Entscheidungsbindung bezüglich des Schaltens aller Auswahl-Transitionen, die erst unter späteren Markierungen des Optimierungsnetzes aktiviert sind. Daher erfüllt diese Realisierung lokaler Zielfunktionsextrema durch Schaltakte von Auswahl-Transitionen exakt das früher aufgestellte Postulat wirkungsminimaler Spielraumschließungen. Dabei erstreckt sich ein lokaler Spielraum auf alle Auswahl-Transitionen, die unter derselben Markierung des Optimierungsnetzes aktiviert sind, und alle Kopien der Optimierungsmarke, mit denen diese aktivierten Transitionen geschaltet werden könnten.

75) Vgl. zu Darstellungen und Anwendungen des Roll back-Konzepts für die Analyse von Entscheidungsbäumen MAGEE (1964a), S. 132; MAGEE (1964b), S. 86ff.; LAUX (1971), S. 40ff. u. 124; HAX,H. (1972), S. 328; RAIFFA (1973), S. 39ff.; RIVETT (1980), S. 81ff.; HAX,H. (1985), S. 178ff.; ZELEWSKI (1986a), S. 686; FRANKE,G. (1988), S. 226f.; BLOHM (1988), S. 251f.; BAMBERG (1989), S. 218ff.; SIEBEN (1990), S. 41f.; KRUSCHWITZ (1990), S. 288ff.; LAUX (1991), S. 261f. u. 269; PERRIDON (1991), S. 124ff.; GÖTZE,U. (1991), S. 329f.

76) Vgl. dazu die Erläuterung, wie jedes Schalten einer Schaltschritt-Transition die alte Ausprägung des Schaltfolgen-Attributs der Optimierungsmarkenkopie *von vorne* um einen Schaltschritt erweitert.

77) Vgl. zum Optimalprinzip, das oftmals auch als Optimalitätsprinzip (principle of optimality) thematisiert wird, BELLMAN (1954), S. 504ff.; BELLMAN (1957a), S. 83ff. u. 291ff.; BELLMAN (1957b), S. 281; BELLMAN (1967), S. 88f.; NEUMANN,K. (1968), S. 60ff. u. 68ff.; BECKMANN,M. (1968), *passim*, insbesondere S. 10ff., 45ff., 63ff. u. 125ff.; STEPAN (1977), S. 87ff.; NEUMANN,K. (1977), S. 31ff., insbesondere S. 33f.; KAMIEN (1981), S. 238ff.; SCHIEMENZ (1982), S. 86ff.; FEICHTINGER (1982a), S. 176; KERN,W. (1987), S. 67; STEPAN (1988), S. 179f.; FRANKE,G. (1988), S. 227; DYCKE (1988), S. 26f.; SCHNEEWEIB,C. (1989b), S. 10f.; BAMBERG (1989), S. 218; ELLINGER (1990a), S. 243ff.; DOMSCHKE (1990), S. 131f.; NEUMANN,K. (1990), S. 169ff.

78) Das Roll back-Konzept läßt sich strenggenommen nicht überall dort anwenden, wo das Optimalprinzip der Dynamischen Programmierung gilt. Vielmehr wird sein Anwendungsbereich auf Graphen eingeschränkt, die keine zyklischen Wege enthalten. Daher wird es vor allem auf Entscheidungsbäume angewendet, die per constructionem azyklische Graphen darstellen.

Die hier untersuchten Optimierungsnetze gehören jedoch im allgemeinen nicht zu den baumartigen Graphen. Denn ihnen fehlt - zumindest im Regelfall - die charakteristische Eigenschaft von Baumgraphen, daß sich jeder Knoten von der Baumwurzel aus auf genau einem Weg erreichen läßt. Optimierungsnetze besitzen zwar einen oder mehrere Wurzelknoten, von denen aus alle anderen Knoten erreicht werden können. Diese Wurzelknoten sind die Stellen, die jeweils eine der erwünschten Endmarkierungen des zugrundeliegenden Netzmodells darstellen. Aber von diesen Wurzelknoten gehen oftmals verschiedene Wege aus. Jeder dieser Wege, die jeweils einen finalen Schaltprozeß des Netzmodells repräsentieren, führt zum Knoten der Ausgangsstelle s_0 . Also kann mindestens dieser Knoten auf mehreren Wegen erreicht werden. Daher handelt es sich bei Optimierungsnetzen im allgemeinen um keine baumartigen Graphen. A fortiori stellen sie auch keine Entscheidungsbäume dar. Dennoch läßt sich das Roll back-Konzept auf Optimierungsnetze anwenden. Denn der Konzepteinsatz setzt lediglich azyklischen Graphen voraus. Die graphische Repräsentation von Optimierungsnetzen erfüllt diese Prämisse. Entweder enthält sie von vornherein keine zyklischen Wege. Oder die oben spezifizierte Konstruktion von Optimierungsnetzen sorgt dafür, daß Optimierungsnetze, deren Graphen zunächst noch zyklische Wege aufweisen, nachträglich von allen Zyklen befreit werden. Daher besitzt jedes vollständig bearbeitete Optimierungsnetz notwendig eine azyklische graphische Repräsentation.

79) Den ersten Ansatzpunkt bildete die nebenläufige Konstruktionsmethode für Erreichbarkeitsgraphen, die bereits vorgestellt wurde.

80) Ein interessanter Ansatz in dieser Richtung findet sich bei FAHRION (1989), S. 514ff. Er stellt eine Programmierung des Roll back-Konzepts vor, die auf der - hier ebenfalls benutzten - Programmiersprache PROLOG basiert. Einerseits wird auf diese Weise ersichtlich, wie sich die hier vorgestellte Auswertung von Optimierungsnetzen so implementieren läßt, daß eine kohärente Einbettung in die auch sonst präferierte PROLOG-Umgebung erfolgt.

Andererseits wurde bereits auf die vielfältigen Konzepte für nebenläufige PROLOG-Implementierungen hingewiesen. Sie könnten herangezogen werden, um FAHRION's Programmierung des Roll back-Konzepts um die oben angesprochene Nebenläufigkeit der Programmausführung zu bereichern.

81) Dabei wird vorausgesetzt, daß als Effizienzmaßstab nur die Operationszeit betrachtet wird, die ein Automatisches Informationsverarbeitungssystem zum Erkennen eines finalen Schaltprozesses insgesamt benötigt. Dadurch werden andere Effizienzeinflüsse ausgeklammert. Beispielsweise bleibt unberücksichtigt, daß die Prozessorvervielfachung bei parallelen Systemarchitekturen gegenüber sequentiellen Architekturen mit jeweils nur einem Prozessor zusätzliche Ressourcen bindet. Darauf wurde schon hingewiesen.

82) Die Besonderheiten und Vorzüge, die parallele Architekturen Automatischer Informationsverarbeitungssysteme gegenüber ihren sequentiellen Pendanten auszeichnen, hat der Verf. an anderer Stelle ausführlicher diskutiert; vgl. ZELEWSKI (1986a), S. 314ff. (vgl. ebenso die entsprechenden Anmerkungen zur nebenläufigen Konstruktion von Erreichbarkeitsgraphen). Er verzichtet daher darauf, sie hier nochmals darzulegen.

83) Wenn dies der Fall ist, läßt sich auch jede nicht-leere Teilmenge der Gesamtheit aller sw*-optimalen Schaltprozesse ermitteln. Dazu braucht lediglich zunächst die Gesamtheit erzeugt zu werden, aus der dann nachträglich die jeweils interessierende Teilmenge herausgelöst wird.

84) Dieser Erreichbarkeitsgraph ist nicht mit dem bewerteten Erreichbarkeitsgraphen des zugrundeliegenden Netzmodells zu verwechseln. Vielmehr wurde der letztgenannte bewertete Erreichbarkeitsgraph zunächst in ein Optimierungsnetz transformiert. Erst die dynamische Struktur dieses Optimierungsnetzes wird durch den oben eingeführten Erreichbarkeitsgraphen repräsentiert. Daher läßt sich der bewertete Erreichbarkeitsgraph des zugrundeliegenden Netzmodells der Objektebene zurechnen, während der Erreichbarkeitsgraph des Optimierungsnetzes zur Metaebene gehört.

85) Das Petrinetz-Konzept zeichnet sich grundsätzlich dadurch aus, die inhärente Nebenläufigkeit von problemlösenden Methoden aufzudecken und bei der Bewältigung von Problemen entsprechend auszunutzen. Dies wurde schon anhand des Problems, Erreichbarkeitsgraphen von Netzmodellen zu konstruieren, exemplarisch ausgeführt. Dabei handelte es sich jedoch nicht - wie nachfolgend diskutiert - um ein Optimierungsproblem. Ein anderes Beispiel betrifft die Aufgabe, Lösungen für linear-ganzzahlige OR-Modelle zu bestimmen, deren Formalzielsysteme keine Extremalziele enthalten. Auch solche Modelle beinhalten keine Optimierungsprobleme. Sie können aber eine große Rolle spielen, um angesichts großer Anzahlen von Satisfizierungs- oder Punktzielen zulässige Problemlösungen aufzufinden. Dies gilt z.B. für kombinierte Personal- und Betriebsmittel-Einsatzplanungen von Luftverkehrsgesellschaften oder für Stundenplanerstellungen von Ausbildungseinrichtungen. Auch für solche OR-Modelle lassen sich mit Hilfe des Petrinetz-Konzepts nebenläufige Lösungsmethoden formulieren. Dies hat der Verf. an anderer Stelle im Detail ausgeführt; vgl. ZELEWSKI (1986c), S. 5ff.; ZELEWSKI (1988b), S. 353ff.

86) Es wird nochmals betont, daß als Effizienzmaßstab nur die Operationszeit zugrundeliegt, die zur Ermittlung eines Optimums benötigt wird. Es geht hier also um das möglichst rasche Erkennen optimaler Problemlösungen. Geschwindigkeitsvorteile nebenläufiger Informationsverarbeitung werden aber zunächst durch zusätzlichen Ressourceneinsatz erkauft, der für die Prozessorvervielfachung in parallelen Systemarchitekturen anfällt. Hinzu kommen die Ressourcen, die für die Reformulierung eines Optimierungsproblems durch ein Netzmodell, für die Konstruktion seines Erreichbarkeitsgraphen und für die Ableitung des letztlich interessierenden Optimierungsnetzes benötigt werden. Bei weiter gefaßten Effizienzmaßstäben müßten alle vorgenannten Ressourcenaspekte gegen den eingangs konstatierten Geschwindigkeitsvorteil aufgewogen werden. Daraus könnte durchaus die Erkenntnis folgen, daß sich die nebenläufige Lösung von Optimierungsproblemen auf der Basis des Petrinetz-Konzepts - zumindest mit den derzeit verfügbaren Automatischen Informationsverarbeitungssystemen - weitaus ineffizienter verhält als alternative Lösungskonzepte. Vgl. dazu die einleitenden Erörterungen des Effizienzdefizits der Erreichbarkeitsanalyse von Netzmodellen.

87) Vgl. dazu die Quellen, die bereits zur Entscheidungsbaumtechnik angeführt wurden.

88) Die Transformation von Entscheidungsbäumen in äquivalente Optimierungsnetze bereitet keine großen Schwierigkeiten. Die zugrundeliegende Transformationsidee hat der Verf. bereits in ZELEWSKI (1986a), S. 680ff., dargelegt. Sie wird hier leicht modifiziert, um den Einsatz von Optimierungsnetzen zu ermöglichen.

Alle Entscheidungsknoten EK_i mit $i \in \{1, \dots, I\}$ und $I \in \mathcal{N}_*$ des Entscheidungsbaums bleiben erhalten. Jede Ausgangskante eines Entscheidungsknotens EK_i repräsentiert eine Entscheidungsalternative $EA_{i,j}$ mit $j \in \{1, \dots, J_i\}$, $J_i \in \mathcal{N}_*$ und $J_i \geq 2$. Die Ausgangskante einer solchen Entscheidungsalternative $EA_{i,j}$ ist mit dem Zielbeitrag $z_{f,i,j}$ gewichtet, den die Entscheidung zugunsten dieser Alternative verursachen würde. Jede Ausgangskante eines Entscheidungsknotens ist die einzige Eingangskante von jeweils einem Zustandsknoten ZK_m mit $m \in \{1, \dots, M\}$ und $M \in \mathcal{N}_*$. Jeder Zustandsknoten kann keine, eine oder mehrere Ausgangskanten besitzen. Falls ein Zustandsknoten keine Ausgangskante besitzt, handelt es sich um einen Endknoten des Entscheidungsbaums. Wenn ein Zustandsknoten mindestens eine Ausgangskante besitzt, so repräsentiert jede Ausgangskante eine mögliche Umweltsituation $US_{m,n}$ mit $n \in \{1, \dots, N_m\}$ und $N_m \in \mathcal{N}_*$. Die Ausgangskante einer solchen Umweltsituation $US_{m,n}$ ist mit der Eintrittswahrscheinlichkeit $w_{m,n}$ der

Umweltsituation im Zustandsknoten ZK_m gewichtet. Für einen Zustandsknoten mit genau einer Ausgangskante beträgt diese Eintrittswahrscheinlichkeit notwendig $w_{m,1}=1$. Alle Zustandsknoten des Entscheidungsbaums werden einschließlich ihrer adjazenten Kanten durch einzelne Kanten oder durch Kantenbündel ersetzt. Für jeden Zustandsknoten ZK_m gilt, wenn seine genau eine Eingangskante mit dem Zielbeitrag $zf_{i,j}$ gewichtet ist:

- Wenn der Zustandsknoten als Endknoten keine Ausgangskante besitzt, dann wird er ersatzlos gestrichen. Seine Eingangskante bleibt mit dem Gewicht $W=zf_{i,j}$ unverändert erhalten.
- Wenn der Zustandsknoten genau eine Ausgangskante mit dem Gewicht $w_{m,1}=1$ besitzt, dann wird der Zustandsknoten samt seiner Ein- und seiner Ausgangskante durch genau eine Kante mit dem Gewicht $W = 1 \cdot zf_{i,j} = zf_{i,j}$ ersetzt.
- Wenn der Zustandsknoten mehrere Ausgangskanten mit den Gewichten $w_{m,n}$ besitzt, dann wird der Komplex aus dem Zustandsknoten und seinen adjazenten Kanten durch genau so viele Kanten ersetzt, wie der Zustandsknoten Ausgangskanten besitzt. Dies führt zu einem Multigraphen. Jede neu gebildete Kante erhält das Gewicht $W = w_{m,n} \cdot zf_{i,j}$.

Das nunmehr vorliegende Derivat des ursprünglichen Entscheidungsbaums umfaßt nur noch Entscheidungsknoten. (Das Derivat stellt keinen Graphen mehr dar, weil einige seiner "Kanten" nicht mehr durch je zwei begrenzende Knoten konstituiert werden.) Jeder Entscheidungsknoten wird als eine Auswahl-Transition dargestellt. Jede Kante, die sich zwischen zwei Auswahl-Transitionen mit dem Gewicht W erstreckt, wird durch ein stellenberandetes Teilnetz so ersetzt, daß gilt: Das Teilnetz besteht aus genau einer Schaltschritt-Transition. Sie kann wie üblich mit zwei Bestimmungsgleichungen beschriftet werden, um optimale Schaltwerte und -folgen zu ermitteln. Hier wird der Übersichtlichkeit halber auf den zweiten Aspekt verzichtet. Also wird jede Schaltschritt-Transition nur mit der einen Bestimmungsgleichung "Wert_neu := Wert_alt + sw_a " beschriftet. Darin wird der Schaltwert sw_a mit dem Gewicht W der substituierten Kante gleichgesetzt. Die Schaltschritt-Transition erhält genau eine Ein- und genau eine Ausgangsstelle. Die Eingangsstelle (Ausgangsstelle) wird über eine Eingangskante (Ausgangskante) mit derjenigen Auswahl-Transition verknüpft, welche den Entscheidungsknoten am Ursprung (an der Spitze) der substituierten Kante ersetzt hat. Jede Kante, die aus der Transformation eines Endknotens hervorgegangen ist, besitzt zwar eine Auswahl-Transition in ihrem Kantenursprung, aber keine Transition an ihrer Kantenspitze. Sie wird bis auf eine Ausnahme genau so behandelt, wie es voranstehend beschrieben wurde. Lediglich unterbleibt es, die neu gebildete Ausgangsstelle mit einer Auswahl-Transition zu verknüpfen, die hier nicht definiert ist.

In dem resultierenden vorläufigen Optimierungsnetz wird die Richtung aller Kanten umgekehrt. Alle Stellen ohne Eingangskanten, die aus der Transformation von Endknoten des Entscheidungsbaums hervorgegangen sind, werden über eine Multiplikator-Transition mit einer zusätzlich eingeführten Endstelle $s_{E/1}=s_E$ verknüpft. Die eine Auswahl-Transition, die für die frühere Wurzel des Entscheidungsbaums eingeführt worden ist, wird um genau eine Ausgangsstelle s_0 ergänzt. Dann liegt ein Optimierungsnetz vor, das den ursprünglich vorgegebenen Entscheidungsbaum äquivalent repräsentiert. Die Abb. 101 bis 103 verdeutlichen den Transformationsprozeß anhand eines einfachen Beispiels. Daraus wird ersichtlich, daß die Umwandlung von Entscheidungsbäumen in Optimierungsnetze bei graphischer Visualisierung wesentlich weniger komplex ausfällt, als es die o.a. natürlichsprachliche Beschreibung erwarten läßt. Allerdings verdeutlichen sie auch, daß der Umfang des Optimierungsnetzes gegenüber dem zugrundeliegenden Entscheidungsbaum beträchtlich angewachsen ist. Dies unterstreichen die früheren Anmerkungen, die das erhebliche Volumen von Optimierungsnetzen ansprachen.

89) In den Schaltakten der Auswahl-Transitionen spiegelt sich das darwinistische Prinzip des Roll back-Konzepts wieder, jeweils alle garantiert nicht-optimalen Teillösungen auszulöschen. Denn alle Kopien der Optimierungsmarke, deren aktuellen Schaltwerte in der lokalen Umgebung einer Auswahl-Transition keinen Extremwert der vorgegebenen Zielfunktion darstellen, gehen beim Schalten dieser Transition unter. Darüber hinaus verdeutlichen mehrere Kopien der Optimierungsmarke, die durch Multiplikator-Transitionen zunächst identisch vervielfacht wurden und danach simultan durch das Optimierungsnetz fließen, die inhärente Nebenläufigkeit von Algorithmen, die der Auswertung von Entscheidungsbäumen dienen. Eine derart plastische Visualisierung algorithmischer Nebenläufigkeit ist dem Verf. seitens anderer Repräsentationskonzepte nicht bekannt.

90) Einerseits gehört die Entscheidungsbaumtechnik nicht zum hier interessierenden Untersuchungsobjekt der Koordinierung komplexer, insbesondere nebenläufiger Systeme. Andererseits wurden Effizienzgesichtspunkte zugunsten der Ausdrucksmächtigkeit in den Hintergrund gerückt.

91) Vgl. Quellen, die in einer früheren Anmerkung zum Konzept der Kontraktnetze angeführt wurden.

92) Die Leistungsfähigkeit bezieht sich auf die gesicherte Ermittlung von Netzoptima. Effizienzaspekte sind damit noch nicht angesprochen.

93) Die Übereinstimmung dieser lokal optimalen Angebotsauswahl fällt besonders offensichtlich aus bezüglich der Charakterisierung von Verhandlungsprotokollen für die Knoten von Kontraktnetzen bei BAKER, A. (1988), S. 105f. (z.B. in den Knoten "min" der Fig. 6 auf S. 105): Dort wird aus allen Angeboten für Ausführungen der gleichen Teilaufgabe jeweils ein kostenminimales Ausführungsangebot ausgewählt.

94) Das Gleiche gilt für die Quellen, die bereits an früherer Stelle als betriebswirtschaftlich orientierte Beiträge zum Kontraktnetz-Konzept erwähnt wurden. Dort werden zwischen den Knoten eines Kontraktnetzes in der Regel keine echten Verhandlungen über auszuführende Teilaufgaben geführt. Statt dessen delegiert ein Netz-knoten eine Teilaufgabe an einen anderen, ex ante fixierten Netz-knoten. Weder werden von verschiedenen Netz-knoten Angebote über alternative Ausführungen der gleichen Teilaufgabe eingeholt. Noch kann der beauftragte Netz-knoten teilautonom darüber entscheiden, ob er ein Angebot zur Teilaufgabenerfüllung unterbreiten möchte. Daher stehen die Knoten des "Kontrakt"netzes in einer Beziehung, die sich im Rahmen der konventionellen Informationsverarbeitung bewegt. Sie läßt sich etwa durch den Aufruf von Unterprogrammen realisieren. Ebenso kommen verteilte Informationsverarbeitungssysteme in Betracht, die nach dem "master/slave"-Prinzip organisiert sind. Der Verzicht auf Kontraktverhandlungen läßt sich für den Fall der Optimierungsnetze sogar innerhalb des Kontraktnetz-Konzepts rechtfertigen. Denn bei DAVIS,R. (1981), S. 34 i.V.m. S. 16 (request-response mechanism), DAVIS,R. (1983), S. 100 i.V.m. S. 81 (request-response mechanism), wird explizit empfohlen, auf Verhandlungen über Aufgabenausführungen zugunsten eines direkten Agentenaufrufs zu verzichten, sofern es sich um sehr einfache Teilaufgaben handelt. Denn dann kann der Ressourceneinsatz für die Verhandlungsprozesse angesichts der erwarteten Ressourceneinsparungen durch Auswahl zwischen alternativen Ausführungsangeboten - im Sinne eines groben Tendenzurteils - nicht mehr gerechtfertigt werden.

Darüber hinaus beansprucht das Kontraktnetz-Konzept, daß die teilautonomen Agenten in der Lage seien, eine vorgegebene Teilaufgabe selbständig in Teilaufgaben zu zerlegen; vgl. DAVIS,R. (1981), S. 6, 8ff. u. 14; DAVIS,R. (1983), S. 66, 68ff. u. 75f. Auch diese Teilaufgabenkomposition bleibt in Optimierungsnetzen unberücksichtigt. Denn die zu erfüllenden Teilaufgaben liegen bereits durch die vorausgesetzten Erreichbarkeitsgraphen und die Konstruktionsmethode für Optimierungsnetze fest. Die eingangs angesprochenen betriebswirtschaftlichen Quellen ziehen die selbständige Aufgabendeckomposition ebensowenig in Betracht. Am Rande sei vermerkt, daß der Dekompositionsanspruch des Kontraktnetzkonzepts höchst fragwürdig erscheint. So wird die Realisierung der Aufgabendeckomposition von DAVIS und SMITH in den beiden o.a. Werken nicht konkret beschrieben. Statt dessen räumen sie sogar an späterer Stelle ein: "Problem decomposition ... is not a well-understood process ... there are relatively few principles that can be used prospectively to produce good decompositions." (DAVIS,R. (1981), S. 35, u. DAVIS,R. (1983), S. 100f.). Dabei geben sie noch nicht einmal an, welche Dekompositionsprinzipien sie meinen und wie sie die Güte von Problemdekompositionen messen wollen. Aufgrund dieser Schwächen vermag das Konzept der Kontraktnetze seinen Anspruch auf Aufgabendeckomposition - zumindest in den o.a. Quellen - nicht überzeugend einzulösen.

95) Die Informationsverarbeitung eines solchen Datenflußautomaten unterliegt dem "push"-Prinzip, bei dem Informationsobjekte ihre eigene Verarbeitung abhängig von ihrem aktuellen Informationsgehalt vorantreiben. Kontraktnetze verwirklichen dagegen in der Regel ein "pull"-Prinzip. Dort zieht eine Teilaufgabe, die erfüllt werden soll, entsprechende Verhandlungen über die Aufgabenausführung nach sich, die letztlich zur Ausführung der Teilaufgabe führen. Vgl. zu dieser "pull"-Charakteristik des Kontraktnetzkonzepts BAKER,A. (1988), S. 102f.

Es wäre jedoch verfehlt, die Ersetzung des "pull"-Prinzips der Kontraktnetze durch das "push"-Prinzip der Optimierungsnetze als einen vermeidenswerten Defekt zu betrachten. Denn die Konstruktion von Optimierungsprinzipien als Datenflußautomaten erlaubt es, in Netzmodellen *optimale* Schaltprozesse zu ermitteln. Die Verhandlungskonzepte von Kontraktnetzen auf der Basis des "pull"-Prinzips stellen eine solche *Prozessoptimierung* in Systemmodellen dagegen oftmals nicht sicher. Vgl. dazu etwa die dürftigen und vage gehaltenen Ausführungen in BAKER,A. (1988), S. 103f., zur analytischen Auswertung von Kontraktnetzen. Statt dessen herrscht die *simulative* Untersuchung von Auswertungsnetzen vor; vgl. etwa BAKER,A. (1988), S. 104ff. Solche Simulationsstudien sind jedoch - wenn vom Sonderfall omnivalenter Simulationsanalysen abgesehen wird - im allgemeinen nicht in der Lage, holistische Modelleigenschaften wie Netzoptima zu ermitteln. Beispielsweise leistet die Auswahl kostenminimaler Ausführungsangebote für einzelne Teilaufgaben, die für ein Verhandlungsprotokoll in einer früheren Anmerkung erwähnt wurde, immer nur eine *lokale* Optimierung. Ein *globales* Kontrakt optimum wird hierdurch in keiner Weise gewährleistet. Auf diese fehlende globale Optimierungsqualität gehen manche Proponenten des Kontraktnetz-Konzepts jedoch überhaupt nicht ein, obwohl sie - wie z.B. BAKER,A. (1988), S. 102 u. 106, - immer wieder von maximalen, minimalen oder optimalen Zielerfüllungen reden. Darüber hinaus müssen in Kontraktnetzen noch spezielle Schwierigkeiten behandelt werden, die aus den nebenläufigen Verhandlungsprozessen der teilautonomen Agenten resultieren. Dabei kann es sich vor allem um Deadlocks und um das zwischenzeitliche Verfallen von unterbreiteten, aber noch nicht akzeptierten Angeboten handeln. Solche Verhandlungsprobleme werden z.B. von DAVIS,R. (1981), S. 8f. u. 36f.; BAKER,A. (1988), S. 102f. (Deadlocks) u. 108f. (Verfallen von Angeboten), diskutiert. Vgl. dazu auch die ausführlichen Erörterungen, die sich mit Deadlocks, Livelocks und Synchronisationskonzepten für deren Vermeidung beschäftigen. Diese Schwierigkeiten treten in den oben vorgestellten Optimierungsnetzen dagegen grundsätzlich nicht auf, weil dort aufgrund des "pull"-Prinzips keine Verhandlungen erfolgen. Allerdings wird eingeräumt, daß andere Vertreter des Kontraktnetz-Konzepts explizit seine mangelnde Optimierungsfähigkeit einräumen. Dabei wird sowohl die Unmöglichkeit anerkannt, aus lokalen Entscheidungen - ohne zusätzliche Vorkehrungen - globale Optima abzuleiten, als auch auf die voranstehend angesprochenen Koordinie-

rungsschwierigkeiten bei nebenläufigen Verhandlungsprozessen Bezug genommen; vgl. DAVIS,R. (1981), S. 36f.; DAVIS,R. (1983), S. 102f.

96) Allerdings wird in der späteren Fallstudie ein Zuordnungsmodul für die Zuordnung zwischen Arbeitsgängen und Bearbeitungsstationen vorgestellt. Es wird so ausgestaltet, daß es sich ebenso für die Ausformulierung eines Schaltagenten benutzen läßt.

5.2.3.5.5 Robustheitsuntersuchungen

Die wesentliche Aufgabe der Robustheitsanalyse eines Netzmodells besteht darin, die Mengen IPROM und NIPROM aller intendierten bzw. nicht-intendierten Schaltprozesse zu ermitteln. Für die Bestimmung der beiden Prozeßmengen bietet es sich wiederum an, den Erreichbarkeitsgraphen des zugrundeliegenden Netzmodells auszuwerten. Dies wird hier jedoch nicht mehr im einzelnen dargelegt.

Statt dessen wird auf die früheren Ausführungen verwiesen, die beschrieben, wie sich in Erreichbarkeitsgraphen grundsätzlich finale oder optimale Schaltprozesse sowie Dead- oder Livelockmarkierungen bestimmen lassen. Die Kenntnisse der vorgenannten Schaltprozesse reichen einerseits aus, um die Menge aller intendierten Schaltprozesse zu ermitteln¹⁾. Insbesondere kann auch auf die Konstruktion von Optimierungsnetzen zurückgegriffen werden, die im voranstehenden Kapitel für die Suche nach optimalen Schaltprozessen erläutert wurden. Andererseits werden die nicht-intendierten Schaltprozesse durch die Gesamtheit aller unerwünschten Dead- und Livelockmarkierungen vollständig determiniert²⁾. Die früher definierten Robustheitsmaße können dann mit Hilfe der beiden Prozeßmengen IPROM und NIPROM ohne Schwierigkeiten berechnet werden. Schon auf diese Weise gestattet die Untersuchung von Erreichbarkeitsgraphen, detaillierte Einblicke in die Robustheit einzelner lokaler Koordinierungsentscheidungen oder ganzer Netzmodelle zu gewinnen.

Darüber hinaus lassen sich die Robustheitsuntersuchungen von Netzmodellen auf dem Fundament der Erreichbarkeitsanalyse so weit ausbauen, daß die Robustheitsanalyse die Qualität einer zweckpluralen Auswertungsrechnung³⁾ erhält. Sie gestattet es, so unterschiedliche Auswertungszwecke zu erfüllen wie die optimierende Netzanalyse einerseits oder wie die projektiven Untersuchungen über die Auswirkungen heuristischer Koordinierungsregeln andererseits. Damit dient die Robustheitsanalyse nicht nur auf der Objektebene zur Beurteilung der Entscheidungs- oder Modellrobustheit. Vielmehr erweist sie sich auf einer Metaebene selbst als ein flexibles - d.h. an unterschiedliche Auswertungszwecke anpassungsfähiges - Untersuchungskonzept.

Ausgangspunkt dieser verfeinerten Robustheitsanalyse ist ein modifizierter Robustheitsbegriff. Er gründet weiterhin auf der Robustheit von lokalen Koordinierungsentscheidungen. Sie erstrecken sich - wie früher dargelegt wurde - darauf, aus denjenigen Schaltschritten, die unter der jeweils betrachteten Netzmarkierung aktiviert sind, genau einen zur Ausführung auszuwählen. Allerdings wird der Erreichbarkeitsgraph des untersuchten Netzmodells im allgemeinen nun nicht mehr vollständig ausgewertet. Statt dessen beruht die Auswahl robuster Schaltschritte jetzt auf (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen. Jeder derart reduzierte Erreichbarkeitsgraph stellt einen Teilerreichbarkeitsgraphen $TRG(M_w, L_{sup}) = (KN_{TRG}, KA_{TRG}; bk_{TRG})$ dar, der durch seinen Wurzelknoten M_w und die zulässige Reichweite L_{sup} aller seiner Schaltwege ausgezeichnet ist⁴⁾. Seine Konstruktion wurde bereits im Kontext der Stelle/Transition-Netze eingeführt⁵⁾. Das Konzept der Teilerreichbarkeitsgraphen umfaßt als Grenzfall auch den vollständigen Erreichbarkeitsgraphen $RG(M_0, SR_S)$ eines Netzmodells⁶⁾. Daher führen die nachfolgenden Erläuterungen trotz ihrer Bezugnahme auf (M_w, L_{sup}) - "reduzierte" Erreichbarkeitsgraphen dennoch zu einer inhaltlichen Erweiterung der Robustheitsanalyse.

Auf einen Teilerreichbarkeitsgraphen lassen sich die ursprünglich betrachteten Mengen IPROM und NIPROM aller intendierten bzw. nicht-intendierten Schaltprozesse zumeist nicht mehr anwenden. Denn es ist nicht sichergestellt, daß die Ausgangsmarkierung M_0 und diejenigen erwünschten Endmarkierungen $M_{E/e}$, die in der Erreichbarkeitsmenge $RM(M_0)$ des zugrundeliegenden Netzmodells enthalten sind, auch als Wurzelknoten bzw. erwünschte Endmarkierungsknoten in den (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen vorkommen. Dies wäre jedoch notwendig, um die oben definierten intendierten und nicht-intendierten Prozeßmengen unverändert

übernehmen zu können. Statt dessen ist es jedoch möglich, die Prozeßmengendefinitionen so abzuwandeln, daß die darin enthaltenen Schaltprozesse der Vorstellung intendierter oder aber nicht-intendierter Prozesse so nahe kommen, wie es die Untersuchung eines Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ inhaltlich zuläßt.

Als nicht-intendierte Schaltprozesse lassen sich alle Prozesse ausgrenzen, die unter der Wurzelknotenmarkierung M_w des betrachteten Teilerreichbarkeitsgraphen starten und zu einer unerwünschten Deadlockmarkierung führen, die ebenso als Markierungsknoten im Teilerreichbarkeitsgraphen enthalten ist. Denn solche Prozesse würden auch im vollständigen Erreichbarkeitsgraphen $\text{RG}(M_0, \text{SR}_S)$ des analysierten Netzmodells in jener unerwünschten Deadlockmarkierung enden. Für Schaltprozesse, die im Teilerreichbarkeitsgraphen zu einer Livelockmarkierung führen, läßt sich dagegen im allgemeinen⁷⁾ nicht ausschließen, sie im Vollerreichbarkeitsgraphen noch so fortsetzen zu können, daß schließlich doch noch eine der erwünschten Endmarkierungen erreicht wird. Daher kommen für einen Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ als nicht-intendierte Schaltprozesse nur solche Prozesse in Betracht, die in unerwünschten Deadlockmarkierungen enden. Folglich gilt für die modifizierte Menge $\text{NIPROM}_{\text{TRG}}$ aller Schaltprozesse, die im (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen vermieden werden sollen:

$$\text{NIPROM}_{\text{TRG}} = \{ \text{PRO}_{w,r} : \exists (L \in \mathcal{N}_0) : L \leq L_{\text{sup}} \wedge \text{PRO}_{w,r}(\text{SF}_L, \text{MF}_L) \in \text{PROM} \wedge \dots$$

$$\text{MF}_L = (M_w, \dots, M_r) \wedge \text{DEAD}(M_r) \wedge M_r \notin \text{ME} \}$$

Im Hinblick auf die intendierten Schaltprozesse wird oftmals der Fall eintreten, daß im Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ überhaupt keine Schaltwege enthalten sind, die sachlich zulässige - also finale - Schaltprozesse darstellen. Dies folgt aus dem oben angesprochenen Sachverhalt, daß die Knotenmenge KN_{TRG} des Teilerreichbarkeitsgraphen weder die Ausgangsmarkierung M_0 noch Markierungen aus der Menge ME erwünschter Endmarkierungen zu umfassen braucht. In einem solchen Fall existieren a fortiori im Teilerreichbarkeitsgraphen auch keine Schaltwege, die formal zulässigen Schaltprozessen entsprechen⁸⁾. Diese Schwierigkeiten lassen sich jedoch überwinden. Zu diesem Zweck werden für die Untersuchung eines Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ folgende Vereinbarungen getroffen:

- Falls die Knotenmenge KN_{TRG} des Teilerreichbarkeitsgraphen mindestens eine erwünschte Endmarkierung $M_{E/e}$ aus der Endmarkierungsmenge ME des zugrundeliegenden Netzmodells enthält, gilt für die Menge ME_{TRG} erwünschter Endmarkierungen, die für die Untersuchung des Teilerreichbarkeitsgraphen relevant ist: Sie umfaßt alle erwünschten Endmarkierungen $M_{E/e}$ aus der Endmarkierungsmenge ME des Netzmodells, die zugleich auch in der Knotenmenge des Teilerreichbarkeitsgraphen enthalten sind: $\text{ME}_{\text{TRG}} = \{ M_r : M_r \in \text{KN}_{\text{TRG}} \wedge M_r \in \text{ME} \}$.
- Wenn die Knotenmenge KN_{TRG} des Teilerreichbarkeitsgraphen keine erwünschte Endmarkierung $M_{E/e}$ aus der Endmarkierungsmenge ME des zugrundeliegenden Netzmodells enthält, umfaßt die Menge ME_{TRG} alle lebendigen Markierungen aus der Knotenmenge des untersuchten Teilerreichbarkeitsgraphen: $\text{ME}_{\text{TRG}} = \{ M_r : M_r \in \text{KN}_{\text{TRG}} \wedge \text{LEB}(M_r) \}$. Es handelt sich dann um vorläufig erwünschte Endmarkierungen, die nur angesichts der begrenzten Reichweite des Teilerreichbarkeitsgraphen TRG als "End"markierungen behandelt werden⁹⁾.
- Es werden nicht diejenigen Formalziele betrachtet, die im zugrundeliegenden Netzmodell die Vorziehungswürdigkeit finaler Schaltprozesse bewerten¹⁰⁾. Statt dessen werden subsidiäre Formalziele herangezogen, die zu den Formalzielen des Netzmodells in einer Mittel-Zweck-Relation stehen. Dann stellt das Ausmaß, in dem die subsidiären Formalziele durch Schaltprozesse mit korrespondierenden Schaltwegen im Teilerreichbarkeitsgraphen erfüllt werden, einen *Indikator* für das Ausmaß dar, in dem sich die Formalziele des Netzmodells durch dort ausgeführte finale Schaltprozesse erfüllen lassen¹¹⁾. Alle Schaltprozesse $\text{PRO}_{w,E/e}(\text{SF}_L, \text{MF}_L)$,

die von der Wurzelmarkierung M_w des Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ zu einer Markierung aus seiner Menge ME_{TRG} (vorläufig) erwünschter Endmarkierungen führt, werden in der Menge $\text{ZFPROM}_{\text{TRG}}$ formal zulässiger Schaltprozesse zusammengefaßt. Das Prädikat $\text{NDOM}(\text{PRO}_{w.E/e})$ der Nichtdominiertheit wird auf die subsidiären Formalziele bezogen, die für den Teilerreichbarkeitsgraphen eingeführt wurden, und dann als $\text{NDOM}_{\text{TRG}}(\text{PRO}_{w.E/e})$ notiert.

Unter den voranstehend skizzierten Vereinbarungen läßt sich die modifizierte Menge $\text{IPROM}_{\text{TRG}}$ aller Schaltprozesse, die vor dem Hintergrund eines Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ intendiert werden, festlegen durch:

$$\text{IPROM}_{\text{TRG}} = \{ \text{PRO}_{w.E/e} : \exists (M_{E/e} \in \text{ME}_{\text{TRG}}) \exists (L \in \mathcal{N}_0) : L \leq L_{\text{sup}} \wedge \dots \\ \text{PRO}_{w.E/e}(\text{SF}_L, \text{MF}_L) \in \text{ZFPROM}_{\text{TRG}} \wedge \text{NDOM}_{\text{TRG}}(\text{PRO}_{w.E/e}) \}$$

Auf dieser Grundlage lassen sich die Funktionen und Prädikate, die früher auf der Grundlage vollständiger Erreichbarkeitsgraphen eingeführt wurden, ohne Schwierigkeiten so modifizieren, daß sie für Robustheitsanalysen auf dem Fundament von (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen analog gelten. Die Funktions- und Prädikatsbezeichnungen werden dabei auf die jeweils vorausgesetzten Wurzelmarkierungen M_w der Teilerreichbarkeitsgraphen und ihre zulässigen Reichweiten L_{sup} bezogen. Für jeden Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}}) = (\text{KN}_{\text{TRG}}, \text{KA}_{\text{TRG}}; \text{bk}_{\text{TRG}})$ gilt:

$$\text{chance}_{\text{TRG}}: \text{KN}_{\text{TRG}} \times \text{SSM} \rightarrow [0;1]$$

$$(M_{\text{nf}}, \text{SS}_a) \rightarrow \text{chance}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a); \quad \text{falls } M_{\text{nf}} \in (\text{KN}_{\text{TRG}} - \text{ME}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}})$$

mit:

$$\text{chance}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a) = \#(\{ \text{PRO}_{\text{nf.E/e}}(\text{SF}_{L/e}, \text{MF}_{L/e}) \}) : \#(\text{IPROM}_{\text{TRG}})$$

und:

$$\begin{aligned} \#(\{ \text{PRO}_{\text{nf.E/e}}(\text{SF}_{L/e}, \text{MF}_{L/e}) \}) &= \dots \\ \#(\{ \text{PRO}_{\text{nf.E/e}}(\text{SF}_{L/e}, \text{MF}_{L/e}) : \text{PRO}_{\text{nf.E/e}}(\text{SF}_{L/e}, \text{MF}_{L/e}) \in \text{PROM} \wedge \dots \\ &(\text{SF}_{L/e} = (\text{SS}_{a(1/e)}, \dots, \text{SS}_{a(L/e)}) \wedge \text{MF}_{L/e} = (M_{r(0/e)}, M_{r(1/e)}, \dots, M_{r(L/e)})) \\ &\rightarrow (\text{SS}_{a(1/e)} = \text{SS}_a \wedge M_{r(0/e)} = M_{\text{nf}} \wedge \dots \\ &(\forall (\text{PRO}_{w.E/e}(\text{SF}_{L/G}, \text{MF}_{L/G}) \in \text{IPROM}_{\text{TRG}}): \dots \\ &\text{SF}_{L/G} = (\text{SS}_{a(1/G)}, \dots, \text{SS}_{a(1/e)}, \dots, \text{SS}_{a(L/e)}) \\ &\wedge \text{MF}_{L/G} = (M_0, \dots, M_{r(0/e)}, M_{r(1/e)}, \dots, M_{r(L/e)})) \}) \end{aligned}$$

risiko_{TRG}: $\text{KN}_{\text{TRG}} \times \text{SSM} \rightarrow [0;1]$

$(M_{\text{nf}}, \text{SS}_a) \rightarrow \text{risiko}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a); \quad \text{falls } M_{\text{nf}} \in (\text{KN}_{\text{TRG}}\text{-ME}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}})$

mit:

$\text{risiko}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a) = \#(\{\text{PRO}_{\text{nf},r}(\text{SF}_{L/r}, \text{MF}_{L/r})\}) : \#(\text{NIPROM}_{\text{TRG}})$

und:

$\#(\{\text{PRO}_{\text{nf},r}(\text{SF}_{L/r}, \text{MF}_{L/r})\}) = \dots$
 $\#(\{\text{PRO}_{\text{nf},r}(\text{SF}_{L/r}, \text{MF}_{L/r}) : \text{PRO}_{\text{nf},r}(\text{SF}_{L/r}, \text{MF}_{L/r}) \in \text{PROM} \wedge \dots$
 $(\text{SF}_{L/r} = (\text{SS}_{a(1/r)}, \dots, \text{SS}_{a(L/r)}) \wedge \text{MF}_{L/r} = (M_{r(0/r)}, M_{r(1/r)}, \dots, M_{r(L/r)})$
 $\rightarrow (\text{SS}_{a(1/r)} = \text{SS}_a \wedge M_{r(0/r)} = M_{\text{nf}} \wedge \dots$
 $(\forall (\text{PRO}_{w,r}(\text{SF}_{L/G}, \text{MF}_{L/G}) \in \text{NIPROM}_{\text{TRG}}) : \dots$
 $\text{SF}_{L/G} = (\text{SS}_{a(1/G)}, \dots, \text{SS}_{a(1/r)}, \dots, \text{SS}_{a(L/r)})$
 $\wedge \text{MF}_{L/G} = (M_0, \dots, M_{r(0/r)}, M_{r(1/r)}, \dots, M_{r(L/r)})\})\})$

rob_{TRG}: $\text{KN}_{\text{TRG}} \times \text{SSM} \rightarrow [-1;+1]$

$(M_{\text{nf}}, \text{SS}_a) \rightarrow \text{rob}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a); \quad \text{falls } M_{\text{nf}} \in (\text{KN}_{\text{TRG}}\text{-ME}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}})$

mit:

$\text{rob}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a) = \alpha_c \cdot \text{chance}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a) - \alpha_r \cdot \text{risiko}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a)$

$\text{ROB}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a)$

$:\Leftrightarrow M_{\text{nf}} \in (\text{KN}_{\text{TRG}}\text{-ME}) \wedge \text{LEB}(M_{\text{nf}}) \wedge \text{AKT}(\text{SS}_a, M_{\text{nf}}) \wedge \dots$

$\text{rob}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_a) = \max \{ \text{rob}_{\text{TRG}}(M_{\text{nf}}, \text{SS}_k) : \text{SS}_k \in \text{SSM} \wedge \text{AKT}(\text{SS}_k, M_{\text{nf}}) \}$

mrob_{TRG}: $\text{KN}_{\text{TRG}} \rightarrow [-1;+1]$

mit:

$$M_r \rightarrow \text{mrob}_{\text{TRG}}(M_r) = \begin{cases} +1; & \text{für } M_r \in \text{ME} \\ \text{rob}_{\text{TRG}}(M_r, \text{SS}_a); & \text{für } \text{LEB}(M_r) \wedge M_r \in (\text{KN}_{\text{TRG}}\text{-ME}) \\ & \wedge \text{ROB}_{\text{TRG}}(M_r, \text{SS}_a) \\ -1; & \text{für } \text{DEAD}(M_r) \\ & \wedge M_r \in (\text{KN}_{\text{TRG}}\text{-ME}) \end{cases}$$

$\underline{\text{grob}}_{\text{TRG}}(\text{SN}) = (\text{mrob}_{\text{TRG}}(M_r) : M_r \in \text{KN}_{\text{TRG}})$

$\text{grob}_w(\text{SN}) = \text{mrob}_{\text{TRG}}(M_w)$

Es ist offensichtlich, daß die voranstehenden Konstrukte für die Robustheitsanalyse auf der Basis von (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ in die ursprünglich definierten Robustheitsmaße auf der Grundlage vollständiger Erreichbarkeitsgraphen $\text{RG}(M_0, \text{SR}_S)$ übergehen, sofern die Definitionen der intendierten und nicht-intendierten Prozeßmengen entsprechend angepaßt¹²⁾ sowie *unechte* Teilerreichbarkeitsgraphen betrachtet werden¹³⁾. Denn der Wur-

zelknoten M_w eines unechten Teilerreichbarkeitsgraphen stimmt mit dem Ausgangsmarkierungsknoten M_0 des vollständigen Erreichbarkeitsgraphen überein: $M_w = M_0$. Die zulässige Reichweite L_{sup} eines unechten Teilerreichbarkeitsgraphen ist notwendig mindestens so groß wie die tatsächliche Reichweite L_{max} des vollständigen Erreichbarkeitsgraphen, so daß stets $L_{sup} \geq L_{max}$ gilt. Folglich erfüllt jeder unechte Teilerreichbarkeitsgraph $TRG(M_w, L_{sup})$ die Beziehung $KN_{TRG} = RM(M_0)$. Hinzu kommt die Substitution der beiden Mengen $IPROM_{TRG}$ und $NIPROM_{TRG}$ intendierter bzw. nicht-intendierter Schaltprozesse durch die beiden analogen Prozeßmengen $IPROM$ bzw. $NIPROM$, die oben für vollständige Erreichbarkeitsgraphen definiert wurden. Die Gesamtheit der vorgenannten Transformationen überführt die Robustheitskonstrukte von unechten Teilerreichbarkeitsgraphen in diejenigen für vollständige Erreichbarkeitsgraphen.

Vollständige Erreichbarkeitsgraphen $RG(M_0, SR_S)$ und Teilerreichbarkeitsgraphen $TRG(M_w, L_{sup})$ zusammen bilden ein reichhaltiges Fundament für die Beurteilung der Robustheit von Netzmodellen. Das hierdurch eröffnete breite Analysespektrum läßt sich mit der Hilfe von fünf herausragenden Analysevarianten verdeutlichen. Die erste Variante besteht darin, auf der Grundlage von vollständigen Erreichbarkeitsgraphen robuste (erste) Schritte zu bestimmen und die Robustheit von einzelnen Auswahlentscheidungen oder ganzen Netzmodellen quantitativ zu beurteilen. Dies wurde bereits oben ausführlich erläutert, so daß hierauf nicht weiter eingegangen zu werden braucht.

Die zweite Variante der Robustheitsanalyse fällt inhaltlich mit der optimierenden Netzanalyse zusammen¹⁴⁾. Dies mag auf der ersten Blick erstaunen, weil Netzoptima und Robustheitsmaße zunächst vollkommen unterschiedlich konzeptualisiert wurden. Dennoch liefert die Robustheitsanalyse eines Netzmodells mindestens einen optimalen Schaltprozeß, wenn folgende Bedingungen erfüllt sind:

- Das Formalzielsystem besteht aus genau einem Extremierungsziel.
- Der Entscheidungsträger verhält sich risikoblind, so daß seine Unsicherheitspräferenz durch die Parameter $\alpha_c=1$ und $\alpha_r=0$ festgelegt ist.
- Es wird ein vollständiger Erreichbarkeitsgraph mit seinen Mengen $IPROM$ und $NIPROM$ intendierter bzw. nicht-intendierter Schaltprozesse¹⁵⁾ zugrundegelegt.
- Unter der Ausgangsmarkierung M_0 wird ein robuster Schaltschritt SS_a ausgewählt, der das Prädikat $ROB(M_0, SS_a)$ erfüllt.

Unter diesen Voraussetzungen garantiert die Auswahl eines robusten ersten (Schalt-)Schritts die Selektion eines optimalen Schaltprozesses¹⁶⁾. Denn die Risikoblindheit des Entscheidungsträgers führt dazu, daß der ausgewählte Schaltschritt SS_a unter der Ausgangsmarkierung M_0 der erste Schaltschritt aus einem Schaltprozeß $PRO_{0, M/e}(SF_L, MF_L)$ ist, der zur Menge $IPROM$ der intendierten Schaltprozesse gehört. Jeder Schaltprozeß aus der Menge $IPROM$, der darin das Prädikat $NDOM(PRO_{0, M/e})$ erfüllt, wird von keinem anderen sachlich zulässigen Schaltprozeß dominiert. Da ein monodimensionales Formalzielsystem mit Extremierungscharakter vorausgesetzt wurde, handelt es sich bei dem Schaltprozeß $PRO_{0, M/e}(SF_L, MF_L)$ mit dem ausgewählten ersten Schaltschritt SS_a nicht nur um einen nicht-dominierten, sondern sogar um einen optimalen Schaltprozeß. Dieser optimale Schaltprozeß $PRO_{0, M/e}(SF_L, MF_L)$ fällt während der Ermittlung des robusten ersten (Schalt-)Schritts SS_a als "Nebenergebnis" an. Daher ist mindestens ein optimaler Schaltprozeß $PRO_{0, M/e}(SF_L, MF_L)$ bekannt, sobald für die Ausgangsmarkierung M_0 des untersuchten Netzmodells ein erster robuster Schaltschritt SS_a berechnet worden ist. Zugleich steht mit dem Schaltwert $sw(PRO_{0, M/e}(SF_L, MF_L))$ dieses Schaltprozesses auch das gesuchte Netzoptimum fest.

Also erweist sich die Robustheitsanalyse unter den o.a. Prämissen tatsächlich als eine optimierende Netzanalyse¹⁷⁾. Über die konventionelle Bestimmung von Netzoptima hinaus, die bereits erläutert wurde, besitzt die optimierende Robustheitsanalyse noch einen charakteristi-

schen Vorzug¹⁸⁾. Denn sie vermag auch nachträglich auf unvorhergesehene Veränderungen der Produktionssituation zu reagieren, die sich in Modifizierungen des Netzmodells und seines Erreichbarkeitsgraphen niederschlagen. Dies gilt allerdings nur für jene günstigen Fälle, in denen sich eine unerwartete Situationsveränderung im Netzmodell als eine Markierungsabweichung niederschlägt¹⁹⁾. Darauf beschränken sich die anschließenden Überlegungen²⁰⁾. Die Behandlung solcher Markierungsabweichungen stellt zugleich die dritte von den fünf oben angekündigten Varianten dar, die sich für die Robustheitsanalyse von Netzmodellen als besonders interessant erweisen.

Eine Markierungsabweichung liegt vor, wenn die aktuelle Netzmarkierung, die aufgrund der unvorhergesehenen Entwicklung der Produktionssituation hervorgerufen worden ist, nicht mit der geplanten Markierung übereinstimmt. Bei der geplanten Markierung handelt es sich um diejenige Markierung des Netzmodells, die bei der Ausführung des ausgewählten optimalen Schaltprozesses $PRO_{0,M/e}(SF_L, MF_L)$ bis zum Eintreten der unerwarteten Situationsveränderung hätte erreicht werden müssen. Eine solche Markierungsabweichung kann hinsichtlich der angestrebten Netzoptimierung zwei unterschiedliche Qualitäten besitzen:

- Entweder handelt es sich um eine unschädliche Markierungsabweichung. Dies ist genau dann der Fall, wenn die aktuelle Netzmarkierung in der Markierungsfolge MF_L von mindestens einem anderen intendierten Schaltprozeß $PRO_{0,M/e}(SF_L, MF_L)$ aus der Prozeßmenge IPROM enthalten ist.
- Oder es liegt eine schädliche Markierungsabweichung vor. Sie resultiert immer dann, wenn in der Prozeßmenge IPROM kein intendierter Schaltprozeß vorkommt, dessen Markierungsfolge die aktuelle Netzmarkierung umfaßt.

Die Behandlung von unschädlichen Markierungsabweichungen bereitet keine Schwierigkeiten. Es braucht lediglich vom ursprünglich ausgewählten optimalen Schaltprozeß $PRO_{0,M/e}(SF_L, MF_L)$ zum neuen, aber weiterhin optimalen Schaltprozeß $PRO_{0,M/e}(SF_L, MF_L)$ übergewechselt zu werden. Hinsichtlich dieser Möglichkeit, auf unvorhergesehene Situationsveränderungen mit einem optimalen Prozeßwechsel reagieren zu können, offenbart sich der besondere Robustheitsbezug der hier entwickelten optimierenden Robustheitsanalyse von Netzmodellen²¹⁾.

Allerdings zeigt die praktische Erfahrung im Umgang mit Netzmodellen, daß ihre Mengen IPROM aller intendierten Schaltprozesse weitaus weniger Elemente umfassen als ihre Mengen PROM aller zulässigen Schaltprozesse. Daher fällt die Chance, daß eine Markierungsabweichung unschädlich bleibt, weil sie durch einen optimalen Prozeßwechsel kompensiert werden kann, im allgemeinen sehr klein aus. Immerhin besitzen Netzmodelle in dieser Hinsicht zwei Vorteile. Erstens gestatten sie, die Option eines optimalen Prozeßwechsels zu nutzen, *falls* sie sich bei einer unerwarteten Entwicklung der Produktionssituation bieten sollte. Zweitens ermöglichen Netzmodelle, die Chance für solche optimalen Prozeßwechsel zu quantifizieren. Sie läßt sich durch ein einfaches Robustheitsmaß $rob_{op}(SN)$ messen. Es stellt für ein vorliegendes Netzmodell SN den Anteil der intendierten optimalen an allen zulässigen, aber nicht zyklisch wiederholten²²⁾ Schaltprozessen fest:

$$rob_{op}(SN) = \#(IPROM) : \#(PROM_B)$$

Zumeist wird die aktuelle Netzmarkierung, die im Netzmodell nach dem Eintreten einer unvorhergesehenen Produktionssituation vorliegt, jedoch nicht zur Markierungsfolge eines optimalen Schaltprozesses gehören. Auf eine solche schädliche Markierungsabweichung kann auf zwei Weisen reagiert werden. Erstens ist es möglich, die aktuelle Netzmarkierung als neue Ausgangsmarkierung anzusetzen. Für das derart aktualisierte Netzmodell wird eine neue optimierende Robustheitsanalyse ausgeführt. Dies bereitet jedoch erheblichen Analyseaufwand²³⁾. Daher wird dieser Weg bei der praktischen Anwendung von Netzmodellen im allgemeinen nicht beschritten werden²⁴⁾.

Statt dessen erweist sich eine zweite Reaktionsmöglichkeit als weitaus interessanter. Bei ihr wird anhand des Erreichbarkeitsgraphen des Netzmodells untersucht, ob vom Knoten der aktuellen Netzmarkierung zu einem Markierungsknoten gelangt werden kann, dessen Markierung in der Markierungsfolge eines intendierten optimalen Schaltprozesses aus der Menge IPROM enthalten ist²⁵⁾. Wenn dies der Fall ist²⁶⁾, wird die aufgefundene Markierung aus dem optimalen Schaltprozeß als Auffangmarkierung bezeichnet. Der Rest des zugehörigen optimalen Schaltprozesses, der mit der Auffangmarkierung startet und zur erwünschten Endmarkierung dieses Schaltprozesses führt, wird entsprechend als Auffangprozeß angesprochen. Unter diesen Voraussetzungen existiert mindestens²⁷⁾ ein Rückkehrprozeß, der die aktuelle Netzmarkierung als Start- und die Auffangmarkierung als Zielmarkierung besitzt. Nach dessen Ausführen kann mit einem der optimalen Schaltprozesse fortgefahren werden, der die Auffangmarkierung enthält. An dessen Ende wird eine der erwünschten Endmarkierungen erreicht. Der Gesamtprozeß, der sich zwischen der Ausgangsmarkierung des Netzmodells und der schließlich erreichten Endmarkierung erstreckt, besteht aus drei Teilprozessen: dem ursprünglich ausgewählten optimalen Schaltprozeß bis zum Eintreten der schädlichen Markierungsabweichung, dem Rückkehrprozeß und dem Auffangprozeß. Dieser Gesamtprozeß gehört per constructionem nicht mehr zur intendierten Prozeßmenge IPROM²⁸⁾. Deshalb muß es sich um einen suboptimalen Schaltprozeß handeln²⁹⁾.

Dennoch wird es oftmals vorteilhafter sein, mit einem suboptimalen Rückkehrprozeß auf eine unvorhergesehene Veränderung der Produktionssituation zu reagieren, als bei Eintritt der Situationsveränderung mit der aktuellen Netzmarkierung als revidierter Ausgangsmarkierung ein neues Netzoptimum zu bestimmen. Denn die Mindererfüllung des Formalzielsystems, die durch die Auswahl des suboptimalen Rückkehrprozesses gegenüber einem neu bestimmten optimalen Schaltprozeß in Kauf genommen wird, muß mit demjenigen Ressourcenmehreinsatz verglichen werden, um den die Ermittlung eines neuen Optimums die Konstruktion eines Rückkehrprozesses übertrifft³⁰⁾. Diese Abwägung wird zwar im konkreten Einzelfall nicht durchgeführt werden. Denn es würde unnötig hohen Aufwand verursachen, *sowohl* ein neues Optimum *als auch* einen Rückkehrprozeß zu bestimmen, um beide Reaktionsmöglichkeiten gegeneinander abzuwägen. Aber Erfahrungen aus dem praktischen Umgang mit Modellen für komplexe Koordinierungsprobleme legen die Erwartungshaltung nahe, daß es tendenziell günstiger ist, nach einer Abweichung zu bereits ermittelten - nunmehr aber suboptimal gewordenen - Problemlösungen zurückzukehren, anstatt eine neue optimale Lösung zu berechnen.

Darüber hinaus bietet sich die Konstruktion von Rückkehrprozessen³¹⁾ an, um die Robustheitsanalyse von Netzmodellen zu einem Instrument der Anpassungsplanung³²⁾ auszubauen, das der Idee des "turnpike scheduling"³³⁾ entspricht. Diese Anpassungsplanung ist keineswegs auf die voranstehend erörterte optimierende Robustheitsanalyse beschränkt. Vielmehr gelten die nachfolgenden Anmerkungen für jede Robustheitsanalyse, also z.B. auch bei der Verfolgung von Satisfizierungszielen und bei mehrdimensionalen Formalzielsystemen. Die Ermittlung eines Rückkehrprozesses kann als ein Koordinierungsproblem sui generis formuliert werden³⁴⁾. Dabei wird das zugrundeliegende Netzmodell als Anpassungsmodell übernommen, allerdings in drei wesentlichen Merkmalen modifiziert. Erstens wird die aktuelle Ausgangsmarkierung, die nach dem Eintreten einer schädlichen Markierungsabweichung vorliegt, als Ausgangsmarkierung des Anpassungsmodells festgelegt. Zweitens werden alle Netzmarkierungen, die in den Markierungsfolgen der intendierten Schaltprozesse aus der Menge IPROM des zugrundeliegenden Netzmodells enthalten sind, als erwünschte Endmarkierungen des Anpassungsmodells ausgezeichnet. Drittens können für die Bearbeitung des Rückkehrproblems anpassungsspezifische Formalziele vorgegeben werden³⁵⁾. Das Rückkehrproblem ist gelöst, sobald im Anpassungsmodell ein Schaltprozeß aufgefunden ist, der von der neuen Ausgangsmarkierung zu einer der neuen erwünschten Endmarkierungen führt und hierbei das Formalzielsystem des Anpassungsmodells in der intendierten Weise erfüllt³⁶⁾. Dieser Schaltprozeß stellt per constructionem den gesuchten Rückkehrprozeß für das zugrundeliegende Netzmodell dar.

Für die Ermittlung eines Rückkehrprozesses im Anpassungsmodell können alle Konzepte angewandt werden, die zur Konstruktion intendierter Schaltprozesse in Netzmodellen geeignet sind. Es liegt nicht mehr im Erkenntnisinteresse dieser Arbeit, das breite Spektrum solcher Konstruktionskonzepte zu entfalten. Statt dessen wird in exemplarischer Weise auf die Konzepte zur Ermittlung optimaler Schaltprozesse verwiesen, die im voranstehenden Kapitel ausführlich beschrieben wurden. Sie lassen sich ebenso einsetzen, um in Anpassungsmodellen optimale³⁷⁾ Rückkehrprozesse zu bestimmen. Alle früher angesprochenen Erkenntnisse aus den Bereichen der Graphentheorie, der Dynamischen Programmierung und der Erforschung Künstlicher Intelligenz, die in die Ermittlung optimaler Schaltprozesse eingebunden werden können, lassen sich daher ebenso einsetzen, um in Anpassungsmodellen optimale Rückkehrprozesse zu suchen. Sie befruchten die Bemühungen, durch Robustheitsuntersuchungen von Netzmodellen auf unvorhergesehene Veränderungen von Produktionssituationen durch entsprechende Rückkehrprozesse zu reagieren. Die derart bereicherte Robustheitsanalyse stellt einen bemerkenswerten Beitrag zur Konzipierung von Anpassungsplanungen dar.

Die beiden letzten Beispiele für Robustheitsuntersuchungen von Netzmodellen verlassen den Bereich optimierender Netzanalysen endgültig³⁸⁾. Statt dessen erstrecken sie sich auf projektive Netzanalysen, mit deren Hilfe die Wirkungen von heuristischen Entscheidungsregeln untersucht werden sollen³⁹⁾. Jeder dieser Heurismen dient dazu, aus dem Entscheidungsspielraum einer aktuell betrachteten Netzmarkierung genau einen der dort aktivierten Schaltschritte als lokale Entscheidungsalternative auszuwählen. Dabei wird zwischen Heurismen mit begrenzter Vorausschau auf der einen Seite und Heurismen ohne Vorausschau auf der anderen Seite unterschieden.

Zunächst wird auf den einfacheren Fall der vorausschaufreien oder "blinden"⁴⁰⁾ Heurismen eingegangen. Sie sind dadurch gekennzeichnet, daß sich das heuristische Entscheidungskriterium ausschließlich auf jene Informationen erstreckt, die unter der aktuellen Netzmarkierung hinsichtlich der dort aktivierten Schaltschritte zur Verfügung stehen. Von allen Auswirkungen, die diese Auswahl einer lokalen Entscheidungsalternative auf zukünftig auswählbare Entscheidungsalternativen⁴¹⁾ auszuüben vermag⁴²⁾, wird dabei abgesehen⁴³⁾. Solche vorausschaufreien Heurismen stellen den Standardfall aller heuristischen Entscheidungsregeln dar, die in betriebswirtschaftlichen Kontexten für die Bearbeitung von Koordinierungsproblemen thematisiert werden⁴⁴⁾.

Die Robustheitsanalyse umfaßt alle vorausschaufreien Entscheidungsregeln, sobald eine charakteristische Einschränkung erfolgt: Ihr werden ausschließlich (M_w, L_{sup}) -reduzierte Erreichbarkeitsgraphen zugrundegelegt, deren zulässige Reichweite den Wert $L_{sup}=1$ annimmt. Dann wird die Auswahl eines robusten Schaltschritts SS_a , der unter der aktuellen Netzmarkierung M_w aktiviert ist, nur noch von jenen Schaltprozessen determiniert, die unter der Markierung M_w starten und maximal⁴⁵⁾ einen Schaltschritt lang sind. Darüber hinaus wird im allgemeinen auch ein risikoblinder Entscheidungsträger unterstellt, der die Unsicherheitspräferenz $\alpha_c=1$ und $\alpha_r=0$ besitzt. Dadurch wird das Risiko, daß Auswahlentscheidungen zu unerwünschten Deadlockmarkierungen führen⁴⁶⁾, nicht explizit beachtet. Dies ist zwar nicht unbedingt erforderlich⁴⁷⁾, vereinfacht aber die heuristisch basierte Robustheitsanalyse⁴⁸⁾.

Unter diesen Voraussetzungen gilt für jede lokale Auswahlentscheidung unter einer aktuellen Netzmarkierung M_w : Falls es sich um eine lebendige Markierung M_w handelt, werden zunächst alle Schaltprozesse $PRO_{w,f}(SF_1, MF_1)$ betrachtet, die durch das Ausführen je eines aktivierten Schaltschritts SS_a zu einer schaltschrittspezifischen unmittelbaren Nachfolgermarkierung M_f führen: $M_w[SS_a]M_f$. Die Schaltwerte $\underline{sw}(PRO_{w,f}(SF_1, MF_1))$ der Schaltprozesse hängen wegen $SF_1=(SS_a)$ nur von den Schaltwerten $\underline{sw}(SS_a)$ der jeweils ausführbaren Schaltschritte ab. Daher wird ausschließlich durch diese schaltschrittspezifischen Schaltwerte $\underline{sw}(SS_a)$ bestimmt, welche Schaltprozesse unter der aktuellen Markierung M_w zur Menge $IPROM_{TRG}$ der dort intendierten Schaltprozesse gehören. Im Regelfall führt die Definition heuristischer Entscheidungsregeln dazu, daß die intendierte Prozeßmenge nur höchstens ein Element⁴⁹⁾ enthält⁵⁰⁾:

$0 \leq \#(\text{IPROM}_{\text{TRG}}) \leq 1$. Sofern die Prozeßmenge $\text{IPROM}_{\text{TRG}}$ einen intendierten Schaltprozeß $\text{PRO}_{w,f}(\text{SF}_1, \text{MF}_1)$ enthält, fällt wegen der Außerachtlassung des Risikoaspekts die Auswahl des robusten Schritts unter der Markierung M_w notwendig auf jenen einen Schaltschritt SS_a , der unter der Markierung M_w aktiviert ist und der den einen intendierten Schaltprozeß einleitet⁵¹). Also wird durch die Robustheitsanalyse, die auf der Grundlage eines $(M_w, 1)$ -reduzierten Erreichbarkeitsgraphen bei Risikoblindheit erfolgt, unter einer lebendigen Markierung M_w genau ein dort aktivierter Schaltschritt SS_a nach Maßgabe der jeweils eingesetzten heuristischen Entscheidungsregeln ausgewählt. Damit ist die oben vorgetragene Behauptung bewiesen, die Robustheitsanalyse erlaube auch, Heurismen zur Lösung lokaler Entscheidungsprobleme anzuwenden.

Allerdings ist auf zwei Besonderheiten hinzuweisen, die von heuristischen Entscheidungsregeln gewöhnlich nicht beachtet werden, sich jedoch im Rahmen der heuristischen Robustheitsanalyse unmittelbar aufdrängen. Sie betreffen beide die Berücksichtigung von Deadlockmarkierungen. Die voranstehende Argumentation betraf ausschließlich lebendige Markierungen M_w . Den ebenso denkmöglichen Fall, daß die aktuelle Markierung M_w tot ist, wird bei heuristischen Konzepten für die Lösung von Koordinierungsproblemen im allgemeinen nicht wahrgenommen⁵²). Für die Robustheitsanalyse bereitet dies jedoch keine Schwierigkeiten. Unter einer toten Markierung M_w ist per definitionem überhaupt kein Schaltschritt aktiviert. Folglich sind die Prozeßmengen $\text{ZFPRO}_{\text{TRG}}$ und a fortiori auch $\text{IPROM}_{\text{TRG}}$ leer. Daher besteht unter der aktuellen Deadlockmarkierung M_w auch kein Entscheidungsspielraum, irgendeinen Schaltschritt auszuwählen.

Die Beachtung von Deadlocks reicht tatsächlich noch weiter. Denn es werden nur solche Schaltprozesse der Länge $L=1$ berücksichtigt, die zu Folgemarkierungen M_f führen, die keine unerwünschten Deadlockmarkierungen darstellen. Dies folgt zwangsläufig aus der Definition der Prozeßmengen $\text{ZFPRO}_{\text{TRG}}$ und $\text{IPROM}_{\text{TRG}}$: Sie können nur solche Schaltprozesse enthalten, die entweder zu einer erwünschten Endmarkierung aus der Endmarkierungsmenge ME des zugrundeliegenden Netzmodells führen oder aber zumindest eine lebendige Markierung M_f erreichen. Daher gelangen bei der Auswahl robuster Schaltschritte zwar weiterhin gewöhnliche "blinde" Heurismen zur Anwendung. Sie verhalten sich jedoch nur in dem Sinn vorausschaufrei, daß Schaltschritte, die in späteren Netzmarkierungen ausgewählt werden könnten, die heuristischen Auswahlentscheidungen unter den aktuellen Netzmarkierungen nicht beeinflussen. Dennoch wird die Auswahlentscheidung um einen - wenn auch geringen - Vorausschauaspekt erweitert. Denn es werden alle Schaltschritte, die wegen ihrer Aktivierung unter der aktuellen Markierung M_w zur Wahl stehen, ausgeschlossen, falls sie zu einer Folgemarkierung führten, die eine unerwünschte Deadlockmarkierung darstellte. Dies bedeutet, daß im Extremfall die intendierte Prozeßmenge $\text{IPROM}_{\text{TRG}}$ auch leer sein kann, obwohl die aktuelle Markierung M_w nicht tot ist. Dieser Fall $\#(\text{IPROM}_{\text{TRG}}) = 0$ tritt ein, wenn alle Schaltschritte, die unter der Markierung M_w aktiviert sind, jeweils eine Deadlockmarkierung als Folgemarkierung M_f hervorbringen würden. Dann wird überhaupt kein Schaltschritt ausgewählt⁵³).

Die Bereicherung von heuristischen Entscheidungsregeln um Vorausschauaspekte läßt sich weiter ausbauen. Dies geschieht durch Heurismen mit begrenzter Vorausschau⁵⁴). Sie unterscheiden sich von den voranstehenden blinden Heurismen dadurch, daß sie bei der Auswahl eines Schaltschritts unter der aktuellen Netzmarkierung berücksichtigen, in welcher Weise sich dies auf zukünftig ausführbare Schaltschritte auswirkt⁵⁵). Dabei wird das Ausmaß, in dem solche Fernwirkungen Beachtung finden, durch eine maximale Vorausschauweite begrenzt. Die maximale Vorausschauweite gibt an, wie lang die Schaltprozesse im untersuchten Netzmodell höchstens sein dürfen, die für die Schaltschrittauswahl unter der aktuellen Netzmarkierung relevant sein sollen. Diese maximale Vorausschauweite wird in der Robustheitsanalyse des Netzmodells als die zulässige Reichweite L_{sup} der zugrundegelegten Teilerreichbarkeitsgraphen $\text{TRG}(M_w, L_{\text{sup}})$ festgelegt⁵⁶). Die Vorausschauweite muß die Bedingung $L_{\text{sup}} \geq 2$ erfüllen, um heuristische Entscheidungsregeln mit begrenzter Vorausschau von allen vorausschaufreien Heurismen überschneidungsfrei abzugrenzen⁵⁷).

Durch eine heuristische Entscheidungsregel mit der Vorausschauweite L_{sup} wird unter der aktuellen Netzmarkierung M_w ein Schaltschritt SS_a ausgewählt, der sich auf der Basis des vorausgesetzten (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen als ein robuster Schritt auszeichnet. Für diese Auswahl gelten die Ausführungen, die zuvor zu blinden Heurismen erfolgten, analog, so daß sie hier nicht noch einmal vorgetragen zu werden brauchen. Allerdings bedeutet jetzt die Voraussetzung eines risikoblinden Entscheidungsträgers nicht mehr, daß alle unerwünschten Deadlockmarkierungen, die in der Knotenmenge KN_{TRG} des (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen enthalten sind, in der intendierten Prozeßmenge $IPROM_{\text{TRG}}$ implizit berücksichtigt werden. Denn unter der Markierung M_w kann durchaus ein Schaltschritt SS_a aktiviert sein, der sowohl der erste Schaltschritt aus einem intendierten Schaltprozeß der Menge $IPROM_{\text{TRG}}$ ist als auch über spätere Schaltschritte schließlich in eine unerwünschte Deadlockmarkierung führen kann. Dies läßt sich nicht ausschließen, weil wegen $L_{\text{sup}} \geq 2$ grundsätzlich auch Schaltprozesse zulässig sind, deren Schaltfolgen mehrere Schaltschritte umfassen. Daher bereitet es bei vorausschauenden Heurismen durchaus einen Unterschied, ob entweder heuristische Auswahlentscheidungen bei Risikoblindheit getroffen werden oder aber das Risiko des Erreichens unerwünschter Deadlockmarkierungen beachtet wird. Im letztgenannten Fall muß für die Beurteilung von Schaltschritten, die unter der aktuellen Markierung M_w aktiviert sind, auch die nicht-intendierte Prozeßmenge $NIPROM_{\text{TRG}}$ ermittelt werden.

Anmerkungen zum Kapitel:

- 1) Vgl. dazu die Definition der intendierten Prozeßmenge IPROM. Falls das Formalzielsystem des untersuchten Netzmodells ausschließlich Satisfizierungsziele enthält, brauchen nur dessen finalen Schaltprozesse ermittelt zu werden. Ihre Gesamtheit fällt mit allen intendierten Schaltprozessen zusammen. Wenn das Formalzielsystem mindestens ein Meliorierungsziel enthält, erweisen sich nur jene finalen Schaltprozesse als intendiert, die gegenüber der zielspezifischen Referenzalternative besserstellen. Sofern das Formalzielsystem mindestens ein Extremierungsziel enthält, interessieren aus der Menge aller finalen Schaltprozesse nur noch jene, die zugleich auch optimale Schaltprozesse darstellen.
- 2) Vgl. die Definition der nicht-intendierten Prozeßmenge NIPROM.
- 3) Die begriffliche Verwandtschaft zur zweckpluralen Grundrechnung, die im Rahmen der früher angesprochenen Relativen Einzelkosten- und Deckungsbeitragsrechnung angestrebt wird, ist hier bewußt gewählt. Vgl. zu zweckpluralen ("zweckneutralen") Grundrechnungen SCHMALENBACH (1948), S. 66ff.; RIEBEL (1959), S. 214 u. 218ff.; SCHMALENBACH (1963), S. 268ff. u. 426ff.; RIEBEL (1964a), S. 84(ff.); RIEBEL (1964b), S. 144f.; LABMANN (1968), S. 21, 24 u. 159f.; HUMMEL, S. (1970), S. 53ff., 64ff., 106ff., 130ff. u. 148f.; RIEBEL (1979a), S. 787ff.; RIEBEL (1979b), S. 863ff.; LABMANN (1980b), S. 343ff., insbesondere S. 344 u. 347; RIEBEL (1981b), S. 462ff.; RIEBEL (1982), S. 102ff.; ZELEWSKI (1986a), S. 103 u. 710f.; RIEBEL (1987), S. 1161ff.; RIEBEL (1989a), S. 255f.; RIEBEL (1990), S. 39ff., 149(ff.), 170ff., 432ff., 444ff., 638ff., 696f., 701 u. 711.
- 4) Ein analoger Ansatz zur Konstruktion mehrerer Teilgraphen findet sich bei RIVETT (1980), S. 86, im Kontext von Entscheidungsbäumen. Dort klingt sogar ein vager Bezug auf robuste erste Schritte an.
- 5) Die dort erfolgten Ausführungen gelten für Synthetische Netze unverändert, da die Konstruktion von Erreichbarkeitsgraphen nicht von der Netzklasse abhängt, zu denen das zugrundeliegende Netzmodell gehört.
- 6) An früherer Stelle wurde gezeigt, daß jeder (M_w, L_{sup}) - "reduzierte" Erreichbarkeitsgraph mit dem einen vollständigen Erreichbarkeitsgraphen eines Netzmodells zusammenfällt, falls $M_w = M_0$ und $L_{sup} \geq L_{max}$ mit L_{max} als tatsächlicher Reichweite des vollständigen Erreichbarkeitsgraphen gelten. Es liegt dann ein unechter Teilerreichbarkeitsgraph vor.
- 7) Dies ist nur in dem Sonderfall möglich, in dem bekannt ist, daß es sich bei dem ausgewerteten Teilerreichbarkeitsgraphen um einen *unechten* Teilerreichbarkeitsgraphen handelt.
- 8) Vgl. dazu die Definitionen finaler (sachlich zulässiger) und formal zulässiger Schaltprozesse.
- 9) Im folgenden wird es die Diktion erleichtern, von den Elementen der Menge ME_{TRG} zu sprechen, ohne sich festzulegen, ob entweder tatsächlich erwünschte Endmarkierungen aus der Endmarkierungsmenge EM des Netzmodells oder aber nur vorläufig erwünschte "End"markierungen aus der Knotenmenge KN_{TRG} eines Teilerreichbarkeitsgraphen TRG betroffen sind. Dann werden alle Elemente aus der Menge ME_{TRG} vereinfacht als erwünschte Endmarkierungen im Teilerreichbarkeitsgraphen TRG bezeichnet.
- 10) Denn solche finalen Schaltprozesse, die von der Ausgangsmarkierung M_0 zu einer der erwünschten Endmarkierungen aus der Menge ME führen, sind für den betrachteten Teilerreichbarkeitsgraphen im allgemeinen nicht definiert. Dies wurde bereits dargelegt.
- 11) Auf die grundsätzlichen Validitätsdefizite, die allen Indikator Konzepten anhaften, wird später im Zusammenhang mit der Beurteilung des Petrinetz-Konzepts näher eingegangen. Die dort vorgetragenen Vorbehalte gelten hier analog für die Indikatorqualität von subsidiären Formalzielen. Dennoch lassen sich die Indikatoren durch ein "faute de mieux-Argument" rechtfertigen: Solange keine zufriedenstellendere Alternative in Sicht ist, um die Zielwirksamkeit von Schaltwegen und von damit korrespondierenden Schaltprozessen im ausgewerteten Netzmodell anhand eines Teilerreichbarkeitsgraphen zu beurteilen, handelt es sich um die "beste" aller bekannten Beurteilungsmöglichkeiten. Welche subsidiären Formalziele im einzelnen herangezogen werden, kann die Indikatorqualität erheblich beeinflussen. Dies läßt sich jedoch im Rahmen eines *allgemeinen* Analysekonzepts nicht näher klären, sondern erfordert einzelfallspezifische Untersuchungen. Diese werden hier jedoch nicht angestrebt. Vgl. dazu die thematische Rahmenlegung zu Beginn dieser Arbeit.
Beispielsweise kann für das Netzmodell eines Produktionssystems das Formalziel verfolgt werden, die durchschnittliche Durchlaufzeit aller darin abgewickelten Aufträge zu minimieren. Als subsidiäres Formalziel für die Betrachtung eines Teilerreichbarkeitsgraphen bietet es sich in diesem Fall an, die Summe aus Warte- und Transportzeiten bei der Auftragsabwicklung zu minimieren. Diese Warte- und Transportzeiten sind auch für nicht-finale Schaltprozesse definiert. Sie stehen in einem sachlich fundierten Indikatorzusammenhang mit dem Durchlaufzeitziel, das für alle finalen Schaltprozesse des Netzmodells letztlich verfolgt wird. Die Güte dieses Indikatorzusammenhangs wird hier nicht weiter diskutiert. Aus betriebswirtschaftlichen Beiträgen zur Ablaufplanung von Produktionsprozessen ist hinlänglich bekannt, daß dieser Indikatorzusammenhang nicht stringent ist. Denn nicht-

finale Teilprozesse mit minimalen Summen aus Warte- und Transportzeiten brauchen keineswegs notwendig zu finalen Gesamtprozessen mit minimalen Durchlaufzeiten zu führen.

12) Es wird in dieser Arbeit stets vorausgesetzt, daß bei der Zugrundelegung vollständiger Erreichbarkeitsgraphen stets die ursprünglich definierten Prozeßmengen IPROM und NIPROM aller intendierten bzw. nicht-intendierten Schaltprozesse verwendet werden. Sie ersetzen dann die analogen Prozeßmengen $IPROM_{TRG}$ und $NIPROM_{TRG}$ aus der Untersuchung von Teilerreichbarkeitsgraphen.

13) Vgl. dazu die Definition unechter Teilerreichbarkeitsgraphen.

14) Daher wird auch von einer optimierenden Robustheitsanalyse gesprochen.

15) Wegen $\alpha_r = 0$ könnte auf die Ermittlung der nicht-intendierten Prozeßmenge NIPROM verzichtet werden. Der damit verbundene potentielle Informationsverlust wurde bereits in einer früheren Anmerkung dargelegt. Dort wurde auch ein modifiziertes Min/Max-Entscheidungskriterium vorgestellt, daß an der Dominanz des Chancenaspekts festhält, aber dennoch unter allen nicht-dominierten - und somit chancenmaximalen - Entscheidungsalternativen diejenige mit dem geringsten "Restrisiko" auswählt. Um diese Option aufrechtzuerhalten, wird hier weiterhin die Prozeßmenge NIPROM einbezogen.

16) Daher kann ebenso davon gesprochen werden, das Ergebnis einer optimierenden Robustheitsanalyse bestehe in der Auswahl eines optimalen Schaltprozesses.

17) Es muß jedoch darauf hingewiesen werden, daß die optimierende Robustheitsanalyse einen wesentlich höheren Untersuchungsaufwand als die - ohnehin schon ressourcenintensive - konventionelle Netzoptimierung bereiten kann. Denn bereits die Ermittlung der Menge IPROM aller intendierten Schaltprozesse, die zu Beginn der Robustheitsanalyse erforderlich ist, entspricht einer vollständigen konventionellen Netzoptimierung. Es handelt sich um die besonders aufwendige Variante der starken Netzoptimierung, da *alle* optimalen Schaltprozesse aufgedeckt werden. Die optimierende Robustheitsanalyse benötigt daher größere Durchführungsressourcen als die konventionelle schwache Netzoptimierung, bei der nur nach *einem* optimalen Schaltprozeß gesucht wird. Dies gilt jedenfalls dann, wenn in dem untersuchten Netzmodell mehrere optimale Schaltprozesse existieren. Darüber hinaus verlangt die optimierende Robustheitsanalyse noch weitere Berechnungsoperationen, bis der robuste erste (Schalt-)Schritt endgültig feststeht. Diese zusätzlichen Berechnungen verursachen allerdings - im Vergleich zur Ermittlung der intendierten Prozeßmenge IPROM - nur relativ geringfügigen Aufwand, falls auf die Bestimmung der Menge NIPROM aller nicht-intendierten Schaltprozesse verzichtet wird. Auf ihre Abundanz wurde bereits hingewiesen. Dort wurde allerdings auch angedeutet, daß sie für eine verfeinerte Netzoptimierung, die auch noch "Restrisiken" abzudecken vermag, erforderlich ist. Schließlich ist darauf hinzuweisen, daß auch die nachstehend erläuterte Option, auf unvorhergesehene Veränderungen der Produktionssituation nachträglich zu reagieren, zusätzlichen Aufwand verursacht. Dies gilt insbesondere dann, wenn wegen schädlicher Markierungsabweichungen die dort beschriebenen Rückkehrprozesse berechnet werden müssen. Die generelle Abwägung, ob die zusätzlichen Ressourcenbedarfe für die voranstehend erwähnten Varianten der Robustheitsanalyse angesichts der variantenspezifischen Erkenntnispotentiale gerechtfertigt sind, kann hier nicht beurteilt werden. Dies hängt vielmehr von den Präferenzen des jeweils agierenden Modellierungsträgers ab.

18) Auf einen weiteren Vorteil wurde bereits hingewiesen: Die optimierende Robustheitsanalyse ist durch die Einbeziehung der Menge NIPROM aller nicht-intendierten Schaltprozesse in der Lage, auch das "Restrisiko" des Verfehlens von erwünschten Endmarkierungen zu erfassen. Dazu braucht lediglich das modifizierte Min/Max-Entscheidungskriterium benutzt zu werden, das in einer früheren Anmerkung vorgestellt wurde.

19) Die günstigen Fälle betreffen alle Situationsveränderungen, in denen weder die Struktur des modellierten Produktionssystems noch das Zielsystem des Entscheidungsträgers verändert werden. Sie erstrecken sich daher nur auf die aktuellen Zustände von Komponenten des Produktionssystems und auf die darin ausgeführten Produktionsprozesse. Beispielsweise kann die unvorhergesehene Störung einer Bearbeitungsstation dazu führen, daß sich der ursprünglich vorgesehene optimale Schaltprozeß nicht mehr ausführen läßt, weil in ihm der gestörten Bearbeitungsstation mindestens ein Arbeitsgang zugeordnet wurde. Dies kann sich im Netzmodell als Veränderung der aktuellen Markierung auswirken. Dabei wird unterstellt, daß im Netzmodell eine Stelle existiert, deren zugehöriges Prädikat die Betriebsbereitschaft der betroffenen Bearbeitungsstation ausdrückt. Diese Stelle büßt ihre Marke, die bislang die Gültigkeit des Betriebsbereitschaftsprädikats repräsentierte, im Zeitpunkt des Bekanntwerdens der Betriebsstörung ein. Ein "spontaner", vom Entscheidungsträger nicht zu beeinflussender Schaltschritt bewirkt im Netzmodell diesen Markenabzug. Dieser Schaltschritt ist im zuvor geplanten optimalen Schaltprozeß nicht mehr enthalten. Im Erreichbarkeitsgraphen schlägt sich dies derart nieder, daß eine Schaltkante vom beabsichtigten Schaltweg fortführt. Es besteht dann ein hohes Risiko, den nunmehr realisierten Schaltprozeß nicht mehr so fortsetzen zu können, daß der gesamte - modifizierte - Schaltprozeß immer noch zur Menge IPROM intendierter optimaler Schaltprozesse gehört. In widrigen Fällen erstreckt sich dagegen die Veränderung der Produktionssituation auf eine strukturelle Änderung des modellierten Produktionssystems oder auf Modifizierungen des Zielsystems des Entscheidungsträgers. Dann muß zu einem neuen, strukturell überarbeiteten Netzmodell übergegangen werden. Da die Robustheitsanalyse stets

nur für das jeweils zugrundegelegte Netzmodell gültig ist, muß sie erneut vollständig ausgeführt werden. Daher beziehen sich die nachfolgenden Ausführungen nicht auf solche widrigen Fälle. Zwar könnte eingewandt werden, daß Modifizierungen des Sachzielsystems - in der Gestalt veränderter Auftragspakete - nur zu wechselnden Ausgangs- und erwünschten Endmarkierungen führten. Analog ließe sich einwerfen, Variationen des Formalzielsystems beträfen nur die Schaltwerte von Transitionen. Beides erlaubte, mit weiterhin "unveränderten" Netzmodellen umzugehen. Dies ist jedoch tatsächlich nicht der Fall. Erstens handelt es sich auch bei modifizierten Ausgangs- und Endmarkierungen bzw. variierten Schaltwerten um *andere* Netzmodelle (vgl. dazu die Definition Synthetischer Netze). Zweitens reagieren die Mengen (N)IPROM aller (nicht-)intendierten Schaltprozesse empfindlich auf Variationen von Ausgangsmarkierungen, erwünschten Endmarkierungen und Formalzielen. Dies folgt unmittelbar aus den o.a. Definitionen der Schaltprozeßmengen. Diese Mengen stellen den zentralen Ansatzpunkt der optimierenden Robustheitsanalyse von Netzmodellen dar. Daher müßte sie auch dann neu abgewickelt werden, wenn neben den vorgenannten Zielsystemmodifizierungen keine strukturellen Veränderungen des modellierten Produktionssystems eingetreten wären.

20) In allen anderen - widrigen - Fällen kann die optimierende Robustheitsanalyse zwar weiterhin angewendet werden. Sie muß dann jedoch vollständig neu erstellt werden. Dies wurde in der voranstehenden Anmerkung aufgezeigt. Eine solche vollständige Analysewiederholung ist ebenso bei der konventionellen Netzoptimierung erforderlich. Daher entfällt der hier erläuterte Vorzug der optimierenden Robustheitsanalyse gegenüber konventionellen Ermittlungen von Netzoptima in jenen widrigen Fällen.

21) Zugleich verdeutlicht diese Option auf optimale Prozeßwechsel, daß sich der hohe Ermittlungsaufwand der optimierenden Robustheitsanalyse, der für die Bestimmung *aller* optimalen Schaltprozesse in der Menge IPROM anfällt, durch einen zusätzlichen Analysenutzen gegenüber der konventionellen Netzoptimierung rechtfertigen läßt. Dies gilt zumindest dann, wenn einerseits die konventionelle Netzoptimierung in ihrer schwachen Variante erfolgt, jeweils nur *einen* optimalen Schaltprozeß zu bestimmen, und andererseits im untersuchten Netzmodell mehrere optimale Schaltprozesse existieren.

22) In einem Netzmodell, in dem zyklische Schaltprozesse zulässig sind, können diese beliebig - potentiell unendlich - oft wiederholt werden. Daher sind dort auch potentiell unendlich viele Schaltprozesse zulässig. Die Anzahl der intendierten Schaltprozesse ist jedoch im allgemeinen endlich. Denn jeder intendierte Schaltprozeß stellt im Regelfall einen endlichen, azyklischen Schaltprozeß dar, der von der Ausgangsmarkierung zu einer der erwünschten Endmarkierungen führt und dabei das Formalzielsystem in der angestrebten - z.B. optimalen - Weise erfüllt. In einem Netzmodell mit endlichem Erreichbarkeitsgraphen können nur endlich viele verschiedene intendierte Schaltprozesse dieser Art existieren. Unter den voranstehenden Voraussetzungen konvergiert der Anteil von endlich vielen intendierten Schaltprozessen an den unendlich vielen zulässigen - auch zyklisch wiederholten - Schaltprozessen gegen Null. Diese Nullkonvergenz ist jedoch zur Beurteilung der Chance optimaler Prozeßwechsel unerwünscht. Denn es interessieren nicht die potentiell unendlich vielen zyklischen, aber nicht-intendierten Schaltprozesse, sondern nur die intendierten, aber im allgemeinen azyklischen Schaltprozesse. Daher erfolgt die oben vorgetragene Einschränkung auf Schaltprozesse, die nicht zyklisch wiederholt werden. Sie enthält von allen zyklisch wiederholbaren Schaltprozessen nur die Basisprozesse, deren Schaltzyklen jeweils genau einmal durchlaufen werden. Die Menge $PROM_b$ aller derart definierten Schaltprozesse wurde bereits eingeführt.

23) Vgl. die Ausführungen zum Aufwand optimierender Robustheitsanalysen.

24) Vgl. auch die Ausführungen zu den Schwächen des konventionellen Planungsverständnisses, unvorhergesehene Veränderungen von Produktionssituationen zu beherrschen. Denn jeder Neuaufwurf der optimierenden Robustheitsanalyse entspricht der konventionellen Neuplanung optimaler Schaltprozesse. Unter ungünstigen Umständen kann sie sogar zum Ärgernis einer "nervösen" Netzoptimierung entarten, bei der fortlaufend Neuplanungen erfolgen, die sich niemals verwirklichen lassen. Dieses Problem wurde bereits als "Nervosität" von PPS-Systemen angesprochen.

25) Die intendierte Prozeßmenge IPROM mußte schon zu Beginn der Robustheitsanalyse ermittelt werden, um die Werte der Funktion "chance" für alle Schaltschritte zu ermitteln, die unter der Ausgangsmarkierung des Netzmodells aktiviert sind.

26) Andernfalls muß die Robustheitsanalyse mit der aktuellen Netzmarkierung als revidierter Ausgangsmarkierung von neuem durchgeführt werden. Fortan wird unterstellt, daß diese Neuauflage der Robustheitsanalyse nicht erforderlich ist.

27) Es ist durchaus möglich, daß zwischen der aktuellen Netzmarkierung und der Auffangmarkierung mehrere Schaltprozesse zulässig sind. Darüber hinaus können im Erreichbarkeitsgraphen des Netzmodells auch mehrere Knoten gefunden werden, die jeweils eine Auffangmarkierung darstellen. Dann existieren auch entsprechend viele unterschiedliche Auffang- und Rückkehrprozesse.

28) Dies folgt aus der Inkonsistenz der hypothetischen Annahme des kontradiktorischen Gegenteils: Wenn der Gesamtprozeß zur intendierten Prozeßmenge IPROM gehören würde, so müßte die aktuelle Netzmarkierung, die bei Eintritt der unvorhergesehenen Situationsveränderung vorliegt, zur Markierungsfolge mindestens eines intendierten

Schaltprozesses gehören. Dies widerspricht aber der Definition der hier betrachteten *schädlichen* Markierungsabweichung. Daher kann es nicht zutreffen, daß der Gesamtprozeß zur Menge IPROM gehört; q.e.d.

29) Da sich der Gesamtprozeß einerseits suboptimal verhält und andererseits durch die Konstruktion des Rückkehrprozesses entsteht, wird fortan auch der Rückkehrprozeß als "suboptimal" bezeichnet. Hierbei handelt es sich jedoch nur um eine verdeutlichende Redeweise. Denn die Qualifizierung von Schaltprozessen als "optimal" oder "suboptimal" ist strenggenommen nur für finale Schaltprozesse definiert, die von der Ausgangsmarkierung bis zu einer der erwünschten Endmarkierungen M_{Be} reichen. Dies trifft auf die hier betrachteten Rückkehrprozesse per constructionem nicht zu.

30) Der Ressourcen"mehr"einsatz kann auch negativ ausfallen. Dies ist dann der Fall, wenn es weniger Aufwand bereitet, ein neues Netzoptimum zu ermitteln, als einen Rückkehrprozeß zu bestimmen. In der Regel liegen solche Verhältnisse jedoch nicht vor.

31) Falls überhaupt kein Rückkehrprozeß existiert, muß ein optimaler Schaltprozeß vollständig neu bestimmt werden. Dann entfallen die nachfolgenden Überlegungen.

32) Bereits anläßlich der Entfaltung des thematischen Bezugsrahmens dieser Arbeit wurde betont, daß Konzepte für Anpassungsplanungen ein Desiderat für die hier untersuchten dynamischen Koordinierungsprobleme darstellen. Vgl. auch den Hinweis, der anläßlich der Diskussion begründungsverwaltender Systeme zu Anpassungsplanungen erfolgte.

33) Vgl. zur Anpassungsstrategie des "turnpike scheduling" VAN DYKE PARUNAK (1987), S. 303f.; ZELEWSKI (1988c), S. 55; ROSE,HA. (1989), S. 83.

Diese Strategie bezieht sich auf Maschinenbelegungsplanungen, deren intendierte Realisierung durch unerwartete Veränderungen der Produktionssituation nicht mehr möglich ist. Es wird dann versucht, nach Eintritt der Produktionsstörung die Ausführung der Produktionsprozesse im Produktionssystem so fortzusetzen, daß ein Wiederanknüpfungspunkt im ursprünglichen Maschinenbelegungsplan erreicht wird. Auch hier wird darauf verzichtet, eine vollständige Neuplanung vorzunehmen. Eine Verschärfung dieser Anpassungsstrategie erfolgt durch den Grundsatz, Plananpassungen an Produktionsstörungen sollten so erfolgen, daß vom zuletzt gültigen Produktionsplan so wenig wie möglich abgewichen werde. Auf diese Anpassung durch kleinstmögliche Abweichungen wurde bereits hingewiesen; vgl. auch STUTE (1978a), S. 83 u. 95; DÖTLING (1981), S. 63, 75 u. 82.

34) Es wird auch kurz als Rückkehrproblem bezeichnet.

35) Beispielsweise kann es sich um Satisfizierungsziele derart handeln, daß die Anzahl der Schaltschritte, die im Anpassungsmodell bis zum Erreichen einer der erwünschten Endmarkierungen ausgeführt werden, nach oben beschränkt wird. Dies führt dazu, daß nur solche Rückkehrprozesse betrachtet werden, deren Schaltfolgen nicht länger als die höchstzulässige Schaltschrittanzahl sind. Dabei kann im Falle von Koordinierungsnetzen auch auf die Analyse der Promptheit von Reaktionsprozessen zurückgegriffen werden, falls es möglich ist, die Rückkehr- als Reaktionsprozesse darzustellen. Vgl. dazu die Ausführungen zu Koordinierungsnetzen.

Ebenso läßt sich an Optimierungsziele denken, die etwa vorschreiben, die Endmarkierungen des Anpassungsmodells mit der kleinstmöglichen Schaltschrittanzahl oder mit minimalen Schaltkosten zu erreichen. Im letztgenannten Fall müssen allerdings die Transitionen des Netzmodells mit Kosten bewertet sein, die durch ihr Schalten verursacht werden. Auf solche Schaltwerte von Transitionen wurde schon im Kontext der Netzoptimierung ausführlich eingegangen. Falls keine Formalziele spezifiziert werden, gilt hingegen die Vereinbarung, daß dem Entscheidungsträger alle Rückkehrprozesse gleich willkommen sind.

36) Falls erkannt wird, daß im Anpassungsmodell kein solcher Schaltprozeß existiert, läßt sich das Rückkehrproblem grundsätzlich nicht lösen. Folglich muß die Anpassungsplanung erfolglos abgebrochen werden. Dann ist nur noch eine Neuplanung möglich. Bei ihr werden im zugrundeliegenden Netzmodell - von der aktuellen Netzmarkierung aus - intendierte Schaltprozesse vollkommen neu ermittelt. Gleiches gilt, wenn die Suche nach Schaltprozessen im Anpassungsmodell einer Schranke maximal einzusetzender Ressourcen unterliegt und dieses Ressourcenlimit erreicht worden ist, ohne daß einer der gesuchten Schaltprozesse gefunden werden konnte.

37) Die Prozeßoptimalität gilt allerdings nur im Rahmen des jeweils untersuchten Anpassungsmodells. Denn es wurde schon oben dargelegt, daß die Gesamtprozesse, zu denen die Rückkehrprozesse bei schädlichen Markierungsabweichungen gehören, in den zugrundeliegenden Netzmodellen stets suboptimale Prozesse darstellen.

38) Bereits die Ermittlung von Rückkehrprozessen brauchte sich nicht auf die Suche nach optimalen Rückkehrprozessen zu beschränken. Allerdings gilt die Devianz von optimierenden Netzanalysen nur hinsichtlich der verfolgten Untersuchungsabsichten. Denn es wird später gezeigt werden, daß in Ausnahmefällen auch heuristische Entscheidungsregeln die unbeabsichtigte Nebenwirkung besitzen können, optimale Schaltprozesse hervorzubringen. Von solchen nicht intendierten Netzoptimierungen auf heuristischer Entscheidungsbasis wird zunächst abgesehen.

39) Daher wird auch von einer heuristischen Robustheitsanalyse gesprochen.

40) Strenggenommen handelt es sich um zukunftsblinde Heurismen. Der Einfachheit halber werden sie jedoch nur als "blind" bezeichnet.

41) Es wird hier bewußt auf "zukünftig auswählbare Entscheidungsalternativen" und nicht auf "zukünftige Entscheidungsspielräume" Bezug genommen. Denn es wird später aufgezeigt werden, daß sich trotz der Anwendung von vorausschaufreien Entscheidungsregeln bei einer lokalen Auswahlentscheidung berücksichtigen läßt, ob sie durch Hervorbringen einer toten Folgemarkierung alle zukünftigen Entscheidungsspielräume vorzeitig schließen würde. Dann könnten anschließend überhaupt keine aktivierten Schaltschritte mehr ausgewählt werden. In dieser speziellen, negativ formulierten Hinsicht lassen also auch blinde Heurismen eine Vorausschau zu, ob alle zukünftigen Entscheidungsspielräume unmittelbar untergehen. Hierdurch erfolgt jedoch keine positive Vorausschau auf Entscheidungsalternativen, die zukünftig im Fall von weiterhin offenen Entscheidungsspielräumen noch zur Auswahl stehen.

42) Vgl. dazu die Konzeptualisierung dynamischer Koordinierungsprobleme.

43) Ebenso wird darauf verzichtet, auf Informationen über den Schaltprozeß, der bis zur aktuellen Markierung hin ausgeführt wurde, explizit Bezug zu nehmen. Solche "historischen" Informationen lassen sich aber dennoch implizit berücksichtigen, indem sie von den Marken des Netzmodells während ihres Fortschaltens gesammelt werden. Beispielsweise können in einem Attribut einer sortierten Marke die Schaltwerte aller Transitionen akkumuliert werden, durch deren Schaltakte die betroffene Marke im Netzmodell fortbewegt wurde.

44) Vgl. dazu die Anmerkungen und Quellenhinweise, die anläßlich der thematischen Rahmenlegung zur betriebswirtschaftlichen Relevanz von heuristischen Entscheidungsregeln erfolgten.

45) Falls zufällig eine tote Netzmarkierung M_w betrachtet wird, ist dort nur noch der entartete Schaltprozeß mit der Länge $L=0$ zulässig.

46) Das Risiko unerwünschter Livelockmarkierungen kann bei der Zugrundelegung von Teilerreichbarkeitsgraphen im allgemeinen nicht erfaßt werden. Dies wurde schon oben begründet.

47) Denn das Risiko unerwünschter Deadlockmarkierungen fließt implizit in die Konstruktion der intendierten Prozeßmenge $IPROM_{TRG}$ ein. Dies wird in einer anderen Anmerkung detailliert dargelegt. Daher bleibt es hier unbeachtlich, ob das Risiko unerwünschter Deadlockmarkierungen vermittels einer Unsicherheitspräferenz mit $\alpha_r > 0$ explizit erfaßt oder aber implizit berücksichtigt wird.

48) Die Vereinfachung liegt darin, daß nachfolgend nur die intendierte Prozeßmenge $IPROM_{TRG}$ nicht aber deren Pendant - die Menge $NIPROM_{TRG}$ der nicht-intendierten Schaltprozesse - berücksichtigt zu werden braucht. Denn bei einem Risikoparameter $\alpha_r = 0$ vermag sich die nicht-intendierte Prozeßmenge $NIPROM_{TRG}$ niemals auf die Auswahl robuster (Schalt-)Schritte auszuwirken.

49) Auf einfache heuristische Entscheidungsregeln trifft diese Feststellung oftmals nicht zu. Statt dessen determinieren sie in zahlreichen Fällen noch keine eindeutige Auswahl zwischen Entscheidungsalternativen. Denn sie werden zumeist mit der Hilfe eines Extremierungsoperators formuliert, der auf ein Alternativenmerkmal angewendet wird. Die Menge aller Alternativen, die für dieses Merkmal eine minimale oder maximale Ausprägung besitzen, muß aber keineswegs einelementig sein. Vgl. zu Beispielen für mehrelementige Ergebnismengen der Regelanwendung - im Kontext von Prozeßkoordinierungen durch Prioritätsregeln - SPUR (1980), S. 327 u. 340f. Zweck der Anwendung von Entscheidungsregeln ist jedoch, aus einer mehrelementigen Alternativenmenge genau eine Entscheidungsalternative auszuwählen. Daher wird versucht, der potentiellen Mehrdeutigkeit von Entscheidungsregeln dadurch zu begegnen, daß mehrere Entscheidungsregeln miteinander hierarchisch kombiniert werden. Vgl. zu solchen hierarchischen Regelkombinationen SPUR (1980), S. 327 (u. 340ff.); SPUR (1981a), S. 116; HINTZ (1987), S. 80f.

Regelhierarchien können, müssen aber nicht die Eindeutigkeit der Auswahlentscheidung gewährleisten. Denn es ist - ohne zusätzliche Vorkehrungen - unmöglich zu garantieren, daß die Aneinanderreihung mehrerer potentiell mehrdeutiger Entscheidungsregeln ein eindeutiges Selektionsergebnis liefert. Deshalb wird oftmals auf das Instrument einer "tie breaking rule" zurückgegriffen. Es ist so konstruiert, daß alle Mehrdeutigkeiten, die auf übergeordneten Hierarchiestufen noch nicht beseitigt werden konnten, auf der untersten Hierarchieebene mit "roher Gewalt" auflöst. Zu diesem Zweck können z.B. alle Entscheidungsalternativen einen identifizierenden, natürlichzahligen Index enthalten. Auf die derart identifizierten Alternativen läßt sich dann auf der untersten Hierarchieebene eine eindeutigkeitsstiftende Regel anwenden. Diese Regel kann etwa diejenige Alternative selektieren, die entweder den minimalen oder aber den maximalen Indexwert aufweist. Von einer solchen eindeutigkeitsgewährleistenden Entscheidungsregelkombination wird fortan ausgegangen. Der Einfachheit halber wird sie weiterhin als heuristische Entscheidungsregel angesprochen. Dies ist insofern zulässig, als hiermit eine komplexe Regel gemeint sein kann, die aus mehreren einfachen Regeln zusammengesetzt ist.

50) In der intendierten Prozeßmenge $IPROM_{TRG}$ wird zunächst die Menge $ZFPROM_{TRG}$ aller formal zulässigen finalen Schaltprozesse ausgewählt. Er umfaßt alle Schaltprozesse, die sowohl alle Sachziele (Finalität) als auch alle Satisfizierungsziele (formale Zulässigkeit) erfüllen. Es wurde jedoch schon oben darauf hingewiesen, daß sich die

Sach- und Formalziele, die für Netzmodelle und deren vollständige Erreichbarkeitsgraphen definiert sind, auf (echte) Teilerreichbarkeitsgraphen im allgemeinen nicht übertragen lassen. Es müssen Ersatzziele betrachtet werden, die zu den vorgenannten Zielen in einem Mittel-Zweck-Verhältnis stehen. Oftmals werden im konkreten Einzelfall solche adäquaten Substitute für Sach- und Satisfizierungsziele nicht bekannt sein. Dann kann aber immerhin noch auf die oben definierte Menge ME_{TRG} zurückgegriffen werden, deren Markierungen in der Knotenmenge KN_{TRG} des Teilerreichbarkeitsgraphen TRG enthalten sind. Sie umfaßt entweder tatsächlich erwünschte Endmarkierungen aus der Endmarkierungsmenge ME des zugrundeliegenden Netzmodells. Oder es gehören zu ihr als vorläufig erwünschte Endmarkierungen alle lebendigen Markierungen aus der Knotenmenge des Teilerreichbarkeitsgraphen. Auf diese Weise wird von vornherein ausgeschlossen, daß aktivierte Schaltschritte SS_a in die lokale Auswahlentscheidung einbezogen werden, obwohl ihre Ausführung unmittelbar - und deshalb zwangsläufig - eine unerwünschte Deadlockmarkierung hervorbringen würde. Daher umfaßt die Prozeßmenge $ZFPROM_{TRG}$ im Teilerreichbarkeitsgraphen TRG alle Schaltprozesse $PRO_{w,f}(SF_1, MF_1)$ mit $SF_1=(SS_a)$ und $AKT(SS_a, M_w)$, die entweder in einer erwünschten Endmarkierung des Netzmodells oder aber zumindest in einer lebendigen Netzmarkierung enden. Falls alle Schaltschritte SS_a , die unter der Markierung M_w aktiviert sind, eine tote Folgemarkierung M_f hervorbringen würden, ist die Prozeßmenge $ZFPROM_{TRG}$ notwendig leer. Andernfalls enthält sie mindestens einen Schaltprozeß mit einer (vorläufig) erwünschten Zielmarkierung M_f . Die eindeutige Auswahl genau eines Schaltprozesses daraus leistet dann das Prädikat $NDOM_{TRG}(PRO_{w,f})$ der Nicht-Dominiertheit. Es stellt die prädikatenlogische Reformulierung der oben angesprochenen Kombination heuristischer Entscheidungsregeln dar, die insgesamt eine eindeutige Auswahlentscheidung garantiert. Daher kann in der Prozeßmenge $ZFPROM_{TRG}$ nur höchstens ein nicht-dominierter Schaltprozeß $PRO_{w,f}(SF_1, MF_1)$ enthalten sein. Dies ist der oben angesprochene höchstens eine intendierte Schaltprozeß aus der Prozeßmenge $IPROM_{TRG}$; q.e.d.

51) Präzise wird diese Zugehörigkeit des Schaltschritts SS_a zum Schaltprozeß aus der einelementigen intendierten Prozeßmenge $IPROM_{TRG}$ ausgedrückt durch:

$$IPROM_{TRG} = \{PRO_{w,f}(SF_1, MF_1)\} \wedge SF_1=(SS_a) \wedge MF_1=(M_w, M_f)$$

52) Dies ergibt sich zwangsläufig aus der mangelhaften Konzeptualisierung von Deadlocks durch konventionelle Koordinierungskonzepte.

53) Bei der konventionellen Anwendung blinder heuristischer Entscheidungsregeln wäre dagegen zunächst einer dieser Schaltschritte ausgewählt worden. Erst im darauf folgenden Zustand M_f des Netzmodells wäre dann erkannt worden, daß es sich um eine unerwünschte Deadlockmarkierung handelt. Der Ressourcenverzehr dieser unnötigen Regelanwendung wird dagegen in der heuristischen Robustheitsanalyse von vornherein vermieden.

54) Diese Heurismen werden auch als vorausschauende oder "look ahead"-Entscheidungsregeln bezeichnet.

55) Ohne speziellen Bezug auf die Auswertung von Netzmodellen liegen vorausschauende Heurismen also dann vor, wenn die Auswahl lokaler Entscheidungsalternativen von ihren Auswirkungen auf zukünftig auswählbare Entscheidungsalternativen beeinflusst wird.

56) Am Rande wird auf eine kontraintuitive Konsequenz dieses Ansatzes hingewiesen: Bereits an früherer Stelle wurde deutlich, daß das Konzept der Teilerreichbarkeitsgraphen $TRG(M_w, L_{sup})$ mit $M_w=M_0$ und $L_{sup} \geq L_{max}$ die vollständigen Erreichbarkeitsgraphen von Netzmodellen umgreift. Auf der Basis vollständiger Erreichbarkeitsgraphen kann die Auswahl robuster erster Schritte in den Ausgangsmarkierungen M_0 so gestaltet werden, daß nur solche Schaltschritte SS_a selektiert werden, die unter den Ausgangsmarkierungen M_0 jeweils optimale Schaltprozesse einleiten. Dies wurde als Einsatzmöglichkeit der Robustheitsanalyse für Optimierungszwecke dargelegt. Folglich bewirken *heuristische* Entscheidungsregeln eine *Optimierung* des untersuchten Netzmodells, falls drei Voraussetzungen erfüllt sind: Erstens werden die Heurismen auf die Ausgangsmarkierung M_0 zur Auswahl eines robusten ersten Schritts angewendet. Zweitens besitzen sie eine Vorausschauweite L_{sup} , die mindestens so groß ist wie die tatsächliche Reichweite L_{max} des Erreichbarkeitsgraphen des Netzmodells. Drittens handelt es sich um eine *optimierende* Robustheitsanalyse, welche die früher aufgeführten Prämissen erfüllt.

Daher trifft die weit verbreitete Ansicht, heuristische Entscheidungsregeln erlaubten *grundsätzlich* nur suboptimale Lösungen von Koordinierungsproblemen, nicht zu. Vgl. zu dieser Ansicht z.B. STREIM (1975), S. 151; SEELBACH (1979), Sp. 26; SPUR (1980), S. 364; NIEB (1980), S. 30; MAIER, U. (1980), S. 55; KNOOP (1986), S. 29 u. 166; MÜLLER, A. (1987), S. 274.

Die behauptete Suboptimalität von Heurismen erweist sich zwar oftmals als zutreffend. Dies gilt insbesondere für die Koordinierung von Produktionsprozessen mit der Hilfe von Prioritätsregeln, die eine typische Ausprägung heuristischer Entscheidungsregeln darstellen. Doch ist die *Generalisierung* dieser Regelsuboptimalität unzulässig. Dies belegt die hier thematisierte Robustheitsanalyse von Netzmodellen. Sie gestattet auch, eine notwendige Bedingung dafür anzugeben, daß sich bei der Abbildung von Koordinierungsproblemen auf Netzmodelle die hier betrachteten Heurismen tatsächlich als suboptimal herausstellen: Die Reichweite L_{sup} der regelgestützten Auswahlentscheidungen in den (M_w, L_{sup}) -reduzierten Erreichbarkeitsgraphen muß *kleiner* sein als die tatsächliche Reichweite L_{max} der voll-

ständigen Erreichbarkeitsgraphen, die zu den untersuchten Netzmodellen gehören: $L_{\text{sup}} < L_{\text{max}}$. Diese Suboptimalitätsbedingung für heuristische Entscheidungsregeln läßt sich dadurch erfüllen, daß folgende Anforderungen zutreffen: Erstens werden nicht-triviale Koordinierungsprobleme untersucht, deren Netzmodelle Erreichbarkeitsgraphen mit tatsächlichen Reichweiten $L_{\text{max}} \geq 3$ besitzen. Diese Voraussetzung läßt sich in der Regel schnell überprüfen, indem untersucht wird, ob in den Netzmodellen mindestens ein Schaltprozeß der Länge $L \geq 3$ ausgeführt werden kann. Zweitens dürfen entweder nur vorausschaufreie Heuristiken angewendet werden. Dann ist wegen $L_{\text{sup}} = 1$ die Beziehung $L_{\text{sup}} < L_{\text{max}}$ auf jeden Fall erfüllt. Oder aber es werden heuristische Entscheidungsregeln mit Vorausschauweiten L_{sup} eingesetzt, die der Anforderung $L_{\text{sup}} < L_{\text{max}}$ genügen. Das letztgenannte Postulat wird tendenziell um so seltener untersucht werden, je größer die zugelassene Vorausschauweite L_{sup} ist. Denn es werden dann um so mehr Ressourcen erforderlich, um nach Schaltprozessen mit der Länge L und $L_{\text{sup}} < L$ zu forschen, deren Existenz die Bedingung $L_{\text{sup}} < L_{\text{max}}$ sicherstellen würde. Heuristische Entscheidungsregeln werden jedoch mit der Absicht eingesetzt, aufwendige Suchprozesse zu erübrigen. Daher nimmt die Gefahr, daß die Erfüllung der Suboptimalitätsbedingung $L_{\text{sup}} < L_{\text{max}}$ unbekannt bleibt, mit wachsender Vorausschauweite L_{sup} von heuristischen Entscheidungsregeln zu.

57) Denn für vorausschaufreie Entscheidungsregeln wurden der heuristischen Robustheitsanalyse ($M_{w, L_{\text{sup}}}$)-reduzierte Erreichbarkeitsgraphen mit der Reichweite $L_{\text{sup}} = 1$ zugrundegelegt.

5.2.3.6 Die Simulationsanalyse

Die simulative Untersuchung¹⁾ dynamischer Netzeigenschaften²⁾ gehört neben der Invariantenanalyse zu den am häufigsten verwandten Analysetechniken des Petrinetz-Konzepts. Dies schlägt sich u.a. auch in den Software-Paketen nieder, die für Netzanalysen angeboten werden. Sie verfügen fast immer über entsprechende Komponenten für die simulative Netzauswertung³⁾. Auch bei der praktischen Modellierung von Koordinierungsproblemen in komplexen Produktionssystemen - insbesondere Flexiblen Fertigungssystemen - genießt die simulative Modellauswertung große Beachtung⁴⁾.

Dennoch wird die Netzsimulation hier erst als letzte Untersuchungsvariante vorgestellt. Sie wird auch - zumindest im Vergleich zur Erreichbarkeitsanalyse - nur knapp skizziert⁵⁾. Beides liegt in der ambivalenten Charakteristik der Simulationsanalyse begründet:

- ❑ Entweder wird die Netzsimulation in ihrer trivialen (monovalenten⁶⁾) Variante betrieben, die keiner eingehenden Erläuterung bedarf.
- ❑ Oder es liegt die anspruchsvollste (omnivalente) Simulationsvariante vor. Sie fällt mit der bereits ausführlich behandelten Konstruktion der Erreichbarkeitsgraphen von Netzen zusammen. Daher erübrigt sich eine detaillierte Darstellung ebenfalls.

Tatsächlich stellen die beiden Varianten der Netzsimulation nur die beiden Antipoden eines Variantenspektrums dar⁷⁾. Der Übersichtlichkeit halber konzentrieren sich die anschließenden Ausführungen jedoch auf die zwei Grenzfälle dieses Spektrums.

Unter einer Simulation wird zumeist verstanden, ein reales Problem als ein System zu konzeptualisieren, diese Systemvorstellung⁸⁾ auf ein formalsprachliches Modell abzubilden und das Verhalten dieses Modells in experimenteller Weise zu untersuchen⁹⁾. Dieser Simulationsbegriff reicht jedoch für die anschließenden Ausführungen nicht aus. Sein Mangel beruht auf der unklaren Abgrenzung gegenüber dem Begriff analytischer Modellauswertung¹⁰⁾. Zwar besteht weitgehend Übereinstimmung, daß es sich bei simulativen und analytischen Konzepten¹¹⁾ um strenge Gegensätze handelt¹²⁾. Worin der prinzipielle Unterschied zwischen beiden liegt, bleibt jedoch zumeist im dunkeln. Er mag für so offensichtlich gehalten werden, daß sich seine Explizierung infolge "Trivialität" erübrige. Tatsächlich verhält sich die Abgrenzung simulativer und analytischer Modellauswertungen jedoch nicht so einfach. Die Abgrenzungsschwierigkeiten offenbaren sich u.a.¹³⁾ hinsichtlich der optimierenden Modellanalyse. Sehr häufig wird die Überzeugung vertreten, Modelloptima könnten nur auf analytische, nicht aber auf simulative Weise ermittelt werden¹⁴⁾. Im Gegensatz dazu steht die nachfolgend belegte Erkenntnis, daß sich durch die *Simulation* von Netzmodellen *optimale* Schaltprozesse und -folgen bestimmen lassen. Daher bedarf es einer präziseren Fassung des Simulationsbegriffs. Ein Analysekonzept wird hier als simulativ qualifiziert, wenn es fünf Anforderungen erfüllt¹⁵⁾:

- ❑ Ausgangspunkt ist die systemorientierte Konzeptualisierung eines realen¹⁶⁾ oder formalen¹⁷⁾ Problems. Sie wird in ein formalsprachlich expliziertes Modell¹⁸⁾ umgesetzt, das als ein endliches Formelsystem vorliegt.
- ❑ Es handelt sich um ein kinetisches oder dynamisches¹⁹⁾ Modell mit einem Ausgangszustand und einem wohldefinierten Modellverhalten.
- ❑ Der Ausgangszustand des Modells wird durch ein variablenfreies Formelsystem²⁰⁾ spezifiziert.
- ❑ Das Modellverhalten wird durch eine nicht-leere Menge von Transformationsregeln oder -funktionen²¹⁾ determiniert. Sie bestimmen für jeden Referenzzustand des Modells dessen Folgezustand.

- Untersuchungsziel ist es, durch das sukzessive Anwenden der Transformationsregeln bzw. -funktionen mindestens²²⁾ eine zulässige Modellverhaltensweise zu erzeugen²³⁾. Durch Auswertung der generierten Modellverhaltensweisen wird festgestellt, ob es sich um intendierte Lösungen des zugrundeliegenden Problems handelt oder ob sie dessen Unlösbarkeit anzeigen²⁴⁾.

Die Simulationsanalyse von Netzmodellen wird allen fünf Voraussetzungen gerecht: Netzmodelle liegen als formalsprachlich explizierte Konstrukte vor. Sie bilden die realen oder formalen Probleme, deren Lösung mit der Konstruktion von Netzmodellen angestrebt wird, in systemorientierter Weise ab. Die Definition jedes Netzmodells läßt sich als ein algebraisch-prädikatenlogisches Formelsystem auffassen. Bei einem Netzmodell handelt es sich stets um ein dynamisches Modell. Sein Ausgangszustand ist durch die Ausgangsmarkierung des Netzes in variablenfreier Weise festgelegt. Das Modellverhalten wird bei Synthetischen Netzen durch die Schaltregel des allgemeinen Übergangsschemas und ihre individualisierten Erscheinungsformen in den transaktionsspezifischen Schaltregeln determiniert²⁵⁾. Ihre Gesamtheit stellt die Transformationsregelmengende des Netzmodells dar. Durch sukzessives Anwenden dieser Schaltregel(n)²⁶⁾ auf den Ausgangszustand des Netzmodells und alle daraus hervorgebrachten Folgezustände wird mindestens ein Schaltprozeß als eine Verhaltensweise des Netzmodells erzeugt²⁷⁾. In der graphischen Visualisierung des Netzmodells äußert sich ein solcher Simulationslauf als ein schaltbedingter Markenfluß durch das Netz. Daher wird die Simulationsanalyse von Netzen oftmals als ein "Markenspiel" (token game)²⁸⁾ oder ähnlich bezeichnet. Die Auswertung aller derart erzeugten Schaltprozesse soll Erkenntnisse darüber vermitteln, welche Lösungen das zugrundeliegende Problem besitzt oder ob es unlösbar ist. Als Netzsimulation wird fortan der gesamte Prozeß verstanden, der die Erzeugung von Schaltprozessen in Simulationsläufen und deren lösungsbezogenen Auswertungen umfaßt.

Die Varianten der Netzsimulation resultieren daraus, daß die Transformation des Referenzzustands eines Netzmodells in seinen Folgezustand keineswegs eindeutig sein muß²⁹⁾. Dies folgt aus dem bereits erwähnten indeterministischen Charakter der Schaltregel. Denn aus der Markierung M_r eines Netzes, das den Referenzzustand "r" des untersuchten Netzmodells darstellt, können durch Anwenden der Schaltregel durchaus mehrere Folgemarkierungen M_f als Folgezustände "f" des Netzmodells hervorgebracht werden. Dies ist immer dann der Fall, wenn unter der Referenzmarkierung M_r mindestens zwei Schaltschritte aktiviert sind, deren Ausführungen unterschiedliche Folgemarkierungen bewirken würden³⁰⁾. Eine solche Referenzmarkierung mit mehrdeutigen Folgemarkierungen stellt den Normalfall für Netzmodelle dar³¹⁾. Sie wird fortan kurz als polykonsekutive Markierung bezeichnet. Die zugehörigen Netze werden als normale Netzmodelle angesprochen³²⁾. Wenn bei der simulativen Untersuchung eines normalen Netzmodells ein Modellzustand erreicht wird, der eine polykonsekutive Markierung darstellt, bieten sich zwei konträre Strategien an³³⁾:

- Es wird genau ein aktivierter Schaltschritt ausgewählt. Alle anderen aktivierten Schaltschritte werden vergessen. Die Netzsimulation wird fortgesetzt, indem der selektierte Schaltschritt ausgeführt und dadurch der nächste Modellzustand ermittelt wird.
- Zunächst wird ebenfalls ein aktivierter Schaltschritt ausgeführt, um durch seine Ausführung den Folgezustand des Netzmodells zu bestimmen. Aber die übrigen Schaltschritte, die unter der polykonsekutiven Markierung aktiviert waren, werden gespeichert. Sie werden später vollständig abgearbeitet, indem von der Markierung aus alle alternativen Simulationsfortsetzungen erforscht werden.

Es können auch beliebige Mischformen aus den beiden Strategien gebildet werden, die sich auf zwei verschiedenen Ebenen bewegen. Einerseits lassen sich bei der Betrachtung derselben polykonsekutiven Markierung zwar die alternativen Schaltwirkungen von mehreren, aber nicht von allen aktivierten Schaltschritten simulieren. Andererseits ist es möglich, wenn ein Netzmodell

mehrere polykonsekutive Markierungen besitzt, unter jeder dieser Referenzmarkierungen eine andere Strategievariante zu verfolgen.

Eine Simulationsanalyse heißt monovalent³⁴⁾, wenn in jedem Zustand des Netzmodells³⁵⁾ genau ein aktivierter Schaltschritt ausgeführt wird und alle anderen aktivierten Schaltschritte vergessen werden. Daher erfolgt während einer monovalenten Simulationsanalyse genau ein Simulationslauf. Dabei wird genau ein Schaltprozeß im zugrundeliegenden Netzmodell ausgeführt und vor dem Hintergrund der jeweils vorgegebenen Analyseaufgabe ausgewertet. Falls ein normales Netzmodell mit mindestens einer polykonsekutiven Markierung als Modellzustand untersucht wird, verhält sich die monovalente Simulationsanalyse notwendig unvollständig³⁶⁾. Denn in ihr wird mindestens ein Schaltschritt, der unter der polykonsekutiven Markierung aktiviert ist, nicht berücksichtigt. Dadurch bleibt die Schaltwirkung dieses Schaltschritts grundsätzlich unbekannt³⁷⁾. Von alternativen Schaltschrittausführungen werden alle - bis auf einen - grundsätzlich nicht beachtet³⁸⁾.

Von einer polyvalenten Simulationsanalyse³⁹⁾ wird dagegen gesprochen, falls zwei Bedingungen erfüllt werden: Erstens muß das untersuchte Netzmodell mindestens eine polykonsekutive Markierung besitzen. Zweitens müssen für mindestens zwei Schaltschritte, die unter dieser Markierung aktiviert sind, die unterschiedlichen Schaltwirkungen in alternativen Simulationsfortsetzungen untersucht werden. Die alternativen Simulationsfortsetzungen werden jeweils in eigenständigen Simulationsläufen erzeugt und ausgewertet. Deshalb erfolgen während einer polyvalenten Simulationsanalyse stets mehrere Simulationsläufe. Entsprechend werden im zugrundeliegenden Netzmodell mehrere alternative Schaltprozesse ausgeführt. Ob die Simulationsläufe und zugehörigen Schaltprozesse entweder zeitlich nacheinander oder aber nebenläufig geschehen, hängt von der Parallelverarbeitungsfähigkeit der jeweils verfügbaren Hardwarearchitektur und Simulationssoftware ab. Den Regelfall stellt derzeit (noch) die sequentielle Abarbeitung dar.

Eine polyvalente Simulationsanalyse wird als vollständig bezeichnet, wenn sie für *alle* polykonsekutiven Markierungen eines Netzmodells die Schaltwirkungen von *allen* dort aktivierten Schaltschritten ermittelt⁴⁰⁾. Sie wird auch kurz als omnivalente Simulationsanalyse⁴¹⁾ bezeichnet. Andernfalls - wenn für mindestens eine polykonsekutive Markierung des Netzmodells die Schaltwirkungen von mindestens einem der dort aktivierten Schaltschritte nicht bestimmt wird - liegt eine unvollständige polyvalente Simulationsanalyse⁴²⁾ vor. Bei der Betrachtung polyvalenter Simulationsanalysen steht fortan ihre vollständige Extremform im Vordergrund des Interesses. Zunächst wird jedoch auf die Mängel von monovalenten Simulationsanalysen eingegangen, um die spätere Vertiefung der omnivalenten Analysevariante vorzubereiten.

Eine monovalente Simulationsanalyse erweist sich als trivial, weil sie über das sukzessive Anwenden der Schaltregel nicht hinausreicht. Mit dieser Analysevariante lassen sich normale Netzmodelle, die über mindestens eine polykonsekutive Markierung verfügen, hinsichtlich ihrer dynamischen Eigenschaften nur in rudimentärer Weise untersuchen⁴³⁾. Die Analysedefizite erstrecken sich auf drei unterschiedliche Aspekte. Sie gelten nicht nur für die monovalente Simulationsanalyse, sondern auch für alle unvollständigen polyvalenten Simulationsanalysen. Der Übersichtlichkeit halber werden sie aber nur anhand der monovalenten Analysevariante thematisiert.

Erstens können alle holistischen dynamischen Netzeigenschaften, die eine vollständige Kenntnis der dynamischen Struktur eines Netzmodells voraussetzen, prinzipiell nicht ermittelt werden. Dazu gehören vor allem die SW*-Optimalität finaler Netzmodelle, das echte Terminieren und die R-Promptheit von Netzmodellen. Denn diese Netzeigenschaften lassen sich nur dadurch beurteilen, daß die Schaltwerte *aller* finalen Schaltprozesse, die Erreichbarkeit einer Endmarkierung von *allen* erreichbaren Markierungen aus bzw. die Einhaltung der Reaktionsspanne R durch *alle* Reaktionsprozesse untersucht werden. Bei der monovalenten Simulationsanalyse eines normalen Netzmodells wird jedoch mindestens ein aktivierter Schaltschritt aus der Unter-

suchung ausgeschlossen⁴⁴). Folglich lassen sich die vorgenannten Allurteile - abgesehen von besonders gelagerten Ausnahmefällen⁴⁵ - nicht treffen.

Zweitens lassen sich alle übrigen dynamischen Netzeigenschaften, die kein vollständiges Wissen über die dynamische Struktur eines Netzmodells erfordern, nur unter günstigen Umständen feststellen. Diese günstigen Umstände liegen immer dann vor, wenn bei der monovalenten Simulation eines normalen Netzmodells zufällig jene Schaltschritte ausgeführt werden, die zum Erkennen der interessierenden Netzeigenschaft ausreichen. Dies läßt sich z.B. anhand der Eigenschaft "Deadlockfreiheit" verdeutlichen. Falls ein Netzmodell tatsächlich eine Deadlockmarkierung besitzt, reicht es zu dessen Erkenntnis aus, einen Schaltprozeß aufzufinden, der in dieser Deadlockmarkierung endet. Eine monovalente Simulationsanalyse kann diesen Schaltprozeß durch die - allerdings blinde - Auswahl aktivierter Schaltschritte zufällig hervorbringen. Nur besteht keinerlei Sicherheit, ob die monovalente Simulationsanalyse tatsächlich auf diesen Schaltprozeß stößt.

Drittens führt die monovalente Simulationsanalyse zu potentiell inkorrekten Ergebnissen⁴⁶, falls ihre Unvollständigkeit und die daraus folgende Einschränkung auf günstige Analysesituationen übersehen oder bewußt ignoriert werden. Dies läßt sich wiederum anhand voranstehenden Beispiels erläutern, in dem es galt, die Deadlockfreiheit eines Netzmodells zu überprüfen. Im ungünstigen Fall enthält das Netzmodell tatsächlich mindestens einen Deadlock. Zufällig werden aber immer jene Schaltschritte ausgewählt, die keine Deadlockmarkierung als Folgemarkierung hervorbringen. Wenn aus dem Nichterreichen einer Deadlockmarkierung gefolgert wird, das untersuchte Netzmodell verhalte sich deadlockfrei⁴⁷, dann widerspricht dies dem tatsächlich vorhandenen Deadlock. Daher ist das Simulationsergebnis inkorrekt.

Aufgrund der voranstehenden Ausführungen ergibt sich für die monovalente Simulationsanalyse eines normalen Netzmodells eine Sachlage, die der früher erörterten Charakteristik der prädikatenlogischen Semi-Entscheidbarkeit ähnelt:

- ❑ Falls während einer Netzsimulation eine Deadlockmarkierung erreicht wird, dann ist die fehlende Deadlockfreiheit des untersuchten Netzmodells korrekt erkannt. Dies ist der oben angesprochene günstige Fall.
- ❑ Andernfalls verläuft die Netzsimulation ungünstig. Es wird bis zu ihrem Ende keine Deadlockmarkierung erreicht. Da die monovalente Simulation normaler Netzmodelle unvollständig ist, kann die Deadlockfreiheit des untersuchten Netzmodells nicht entschieden werden:
 - Entweder verhält sich das untersuchte Netzmodell tatsächlich deadlockfrei.
 - Oder das Netzmodell enthält mindestens einen Deadlock. Aber während der Netzsimulation wurde keine seiner Deadlockmarkierungen erreicht.
- ❑ Sofern das Nichterreichen einer Deadlockmarkierung als Deadlockfreiheit des untersuchten Netzmodells ausgelegt wird, erweist sich dieses Simulationsergebnis als potentiell inkorrekt⁴⁸).

Folglich gilt für die monovalente Netzsimulation: Mit ihrer Hilfe lassen sich einige interessante dynamische Netzeigenschaften untersuchen, aber keineswegs deren Gesamtheit. Diejenigen Netzeigenschaften, deren Analyse grundsätzlich möglich ist, können nur unter günstigen Umständen korrekt beurteilt werden. Falls über die letztgenannten Eigenschaften dennoch unter allen Umständen Urteile abgegeben werden, so sind diese potentiell inkorrekt. Diese dreifache Einschränkung des Erkenntnispotentials monovalenter Simulationsanalysen wurde oben als "rudimentär" umschrieben.

Um so mehr erstaunt es, daß sich die Netzsimulationen, die in der Netzliteratur so breiten Raum finden, nur auf die triviale Variante der monovalenten Simulationsanalyse erstrecken. Die wesentlich leistungsfähigere, anschließend näher betrachtete omnivalente Simulationsanalyse findet dagegen dort keine Beachtung. Als pars pro toto wird nur auf die simulative Analyse von

Netzmodellen verwiesen, mit deren Hilfe die Korrektheit - z.B. Deadlockfreiheit - von Telekommunikationsprotokollen⁴⁹⁾ "nachgewiesen" werden soll⁵⁰⁾. Aufgrund der oben dargelegten Zusammenhänge ist eine monovalente Netzsimulation hierzu jedoch grundsätzlich nicht in der Lage⁵¹⁾. Allenfalls vermag sie die erhoffte Deadlockfreiheit zu widerlegen, indem im untersuchten Netzmodell günstigenfalls ein Schaltprozeß ausgeführt wird, der in einer Deadlockmarkierung endet.

Die voranstehende Kritik an der monovalenten Simulationsanalyse von Netzmodellen bezieht sich jedoch nur auf das Außerachtlassen des Erkenntnispotentials ihres omnivalenten Pendant. Davon unberührt bleibt der Sachverhalt, daß sich die monovalente Simulationsanalyse ohne Einschränkungen für projektive Modellanalysen eignet. Sie untersuchen, in welcher Weise sich ein Schaltprozeß, der im Netzmodell durch einen Simulationslauf ausgeführt wird, auf die sachlichen und formalen Zielsetzungen des Modellierungsträgers auswirkt. Die Konsequenzen der Prozeßausführung werden auf das Erfüllungsausmaß des vorgegebenen Zielsystems "projiziert". Dadurch können z.B. "What if"-Fragen beantwortet werden⁵²⁾. Unvollständige polyvalente Simulationsanalysen lassen sich in analoger Weise für komparative Modellanalysen einsetzen. Dann werden in mehreren Simulationsläufen die zielsystemrelevanten Auswirkungen mehrerer alternativer Schaltprozesse simuliert. Sie können nach dem Abschluß aller involvierten Simulationsläufe miteinander verglichen werden.

Projektive und komparative Analysen von Netzmodellen sind durchaus von praktischen Interesse. Beispielsweise kann ein Netzmodell gestaltet werden, um in ihm die Zielwirksamkeit von Entscheidungsregeln zu untersuchen⁵³⁾. Es wird jeweils eine nicht-leere Regelmenge⁵⁴⁾ betrachtet, die es gestattet, zumindest einen Teil der Schaltkonflikte des Netzmodells aufzulösen. Die Regelmenge trägt dann zur Definition einer Konfliktstrategie bei, die - als Bestandteil einer umfassender definierten Schaltstrategie - das Schaltverhalten des Netzmodells bei vorgegebener Ausgangsmarkierung eindeutig determiniert⁵⁵⁾. Auf dieser Grundlage lassen sich im Netzmodell Schaltprozesse für unterschiedliche Ausgangsmarkierungen ausführen. Dabei werden die Auswirkungen der Prozeßausführungen auf das sachliche und formale Zielsystem des Modellierungsträgers projiziert.

Die Simulationsergebnisse werden in einer projektiven Simulationsanalyse zu einem Urteil über die Zielwirksamkeit der jeweils vorausgesetzten Entscheidungsregelmenge zusammengefaßt. Dieses Urteil über die Regelmengenwirksamkeit läßt sich auch situationsspezifisch ausdifferenzieren. Dabei wird den untersuchten Ausgangsmarkierungen des Netzmodells eine Klasseneinteilung derart überlagert, daß untereinander ähnliche Ausgangsmarkierungen zu jeweils einer gemeinsamen Simulationssituation aggregiert sind. In einer komparativen Simulationsanalyse werden alternative Entscheidungsregelmengen hinsichtlich ihrer, gegebenenfalls situationsspezifisch differenzierten Zielwirksamkeit verglichen⁵⁶⁾. Ein simulativ ausgewertetes Netzmodell bietet sich daher als eine Testumgebung an, in der sich Entscheidungsregelmengen durch projektive oder komparative Analysen beurteilen lassen.

Das Erkenntnispotential der monovalenten und der unvollständigen polyvalenten Simulationsanalyse, das durch die voranstehend skizzierten projektiven und komparativen Auswertungsmöglichkeiten bereits recht ansehnlich ausfällt, kann durch den Übergang zur omnivalenten Simulationsanalyse noch beträchtlich ausgeweitet werden. Bei der omnivalenten Simulationsanalyse werden - im Gegensatz zu ihrer monovalenten Antipode - die Schaltwirkungen *aller* Schaltschritte berücksichtigt, die unter *allen* erreichbaren Markierungen eines untersuchten Netzmodells aktiviert sind. Dadurch wird die Gesamtheit aller erreichbaren Markierungen des Netzmodells und aller zulässigen Schaltschrittausführungen erzeugt. Genau das leistet aber auch die Konstruktion des Erreichbarkeitsgraphen desselben Netzmodells. Daher besteht zwischen der omnivalenten Simulationsanalyse und der Konstruktion von Erreichbarkeitsgraphen nur ein terminologischer, aber kein materieller Unterschied⁵⁷⁾. Alle Untersuchungskonzepte, die in den voranstehenden Kapiteln für die Auswertung eines Erreichbarkeitsgraphen beschrieben wurden, lassen sich daher auch auf das Ergebnis einer omnivalenten Simulationsanalyse anwenden.

Dies hat zwei bemerkenswerte Konsequenzen. Einerseits ist die polyvalente Simulationsanalyse selbst dann, wenn sie in ihrer vollständigen Variante ausgeführt wird, wesentlich leistungärmer als die Erreichbarkeitsanalyse. Denn die omnivalente Simulationsanalyse vollbringt nur die Vorstufe der Erreichbarkeitsanalyse: Sie expliziert - wie die Konstruktion des Erreichbarkeitsgraphen - die vollständige dynamische Struktur des untersuchten Netzmodells durch die Generierung aller erreichbaren Markierungen und aller zulässigen Schaltschrittausführungen. Daraus allein folgt aber noch keine Erkenntnis über eine der früher vorgestellten interessanten dynamischen Netzeigenschaften. Andererseits läßt sich die omnivalente Simulationsanalyse mit allen Untersuchungskonzepten der Erreichbarkeitsanalyse kombinieren. Mit Hilfe der derart erweiterten omnivalenten Simulationsanalyse können im Prinzip alle Netzeigenschaften untersucht werden, die auch der Erreichbarkeitsanalyse zugänglich sind. Auf die Wiederholung dieses Erkenntnispotentials wird hier verzichtet, weil es schon früher im Kontext der Erreichbarkeitsanalyse behandelt worden ist.

Lediglich auf eine Besonderheit wird hingewiesen: Mit Hilfe der erweiterten omnivalenten Simulationsanalyse bereitet es keine Schwierigkeiten, die holistischen dynamischen Eigenschaften von Netzmodellen zu ermitteln. Dies gilt insbesondere auch für optimale Schaltwerte und -folgen. Dies widerspricht zwar der eingangs vorgetragenen Ansicht, daß sich optimale Modellösungen nur durch analytische, nicht aber durch simulative Modellauswertungen bestimmen ließen. Tatsächlich liefert aber die omnivalente Simulation eines finalen Netzmodells alle Informationen über erreichbare Markierungen und markierungstransformierende Schaltschritte, die zur Konstruktion eines Optimierungsnetzes erforderlich sind. Die Ermittlung eines finalen Schaltprozesses in diesem Optimierungsnetz, der an seinem Ende das gesuchte Netzoptimum hervorbringt, besteht lediglich aus sukzessiven Anwendungen der Schaltregel auf die Markierungen des Optimierungsnetzes. Daher wird dieser optimale Schaltprozeß durch eine Simulation des Optimierungsnetzes gewonnen⁵⁸⁾. Folglich wird das Netzoptimum des untersuchten Netzmodells auf rein simulative - wenn auch recht komplexe - Weise gewonnen. Die Simulationsanalyse von Netzmodellen eignet sich daher auch für optimierende Modellauswertungen.

Es könnte jedoch der Einwand erhoben werden, das Petrinetz-Konzept stelle im Grunde nichts anderes als das Fundament einer⁵⁹⁾ formalen Simulationssprache⁶⁰⁾ für die automaten-gestützte Gestaltung und Auswertung von Simulationsmodellen dar. Daher könne sie allenfalls so viel wie - aber auf keinen Fall mehr als - andere, konventionelle Simulationssprachen leisten. Jene konventionellen Simulationssprachen würden bereits durch umfangreiche Softwarepakete unterstützt und erfreuten sich eines großen Bekanntheitsgrads. Folglich sei es nicht einzusehen, warum anstelle dieser konventionellen Simulationssprachen auf die simulative Auswertung von Netzmodellen zurückgegriffen werden solle.

Diese Einlassung trifft hinsichtlich ihres ersten Teilaspekts, daß sich das Petrinetz-Konzept als Fundament einer Simulationssprache betrachten läßt, grundsätzlich zu. Denn das sukzessive Anwenden der Schaltregel auf ein Netzmodell bringt einen Schaltprozeß hervor, der sich ebenso als ein Simulationslauf des betrachteten Netzmodells auffassen läßt. Daher kann die Definition eines Petrinetzes⁶¹⁾ als die implizite Festlegung einer petrinetzgestützten Simulationssprache betrachtet werden⁶²⁾: Diese Simulationssprache umfaßt als Sprachextension genau diejenigen Simulationsmodelle, die als Netzmodelle die vorausgesetzte Netzdefinition erfüllen. Darüber hinaus ist es möglich, mehrere tiefgreifende Übereinstimmungen zwischen dem Petrinetz-Konzept einerseits und einigen speziellen, aber weit verbreiteten Simulationskonzepten andererseits zu identifizieren. Hierzu gehören insbesondere:

- die Unterscheidung zwischen aktiven und passiven Modellkomponenten⁶³⁾;
- die explizite, unmittelbare Modellierung beweglicher Objekte⁶⁴⁾;
- die Konzeptualisierung der dynamischen Modellstruktur durch Systemzustände⁶⁵⁾ und zustandstransformierende Ereignisgeschehnisse⁶⁶⁾ auf einer diskreten Zeitskala⁶⁷⁾;

- die Determinierung der Zustandstransformationen durch lokal definierte Übergangsregeln⁶⁸;
- die Modellgestaltung auf der Basis des objektorientierten Ansatzes⁶⁹.

Die oben vorgetragene These, die Simulationsanalyse von Netzmodellen könne allenfalls so viel wie die Verwendung von bereits etablierten Simulationssprachen leisten, kann sogar noch verschärft werden. Denn konventionelle Simulationssprachen besitzen - wie alle Höheren Programmiersprachen - die Ausdrucksmächtigkeit von TURING-Automaten⁷⁰. Petrinetze fehlt dagegen in ihren bisher vorgestellten Varianten der Stelle/Transition-Netze und der Synthetischen Netze diese TURING-Mächtigkeit. Insofern verhält sich die petrinetzgestützte Simulationssprache zunächst ausdrucksärmer als konventionelle Simulationssprachen. Es wird jedoch später aufgezeigt, daß die Ergänzung von nur einem neuartigen Netzkonstrukt - den Nulltestkanten - ausreicht, um Petrinetze ebenso die Ausdrucksfähigkeit von TURING-Automaten zu verleihen. Daher läßt sich die petrinetzgestützte Simulationssprache nachträglich auf das Ausdrucksvermögen ihrer konventionellen Pendanten anheben.

Allerdings gilt die grundsätzliche Gleichmächtigkeit des Ausdrucksvermögens nicht in jeder Hinsicht. Denn die Gestaltung von Netzmodellen erweist sich prinzipiell ausdrucksärmer als manche konventionelle Simulationssprachen, wenn die Unterscheidung zwischen diskreten⁷¹ und kontinuierlichen⁷² Simulationsmodellen betrachtet wird: Die Simulationsanalyse von Netzmodellen deckt *nur* die Klasse der diskreten Simulationsmodelle ab. Kontinuierliche Simulationsmodelle, die mit der Hilfe konventioneller Simulationssprachen oftmals ebenso gestaltet und ausgewertet werden können⁷³, scheiden dagegen aus dem Leistungsspektrum einer petrinetzgestützten Simulationssprache von vornherein aus. Denn das Petrinetz-Konzept beruht auf einem Prozeßbegriff⁷⁴, der sich ausschließlich mit diskreten Simulationsmodellen vereinbaren läßt. Die Schaltprozesse, die in Netzmodellen ausgeführt werden können, beruhen auf Schaltakten von Transitionen. Diese Schaltakte stellen zeitlose Ereignisgeschehnisse dar. Sie bewirken einen abrupten Übergang zwischen zwei Zuständen eines Netzmodells: seiner Referenz- und seiner Folgemarkierung. Damit entsprechen alle Simulationsmodelle, die mit Hilfe einer petrinetzgestützten Simulationssprache formuliert worden sind, dem diskreten, aber nicht dem kontinuierlichen Modellierungsansatz. Deshalb stehen alle anschließenden Ausführungen unter dem Vorbehalt, sich grundsätzlich nur auf diskrete Simulationsmodelle zu beziehen.

Der Verf. schätzt das Unvermögen des Petrinetz-Konzepts, kontinuierliche Simulationsmodelle zu überdecken, jedoch nicht als eine gravierende Konzeptschwäche ein. Denn durch beliebig kleine Festlegung der Zeiteinheit läßt sich die Zustandsfolge diskreter Simulationsmodelle in jedem erwünschten - aber finiten - Ausmaß an den Zeitfluß kontinuierlicher Simulationsmodelle annähern⁷⁵. Bedeutsamer erscheinen ihm die Konsequenzen, die aus dem oben vorgetragenen Simulationssprachen-Argument gefolgert wurden. Träfen sie zu, so wären die weit verbreiteten Simulationsanalysen von Netzmodellen in der Tat überflüssig. Tatsächlich werden sie dem Petrinetz-Konzept jedoch aus mehreren Gründen nicht gerecht.

Am schwersten wiegt, daß die fundamentale Diskrepanz vernachlässigt wird, die zwischen mono- und omnivalenten Simulationsanalysen besteht. Konventionelle Simulationssprachen sehen zunächst⁷⁶ nur die monovalente Auswertung von Simulationsmodellen vor. Das Petrinetz-Konzept zeichnet sich hingegen dadurch aus, auch dann, wenn es als Fundament einer Simulationssprache eingestuft wird, neben der monovalenten Simulationsanalyse auch die omnivalente Simulationsanalyse zuzulassen⁷⁷. Dadurch wird das gesamte Erkenntnispotential der omnivalenten Simulationsanalyse erschlossen, das um die weitreichenden Untersuchungskonzepte der Erreichbarkeitsanalyse erweitert ist. Es wurde oben dargelegt. Insbesondere unterliegt es nicht mehr den drei charakteristischen Einschränkungen aller monovalenten Simulationsanalysen: der Unfähigkeit, holistische dynamische Eigenschaften zu überprüfen, der Abhängigkeit von günstigen Analysefällen und der potentiellen Inkorrektheit bei Ausdehnung auf alle Analysefälle. Damit reicht die prinzipielle Leistungsfähigkeit der petrinetzgestützten Simulationssprache weit über das Vermögen konventioneller Simulationssprachen hinaus⁷⁸.

Daran ändert sich auch nichts, wenn eine konventionelle Simulationssprache im Rahmen der stochastischen Simulation eingesetzt wird. Denn die stochastische Simulationsanalyse⁷⁹⁾ erweitert zwar den monovalenten zu einem polyvalenten Analyseansatz. Aber diese polyvalente Analysevariante bleibt weiterhin unvollständig⁸⁰⁾. Daher vermag auch sie das Leistungsvermögen der omnivalenten Simulationsanalyse von Netzmodellen grundsätzlich nicht zu erreichen.

Ausgangspunkt der stochastischen Simulationsanalyse ist ein parametrisches Simulationsmodell⁸¹⁾. In ihm wird mindestens eine Konstante aus der variablenfreien Spezifikation des Ausgangszustands eines gewöhnlichen Simulationsmodells⁸²⁾ als "Parameter"⁸³⁾ behandelt⁸⁴⁾. Er vermag alternative Ausprägungen aus einem wohlbestimmten Definitionsbereich anzunehmen. Daher stellt er strenggenommen eine Variable dar⁸⁵⁾. Anschließend wird allen Ausprägungen desselben Parameters eine Wahrscheinlichkeits- oder eine Dichtefunktion überlagert je nachdem, ob er einen diskreten bzw. kontinuierlichen Definitionsbereich besitzt. Diese Überlagerung erfolgt für alle eingeführten Parameter. Dadurch geht ein parametrisches Simulationsmodell in die spezielle Gestalt eines stochastischen Simulationsmodells über⁸⁶⁾. Eine Parameterkonfiguration weist jedem Parameter des stochastischen Simulationsmodells genau eine Ausprägung aus dem Definitionsbereich des Parameters zu⁸⁷⁾. Deshalb legt jede Parameterkonfiguration für ein stochastisches Simulationsmodell genau einen variablenfreien Ausgangszustand fest. Insofern handelt es sich bei einem stochastischen Simulationsmodell um eine Familie von gewöhnlichen Simulationsmodellen, die sich nur durch die unterschiedlichen Parameterfestlegungen ihrer Ausgangszustände voneinander unterscheiden. Die Familienmitglieder werden daher auch als parametrische Modellvarianten angesprochen.

Bei der Simulationsanalyse eines stochastischen Simulationsmodells werden endlich viele⁸⁸⁾ Parameterkonfigurationen mit der Hilfe eines Zufallsgenerators erzeugt. Dabei werden der Konfigurationsgenerierung die oben angeführten Wahrscheinlichkeits- bzw. Dichtefunktionen überlagert. Für jede Parameterkonfiguration wird die zugehörige parametrische Modellvariante in einem eigenständigen Simulationslauf wie jedes gewöhnliche Simulationsmodell untersucht. Die Ergebnisse der einzelnen Simulationsläufe werden abschließend zu einer relativen Häufigkeitsverteilung überlagert⁸⁹⁾. Diese Häufigkeitsverteilung der Simulationsergebnisse läßt sich als eine Schätzung der untersuchten stochastischen Modelleigenschaften interpretieren⁹⁰⁾. Allerdings stellt - bei der Anwendung konventioneller Simulationssprachen - jeder involvierte Simulationslauf weiterhin eine monovalente Simulationsanalyse der jeweils generierten parametrischen Modellvariante dar⁹¹⁾. Daher bleiben konventionelle Analysen von stochastischen Simulationsmodellen auf das Erkenntnisvermögen von iterierten monovalenten Simulationsanalysen⁹²⁾ beschränkt. Sie besitzen die Qualität einer speziellen unvollständigen polyvalenten Simulationsanalyse⁹³⁾.

Die Unvollständigkeit von stochastischen Simulationsanalysen resultiert aus zwei verschiedenen Ursachen. Erstens werden für eine *gegebene* Parameterkonfiguration innerhalb desselben Simulationslaufs *keine alternativen* Simulationsfortsetzungen berücksichtigt. Schon aus diesem Grunde erfolgt keine *vollständige* polyvalente Simulationsanalyse. Zweitens schöpfen die endlich vielen generierten Parameterkonfigurationen im allgemeinen⁹⁴⁾ den Konfigurationsraum aller kombinatorisch möglichen Parameterkonfigurationen nicht aus. Daher wird das Verhaltenspotential eines stochastischen Simulationsmodells im Regelfall nur stichprobenartig - und somit unvollständig - erforscht.

Die stochastische Simulation reicht zwar aus, um Schätzwerte für stochastische Modelleigenschaften zu ermitteln. Aber hinsichtlich der interessanten dynamischen Modelleigenschaften, die früher für Netzmodelle herausgestellt wurden, bleibt der stochastische Simulationsansatz wegen seiner doppelten Unvollständigkeit auf das rudimentäre Erkenntnispotential aller unvollständigen mono- oder polyvalenten Simulationsanalysen beschränkt. Der wesentliche Grund dafür ist die erstgenannte Unvollständigkeitsursache⁹⁵⁾: die lediglich monovalente Simulationsanalyse einer parametrischen Modellvariante während eines Simulationslaufs. Die Iteration solcher monovalenten Simulationsläufe für unterschiedliche Parameterkonfigurationen vermag die

dreifache Einschränkung, der alle unvollständigen Simulationsanalysen unterliegen, nicht zu überwinden. Denn für jede untersuchte parametrische Modellvariante gilt wiederum: Holistische dynamische Eigenschaften, welche die Kenntnis der *gesamten* dynamischen Struktur der Modellvariante voraussetzen, lassen sich grundsätzlich nicht erkennen. Andere Modelleigenschaften können nur unter *günstigen* Umständen korrekt erkannt werden. Falls sie dennoch unter allen Umständen beurteilt werden, fallen die Simulationsergebnisse potentiell inkorrekt aus. Die wiederholte Untersuchung verschiedener Modellvarianten ändert an diesen Einschränkungen nichts. Denn es werden nur die Ausgangszustände des - ansonsten gleichen - Simulationsmodells variiert, ohne innerhalb desselben Simulationslaufs alternative Simulationsfortsetzungen zu berücksichtigen.

Die erweiterte omnivalente Simulationsanalyse von Netzmodellen läßt sich dagegen mit der stochastischen Simulationsanalyse so verknüpfen, daß die dreifachen Erkenntnisstrahlen iterierter monovalenter Simulationsläufe überwunden werden⁹⁶). Dann wird ein stochastisches Netzmodell für jede seiner Parameterkonfigurationen einer eigenständigen erweiterten omnivalenten Simulationsanalyse unterzogen. Bei jedem dieser omnivalenten Simulationsläufe werden alle alternativen Simulationsfortsetzungen berücksichtigt, die für die jeweils untersuchte Parameterkonfiguration des Netzmodells möglich sind. In diesem Zusammenhang wird insbesondere auf die Kombination des Petrinetz-Konzepts mit dem ATMS-Konzept begründungsverwaltender Systeme aus der Erforschung Künstlicher Intelligenz verwiesen. Dabei kann jede Parameterkonfiguration zusammen mit allen hierzu alternativen Simulationsfortsetzungen in der Gestalt von hypothetischen Annahmen nebenläufig bearbeitet werden. Die daraus gewonnenen, konfigurationsspezifischen Simulationsergebnisse werden für alle generierten Parameterkonfigurationen des Netzmodells zusammengetragen und in der oben skizzierten, konventionellen Weise zu einer relativen Häufigkeitsverteilung überlagert.

Selbst wenn "nur" eine einfache monovalente Simulation von Netzmodellen in Betracht gezogen wird, läßt sich der oben vorgetragene Einwand gegenüber dem Petrinetz-Konzept dennoch nicht uneingeschränkt aufrechterhalten. Denn dann kann geltend gemacht werden, daß konventionelle Simulationssprachen auf ihr eigenes - begrenztes - Leistungsvermögen beschränkt sind. Eine petrinetzgestützte Simulationssprache kann hingegen mit allen anderen Analysetechniken kombiniert werden, die für Netzmodelle in den voranstehenden Kapiteln vorgestellt wurden. Daher ist es möglich, ein Simulationsmodell, das als ein Petrinetz gestaltet wurde, nicht nur mit Hilfe der (monovalenten) Simulationsanalyse auszuwerten. Statt dessen läßt sich dasselbe "Simulations"modell um eine Deduktions-, Invarianten-, Reduktions- oder Erreichbarkeitsanalyse bereichern. Auf diese Weise kann mit der Hilfe einer "Simulations"sprache ein Modell gestaltet werden, das sich nachträglich - ohne eine Reformulierung des Modells - auch mit nicht-simulativen Analysetechniken untersuchen läßt. Dieses Kombinationspotential mit alternativen Analysetechniken besitzen konventionelle Simulationssprachen entweder überhaupt nicht⁹⁷) oder aber nur in wesentlich geringerem Ausmaß⁹⁸).

Neben der analytischen Bereicherungsmöglichkeit weist die petrinetzgestützte Simulationssprache weitere Vorzüge auf. Dazu gehört vor allem das breite theoretische Fundament des Petrinetz-Konzepts, das in ähnlicher Qualität für keine der konventionellen Simulationssprachen bekannt ist. In exemplarischer Weise werden als theoretische Grundlagen des Petrinetz-Konzepts hervorgehoben:

- ❑ seine Verwurzelung in der Prädikatenlogik 1. Stufe;
- ❑ seine Einbindung des algebraischen Signaturkonzepts;
- ❑ die Möglichkeit, in Netztheoremen für deklarative Netzmodelle das Inferenzkonzept der kombinierten Unifizierung und Resolution sortierter algebraisch-prädikatenlogischer Formeln einzubeziehen;
- ❑ die Existenz einer axiomatisch fundierten Netztheorie.

Darüber hinaus ist auch die kognitive Adäquanz einer petrinetzgestützten Simulationssprache hervorzuheben. Sie beruht auf der "Animation" von Netzsimulationen durch die Visualisierung des Markenflusses. Er durchströmt die graphischen Repräsentationen der jeweils untersuchten Netze. Auf die kognitionspsychologischen Vorzüge von graphischen Visualisierungen wurde bereits im Zusammenhang mit Allgemeinen Netzen hingewiesen. Sie bezogen sich jedoch nur auf den bipartiten Graphen, der ein Allgemeines Netz repräsentiert. Die Visualisierung eines solchen Graphen besitzt einen statischen Charakter, weil er nur die invariante topologische Netzstruktur wiedergibt. Bei der Visualisierung des Markenflusses, der ein Stelle/Transition-Netz oder ein Synthetisches Netz durchsetzt, kommt jedoch die kinetische Komponente der beweglichen Markenkopien hinzu. Mit ihrer Hilfe lassen sich auch Netzverhaltensweisen in der Anschauungsform "Zeit" eindrucksvoll veranschaulichen⁹⁹). Dabei kann auf Unterstützungsleistungen der CAA-Technik¹⁰⁰) zurückgegriffen werden, um den Markenfluß in einem Netzmodell zu visualisieren¹⁰¹).

Aufgrund ihrer graphischen Animationsqualität eignen sich Netzsimulationen hervorragend, um die interaktive Nutzung von PPS-Systemen¹⁰²) im Dialogbetrieb¹⁰³) in benutzerfreundlicher Weise zu unterstützen. Dabei wird das graphisch visualisierte Netzmodell eines Produktionssystems auf einem hochauflösenden Graphikbildschirm präsentiert¹⁰⁴). Er stellt die Hauptkomponente¹⁰⁵) einer netzbasierten Graphikschnittstelle¹⁰⁶) zwischen einem PPS-System und seinem Benutzer dar. Auf diesem Graphikbildschirm zeigen die Marken, die im Netzmodell Werkstücke oder Werkzeuge repräsentieren, die aktuelle Verteilung dieser Objekte im Produktionssystem an¹⁰⁷). Das Fortschalten der Marken durch Schaltakte von Transitionen veranschaulicht den Fluß von Werkstücken und Werkzeugen durch das Produktionssystem in plastischer Weise. Der Benutzer eines interaktiven PPS-Systems kann zunächst die Auswirkungen von Koordinierungsentscheidungen¹⁰⁸), die er in Betracht zieht, im Netzmodell des Produktionssystems simulieren¹⁰⁹). Anschließend lassen sich die Simulationsergebnisse an der netzbasierten Graphikschnittstelle des PPS-Systems als Markenverteilung oder -fluß benutzerfreundlich präsentieren. Diese graphische Animation von Entscheidungswirkungen verschafft dem Benutzer einen transparenten Überblick über die Konsequenzen seiner Koordinierungsoptionen.

Die Kombination aus einem simulativ ausgewerteten Netzmodell, das ein zugrundeliegendes Produktionssystem repräsentiert, und aus einem interaktiven PPS-System, das die visualisierte Netzdarstellung an einer Graphikschnittstelle präsentiert, stellt ein netzbasiertes Simulationssystem zur Unterstützung von Koordinierungsentscheidungen dar. Wegen der kognitiven Adäquanz seiner graphischen Animationsfähigkeiten leistet ein solches entscheidungsunterstützendes System einen bemerkenswerten Beitrag zur Entwicklung von benutzerfreundlichen PPS-Systemen¹¹⁰), die auf dem Prinzip interaktiver Simulation beruhen.

Neuere Entwicklungen auf dem Gebiet konventioneller Simulationssprachen weisen in die gleiche Richtung. Auch dort werden in ersten Ansätzen Animationskomponenten angeboten, die es ermöglichen, Objektflüsse in den modellierten Systemen graphisch zu veranschaulichen¹¹¹). Allerdings handelt es sich bei diesen Animationskomponenten um Ergänzungen, die auf die Simulationssprachen nur "aufgesetzt" sind. Es besteht keine enge konzeptionelle Verknüpfung zwischen Sprachkonstruktion und Objektfluß. Bei einer petrinetzgestützten Simulationssprache ist das Gegenteil der Fall. Dort wird der Objektfluß als das Fortschalten von Markenkopien unmittelbar in zwei Konstituenten der Netzdefinition - den Marken und der Schaltregel - verankert. Daher kann die kognitionspsychologisch willkommene Visualisierung von Objektflüssen in Simulationsmodellen auf ein wohldefiniertes konzeptionelles Fundament zurückgeführt werden, wenn der Modellgestaltung Petrinetze zugrundeliegen. Für konventionelle Simulationssprachen läßt sich eine ähnliche Animationsbasis nicht erkennen.

Anmerkungen zum Kapitel:

1) Vgl. zur Simulationsanalyse von Netzen HOLT, A. (1968), S. 282ff.; SHAPIRO, R. (1971), S. 275ff.; NOE (1971), S. 369ff.; JOHNSON, R.R. (1972), S. 161 u. 167ff.; NOE (1972), S. 751ff.; AGERWALA (1973), S. 81; SHAPIRO, R. (1975b), S. 99; DADDA (1976c), S. 49 u. 64; WINTER, R. (1977), S. 95 u. 99ff.; BAER, J. (1977), S. 395 u. 398ff.; PERNARDS (1977), S. 212ff.; KWAN (1977a), S. 608; COX, L. (1978), S. 901ff.; GONZALEZ, T. (1978), S. 92ff.; NIETERS (1978), S. 32f.; HAN (1978a), S. 166; SZLANCO (1978), S. 75; ELLIS, C.J. (1979), S. 28ff.; HAN (1979), S. 271; PAKAS-SKEWES (1979), S. VII; RAMAMOORTHY (1980), S. 440f.; WINAND (1980), S. 1253; ANDRE (1980), S. 332; HURA (1981b), S. 83f.; MÜLLER, B. (1981), S. 202; KRÄMER (1981), S. 470; EBERT, J. (1981), S. 327; ROLIN (1982), S. 11f. u. 21; FLORIN (1982), S. 280; HACKMANN (1982), S. 85; HURA (1982c), S. 437; FREEDMAN (1988b), S. 334; VON KLEIST-RETZOW (1991), S. 257; VON MARTIAL (1991a), S. 315 u. 324; SCHEER (1991d), S. 130 (nur ein knapper Hinweis); ROSENSTENGEL (1991), S. 100f.

Eine besondere Rolle spielt die Klasse der Auswertungsnetze. Sie wurde speziell für die Aufgabe entwickelt, Automatische Informationsverarbeitungssysteme zu modellieren. Mit Hilfe der Netzmodelle soll die Verarbeitungsleistung der Systeme studiert und - gegebenenfalls - verbessert werden; vgl. z.B. NOE (1971), S. 371; BAER, J. (1977), S. 395.

In den Frühzeiten des Petrinetz-Konzepts galt die simulative Untersuchung von Netzmodellen sogar als die einzige "praktikable" Auswertungstechnik. Denn für die "analytische" Netzauswertung standen noch keine hinreichend leistungsfähigen Algorithmen zur Verfügung. Vgl. dazu SHAPIRO, R. (1975b), S. 99. Seitdem konnte zwar das Leistungsvermögen der Auswertungsalternativen, die in den voranstehenden Kapiteln erläutert wurden, erheblich gesteigert werden. Aber die Simulationsanalyse spielt für das Studium von Netzmodellen weiterhin eine herausragende Rolle. Dies belegen die o.a. Quellen. Zur Beliebtheit der simulativen Auswertung von Netzmodellen trägt vor allem bei, daß sich die atomaren Netzkomponenten - die Stellen und Transitionen - auf einfache Weise durch entsprechende Konstrukte von Simulationssprachen implementieren lassen. So findet sich schon bei NOE (1971), S. 370, die Anregung, die Transitionen eines Netzes direkt als Routinen einer Simulationssprache zu implementieren. In der Folgezeit wurde erfreute sich die Simulationssprache SIMULA besonderer Beliebtheit bei dem Vorhaben, Netzmodelle zu implementieren; vgl. SCHUMACHER (1975), S. 409f.; RAMÖLLER (1976), S. 16ff. Darüber hinaus diskutiert SCHUMACHER (1975), S. 410, weitere Simulationssprachen hinsichtlich ihrer Eignung, für die Simulation von Netzmodellen eingesetzt zu werden.

2) Sie wird auch kurz als Netzsimulation bezeichnet.

3) Vgl. zu solchen Softwarepaketen mit der Fähigkeit zur Netzsimulation RAZOUK (1985c), S. 20f. (P-NUT).

4) Der Einsatz von Simulationen im Rahmen von PPS-Systemen oder Maschinenbelegungsplanungen wurde bereits zu Beginn dieser Arbeit in einer Anmerkung anhand zahlreicher Quellen belegt. Gleiches gilt für die Anwendung simulativer Konzepte, die zur Koordinierung von Produktionsprozessen in Flexiblen Fertigungssystemen herangezogen werden (Quellenbelege in derselben Anmerkung). Die erhebliche Aufmerksamkeit für Simulationsstudien wird im allgemeinen dadurch gerechtfertigt, daß sich mit ihrer Hilfe Modellierungsaspekte untersuchen lassen, die mit anderen - "analytischen" - Modellierungen in der Regel nicht erfaßt werden können. Dies wird vor allem darauf zurückgeführt, daß Simulationen realistischere Problemkonzeptualisierungen ermöglichen, weil sie auf viele der sonst üblichen Idealisierungen von "analytischen" Modellen verzichten. Typische Beispiele für diesen Argumentationsstrang finden sich bei CLARKSON (1960), S. 923, 926 u. 930, sowie NYHUIS, P. (1989), S. 40. Auf die Schwierigkeit, die hierbei unterstellte Antipode "analytischer" Modellierungen inhaltlich zu konkretisieren, wird in Kürze zurückgekommen.

Aus einer anderen Perspektive wurde die Bedeutung von Simulationen bereits gewürdigt. Dort wurde das simulative Erforschen eines Produktionsmodells als ein Instrument vorgestellt, das es gestattet, *lokale* Entscheidungsalternativen durch die Vorausschau ihrer hypothetischen Konsequenzen in späteren Zuständen des Produktionssystems zu beurteilen. Darüber hinaus läßt sich die Simulation von Produktionsmodellen ebenso in globaler Weise benutzen, um zu überprüfen, ob ein Produktionsplan in seiner Gesamtheit verwirklicht werden kann. Das entspricht der simulativen Beurteilung der Zulässigkeit einer *globalen* Entscheidungsalternative. Vgl. MÜLLER, A. (1987), S. 350, 352 u. 358ff., die im Rahmen eines Simulationsmodells der Frage nach der "Realisierbarkeit von Produktionsplänen" (S. 352) nachgeht.

5) Allerdings wird der Abgrenzung gegenüber konventionellen Simulationssprachen an anderer Stelle größere Beachtung gewidmet. Dies resultiert aus einem möglichen Einwand, der scheinbar nahelegt, auf Simulationsanalysen von Netzmodellen grundsätzlich zu verzichten.

6) Die Simulationsattribute der Mono-, Poly- und Omnivalenz werden in diesem Kapitel noch eingehend erläutert.

7) Auf welche Weise sich dieses Spektrum aus den beiden hier behandelten Antipoden gewinnen läßt, wird an anderer Stelle anhand der dort skizzierten Mischstrategien angesprochen. Daraus folgt ein Strategiespektrum unvollständig polyvalenter Simulationsanalysen.

8) Diese Systemvorstellung kann als internes mentales Modell oder auch als explizites, aber noch natürlichsprachlich umschriebenes Modell vorliegen.

9) Vgl. insbesondere TEMPELMEIER, H. (1989a), S. 1; vgl. daneben auch mit ähnlichen Simulationsauffassungen MÜLLER-MERBACH (1973), S. 451; KERN, W. (1987), S. 116; HINTZ (1987), S. 181; MÜLLER, A. (1987), S. 346.

In den vorgenannten Quellen wird der Aspekt der Realproblemabbildung oftmals ausgeklammert. Dann läßt sich eine Simulation kurz als Experimentieren mit dem Modell eines realen Systems betrachten. Darüber hinaus wird im allgemeinen stillschweigend vorausgesetzt, daß das Systemmodell explizit vorliegt und mit Hilfe eines Automatischen Informationsverarbeitungssystems implementiert ist. Dadurch werden Gedankenexperimente als Umgang mit mentalen Modellen von realen Systemen aus dem Simulationsbegriff ausgegrenzt. Zu den seltenen Ausnahmen, die Gedankenexperimente als Spielart von Simulationen ausdrücklich berücksichtigen, gehört dagegen GOMEZ, P. (1978), S. 141. Der Verf. schließt sich jedoch im folgenden der Prämisse an, daß stets eine automatengestützte Modellimplementierung vorliegt. Es werden also ausschließlich "Computersimulationen" im Sinne von GOMEZ, P. (1978), S. 142, betrachtet. Unter dieser Voraussetzung kann das Experimentieren mit dem Systemmodell auch als Interaktion eines Experimentators mit einem Automatischen Informationsverarbeitungssystem beschrieben werden. Ein vollkommen anders ausgerichtetes, hier nicht berücksichtigtes Simulationsverständnis vertritt STACHOWIAK (1973), S. 341f.

10) Das Begriffsfeld analytischer Modellauswertungen umfaßt vielfältige Varianten, die hier nicht weiter differenziert werden. Dazu gehören z.B. analytische Auswertungsmethoden oder analytische Lösungsverfahren. Alle vorgenannten Begriffsvarianten werden im Regelfall verwendet, ohne dabei zu präzisieren, was denn eine Modellauswertung, eine Auswertungsmethode bzw. ein Lösungsverfahren als "analytisch" qualifiziert. Vgl. zu solchen unreflektierten Verwendungen des Begriffs "analytisch" CLARKSON (1960), S. 922; TEMPELMEIER, H. (1989a), S. 1; HEINHOLD (1989), S. 691.

Zu den wenigen, hier jedoch nicht weiterhelfenden Versuchen, den Begriffsinhalt von "analytischen" Konzepten auszuleuchten, gehört z.B. HANSSMANN (1987), S. 86. Analytische Modelle sollen seiner Ansicht nach durch mathematische Standardoperationen "direkt" berechenbar sein. Was die direkte Berechenbarkeit konkret bedeuten soll, bleibt jedoch im Dunkeln. Davon abgesehen erscheint die Voraussetzung direkter Berechenbarkeit angesichts komplizierter "analytischer" Modelle fragwürdig. Der gleiche Einwand gilt gegenüber der Einlassung von HILTY (1987), S. 216, bei analytischen Berechnungen lasse sich "das Ergebnis in einem Schritt" feststellen. Denn das berechnende Auswerten analytischer Modelle erfordert im allgemeinen eine Vielzahl von Umformungsschritten. Daher könnte auf den ersten Blick eher den Auffassungen von JANICH (1974), S. 85, und SCHURZ (1982), S. 322, zugestimmt werden. Sie betrachten Konstruktumformungen als analytisch, falls es sich um logische oder definitorische Umformungen handelt. Allerdings fällt es dem Verf. schwer, im Hinblick auf die "logischen" Umformungen zu beurteilen, ob darin die mannigfaltigen mathematischen Umformungen - wie z.B. Äquivalenztransformationen - und die Anwendungen des Übergangsschemas von Synthetischen Netzen eingeschlossen sind. Wenn dies der Fall ist, besitzen letztlich auch Simulationsmodelle einen "analytischen" Charakter, weil ihre Auswertungen ebenso auf mathematischen Umformungen beruhen. Andernfalls ließen sich kaum analytische Modelle identifizieren, deren Auswertung nur logische und definitorische, aber keine mathematischen Umformungen umgreifen darf. Folglich führt auch die Auszeichnung bestimmter "analytischer" Konstruktumformungen zu keiner befriedigenden Abgrenzung zwischen analytischen und simulativen Konzepten.

Eine weitere Differenzierungsvariante hat ROSKI (1984), S. 515, vorgelegt. Er spricht von einer analytischen Lösung, wenn sie auf eine "formelmäßige, und nicht rein numerische" Weise gewonnen worden ist. Dabei bleibt aber unklar, wie die Lösung eines Formelsystems zu klassifizieren ist, das aus Restriktionsformeln besteht und durch "Restriktionspropagierung" gelöst wird. Darüber hinaus fielen etliche Algorithmen des Operations Research aus dem Bereich analytischer Lösungsverfahren, weil sie auf rein numerischen Operationen beruhen. Dazu gehören z.B. Branch and bound-Algorithmen zur Lösung kombinatorischer Optimierungsmodelle ebenso wie der Simplex-Algorithmus. Der Verf. zweifelt, ob ROSKI so weit gehen möchte, solche Algorithmen tatsächlich als Simulationsverfahren zu klassifizieren. Schließlich tauchen auch erhebliche Schwierigkeiten auf, wenn aus ROSKI's Definition analytischer Lösungen gefolgert wird, daß Simulationen rein numerische Lösungsverfahren darstellen. Diese Simulationsauffassung ist zwar weit verbreitet. Aber sie ignoriert, daß schon seit längerer Zeit nicht-numerische oder "symbolische" Simulationsverfahren diskutiert werden. Vgl. dazu z.B. den frühen Beitrag von CLARKSON und SIMON in CLARKSON (1960), S. 924f(f). Sie sprechen darin explizit von "nonnumerical simulation" (S. 924).

Ebensowenig hilft ein Rekurs auf die mathematische Analysis weiter. Denn einige als "analytisch" etablierte Methoden für die Lösung von Modellen können nicht aus dem Kalkül der Analysis abgeleitet werden. Hierzu gehören vor allem die kombinatorischen Lösungsmethoden, die etwa bei KRYCHA (1972), S. 76ff., und NIEB (1980), S. 28, als analytisch eingestuft werden. Vgl. zu weiteren Definitionsansätzen für analytische Konzepte KERN, W. (1987), S. 30 u. 75 (analytische Optimierungskonzepte); HERDEN (1990), S. 15 (analytisches Wissen sowie analytische Problem-beschreibungs- und -lösungsansätze). Sie leiden unter ähnlichen Mängeln wie die oben erörterten.

11) Um Mißverständnisse zu vermeiden, wird ausdrücklich darauf hingewiesen, daß die Formulierungen "analytisches Konzept" und "Analysekonzept" (Analysetechnik) in dieser Arbeit deutlich unterschieden werden. In der For-

mulierung "Analysekonzept" wird der Analysebegriff in naiv-umgangssprachlicher Weise synonym zum allgemeinen Untersuchungsbegriff verwendet. Die Formulierung "analytisches Konzept" gebraucht dagegen das Attribut "analytisch" in einem speziellen, von simulativen Untersuchungskonzepten abgegrenzten Verständnis. Vgl. dazu die Präzisierung analytischer Konzepte.

12) Vgl. CLARKSON (1960), S. 922; HANSSMANN (1987), S. 86; HILTY (1987), S. 216ff.; TEMPELMEIER, H. (1989a), S. 1; HEINHOLD (1989), S. 702, Anmk. 21; RAMAN (1989b), S. 15.

13) Daneben läßt sich auch ohne größere Schwierigkeiten zeigen, daß Konzepte, die gewöhnlich als "analytisch" eingestuft werden, dennoch die oben angeführte konventionelle Simulationsdefinition erfüllen können. Beispielsweise werden Warteschlangenmodelle oder Modelle aus dem Bereich der Dynamischen Optimierung betrachtet. Es handelt sich jeweils um kinetische oder dynamische Modelle, die reale Probleme in systemorientierter Weise abbilden. Auch das Verhalten solcher Modelle läßt sich experimentell auswerten, indem etwa die - ansonsten unveränderten - Modelle mit unterschiedlichen Restriktionen oder alternativen Zielfunktionsformulierungen "durchgespielt" werden. Das experimentelle Untersuchen von Modellverhaltensweisen kann also alleine - d.h. *ohne präzisierende Zusätze* - ausreichen, um eine Simulation gegenüber analytischen Modellauswertungen auszuzeichnen. Der Unterschied zwischen beiden Untersuchungskonzepten muß in einem Aliud liegen.

14) Vgl. zur weit verbreiteten Auffassung, mit der Hilfe von Simulationsanalysen ließen sich grundsätzlich keine optimalen Modelllösungen ermitteln, MEYER ZU SELHAUSEN (1980b), S. 135; PRESSMAR (1982), S. 342 (Punkt III); KERN, W. (1987), S. 116f.; TEMPELMEIER, H. (1989a), S. 1; HAHN, D. (1990), S. 236; KLEINER, F. (1991), S. 124. Zu den seltenen Ausnahmen, die zugestehen, mit Simulationsmodellen auch optimale Modelllösungen ermitteln zu können, gehört dagegen GRZESIK (1987), S. 15.

15) Ein Untersuchungskonzept wird hingegen als analytisch eingestuft, wenn es folgende Bedingungen erfüllt:

- Ausgangspunkt ist die systemorientierte Konzeptualisierung eines realen oder formalen Problems. Sie wird in ein formalsprachlich expliziertes Modell umgesetzt, das als ein endliches Formelsystem vorliegt.
- Es handelt sich um ein problemspezifizierendes Formelsystem mit einem Ausgangszustand, mindestens einem erwünschten Endzustand und einer nicht-leeren Menge von Operatoren. Die Operatoren lassen sich als Transformationsregeln oder -funktionen auffassen.
- Das problemspezifizierende Formelsystem enthält in seinem Ausgangszustand mindestens eine Formel, in deren Argument mindestens eine Variable vorkommt. Bei der Variablen kann es sich z.B. um eine Entscheidungsvariable aus einem konventionell definierten Entscheidungsmodell handeln, das ein zugrundeliegendes Entscheidungsproblem als ein Formelsystem spezifiziert.
- Jeder erwünschte Endzustand des problemspezifizierenden Formelsystems zeichnet sich dadurch aus, daß in ihm jeder Variable aus dem Ausgangszustand des Formelsystems genau eine Konstante zugeordnet ist. Die zugehörige Variablenbelegung stellt eine der gesuchten Problemlösungen dar.
- Die Operatoren (Transformationsregeln oder -funktionen) legen fest, wie sich ein beliebiger Referenzzustand des Formelsystems in einen Folgezustand transformieren läßt.
- Untersuchungsziel ist es, durch das sukzessive Anwenden der Operatoren (Transformationsregeln oder -funktionen) mindestens einen der erwünschten Endzustände des problemspezifizierenden Formelsystems zu erreichen. Es wird also angestrebt, mindestens eine Belegung aller Variablen aus dem Ausgangszustand des Formelsystems durch Konstanten aufzufinden und dadurch die jeweils zugehörige(n) Problemlösung(en) zu erkennen. Wenn der Nachweis gelingt, daß dies grundsätzlich unmöglich ist, steht die Unlösbarkeit des untersuchten Problems fest.

Der wesentliche Unterschied zwischen analytischen und simulativen Konzepten erstreckt sich auf zwei interdependente Aspekte: Erstens wird von analytischen Konzepten ein Formelsystem mit *variablenbehaftetem* Ausgangszustand vorausgesetzt, von simulativen Konzepten dagegen ein Formelsystem mit *variablenfreiem* Ausgangszustand. Zweitens verfolgen analytische Konzepte das Ziel, Variablenbelegungen durch Konstanten aufzuspüren, während von simulativen Konzepten Modellverhaltensweisen erzeugt und ausgewertet werden. Im Zusammenhang mit diesen beiden grundsätzlichen Differenzen stehen zwei weitere Randaspekte: Erstens beziehen sich simulative Konzepte auf Formelsysteme, die kinetische oder dynamische Modelle darstellen. Analytische Konzepte beruhen dagegen auf problemspezifizierenden Systemen. Zweitens ist die Angabe von mindestens einem erwünschten Endzustand für jedes problemspezifizierende System obligatorisch. Für kinetische oder dynamische Modelle besitzen solche Endzustände jedoch nur fakultativen Charakter, zumeist fehlen sie sogar. Daher werden sie in der Definition von Simulationsanalysen auch nicht als konstitutive Komponenten aufgeführt.

Aufgrund der hier vorgetragenen Festlegungen verhalten sich analytische und simulative Untersuchungskonzepte nicht notwendig kontradiktorisch. Beispielsweise läßt sich die Konsistenz eines variablenfreien Formelsystems untersuchen, das kein kinetisches oder dynamisches Modell definiert. Bei einem solchen Formelsystem kann es sich etwa um eine zusammengesetzte aussagenlogische Formel handeln. Da weder ein kinetisches noch ein dynamisches Modell untersucht wird, scheidet die Simulationsanalyse aus. Da das Formelsystem in seinem Ausgangszustand keine Variablen enthält, kann seine Analyse auch keinen analytischen Charakter besitzen. Allerdings vermutet der

Verf., daß alle methodischen Konkretisierungen der jeweils eingesetzten Untersuchungskonzepte die ursprüngliche Analyseaufgabe durch eine äquivalente Aufgabe ersetzen werden, die dann letztlich doch wieder auf analytische oder simulative Weise gelöst wird. Dies läßt sich anhand der voranstehenden Konsistenzprüfung einer zusammengesetzten aussagenlogischen Formel verdeutlichen. Als Untersuchungsmethoden bieten sich zwei Alternativen an. Einerseits kann die Formel in konjunktiver Normalform notiert und daraus seine Klauselmengemenge abgeleitet werden. Da alle Klauseln aussagenlogische Formeln darstellen, handelt es sich bei der Klauselmengemenge um ein variablenfreies Formelsystem. Die Konsistenz dieser Klauselmengemenge läßt sich mit Hilfe des Resolutionskonzepts entscheiden. Dabei wird in einem abstrakten Inferenzraum nach der leeren Klausel gesucht. Dieser Inferenzraum läßt sich als ein dynamisches Modell vorstellen. Sein Ausgangszustand ist die ursprünglich gegebene Klauselmengemenge. Seine Transformationsregelmengemenge umfaßt nur die eine Resolutionsregel der Resolventenbildung. Es wird in simulativer Weise nach mindestens einer Modellverhaltensweise geforscht, die in der leeren Klauselmengemenge endet. Andererseits läßt sich aber auch für die zusammengesetzte aussagenlogische Formel eine Wahrheitwertetafel bilden. Sie stellt ein metasprachliches Formelsystem dar. Für alle objektsprachlichen atomaren Formeln, aus denen die zu untersuchende objektsprachliche Formel zusammengesetzt ist, enthält die Wahrheitwertetafel als Konstanten deren zwei alternativen Wahrheitswerte. Die Wahrheitswerte aller daraus zusammengesetzten Formeln stellen dagegen zunächst Variablen dar. Der Ausgangszustand des Formelsystems "Wahrheitwertetafel" enthält also Variablen. In dem einen erwünschten Endzustand des Formelsystems der Wahrheitwertetafel besitzen hingegen alle zusammengesetzten objektsprachlichen Formeln genau einen konstanten Wahrheitwert. Falls in diesem Endzustand die ursprüngliche zusammengesetzte objektsprachliche Formel unter allen Wahrheitwertekombinationen ihrer atomaren Formeln den konstanten Wahrheitwert "falsch" besitzt, ist diese Formel inkonsistent. (Andernfalls ist sie konsistent). Daher wurde diesmal auf analytische Weise die Inkonsistenz der zusammengesetzten aussagenlogischen Formel untersucht.

Eine interessante Ausweitung erfährt die voranstehend skizzierte Inkonsistenzuntersuchung einer zusammengesetzten Formel, wenn es sich dabei um eine prädikatenlogische Formel handelt, in deren Argumenten mindestens eine Variable enthalten ist. Dies bedeutet nicht nur, daß im o.a. simulativen Fall das Resolutions- mit dem Unifizierungskonzept kombiniert werden muß, um im Inferenzraum auch die erforderlichen Variablenbelegungen durchführen zu können. Vielmehr erhält das dynamische Modell, das den Inferenzraum darstellt, jetzt auch einen Ausgangszustand, dessen Formelsystem (Klauselmengemenge) mindestens eine Variable umfaßt. Daher kann es sich bei der Suche nach der leeren Klausel nicht mehr um ein simulatives Untersuchungskonzept handeln. Statt dessen erfüllt es alle Bedingungen, die oben für analytische Untersuchungskonzepte aufgestellt wurden.

Dieser Umschlag von der simulativen in die analytische Konzeptklasse mag insofern unbefriedigend wirken, als eine gleichartige Analyseaufgabe - die Inkonsistenzprüfung einer Formel - nicht mit einem grundsätzlich neuartigen, sondern mit einem lediglich - um die Unifizierungskomponente - erweiterten Untersuchungskonzept erfüllt wird. Der Verf. sieht hierin jedoch keine fundamentale Schwäche, sondern nur eine unerwartete, kontraintuitive Folge der oben unterbreiteten Definitionen analytischer und synthetischer Konzepte. Wer diese Folge beseitigen wollte, müßte alternative Konzeptdefinitionen vorlegen, die mindestens drei Anforderungen gerecht werden: Sie sollten ähnlich trennscharf ausfallen, das "Experimentieren" mit "analytischen" Modellen befriedigend einordnen und das Optimieren von Netzmodellen auf das simulative Ausführen von Schaltprozessen in speziellen Netzmodellen (Optimierungsnetzen) zurückführen können. Der Verf. sieht eine solche Alternative nicht, ohne ihre Existenzmöglichkeit in Abrede zu stellen.

16) Hinsichtlich des Realproblembezugs stimmt diese erste Anforderung mit dem o.a. Simulationsverständnis überein.

17) Auch formale Probleme können so modelliert werden, daß sich ihre Lösungen auf simulative Weise bestimmen lassen. Z.B. kann ein formales Problem darin bestehen, für einen abstrakten Automaten - etwa einen TURING-Automaten - aufzuzeigen, daß er von seinem Ausgangszustand aus einen erwünschten Endzustand erreichen kann. Dieser Automat wurde bereits als ein System konzeptualisiert. Er läßt sich durch ein Netzmodell abbilden. Dann gilt es, innerhalb des Netzmodells mindestens einen Schaltprozeß zu identifizieren, der von der Ausgangsmarkierung des Netzes zu derjenigen Endmarkierung führt, die den erwünschten Endzustand des modellierten Automaten abbildet. Das ursprüngliche Formalproblem läßt sich - unter günstigen Umständen - auf simulative Weise dadurch lösen, daß im Netzmodell die Schaltschritte aus einer Schaltfolge ausgeführt werden, die tatsächlich den gesuchten Schaltprozeß hervorbringt. Solche Schaltschrittausführungen, die auf Anwendungen der Schaltregel beruhen, stellen den Kern der Simulation von Netzmodellen dar. Folglich kann ein *formales* Problem durch die Simulation eines problemabbildenden (Netz-)Modells gelöst werden. (Auf die simulative Untersuchung eines anderen formalen Problems - die Inkonsistenz einer zusammengesetzten aussagenlogischen Formel - wird in einer anderen Anmerkung eingegangen.)

Mit dem voranstehenden Beispiel soll verdeutlicht werden, daß die Eigenart des modellierten Problems für die simulative Untersuchung eines Modells überhaupt keine Rolle spielt. Damit von einer Simulation im oben präzisierten Sinne gesprochen werden kann, ist es lediglich erforderlich, daß ein systemtheoretisch konzeptualisiertes und formalsprachlich expliziertes Modell vorliegt. Ob dieses Modell ein Real- oder ein Formalproblem abbildet, ist für die Unterscheidung zwischen simulativer und analytischer Modellauswertung unerheblich.

18) Jedes Modell, das auf simulative Weise untersucht wird, wird fortan auch als ein Simulationsmodell angesprochen. Entsprechend heißen alle Methoden und Instrumente, die zur Untersuchung eines Simulationsmodells eingesetzt werden, auch Simulationsmethoden bzw. -instrumente. Analog dazu wird ein Modell, das auf analytische Art untersucht wird, als ein analytisches Modell bezeichnet. Die dabei angewandten Methoden und Instrumente werden als analytische Auswertungsmethoden bzw. -instrumente thematisiert. (Die Begriffe der Analysemethoden und -instrumente werden dagegen in allgemeiner Weise als Synonyma für Untersuchungsmethoden bzw. -instrumente gebraucht.)

Dabei ist auf die frühere Trennung zwischen Modell- und Methodenbegriff zu achten: Ein simulatives oder analytisches Untersuchungskonzept wird zunächst nur durch entsprechende simulative bzw. analytische Methoden (Instrumente) unterstützt. Ein Modell ist - solange keine Präzisierungen der Modellgestalt vorgenommen werden - dagegen sowohl der simulativen als auch der analytischen Untersuchung zugänglich. Das gleiche originäre Modell kann daher sowohl in simulativer als auch in analytischer Weise untersucht werden, falls es sich an die spezifischen Voraussetzungen dieser beiden Untersuchungskategorien anpassen läßt. Beispielsweise kann ein Simulationsmodell, dessen Ausgangszustand durch ein variablenfreies Formelsystem ausgedrückt wird, in ein analytisches Modell überführt werden, indem im Argument mindestens einer Formel eine Konstante durch eine Variable ersetzt wird. Umgekehrt ist es möglich, ein analytisches Modell dadurch als ein Simulationsmodell zu reformulieren, daß im Formelsystem des Ausgangszustands des ersten Modells alle Variablen durch jeweils eine Konstante belegt werden.

19) Die anschließenden Ausführungen beziehen sich nur noch auf dynamische Systeme, weil Netzmodelle wegen des dynamischen Charakters ihrer Schaltregel zu dieser Systemkategorie zählen.

20) Variablenfreie Formelsysteme lassen sich ebenso als Formelsysteme ansprechen, die ausschließlich aus Grundtermformeln bestehen.

21) Anschließend werden nur noch die Transformationsregeln angesprochen, weil die Zustandstransformation von Netzmodellen durch deren *Schaltregeln* geleistet wird.

22) Es können auch mehrere alternative Modellverhaltensweisen generiert und ausgewertet werden. Dadurch werden polyvalente Simulationsanalysen einbezogen. Gleiches gilt für die mehrfach iterierte monovalente Simulationsanalysen von stochastischen Simulationsmodellen.

23) Jede zusammenhängende Folge von Simulationsschritten, die in ihrer Gesamtheit genau eine Modellverhaltensweise generiert, wird als ein Simulationslauf bezeichnet.

24) Die Einsicht, daß sich ein Problem in der intendierten Weise grundsätzlich nicht lösen läßt, wird im allgemeinen als Erkenntnispotential von Simulationsanalysen gewürdigt. Die späteren Ausführungen zu omnivalenten Simulationsanalysen werden aber zeigen, daß solche Unlösbarkeitserkenntnisse durchaus auf simulative Weise gewonnen werden können.

25) Bei Stelle/Transition-Netzen handelt es sich um die früher erläuterten, alternativen Schaltregelausprägungen.

26) Der Einfachheit halber wird fortan nur noch von "der" Schaltregel gesprochen. Gemeint ist damit jedoch der generische Schaltregelbegriff, der alle vorgenannten Regelindividualisierungen und Schaltregelausprägungen umgreift.

27) Bei jeder Schaltregelanwendung wird ein aktivierter Schaltschritt ausgeführt, dessen Schaltwirkung einen Folgezustand des Netzmodells hervorbringt. Diesbezüglich wird auch von einer Simulation der Schaltwirkungen von Schaltschritten gesprochen.

28) Vgl. FIDELAK (1988b), S. 8; PAGNONI (1990), S. 163; VON KLEIST-RETZOW (1991), S. 256.

Der Begriff des "Spielens" assoziiert den experimentellen Aspekt der eingangs angeführten Annäherung an den Simulationsbegriff.

29) Vgl. dazu auch am Rande des Petrinetz-Konzepts mit verallgemeinernder Einbettung in Erkenntnisse der KI-Forschung FIDELAK (1988b), S. 24.

30) Die letztgenannte einschränkende Bedingung ist erforderlich, weil in Ausnahmefällen verschiedene Schaltschritte, die unter derselben Referenzmarkierung aktiviert sind, zur selben Folgemarkierung führen können. Dies folgt aus der Möglichkeit, daß die Erreichbarkeitsgraphen von Netzen Multigraphen darstellen.

31) Dies läßt sich schon anhand der wenigen Erreichbarkeitsgraphen erkennen, die in dieser Arbeit für Netze vorgelegt werden. In ihnen besitzen nahezu alle Markierungsknoten, die jeweils eine Referenzmarkierung der zugrundeliegenden Netze repräsentieren, mehrere Folgeknoten.

32) Alle Netzmodelle, die keine polykonsekutive Markierung besitzen, werden dagegen als anomale Netzmodelle bezeichnet.

- 33) Die beiden Strategieantipoden fallen allerdings zusammen, falls ein Netzmodell überhaupt keine polykonsekutive Markierung besitzt. Dann können auch nicht die u.a. Mischstrategien gebildet werden. Von diesem Sonderfall wird jedoch nachfolgend angesehen, weil er für Netzmodelle nur äußerst selten eintritt. Er setzt Netzmodelle voraus, deren Erreichbarkeitsgraphen keine Markierungsknoten mit mehreren Folgeknoten besitzen. Ein totes Netz, dessen Ausgangsmarkierung einen Deadlock darstellt, erfüllt z.B. diese Eigenschaft. Dabei handelt es sich jedoch um einen theoretischen Grenzfall, der für die Modellierungspraxis kaum eine Rolle spielt.
- 34) Die monovalente Simulationsanalyse wird in der Simulationsliteratur zumeist als deterministisches Simulationskonzept thematisiert. Der Verf. bevorzugt jedoch die oben eingeführte Bezeichnung, weil sie zu einer schlüssigeren Terminologie führt. Denn es wird später gezeigt, daß die stochastische Simulationsanalyse keineswegs - wie es ihre Bezeichnung erwarten ließe - im Gegensatz zur deterministischen Simulationsanalyse steht. Statt dessen beruht die (konventionelle) stochastische Simulationsanalyse auf mehrfach wiederholten, von Wahrscheinlichkeitsverteilungen oder Dichtefunktionen überlagerten Ausführungen einzelner monovalenter Simulationsanalysen. Vgl. zum deterministischen Simulationskonzept HAUKE (1973), S. 14f.; KNOOP (1986), S. 136; HINTZ (1987), S. 181; MÜLLER, A. (1987), S. 348ff.; GRZESIK (1987), S. 14ff. (dort als Alternativensimulation). Mitunter wird die monovalente oder deterministische Simulation als Inbegriff jeder Simulationsanalyse behandelt; vgl. z.B. FIDELAK (1988b), S. 24: Dort wird von der simulativen Analyse eines Modells nur dann gesprochen, wenn der Simulationsablauf *eindeutig* ist.
- 35) Es wird nicht gefordert, daß es sich bei diesen Modellzuständen um polykonsekutive Markierungen handeln muß. Daher gehört die Simulationsanalyse jedes anomalen Netzmodells, das überhaupt keine polykonsekutiven Markierungen besitzt, notwendig zu den monovalenten Netzsimulationen.
- 36) Diese prinzipielle Unvollständigkeit aller (monovalenten) Simulationsanalysen von (normalen) Netzmodellen bringt FAVREL (1987), S. 282, zum Ausdruck, wenn er feststellt: "The simulation is a 'trial' only, and *it never covers all the possible cases.*" (kursive Hervorhebung durch den Verf.). Vgl. daneben zur Unvollständigkeit simulativer Analyseansätze - außerhalb des Kontextes von Petrinetzen - auch BOUSSIN (1978), S. 1528.
- 37) Dies betrifft alle Modellzustände, die als Markierungen des Netzmodells *nur* dadurch erreicht werden können, daß der vernachlässigte Schaltschritt ausgeführt wird. Das braucht keineswegs für alle Modellzustände zu gelten, die auf die Ausführung des ausgeklammerten Schaltschritts unmittelbar oder mittelbar folgen. Denn diese Modellzustände können auch über andere, in der monovalenten Netzsimulation enthaltene Schaltschritte erreicht werden.
- 38) Dies entspricht der Einstellung der konventionellen Simulationsanalyse, keine Entscheidungsalternativen zu beachten, sondern nur jeweils *eine* vorgegebene "Alternative" zu untersuchen; vgl. TEMPELMEIER, H. (1989a), S. 1.
- 39) Die polyvalente Simulationsanalyse umgreift die stochastische Simulationsanalyse als eine spezielle Ausprägung.
- 40) Für die hier vorausgesetzten Netzmodelle mit *endlichen* Erreichbarkeitsgraphen, ist dies grundsätzlich immer möglich. Dazu reicht es aus, bei der Simulationsanalyse eines Netzmodells dessen Erreichbarkeitsgraphen *vollständig* zu erforschen. Eine derart omnivalente Simulationsanalyse klingt bei RAZOUK (1985c), S. 4f., an: "The most commonly used method of analyzing systems ... is to explore paths through the state-space. This technique is commonly referred to as simulation. One should not lose sight of the fact that path exploration and exhaustive state exploration are strongly related: ... exhaustive state exploration attempts to reconstruct the complete state-space ... by exploring all paths out of every state." Vgl. ebenso den Fall vollständiger Alternativensimulation bei GRZESIK (1987), S. 15.
- 41) Omnivalente Simulationsanalysen werden in Werken, die sich mit der Simulation von Produktionssystemen beschäftigen, im allgemeinen nicht berücksichtigt. Zu den seltenen Ausnahmen, in denen das Konzept der omnivalenten Simulation angewendet wird, zählen die Ausführungen von SPUR (1980), S. 309ff. u. 348, zu einer besonderen System-Einlastungsregel. Sie fordert, bei der Entscheidung, Produktionsaufträge in ein Produktionssystem einzuschleusen, zunächst für jeden Auftrag die Konsequenzen seiner hypothetischen Einlastung in einem Modell des Produktionssystems vollständig zu simulieren. Erst wenn diese umfassenden Konsequenzsimulationen für alle Aufträge vorliegen, deren Einlastung in das Produktionssystem zur Disposition steht, wird anhand der Simulationsergebnisse einer von diesen Aufträgen tatsächlich ausgewählt.
- 42) Die unvollständigen polyvalenten Simulationsanalysen bilden das Spektrum simulativer Untersuchungen eines Netzmodells, das sich zwischen den Antipoden der monovalenten Simulationsanalyse auf der einen Seite und der omnivalenten Simulationsanalyse auf der anderen Seite erstreckt. Auf diese beiden Antipoden bezog sich die polare Fallunterscheidung zu Beginn dieses Kapitels.
- 43) Anomale Netzmodelle, in denen keine polykonsekutive Markierung erreicht werden kann, lassen sich dagegen mit Hilfe der monovalenten Simulationsanalyse in der gleichen Weise untersuchen, wie es unten für die omnivalente Simulationsanalyse beschrieben wird: Die monovalente Simulationsanalyse allein liefert zwar im allgemeinen noch keine Erkenntnisse über dynamische Eigenschaften des betrachteten anomalen Netzmodells. Aber sie bringt alle

Informationen über erreichbare Markierungen und zulässige Schrittausführungen hervor, die auch im Erreichbarkeitsgraphen des anomalen Netzmodells enthalten wären. Daher lassen sich auf die Ergebnisse monovalenter Simulationsanalysen alle Untersuchungskonzepte anwenden, die bereits im Rahmen der Erreichbarkeitsanalyse von Netzen vorgestellt wurden. Dies gilt allerdings nur unter der o.a. Voraussetzung anomaler Netzmodelle ohne polykonsekutive Markierungen.

Allerdings existiert auch ein Grenzfall anomaler Netzmodelle, in dem bereits die monovalente Simulationsanalyse die Erkenntnis einer dynamischen Netzeigenschaft vermittelt. Es handelt sich um die extreme Eigenschaft, daß ein totes Netz vorliegt. Denn in diesem Fall endet die monovalente Simulation des anomalen Netzmodells bereits in dessen Ausgangszustand, der überhaupt keinen zulässigen Folgezustand besitzt.

44) Dies folgt unmittelbar aus den Definitionen monovalenter Simulationsanalysen und normaler Netzmodelle.

45) Ein solcher Sonderfall läge z.B. vor, wenn die polykonsekutiven Markierungen eines finalen Netzmodells zu keinem seiner finalen Schaltprozesse gehörten. Ein solches Netzmodell könnte nur genau einen finalen Schaltprozeß besitzen. Dessen optimaler Schaltwert ließe sich als Netzoptimum auch durch eine deterministische Simulationsanalyse ermitteln.

46) Auf diese potentielle Inkorrektheit aller (monovalenten) Simulationsanalysen von (normalen) Netzmodellen scheint FAVREL (1987), S. 282, anzuspielen, wenn er anmerkt: "... the properties deduced by ... simulation *are not 100% sure*. The simulation is a 'trial' only ..." (kursive Hervorhebung durch den Verf.). Daran ändert sich auch nichts, wenn zu polyvalenten Simulationsanalysen übergegangen wird, solange diese unvollständig bleiben. So ordnet KOCHAN, D. (1986), S. 134, allen wiederholten Simulationsläufen von stochastischen Simulationen, die unvollständige polyvalente Simulationsanalysen besonderer Art darstellen, zu Recht einen unvermeidbaren "experimental error" zu.

47) Diese Schlußfolgerung entspricht dem "negation by failure"-Prinzip der logischen Programmierung. Eine solche Schlußfolgerung scheint z.B. in VON KLEIST-RETZOW (1991), S. 257, gezogen worden zu sein: Dort wird behauptet, mit Hilfe einer (vermutlich polyvalenten) Netzsimulation die Deadlockfreiheit (Verklemmungsfreiheit) eines Netzmodells nachgewiesen zu haben. Die Ausführung einer poly- anstelle einer monovalenten Simulationsanalyse ändert nichts daran, daß ebenso das Folgerungsprinzip des "negation by failure" präsupponiert werden muß. Vgl. dazu die Erläuterung hinsichtlich der Ähnlichkeit von poly- und monovalenten Simulationsanalysen.

48) Es ist tatsächlich inkorrekt (korrekt), wenn das Netzmodell tatsächlich mindestens einen (keinen) Deadlock besitzt. Dies konnte aber während der monovalenten Simulationsanalyse aufgrund ihrer immanenten Unvollständigkeit nicht sicher erkannt werden.

49) Protokolle legen syntaktische, semantische und pragmatische Vereinbarungen fest, die den Austausch von Nachrichten zwischen Kommunikationspartnern koordinieren. Diese Protokolle spielen vor allem im Bereich der Telekommunikation eine herausragende Rolle. Vgl. zur Funktion und Ausgestaltung solcher Kommunikationsprotokolle HERRTWICH (1989b), S. 387ff.; BEKHI (1989), S. 244f.; HOLZMANN (1991), insbesondere S. 12f. u. 19ff.

50) Netze für die Modellierung von Telekommunikationsprotokollen verwenden z.B. RAZOUK (1985c), S. 21 i.V.m. S. 5f.; BEKHI (1989), S. 246ff.; ANTON (1989), S. 144ff.; HOLZMANN (1991), S. 163 u. 183ff. (allerdings auf S. 185f. mit erheblichen Vorbehalten). Vgl. auch VON KLEIST-RETZOW (1991), S. 257, zur simulativ "nachgewiesenen" Deadlockfreiheit eines Netzmodells. Allerdings handelt es sich dort nicht um Telekommunikationsprotokolle, sondern um die Modellierung von Vorgangsbearbeitungen in Büros.

51) Die Voraussetzung, daß es sich bei den untersuchten Netzmodellen um normale Netzmodelle mit mindestens einer polykonsekutiven Markierung handeln muß, ist - soweit es der Verf. zu überblicken vermag - immer erfüllt.

52) Vgl. zur speziellen Anwendung von Simulationsmodellen, "What if"-Fragen zu beantworten, MEYER ZU SELHAUSEN (1980b), S. 135; BACK-HOCK (1991a), S. 98.

53) Insbesondere kann die Zielwirksamkeit von Prioritätsregeln analysiert werden. Vgl. zu der großen Bedeutung, die der Simulation bei der Prozeßkoordinierung mit Prioritätsregeln zugemessen wird, SEELBACH (1979), Sp. 26; MÜLLER, H. (1987), S. 1665ff., insbesondere S. 1667; MÜLLER, A. (1987), S. 275.

54) Die Regelmenge kann auch aus nur genau einer Entscheidungsregel bestehen. Daher gelten die anschließenden Ausführungen, die sich auf den allgemeinen Fall einer nicht-leeren Regelmenge beziehen, ebenso für den speziellen Fall einer einzelnen Entscheidungsregel.

55) Falls die zu untersuchende Regelmenge nicht ausreicht, um eine solche Konfliktstrategie eindeutig festzulegen, wird sie durch entsprechende Ergänzungsregeln vervollständigt.

56) Dieser Ansatz findet sich bereits bei HOLT, A. (1968), S. 283f. (wiederholte Netzsimulation mit alternativen Strategien für Konfliktauflösungen).

57) Die omnivalente Simulationsanalyse besitzt die Charakteristik eines nondeterministischen Algorithmus, der sich durch einen nondeterministischen TURING-Automaten ausführen läßt. Die gleiche Eigenschaft wurde bereits für die Konstruktion von Erreichbarkeitsgraphen aufgezeigt. Aus dieser Perspektive erweisen sich omnivalente Simulationsanalysen und Konstruktionen von Erreichbarkeitsgraphen im nondeterministischen Sinne als "TURING-äquivalent".

Aus der materiellen Unterschiedslosigkeit von omnivalenter Simulationsanalyse und Konstruktion eines Erreichbarkeitsgraphen läßt sich auch ein terminologischer Umkehrschluß ziehen: Die Konstruktion des Erreichbarkeitsgraphen eines beliebigen Netzmodells kann ebenso als omnivalente Simulation desselben Netzmodells thematisiert werden. Erreichbarkeits- und Simulationsanalyse eines Netzmodells besitzen daher einen gemeinsamen Überlappungsbereich. Sie fallen allerdings auch nicht zusammen. Denn die Überlappung bezieht sich zunächst nur auf die Konstruktion von Erreichbarkeitsgraphen - nicht aber deren Auswertung - und nur auf die omnivalente Simulationsanalyse - nicht aber auf die unvollständige polyvalente Simulationsanalyse oder auf die monovalente Simulationsanalyse. Später wird allerdings der Überlappungsbereich dadurch ausgedehnt, daß die omnivalente Simulationsanalyse um die Auswertungskonzepte für Erreichbarkeitsgraphen erweitert wird.

58) Falls nur ein Netzoptimum interessiert, reicht sogar eine monovalente Simulation des Optimierungsnetzes aus, die so lange fortgesetzt wird, bis die Markierung der Endstelle erreicht wird. Dann beruht die Ermittlung des Optimierungsnetzes aber immer noch auf der omnivalenten Simulation des zugrundeliegenden Netzmodells. Wenn alle Netzoptima ermittelt werden sollen, ist auch eine omnivalente Simulationsanalyse des Optimierungsnetzes erforderlich, um alle seine alternativen optimalen Schaltprozesse bestimmen zu können.

59) Es wird hier der Einfachheit halber nur von "einer" petrinetzgestützten Simulationssprache geredet. Strenggenommen liegt jedoch eine variantenreiche Sprachfamilie vor. Jedes ihrer Mitglieder zeichnet sich durch die jeweils zugrundeliegende Netzklasse und diejenige Schaltregelvariante aus, die für die Simulation der klassenzugehörigen Netze vorgesehen ist. Auf diese präzisierende Betrachtungsweise wird fortan verzichtet, weil die Nuancen unterschiedlicher Sprachvarianten, die für die Simulation von Netzmodellen ersonnen werden können, nicht näher interessieren.

60) Es wird in diesem Kapitel vorausgesetzt, daß alle Modelle, die simulativ ausgewertet werden sollen, auf Automatischen Informationsverarbeitungssystemen implementiert vorliegen. Dadurch werden Gedankenexperimente, in denen sich mentale Modelle "simulieren" lassen, aus dem Simulationsbegriff ausgegrenzt. Die Implementierung eines Simulationsmodells auf einem Automatischen Informationsverarbeitungssystem geschieht jeweils durch die Konstrukte einer formalen Sprache. Aufgrund ihres speziellen Anwendungskontexts wird diese Sprache als Simulationssprache bezeichnet.

Simulationssprachen stellen allerdings keine notwendige Bedingung, sondern nur eine komfortable Unterstützung des Umgangs mit Simulationsmodellen dar. Grundsätzlich können automatengestützte Simulationsmodelle mit Hilfe jeder Programmiersprache implementiert werden. Vgl. zu solchen Implementierungen ohne Rückgriff auf spezielle Simulationssprachen z.B. MÜLLER, A. (1987), S. 351f.

61) Im Rahmen des Petrinetz-Konzepts wird nicht "das" Petrinetz schlechthin, sondern eine Vielzahl unterschiedlicher Netzklassen definiert. Dadurch werden strenggenommen entsprechend viele verschiedene petrinetzgestützte Simulationssprachen festgelegt.

62) Vgl. dazu auch die konzeptionell verwandte Definition einer topologischen Netzsprache im Zusammenhang mit dem algebraischen Signaturkonzept.

63) Vgl. in bezug auf Petrinetze die Differenzierung zwischen aktiven Transitionen und passiven Stellen; vgl. im Hinblick auf Simulationskonzepte KNOOP (1986), S. 136f. u. 144; TEMPELMEIER, H. (1989a), S. 31 (Unterscheidung von Komponenten, die entweder Leistungen abzugeben vermögen oder nicht).

64) Vgl. in bezug auf Petrinetze die Ausführungen zu Marken; vgl. im Hinblick auf Simulationskonzepte KNOOP (1986), S. 136 u. 144; TEMPELMEIER, H. (1989a), S. 30f. u. 56ff.).

65) Vgl. im Hinblick auf Simulationskonzepte KNOOP (1986), S. 136; KOCHAN, D. (1986), S. 133 ("snapshots").

66) Vgl. in bezug auf Petrinetze die Deutung des Schaltens von Transitionen als Eintreten von Ereignissen; vgl. im Hinblick auf Simulationskonzepte SPUR (1980), S. 361ff., 370f., 380ff. u. 396ff.; SELIGER (1983), S. 52ff.; HARTLEY (1984), S. 256; KNOOP (1986), S. 136; MÜLLER, A. (1987), S. 358. Vgl. allerdings auch die Erläuterung, daß Simulationssprachen nicht an zustandstransformierenden Ereignisgeschehnissen anknüpfen, falls sie sich auf kontinuierliche Simulationsmodelle beziehen.

67) Vgl. in bezug auf Petrinetze die Deutung des Schaltens von Transitionen als Eintreten von Ereignissen; vgl. im Hinblick auf Simulationskonzepte SELIGER (1983), S. 52 u. 54.; HARTLEY (1984), S. 256. Vgl. allerdings wiederum den Hinweis, daß Simulationssprachen für kontinuierliche Simulationsmodelle auf einer ebenso kontinuierlichen Zeitskala beruhen.

68) Vgl. in bezug auf Petrinetze die lokal definierten Schaltregeln für einzelne Transitionen. Vgl. im Hinblick auf Simulationskonzepte die Zustandstransformation durch lokale Entscheidungsregeln bei KNOOP (1986), S. 137. Beachtenswert ist auch der explizite Hinweis bei KOCHAN, D. (1986), S. 133, daß für die Beschreibung der zustands-transformierenden Regeln von Simulationsmodellen Petrinetze zunehmend breite Beachtung fänden.

69) Die Objektorientierung wohnt zwar dem Petrinetz-Konzept nicht notwendig inne, wurde aber dieser Arbeit als Konkretisierung des systemtheoretischen Strukturierungsansatzes zugrundegelegt.

70) TURING-Automaten und ihre herausragende Bedeutung für die Gestaltung oder Beurteilung von Konzepten der Automatischen Informationsverarbeitung wurden bereits behandelt.

71) Ein Simulationsmodell heißt diskret, wenn die dynamische Modellstruktur durch Ereignisse konstituiert wird. Dabei wird der atomare, zeitlose Ereignisbegriff vorausgesetzt, der an früherer Stelle eingeführt wurde. Jedes Geschehnis eines solchen Ereignisses bewirkt den Übergang von einem Referenzzustand des Simulationsmodells zu seinem Folgezustand. Zwischen zwei Ereignisgeschehnissen, die unmittelbar aufeinander folgen, liegt derselbe Zustand des Simulationsmodells vor. Die Zeitdauer, während derer ein Modellzustand unverändert vorliegt, wird grundsätzlich nicht festgelegt, sondern nur durch die Geschehnisse des zustandseinleitenden nach vorne und des zustandsbeendenden Ereignisses nach hinten begrenzt. Da die zustandstransformierenden Ereignisse zeitlos geschehen, erfolgt ein abrupter, diskontinuierlicher Übergang zwischen zwei Modellzuständen, die unmittelbar aufeinander folgen. Die Simulationszeit eines diskreten Simulationsmodells wird jeweils durch den Eintritt eines solchen abrupten Zustandsübergangs fortgeschaltet. Sie wird daher auf einer diskreten Zeitskala gemessen. (Darauf läßt sich die oben benutzte Bezeichnung "diskretes" Simulationsmodell zurückführen).

Das Konzept diskreter Simulationsmodelle überdeckt die Varianten der ereignis-, der prozeß- und der aktivitätsorientierten Simulationsmodelle, die des öfteren in der einschlägigen Simulationsliteratur unterschieden werden. Vgl. zu diskreten, ereignis-, prozeß- oder aktivitätsorientierten Simulationskonzepten KOCHAN, D. (1986), S. 153f.; HILTY (1987), S. 218; TEMPELMEIER, H. (1989a), S. 6ff. u. 13.

72) Ein Simulationsmodell wird als kontinuierlich bezeichnet, wenn die dynamische Modellstruktur durch Funktionen konstituiert wird, deren Funktionswerte in Abhängigkeit von der Simulationszeit stetig variieren. Dabei wird die Simulationszeit selbst als eine Variable mit einem kontinuierlichen - in der Regel nicht-negativen reellzahligen - Definitionsbereich vorausgesetzt. Die Simulationszeit von kontinuierlichen Simulationsmodellen wird daher auf einer kontinuierlichen Zeitskala gemessen. Die derart festgelegten zeitkontinuierlichen Funktionen ersetzen die zeitlosen, zustandstransformierenden Ereignisse von diskreten Simulationsmodellen. Statt dessen werden in kontinuierlichen Simulationsmodellen die Modellzustände entsprechend zum Fluß der Simulationszeit kontinuierlich variiert. Vgl. zu Konzepten für kontinuierliche Simulationsmodelle HILTY (1987), S. 218; TEMPELMEIER, H. (1989a), S. 6 u. 13.

73) Vgl. TEMPELMEIER, H. (1989a), S. 13.

74) Dieser Prozeßbegriff wurde bereits bei der Entfaltung des systemtheoretischen Bezugsrahmens vorgestellt. Er wurde später durch das Konzept der Schaltprozesse für Stelle/Transition-Netze bzw. Synthetische Netze konkretisiert.

75) Vgl. dazu die Anmerkung zur Fixierung der Zeiteinheit für Zeitnetze.

76) Eine Erweiterung von mono- zu unvollständig polyvalenten Simulationsanalysen wird anschließend anhand der stochastischen Simulationsanalyse behandelt. Dort wird jedoch gezeigt, daß die hier vorgetragene Kritik gegenüber der monovalenten Simulationsanalyse konventioneller Simulationssprachen im wesentlichen nicht berührt wird.

77) Das gilt auch für alle Analysevarianten, die zwischen diesen beiden Antipoden liegen. Sie interessieren hier jedoch nicht weiter.

78) Allerdings wird dieses *prinzipielle* Leistungspotential der omnivalenten Simulationsanalyse von Netzmodellen bei praktischen simulativen Untersuchungen von Netzmodellen in der Regel nicht ausgeschöpft. Es wurde bereits darauf hingewiesen, daß in der einschlägigen Netzliteratur die monovalenten Simulationsanalysen dominieren.

79) Vgl. zum stochastischen Simulationskonzept HERTZ (1964), S. 99ff.; GUPTA, S. (1968), S. B-18f.; HAUK (1973), S. 14f.; MOORE, L. (1976), S. 37ff.; PRITSKER (1977), S. 66ff.; MÜLLER-MERBACH (1978), S. 423f.; MOELLER (1981), S. 682; KNOOP (1986), S. 135f.; KOCHAN, D. (1986), S. 134; PEARL (1987), S. 245ff. (Allerdings präsentiert die Autorin eine Konzeptvariante aus dem Bereich der KI-Forschung, die von konventionellen stochastischen Simulationen erheblich abweicht. Auf die Spezifika dieser Variante wird hier nicht weiter Bezug genommen.); LÖWE (1987), S. 16ff.; MÜLLER, A. (1987), S. 348f.; GRZESIK (1987), S. 16f. (als Risikosimulation); TEMPELMEIER, H. (1989a), S. 167ff.; CHIN, H. (1989), S. 131ff.

80) Vgl. GRZESIK (1987), S. 16.

81) Vgl. zur parametrischen Formulierung von Simulationsmodellen TEMPELMEIER, H. (1989a), S. 1f., 49ff. (Allerdings dominiert auf S. 49ff. der Aspekt parametrisch festgelegter Wahrscheinlichkeitsverteilungen, der für die Festlegung unterschiedlicher Ausgangszustände eines Simulationsmodells nur eine Möglichkeit unter vielen anderen darstellt.) u. 173.

82) Simulationsanalysen wurden oben auf kinetische oder dynamische Modelle bezogen, deren Ausgangszustände jeweils durch *variablenfreie* Formeln spezifiziert sind. Solche Modelle stellen gewöhnliche Simulationsmodelle dar. Die nachfolgend vorgestellten parametrischen Simulationsmodelle widersprechen jedoch *prima facie* der Definition simulativer Auswertungskonzepte. Denn die Ausgangszustände der parametrischen Simulationsmodelle sind durch Formeln mit *mindestens einer Variablen* festgelegt. Auf den Variablencharakter der Modellparameter wurde bereits hingewiesen. Dieser scheinbare Widerspruch läßt sich jedoch auflösen: Kein parametrisches Simulationsmodell wird unmittelbar in seinem variablenbehafteten Ausgangszustand untersucht. Statt dessen wird in jedem Simulationslauf genau eine parametrische Modellvariante ausgewertet, deren Ausgangszustand durch eine variantenspezifische Parameterkonfiguration vollständig fixiert ist. Es werden lediglich die monovalenten Simulationsanalysen dieser Modellvarianten, die jeweils variablenfreie Ausgangszustände besitzen, für unterschiedliche Parameterkonfigurationen wiederholt (vgl. z.B. GRZESIK (1987), S. 16). Daher liegt weiterhin ein simulatives Untersuchungskonzept im eingangs definierten Sinne vor.

83) Der hier verwendete Parameterbegriff entspricht zunächst nicht demjenigen, der in einer früheren Anmerkung verwendet wurde. Dort wurde von einem Parameter nur in Hinblick auf *quantitativ* bestimmte Attributausprägungen gesprochen. Dies braucht hier jedoch nicht der Fall sein, weil z.B. auch qualitative Attributausprägungen - etwa die Ausprägungen einer "Markenfarbe" - für die Modellparametrisierung zugelassen werden. Statt dessen handelt es sich hier bei einem "Parameter" - aus der Perspektive von Prädikatenlogik und Signaturkonzept - um nichts anderes als um eine Variable. Dennoch besteht ein innerer Zusammenhang zwischen beiden Parameterauffassungen. Denn auch hinter den hier betrachteten Parametern steht ein quantitatives Konzept: Ihnen wird anschließend eine quantitativ bestimmte Wahrscheinlichkeits- oder Dichtefunktion überlagert.

84) Falls ein Netzmodell vorliegt, kann es sich z.B. um die Anzahl von Markenkopien handeln, die sich unter der Ausgangsmarkierung des Netzmodells auf einer seiner Stellen befindet. Ebenso kommt ein Markenattribut in Betracht. Für die Kopie der betroffenen Attributmarke, die sich unter der Ausgangsmarkierung des Netzmodells auf einer seiner Stellen befindet, lassen sich unterschiedliche Ausprägungen festlegen.

85) Vgl. zum Variablencharakter der Parameter aus stochastischen Simulationsmodellen TEMPELMEIER, H. (1989a), S. 1 u. 4.

86) Ein stochastisches Simulationsmodell ist also ein Unterfall der parametrischen Simulationsmodelle. Es gibt auch parametrische Simulationsmodelle ohne stochastischen Charakter. Ein Beispiel dafür wurde schon in einer früheren Anmerkung erwähnt: Es handelt sich um die Ermittlung friktionsfreier Lagerkapazitäten. Dabei wird das Simulationsmodell eines Produktionsmodells mit unterschiedlichen Ausprägungen des Parameters "Lagerkapazität" ausgewertet. Es wurde in der o.a. Anmerkung darauf hingewiesen, daß dieser spezielle Anwendungsfall von parametrischen Simulationsmodellen im Bereich des Petrinetz-Konzepts keine Rolle spielt. Dennoch lassen sich auch für Petrinetze Modellierungsaufgaben vorstellen, die mit der Hilfe von parametrischen Simulationsmodellen erfüllt werden können, ohne dabei auf stochastische Modellierungen zurückzugreifen. Dazu gehört z.B. die Aufgabe festzustellen, wie viele Transportroboter oder Werkstückträger in einem modellierten Produktionssystem bereitgehalten werden müssen, um ein vorgegebenes Auftragspaket so abwickeln zu können, daß den Zielvorstellungen des Entscheidungsträgers möglichst gut entsprochen wird. Diese Modellierungsaufgabe läßt sich grundsätzlich dadurch erfüllen, daß ein Produktionsmodell mit unterschiedlichen Ausstattungen an Transportrobotern oder Werkstückträgern simuliert wird; vgl. KNOOP (1986), S. 132 u. 166. Die Anzahlen der verfügbaren Transportroboter oder Werkstückträger stellen dann die Parameter des Simulationsmodells dar. Wenn das Simulationsmodell als ein Netzmodell ausgestaltet wird, lassen sich diese Transportroboter- oder Werkstückträgeranzahlen durch entsprechende Anzahlen von Kopien einer Transportmittel- bzw. einer Werkstückmarke erfassen, die im Netzmodell unter dessen Ausgangsmarkierung vorhanden sind. Vgl. dazu die Verwendung von Transportmittel- und Werkstückmarken in der später präsentierten Fallstudie. Die Auswertung des Netzmodells geschieht dann, indem seine Ausgangsmarkierung hinsichtlich der Transportmittel- oder Werkstückmarkenkopien "parametrisch variiert" wird. Für jede der alternativen Ausgangsmarkierungen findet eine eigenständige Simulationsanalyse des Netzmodells statt. Durch Vergleich der Simulationsergebnisse läßt sich feststellen, welche der untersuchten Anfangsausstattungen mit Transportrobotern oder Werkstückträgern am günstigsten ist. Auf solche parametrischen, aber dennoch nicht stochastischen Simulationsmodelle wird hier nicht weiter eingegangen. Sie sind zwar durchaus von praktischem Interesse, gehören aber nicht zur stochastischen Simulationsanalyse.

87) Die Parameterkonfiguration stellt aus prädikatenlogischer Sicht eine Variablenbelegung dar.

88) Die Anzahl der Simulationsläufe, mit denen jeweils eine Parameterkonfiguration ausgewertet wird, kann allerdings sehr groß ausfallen. Beispielsweise nennen MOORE, L. (1976), S. 63, und TAYLOR, B. (1980a), S. 54, jeweils

einen Wert von 1.000 Simulationsläufen für die Auswertung von nur einem einzelnen stochastischen (GERT-)Simulationsmodell.

89) Plastische Beispiele solcher Häufigkeitsverteilungen finden sich z.B. bei MOELLER (1981), S. 686 (Fig. 1) u. 690ff. (Fig. 3ff.).

90) Vgl. PRITSKER (1977), S. 66 u. 68; PEARL (1987), S. 246f.; HAAS (1987), S. 30 (in einer statistisch anspruchsvolleren Artikulation mit Konfidenzintervallen); TEMPELMEIER, H. (1989a), S. 167; CHIN, H. (1989), S. 131.

Dabei wird vorausgesetzt, daß die Simulationsläufe des Netzmodells mit stochastisch variierten Parameterkonfigurationen mindestens so lange wiederholt werden, bis sich die Häufigkeitsverteilung der Simulationsergebnisse stabilisiert.

91) Für einen *vorgegebenen* Ausgangszustand des stochastischen Simulationsmodells, der durch die jeweils generierte Parameterkonfiguration festgelegt ist, wird nur eine monovalente Simulationsanalyse vorgenommen. Daher ist es im Rahmen der konventionellen stochastischen Simulation unmöglich, Entscheidungsalternativen zu berücksichtigen, die sich *nicht* auf die Auswahl unter verschiedenen Ausgangszuständen erstrecken. Der Simulationslauf für eine fixierte parametrische Modellvariante ist nicht in der Lage, alternative Simulationsfortsetzungen einzubeziehen. Hierin liegt eine wesentliche Einschränkung gegenüber der Simulationsmöglichkeit von Netzmodellen. Denn in ihrer polyvalenten Simulationsanalyse bereitet es keine Schwierigkeiten, alternative Simulationsfortsetzungen zu erfassen. Dazu brauchen lediglich alternative Schaltwege im Erreichbarkeitsgraphen des untersuchten Netzes verfolgt zu werden, die von einem gemeinsamen Markierungsknoten ausgehen. Denn die vollständige polyvalente Simulationsanalyse eines Netzmodells fällt mit der Konstruktion seines Erreichbarkeitsgraphen zusammen.

92) Vgl. GUPTA, S. (1968), S. B-19 ("Repetent simulation"); GRZESIK (1987), S. 15.

93) Die Polyvalenz der stochastischen Simulation resultiert aus ihrem Ansatz, für *mehrere* Parameterkonfigurationen des gleichen stochastischen Simulationsmodells jeweils einen (monovalenten) Simulationslauf auszuführen. Die zugehörigen alternativen Ausgangszustände des stochastischen Simulationsmodells können als gemeinsame "Folgezustände" eines vorgelagerten, fiktiven "Prä-Ausgangszustands" interpretiert werden. Dieser Prä-Ausgangszustand verhält sich per constructionem polykonsekutiv. Da während der stochastischen Simulation alle alternativen Folgezustände dieses einen fiktiven Prä-Ausgangszustands untersucht werden, liegt aus dieser abstrakten, alle Parameterkonfigurationen einbeziehenden Perspektive eine polyvalente Simulationsanalyse des stochastischen Simulationsmodells vor.

94) In Ausnahmefällen kann der Konfigurationsraum durch eine stochastische Simulationsanalyse vollständig erforscht werden. Dies ist genau dann der Fall, wenn zwei Voraussetzungen erfüllt sind. Erstens muß für *jeden* Parameter ein *diskreter* Definitionsbereich mit endlich vielen zulässigen Ausprägungen vorgesehen sein. Dann ist der Konfigurationsraum aller kombinatorisch möglichen Parameterkonfigurationen endlich. Diese Prämisse erscheint im Rahmen dieser Arbeit nicht als problematisch, weil analoge Vereinbarungen für die Endlichkeit der Definitionsbereiche aller Markenattribute und für die Endlichkeit (Beschränktheit) der Anzahlen aller Markenkopien auf den Stellen eines Netzes getroffen wurden. Zweitens muß aber auch gefordert werden, daß die endlich vielen Parameterkonfigurationen, die während einer stochastischen Simulationsanalyse generiert werden, den endlichen Konfigurationsraum tatsächlich *vollständig* ausschöpfen. Diese Prämisse wird von konventionellen Simulationsanalysen jedoch im allgemeinen wegen des damit verbundenen hohen Simulationsaufwands nicht erfüllt.

95) Die zweite Unvollständigkeitsursache - die unvollständige Ausschöpfung des Konfigurationsraums - wird hier nicht als wesentlich betrachtet, weil sie keinen grundsätzlichen Unterschied zwischen konventionellen und petrinetzgestützten Simulationssprachen etabliert. Denn Netzmodelle lassen nur dann eine vollständige polyvalente Simulationsanalyse zu, wenn sie endliche Erreichbarkeitsgraphen besitzen. Dazu müssen Endlichkeitsprämissen erfüllt sein, die bereits in einer früheren Anmerkung angesprochen wurden. Wenn solche Finitheitsvoraussetzungen konzipiert werden, dann lassen sich aber auch analoge Ausnahmefälle konstruieren, in denen stochastische Simulationsanalysen den Konfigurationsraum kombinatorisch möglicher Parameterkonfigurationen vollständig auszuschöpfen vermögen. Darauf wurde ebenfalls schon in der vorgenannten Anmerkung hingewiesen. Dort wurde zwar das pragmatische Argument vorgetragen, auf die prinzipiell vorstellbare vollständige Erforschung des endlichen Konfigurationsraums werde dennoch verzichtet, um den Ressourceneinsatz für die stochastische Simulationsanalyse niedrig zu halten. Doch läßt sich dieses Argument ebenso gegen die vollständige polyvalente Simulationsanalyse eines Netzmodells richten. Denn auch sein endlicher Erreichbarkeitsgraph nimmt im allgemeinen so großen Umfang an, daß seine vollständige Erforschung oftmals wegen praktischer Ressourcenbeschränkungen nicht in Betracht kommt. Daher läßt sich die zweite Unvollständigkeitsursache nicht nur gegen stochastische Simulationsanalysen auf der Basis konventioneller Simulationssprachen, sondern ebenso gegen petrinetzgestützte Simulationssprachen richten.

96) Bezüglich der zweiten Unvollständigkeitsursache stochastischer Simulationsanalysen, welche die Ausschöpfung des Konfigurationsraums kombinatorisch möglicher Parameterkonfigurationen betraf, wird auf frühere Erläuterungen verwiesen.

97) Diese Ansicht vertritt pointiert FAVREL (1987), S. 282, hinsichtlich der Gestaltung von Simulationsmodellen mit der Hilfe von Petrinetzen: "You can deduce properties of the system described by Petri nets without simulation of it. In a language simulation, you can say nothing about the system without simulation."

98) Beispielsweise lassen sich Modelle, die auf dem Fundament der GERT-Netzplantechnik gestaltet wurden, sowohl mit Hilfe der Simulationssprache QGERT implementieren und auswerten als auch durch die MRP-Technik analysieren. Diese Kombinationsmöglichkeit einer Simulationssprache mit nur einer alternativen Analysetechnik fällt jedoch weitaus bescheidener aus als das o.a. Potential, eine petrinetzgestützte Simulationssprache mit vier unterschiedlichen Analysetechniken zusammenzuführen. Noch wichtiger ist jedoch die außerordentlich rigide Voraussetzung, die GERT-Modelle erfüllen müssen, damit die MRP-Technik überhaupt auf sie angewandt werden kann. Es muß sich um GERT-Modelle vom STEOR-Typ handeln, die ausschließlich stochastische Knoten mit "exklusivem oder"-Ausgang besitzen. Vgl. NEUMANN, K. (1990), S. 56ff., insbesondere S. 80ff. u. 90ff. Das Ausdrucksvermögen solcher GERT-Modelle vom STEOR-Typ ist gegenüber Netzmodellen drastisch eingeschränkt. Darauf wird später näher eingegangen, wenn Petrinetz-Konzept und Netzplantechnik miteinander verglichen werden. Vgl. zur Anwendung der Simulationssprache QGERT auf Maschinenbelegungsprobleme bei Werkstattfertigung z.B. TAYLOR, B. (1982), S. 846ff.

99) Vgl. MOALLA (1976b), S. 117 (ansatzweise); GOTTSCHALK, W. (1977), S. 877; AGERWALA (1978b), S. 309 (als Anschaulichkeit der Darstellung von Kontrollflüssen); HACKMANN (1981), S. 372; ROSENSTENGEL (1991), S. 5f.; PAGNONI (1990), S. 163.

100) CAA steht für: Computer Aided Animation. Vgl. zum Leistungspotential von automatengestützten Animationsstechniken WECK (1991e), S. 117ff.

101) Die Animation von Markenflüssen wird von mehreren Softwarepaketen geleistet, die für die Simulationsanalyse von Netzen angeboten werden. Vgl. z.B. RAZOUK (1985c), S. 20f. (dort noch als Vorhaben angekündigt).

102) PPS-Systeme werden hier als Automatische Informationsverarbeitungssysteme thematisiert, die der Planung und Steuerung von Produktionssystemen dienen. Sofern diese Informationsverarbeitungssysteme im Dialogbetrieb genutzt werden, wird kurz von interaktiven PPS-Systemen gesprochen.

Vgl. zum Dialogbetrieb oder zur interaktiven Nutzung von PPS-Systemen HORMANN, D. (1973), S. 80ff., insbesondere S. 82 (bezüglich der dort erörterten "flexiblen Simulation mit Dialogmöglichkeit"); BECKER, J. (1978), S. 515f. u. 517f.; WICHARZ (1983), S. 366ff. (allerdings ohne explizite Nennung der Dialogverarbeitung); SELIGER (1983), S. 26, 62ff. u. 139; KAZMAIER (1984), S. 165 u. 168; GÜNTHER, H. (1986), S. 242; ALDINGER (1985a), S. 38f., 43, 45, 48f., 123, 128f. u. 134; HELBERG (1987), S. 86; KREIMEIER (1988), S. 394f.; SCHEER (1990c), S. 80, 196 u. 205; WINTER, R. O. (1991), S. 6, 13, 117 u. 176.

Vgl. auch mit speziellem Bezug auf Flexible Fertigungssysteme FRAUNHOFER-INSTITUT FÜR SYSTEMTECHNIK (1982), S. 47 u. 52; HWANG, S. (1984), S. 841f. u. 845ff.; HARTLEY (1984), S. 258; MERTINS (1985a), S. 160ff.; MERTINS (1985b), S. 255ff.; KOCHAN, D. (1986), S. 76, 139 u. - in die Zukunft blickend - S. 146; WILDEMANN (1987a), S. 355, 357, 361 u. 363; HINTZ (1987), S. 4f.

103) Vgl. allgemein zur Dialogverarbeitung als interaktiver Betriebsweise von Automatischen Informationsverarbeitungssystemen HEIBEY (1977), S. 92; SCHEER (1990c), S. 54, 60ff., 71ff. u. 81ff.

104) Der Graphikbildschirm muß an ein Automatisches Informationsverarbeitungssystem mit einer beträchtlichen Informationsverarbeitungskapazität angeschlossen sein. Dies ist erforderlich, um die visualisierten Netzrepräsentationen am Bildschirm so schnell aufbauen und an Veränderungen der Markenverteilung im zugrundeliegenden Netzmodell so rasch anpassen zu können, daß der Benutzer von den Verzögerungszeiten nicht irritiert wird. Im allgemeinen werden Zeitspannen für Bildaufbau und -veränderung im Bereich von einigen wenigen Zehntelsekunden als akzeptabel empfunden. Darauf wird später noch näher eingegangen. Um derart geringe Verzögerungszeiten zu erzielen, müssen Automatische Informationsverarbeitungssysteme aus der Klasse der "Workstations" eingesetzt werden, um netzbasierte Graphikschnittstellen in benutzerfreundlicher Weise ausgestalten zu können. Dies verursacht erhebliche Investitionskosten. Darauf wird später anlässlich der Beurteilung des Petrinetz-Konzepts zurückgekommen.

105) Auf das Automatische Informationsverarbeitungssystem, an das der Graphikbildschirm angeschlossen sein muß, wurde bereits in der voranstehenden Anmerkung hingewiesen.

106) Vgl. zur graphischen Ausgestaltung der Benutzerschnittstellen von PPS-Systemen (insbesondere Leitstandsystemen) SELIGER (1983), S. 63 u. 66; ALDINGER (1985a), S. 37, Fn. 1, u. S. 90f.; MAIER-ROTHE (1985), S. 137; KOCHAN, D. (1986), S. 76 u. 146; NIEDERHAUSEN (1988), S. 402f.; KREIMEIER (1988), S. 394f.; FLEIG (1991), S. 98 i. V.m. Bild 1 auf S. 97;

Vgl. auch zu ähnlichen Bestrebungen, die Benutzerschnittstellen von komplexen Maschinen(system)steuerungen mit weitreichenden Graphikfähigkeiten zu versehen, FORRER (1990), S. CA 111ff.; BRÜGGEMANN (1991), S. 229.

107) Vgl. PERL (1980), S. 11.1.

108) Anstatt die Auswirkungen einzelner Koordinierungsentscheidungen zu simulieren, lassen sich ebenso die Zielwirksamkeiten alternativer Koordinierungsstrategien simulativ untersuchen. Dazu gehört vor allem die Analyse, wie sich (Kombinationen von) Prioritätsregeln in verschiedenen Produktionssituationen auf unterschiedliche Formalziele auszuwirken vermögen. Dieser Analyseansatz wurde bereits erörtert. Weiterführende simulative Untersuchungen von alternativen Koordinierungsstrategien finden sich bei MULLER, H. (1987), S. 1660 u. 1665ff.; vgl. daneben auch KREIMEIER (1988), S. 394f. (S. 394: "Auswahl einer geeigneten Planungsstrategie"; S. 395: "verschiedene Abarbeitungsvorgehensweisen"); NIEDERHAUSEN (1988), S. 403 ("Auswahl der Planungsstrategien").

109) Simulationsstudien werden häufig genutzt, um in Produktionssystemmodellen zu ermitteln, welche Konsequenzen Entscheidungsalternativen bei der Prozeßkoordinierung nach sich ziehen. Vgl. dazu SPUR (1980), S. 309ff., insbesondere S. 348ff., 362ff. u. 383ff.; EVERSHEIM (1981), S. 172f.; HWANG, S. (1984), S. 848; MERTINS (1985a), S. 101, der diesbezüglich von einer experimentellen Optimierung der Maschinenbelegung spricht; KNOOP (1986), S. 165ff. i.V.m. S. 65; WILDEMANN (1988c), S. 63; NIEDERHAUSEN (1988), S. 400ff., insbesondere S. 402f.; KERN, W. (1990a), S. 330; SCHEER (1990c), S. 205 ("Der Disponent kann ... Alternativen in ihren Wirkungen so lange simulieren, bis ein befriedigendes Ergebnis erzielt wird."); WINTER, RO. (1991), S. 211. Vgl. ebenso - jedoch ohne Bezug auf Prozeßkoordinierungen in Produktionssystemen - GOMEZ, P. (1978), S. 141. Vgl. des weiteren die Ausführungen zur vorausschauenden Simulation der hypothetischen Konsequenzen von lokalen Entscheidungsalternativen.

110) Die Qualität menschlicher Koordinierungsleistungen hängt bei interaktiver Informationsverarbeitung u.a. davon ab, wie an der Benutzerschnittstelle die zugrundeliegende Informationsbasis repräsentiert und in welcher Weise der Zugriff auf sie gestaltet wird; vgl. FRAUNHOFER-INSTITUT FÜR SYSTEMTECHNIK (1982), S. 155. Mit der Eignung unterschiedlicher Repräsentationsformen von Informationen für die Werkstattsteuerung befaßt sich z.B. MERTINS (1985a), S. 113f.

Als eine herausragende Determinante der menschlichen Koordinierungsqualität gilt die Transparenz von Informationsrepräsentation und -zugriff. STUTE (1978a), S. 64, stellt z.B. fest: "Dem Bedienungspersonal müssen die jeweils wichtigen Informationen ... unmittelbar, auf das Notwendige beschränkt und leicht verständlich dargeboten werden, um ihm die ... Beurteilung des Prozeßgeschehens zu erleichtern und so die Voraussetzung für Eingriffe zu schaffen." Ähnliche Transparenzforderungen vertreten TWELLMANN (1979), S. 66; MAIER, U. (1980), S. 61; MERTINS (1985a), S. 8 u. 101; ALDINGER (1985a), S. 21 u. 28; WECK (1991e), S. 116 u. - insbesondere - S. 128; vgl. auch HELBERG (1987), S. 169f., der in indirekter Weise die mangelhafte Transparenz konventioneller PPS-Systeme beklagt. Ausführliche Diskussionen der Möglichkeiten und Schwierigkeiten, transparente Benutzerschnittstellen für die interaktive Informationsverarbeitung zu gestalten, finden sich bei TWELLMANN (1979), S. 66ff.; MERTINS (1985a), S. 115ff. u. 118ff., der sich speziell mit der ergonomischen Ausgestaltung von Steuerungssoftware und Dispositionsarbeitsplätzen für die interaktive Werkstattsteuerung befaßt; MULLER, H. (1987), S. 1665ff.

Allgemein herrscht die Ansicht vor, die Transparenz von Informationen werde durch ihre graphische Visualisierung besonders gefördert. Vgl. dazu die Quellen, die bereits die kognitive Adäquanz von graphischen Informationsdarstellungen belegten. Vgl. darüber hinaus zur Hervorhebung der Benutzerfreundlichkeit, Anschaulichkeit oder Transparenz von Informationsverarbeitungssystemen mit graphischen Benutzerschnittstellen oder -oberflächen DIRUF (1983), S. 239f., 242 u. 244; DIRUF (1984), S. 126; BEHRENDT (1988), S. 32 ("Entscheidungen verständlicher machen heißt aber, Entscheidungen sichtbar machen. Diese Visualisierung der Entscheidungsfindung ... wird ... mit Hilfe einer grafikgesteuerten, benutzerfreundlichen Systemoberfläche erreicht."); KRALLMANN (1989b), S. 3; SCHEER (1990c), S. 201 u. 253; GLOVER (1990), S. 24; FORRER (1990), S. CA 111; BACK-HOCK (1991a), S. 98; WECK (1991d), S. 6f.; WECK (1991e), S. 116ff. (mit speziellem Bezug auf die Steuerung flexibler Fertigungssysteme); WINTER, RO. (1991), S. 158; FLEIG (1991), S. 98.

Einen krassen Gegensatz zu dieser graphischen Informationsrepräsentation bildet die Darstellung von Informationen durch Listen. Sie sind im Rahmen der konventionellen Informationsverarbeitung noch oftmals üblich. Dies gilt insbesondere auch für die Präsentation von Produktionsplänen an den Benutzerschnittstellen konventioneller PPS-Systeme. Vgl. zu solchen Listendarstellungen STUTE (1978a), S. 67 u. 78 (Maschinenbelegungspläne); SCHICKHOFF (1991), S. 99, Bild 1 (Maschinenbelegungsplan) und Bild 2 (Arbeitsplatzstatus).

Die Informationsrepräsentation durch Listen zeichnet sich durch ein hohes Ausmaß an Intransparenz aus. Sie stellen "Datenfriedhöfe" dar, die zwar von Automatischen Informationsverarbeitungssystemen leicht behandelt werden können, aber nicht den menschlichen Rezeptions- und Kognitionsstrukturen gerecht werden. Vgl. dazu auch MOSNER (1991), S. 138 u. 140. Dort werden die Unübersichtlichkeit und Undurchschaubarkeit konventioneller Simulationssysteme kritisiert, die ihre Ergebnispräsentationen auf Listendarstellungen beschränken.

Eine Automatische Informationsverarbeitung, die dem Ziel der Benutzerfreundlichkeit gerecht werden soll, muß sich daher von der vorherrschenden Informationsrepräsentation in Listenform befreien. Einen Weg, dieses Ziel zu erreichen, stellt die oben angeführte graphische Darstellungsweise dar.

111) Dazu gehören z.B. die Animationskomponenten CINEMA und SIMANIM für die Simulationssprache SIMAN; vgl. SYSTEMS MODELING CORPORATION (o.J.b), S. 1ff.; SYSTEMS MODELING CORPORATION (o.J.a), S. 3 (bezüglich des dort angesprochenen "part flow"; allerdings wird auf S. 11 die Visualisierung von Objektflüssen nicht mehr angesprochen); TEMPELMEIER, H. (1989a), S. 12, 19, 21 u. 194ff.

Vgl. darüber hinaus zur graphischen Animation von Simulationsprozessen und -ergebnissen MULLER,H. (1987), S. 1660, 1665 u. 1667; KOCH,M. (1990), S. 549ff.; DÖRNHÖFER (1991), S. 129ff.; MOSNER (1991), S. 141ff.; BACK-HOCK (1991a), S. 98.

5.2.4 Beiträge der Netzanalyse zur Prädikatenlogik

Deklarative Netzmodelle können benutzt werden, um die logischen Konsequenzen zu erforschen, die sich aus den zugrundeliegenden deklarativen Objektmodellen grundsätzlich gewinnen lassen¹⁾. Aus dieser Perspektive ermöglicht die Netzanalyse eine Exploration des Inferenzpotentials prädikatenlogischer Objektmodellierungen. Hierdurch wird die Chance eröffnet, netzorientierte Denkmuster in logisch ausgerichtete Kognitionsraster einfließen zu lassen. Obgleich dies ein theoretisch interessantes Unterfangen zu sein scheint²⁾, beschränken sich die anschließenden Ausführungen auf eine Skizze der wesentlichen Aspekte³⁾. Denn in dieser Arbeit steht das Inferenzpotential deklarativer Netzmodelle nicht im Vordergrund⁴⁾. Dennoch wird darauf kurz eingegangen, um die spätere Beurteilung des Petrinetz-Konzepts auf eine breitere Basis zu stellen. Dabei steht das Kriterium der konzeptionellen Fruchtbarkeit im Vordergrund.

Ein Objektmodell, das ein prädikatenlogisches Formelsystem in konjunktiver Normalform darstellt, wird durch das daraus abgeleitete deklarative Netzmodell logisch äquivalent repräsentiert⁵⁾. Damit ist gemeint, daß jede Formel "p", die eine logische Konsequenz der Formelmenge FM des Objektmodells ist, auch im Netzmodell abgeleitet werden kann und umgekehrt⁶⁾. Unter Rückgriff auf die frühere ausführliche Behandlung der operationalen Semantik der Prädikatenlogik läßt sich diese Ableitbarkeit einer Formel "p" aus der Formelmenge FM als Beweisbarkeit (Allgemeingültigkeit) des Subjugats "FM → p" reformulieren. Ein solches Subjugat wird in einem deklarativen Netzmodell mittels der bereits erläuterten indirekten Technik des Refutationsprinzips bewiesen:

- Es wird das Negat $\neg p$ der interessierenden Formel "p" gebildet.
- Das Negat $\neg p$ wird in der üblichen Weise als Teilnetz repräsentiert.
- Es wird das erweiterte deklarative Netzmodell betrachtet, das die Konjugatformel "FM ∧ ($\neg p$)" repräsentiert. Die Modellerweiterung geschieht dadurch, daß Stellen, die mit denselben Prädikatssymbolen beschriftet sind, miteinander identifiziert werden.
- Es wird versucht, im erweiterten Netzmodell die Inkonsistenz der Konjugatformel "FM ∧ ($\neg p$)" nachzuweisen.
- Falls dieser Versuch erfolgreich endet, ist die ursprüngliche Beweisbarkeitsbehauptung für die Subjugatformel "FM → p" indirekt bestätigt⁷⁾. Dann ist die fragliche Ableitbarkeit der Formel "p" positiv entschieden.
- Wenn es dagegen unmöglich ist, im erweiterten Netzmodell jemals die Inkonsistenz der Konjugatformel "FM ∧ ($\neg p$)" abzuleiten, so ist die Beweisbarkeitsbehauptung der ursprünglichen Subjugatformel "FM → p" widerlegt. Die Ableitbarkeit der Formel "p" ist negativ entschieden.
- Es kann auch der Fall eintreten, daß sich innerhalb endlicher Ressourcenschranken weder die Inkonsistenz der Konjugatformel definitiv aufzeigen noch die prinzipielle Unmöglichkeit ihrer Inkonsistenzableitung erkennen läßt. Dann ist die Ableitbarkeit der Formel "p" weder positiv noch negativ entschieden. Dies entspricht der früher diskutierten prädikatenlogischen Semi-Entscheidbarkeit.

Für den zentralen Inkonsistenznachweis wurden im Petrinetz-Konzept bislang zwei Nachweistekniken entwickelt. Beide knüpfen an der markenfreien Ausgangsmarkierung des erweiterten deklarativen Netzmodell für die Konjugatformel "FM ∧ ($\neg p$)" an. Sie unterscheiden sich aber durch die jeweils angewandten Analyseinstrumente.

Von LAUTENBACH stammt das invariantenorientierte Netztheorem⁸⁾: Ihm zufolge ist ein Netzmodell - unter Vernachlässigung beweistechnischer Feinheiten⁹⁾ - genau dann inkonsistent, wenn es mindestens eine T-Invariante besitzt, die sich als ein Schaltvektor interpretieren läßt, welcher die markenfreie Ausgangsmarkierung des Netzmodells in nicht-trivialer Weise¹⁰⁾ reproduziert. Die T-Invariante \underline{it} ist für ein Netzmodell mit insgesamt N Transitionen t_n als ein N -stelliger Spaltenvektor definiert, dessen Produkt mit der Inzidenzmatrix \underline{C} den N -stelligen Nullspaltenvektor $\underline{0}$ ergibt: $\underline{C} \cdot \underline{it} = \underline{0}$.

MURATA und ZHANG haben dagegen ein erreichbarkeitsbezogenes Netztheorem¹¹⁾ bewiesen: Falls in einem Netzmodell eine Schaltfolge konstruiert werden kann, die unter der markenfreien Ausgangsmarkierung aktiviert ist und durch das Schalten einer charakteristischen Zieltransition endet¹²⁾, dann ist das Netzmodell inkonsistent¹³⁾. Dabei ist die Zieltransition genau diejenige Transition, die im erweiterten Netzmodell zum Teilnetz für das Negat " $\neg p$ " aus der Konjugatformel " $FM \wedge (\neg p)$ " gehört. Es läßt sich zeigen, daß das Ausführen dieser Schaltfolge die markenfreie Ausgangsmarkierung des Netzmodells zu reproduzieren vermag¹⁴⁾.

Das Netztheorem von MURATA und ZHANG besitzt dieselbe Qualität wie LAUTENBACH's Netztheorem: Die Ableitbarkeit einer Formel " p " wird dadurch aufgezeigt, daß für das erweiterte Netzmodell eine nicht-triviale Schaltfolge konstruiert wird, welche die markenfreie Ausgangsmarkierung (Nullmarkierung) reproduziert. Daher lassen sich beide Netztheoreme als eng verwandte Ansätze für die Analyse einer dynamischen Netzeigenschaft begreifen:

- Es wird ein Netzmodell für die Repräsentation einer Ableitbarkeitsbehauptung konstruiert.
- Das Netzmodell erhält als besondere Ausgangsmarkierung M_0 die Nullmarkierung, unter der jeder Stelle die leere Marken(multi)menge zugeordnet wird: $M_0(s_m) = \emptyset$ für alle $s_m \in S$.
- Die Nullmarkierung wird zugleich als einziges Element aus der Menge intendierter Endemarkierungen ausgezeichnet: $ME = \{M_E\}$ und $M_E = M_0$.
- Es wird untersucht, ob sich das Netzmodell final verhält, wenn die Nullschaltfolge außer Acht bleibt.
- Im positiven Fall existiert mindestens eine nicht-triviale Schaltfolge, mit der sich die Nullmarkierung reproduzieren läßt. Dann ist die Ableitbarkeitsbehauptung bestätigt.
- Andernfalls existiert keine solche Schaltfolge, und die Ableitbarkeitsbehauptung ist widerlegt.

Aufgrund der immanenten Schwierigkeiten der Invariantenanalyse, die bereits erörtert wurden, besitzt das Netztheorem von MURATA und ZHANG den Vorzug, ohne Netzinvarianten auszukommen. Statt dessen setzt es lediglich das Instrument der Erreichbarkeitsgraphen voraus, mit deren Hilfe sich entscheiden läßt, ob die voranstehend definierte Schaltfolge für die Reproduktion der markenfreien Ausgangsmarkierung konstruiert werden kann. Denn diese Schaltfolge muß, falls sie existiert, im Erreichbarkeitsgraph zu einem zyklischen Schaltweg gehören, für den gilt: Er beginnt und endet im Ursprung des Erreichbarkeitsgraphen, der die markenfreie Ausgangsmarkierung des Netzmodells wiedergibt. Die letzte Kante des Schaltwegs, die unmittelbar vor der Vollendung des Schaltzyklus durchlaufen wird, ist mit dem Schaltakt der charakteristischen Zieltransition beschriftet.

Mit der Hilfe der beiden Netztheoreme bietet sich das Petrinetz-Konzept an, um die Analyse prädikatenlogischer Objektmodelle zu bereichern. Es läßt sich untersuchen, ob beliebige Formeln " p " logische Konsequenzen derjenigen Formelmenge FM sind, die das Formelsystem eines Objektmodells konstituieren¹⁵⁾. Dabei wird mit dem Beitrag des Petrinetz-Konzepts keineswegs der Anspruch erhoben, Ableitbarkeitsbehauptungen bestätigen oder widerlegen zu können, die sich nicht ebenso mit konventionellen prädikatenlogischen Instrumenten überprüfen ließen. Dies kann gar nicht der Fall sein, da für die Prädikatenlogik bereits mehrere vollständige und korrekte, zudem sogar automatisch ausführbare Inferenzkonzepte zur Verfügung stehen. Insofern

stellt die logische Analyse von Netzmodellen mittels der beiden Netztheoreme keine Erweiterung der Prädikatenlogik dar. Dennoch kann sich das Petrinetz-Konzept in drei Richtungen für die Analyse prädikatenlogischer Objektmodelle als fruchtbar erweisen¹⁶⁾: Es bietet Ansatzpunkte, um Transparenz, Effizienz und Anwendungsbreite prädikatenlogischer Modellanalysen zu vergrößern.

Prädikatenlogische Inferenzen stellen oftmals recht intransparente Formeltransformationen dar. Dies kann einerseits am beträchtlichen Volumen solcher Beweisführungen liegen. Es resultiert vor allem aus dem großen kombinatorischen Suchraum, der durch die Gesamtheit aller möglichen Variablenunifizierungen eröffnet wird. Andererseits kann zur Unübersichtlichkeit von Inferenzprozessen ebenso das Refutationsprinzip beitragen. Ihm zufolge wird ein behaupteter Sachverhalt nicht direkt bewiesen, sondern durch Ableiten einer Inkonsistenz indirekt erschlossen. Die Tendenz zur Intransparenz bleibt so lange unerheblich, wie nur das Ergebnis von prädikatenlogischen Inferenzprozeduren interessiert. Dies muß aber keineswegs der Fall sein. Statt dessen kann ein Modellbenutzer auch wünschen, das Zustandekommen von Inferenzen in prädikatenlogisch fundierten Modellen kritisch nachzuvollziehen. In diesem Zusammenhang wird vor allem an die Erklärungskomponente von Expertensystemen erinnert. Sie werden oftmals als einer der herausragenden Beiträge angeführt, welche die Erforschung Künstlicher Intelligenz zur Gestaltung benutzerfreundlicher Informationsverarbeitungssysteme leisten könnte¹⁷⁾.

Falls die Erklärung einer prädikatenlogischen Inferenz erforderlich ist, erlangt die Transparenz des Inferenzprozesses erhebliches Gewicht. In dieser Hinsicht bietet das Netztheorem von MURATA und ZHANG zweifache Unterstützungsleistung an. Erstens erlaubt es, den prädikatenlogischen Inferenzprozeß, der zur Erkenntnis einer logischen Konsequenz eines prädikatenlogischen Objektmodells geführt hat, als Schaltfolge im zugehörigen Netzmodell zu identifizieren¹⁸⁾. Dabei kann die inferenzabbildende Schaltfolge sowohl im visualisierten Erreichbarkeitsgraphen des Netzmodells als auch in der graphischen Repräsentation des Netzmodells nachvollzogen werden. Die besondere Anschaulichkeit solcher visualisierter Darstellungsformen wurde bereits mehrfach herausgestellt. Falls das Netzmodell auf einem Automatischen Informationsverarbeitungssystem mit einer leistungsfähigen graphischen Benutzeroberfläche implementiert ist, kann die inferenzabbildende Schaltfolge sogar mit subtilen Techniken der "Computer-Animation" plastisch simuliert werden.

Zweitens wirkt das Netztheorem von MURATA und ZHANG auch dadurch transparenzfördernd, daß sich die Schaltfolge im Netzmodell und die korrespondierende Inferenz im prädikatenlogischen Objektmodell zueinander invers verhalten¹⁹⁾: Der Inferenzprozeß, der beim Beweis einer logischen Konsequenz "p" das kombinierte Unifizierungs- und Resolutionskonzept realisiert, startet mit der negierten Formel " $\neg p$ "²⁰⁾ und endet erfolgreich, wenn die Leerklausele " \emptyset " abgeleitet wird. Für einen gewöhnlichen Modellbenutzer wird es aber nur schwer nachzuvollziehen sein, warum gerade darin ein erfolgreicher Inferenzabschluß bestehen soll. Die inferenzabbildende Schaltfolge verläuft jedoch aus logischer Perspektive umgekehrt zur Abfolge der einzelnen Inferenzschritte. Sie beginnt mit der markenfreien Ausgangsmarkierung des Netzmodells²¹⁾ und schließt mit dem Schalten derjenigen Transition ab, die zum Teilnetz für die negierte Formel " $\neg p$ " gehört. Dabei zieht der Schalttakt dieser letzten Transition von deren Eingangsstelle genau jene Marke ab, die das interessierende Formelvorkommen "p" repräsentiert²²⁾. Dieser Markenzugriff am Ende der Schaltfolge kann benutzt werden, um den erfolgreichen Nachweis der logischen Konsequenz "p" anschaulich darzustellen.

Die transparenzfördernde Abbildung von Inferenzprozessen durch Schaltfolgen in Netzmodellen läßt sich auf das Konsistenzmonitoring prädikatenlogischer Objektmodelle übertragen. Seine Bedeutung für die Integritätsüberwachung komplexer Objektmodellierungen wurde an früherer Stelle dargelegt. Auch hier kann die Benutzerfreundlichkeit der Modelle erheblich gesteigert werden, wenn sich Integritätsverletzungen nicht nur mittels prädikatenlogischer Inferenzen aufdecken, sondern ebenso hinsichtlich ihres Zustandekommens veranschaulichen lassen. Zu

diesem Zweck kann wiederum auf die Schaltfolgen zurückgegriffen werden, deren Ausführungen in einem Netzmodell Integritätsverletzungen hervorbringen.

Prädikatenlogische Inferenzprozesse leiden aber nicht nur an ihrer zuvor thematisierten Intransparenz. Hinzu kommen auch erhebliche Effizienzprobleme. Darauf wurde schon früher hingewiesen. Hier interessiert nur ein spezieller Aspekt, der sich auf die konventionelle Implementierung von Inferenzprozessen durch sequentielle Kontrollstrukturen der Programmiersprache PROLOG erstreckt. Die Inferenzgeschwindigkeit kann dabei sensitiv auf die Reihenfolge reagieren, in der Klauseln und Literale bei den Unifizierungs- und Resolutionsoperationen berücksichtigt werden. Diese Inferenzreihenfolge ist in den konventionellen PROLOG-Kontrollstrukturen unabhängig von der jeweils bearbeiteten Inferenzaufgabe fest implementiert. Es besteht jedoch die Hoffnung, die Effizienz von prädikatenlogischen Inferenzprozessen dadurch zu erhöhen, daß die Kontrollstrukturen mit Wissen über diejenigen logischen Abhängigkeiten angereichert werden, die zwischen den Klauseln und Literalen des prädikatenlogischen Formelsystems eines deklarativen Objektmodells bestehen²³). Falls dies gelingt, könnte die Inferenzreihenfolge an die logische Struktur der jeweils vorliegenden Inferenzaufgabe angepaßt werden.

Netzmodelle, die aus deklarativen Objektmodellen abgeleitet sind, enthalten implizit Teile des Wissens über die logische Modellstruktur. Denn die Netztopologie umfaßt in der Flußrelation Informationen darüber, wie die klauselrepräsentierenden Transitionen und die literalrepräsentierenden Stellen miteinander zusammenhängen. Daher läßt sich grundsätzlich vorstellen, dieses netztopologische Wissen über prädikatenlogische Formelzusammenhänge zu nutzen, um adaptive Kontrollstrukturen für prädikatenlogische Inferenzprozesse zu gestalten²⁴). Dies eröffnet den Ausblick, mit der Hilfe von Netzmodellen die Effizienz von Inferenzkonzepten in wissensbasierter Weise zu erhöhen.

Wenn sich dieser Gedankenansatz konkret umsetzen läßt, bilden Netzmodelle eine interessante konzeptionelle Schnittstelle zwischen der konventionellen Prädikatenlogik einerseits und der KI-Forschung andererseits. Sie würden durch ihre Repräsentation von Wissen über den logischen Formelzusammenhang in einem prädikatenlogischen Objektmodell den Anspruch der Künstlichen Intelligenz unterstützen, konventionelle Informationsverarbeitungsprozesse durch Auswertung von verarbeitungsrelevantem Wissen fortzuentwickeln. Insofern verheißen Netzmodelle befruchtende Impulse nicht nur für die konventionelle Prädikatenlogik, sondern auch für die Erforschung Künstlicher Intelligenz²⁵).

Darüber hinaus könnte noch untersucht werden, ob sich die Möglichkeit nebenläufigen Schaltens von Transitionen in Netzmodellen ausnutzen läßt, um die Effizienz prädikatenlogischer Inferenzprozesse zu erhöhen. Denn das Netztheorem von MURATA und ZHANG muß nicht darauf beschränkt bleiben, in einem Netzmodell Schaltfolgen zu konstruieren, die mit sequentiellen Inferenzprozessen im zugrundeliegenden Objektmodell korrespondieren. Vielmehr könnte es auf nebenläufige Schaltfolgen ausgedehnt werden. Dann wäre der Frage nachzugehen, ob sich daraus fruchtbare Erkenntnisse für die Gestaltung paralleler Inferenzprozesse im Objektmodell ableiten lassen²⁶). Ebenso böte es sich an zu erforschen, ob LAUTENBACH's Netztheorem Ansatzpunkte liefert, um prädikatenlogische Inferenzprozesse durch numerische Techniken der Invariantenanalyse zu unterstützen²⁷). Der Verf. verzichtet darauf, diese Ausblicke auf potentielle Beiträge des Petrinetz-Konzepts zur Effizienzsteigerung prädikatenlogischer Inferenzen zu vertiefen. Sie liegen außerhalb des Erkenntnisinteresses dieser Arbeit.

Schließlich empfehlen sich Netzmodelle, um die Anwendungsbreite prädikatenlogischer Modellanalysen auszudehnen. Inferenzprozeduren, mit denen sich die logischen Konsequenzen von prädikatenlogischen Objektmodellen untersuchen lassen, bereiten tendenziell sehr großen Aufwand. Darüber hinaus werden sie schnell undurchsichtig, wenn versucht wird, sie durch rein menschliche Denkarbeit auszuführen. Daher bedeutete das Angebot prädikatenlogischer Programmiersprachen²⁸), solche Inferenzprozeduren automatisch ausführen zu lassen, eine bedeutende Ausweitung der praktischen Anwendungsmöglichkeit prädikatenlogischer Modellierungs-

konzepte. Doch unterliegt die Option, das Inferenzpotential prädikatenlogischer Objektmodelle automatengestützt zu erforschen, in der Regel einer wesentlichen Einschränkung: Konventionelle prädikatenlogische Programmiersprachen ermöglichen nur, deklarative Objektmodelle zu implementieren, die das Ausdrucksvermögen von HORN-Klauseln nicht übersteigen²⁹⁾.

Die automatengestützte Analyse implementierter Netzmodelle gestattet es nun, die Einschränkung auf HORN-Klauseln zu überwinden. Dabei wird auf die Schnittstellen-Funktion zurückgegriffen, die Netzmodelle im Modellierungsprozeß zwischen Objektmodellierung und Modellimplementierung einnehmen. Ausgangspunkt ist eine deklarative Objektmodellierung, die das volle prädikatenlogische Ausdruckspotential ausschöpft. Neben den sonst üblichen HORN-Klauseln dürfen ebenso allgemeine Klauseln verwendet werden, die mehrere positive Literale umfassen. Die resultierenden Objektmodelle werden auf äquivalente Netzmodelle abgebildet und schließlich ohne Informationsverlust oder -verzerrung in einer prädikatenlogischen Programmiersprache implementiert. Es wurde schon früher herausgestellt, daß im Verlauf dieser zweistufigen Transformation Objektmodelle mit allgemeinen Klauseln in implementierte Netzmodelle mit HORN-Klauseln überführt werden. Die letztgenannten lassen sich ohne Schwierigkeiten automatengestützt analysieren.

Daher können nunmehr auch die logischen Konsequenzen von solchen Objektmodellen erforscht werden, welche die HORN-Klausel-Restriktion der prädikatenlogischen Programmierung verletzen, ohne auf die effiziente Unterstützung der Automatischen Informationsverarbeitung verzichten zu müssen. Dadurch wird das Analysepotential konventioneller prädikatenlogischer Objektmodellierungen beträchtlich bereichert. Auf diese Weise gelingt es, den automatengestützten Anwendungsbereich von prädikatenlogischen Modellierungs- und Analyseansätzen auszuweiten. Zugleich wird die praktische Anwendung prädikatenlogischer Konzepte stimuliert.

Am Rande wird darauf hingewiesen, daß sich Synthetische Netze auch als didaktisches Instrument anbieten. Mit ihrer Hilfe lassen sich komplexe prädikatenlogische Sachverhalte veranschaulichen. Als Beispiel dient eine prädikatenlogische Antinomie. Sie resultiert aus einer reflexiven Definition, bei der das Definiens das Definiendum umschließt³⁰⁾. Eine solche Antinomie läßt sich in einem Synthetischen Netz mit einfachsten Mitteln ausdrücken. Es braucht lediglich das Aktivierungsprädikat einer Transition auf reflexive Weise definiert zu werden. Zu diesem Zweck wird das Netz aus der nachfolgenden Abb. 104 eingeführt³¹⁾. In ihm ist eine Zählermarke definiert. Sie besitzt genau ein Attribut der Sorte "aktivierungsanzahl". Das Netz umfaßt genau eine Transition t_1 mit genau einer Eingangsstelle s_1 : $T = \{t_1\}$ und $S = \{s_1\}$. Für die Transition wird als Schaltvoraussetzung vereinbart: Die Transition ist genau dann aktiviert, wenn auf ihrer Eingangsstelle eine Kopie der Zählermarke liegt, deren Attributausprägung die Anzahl "aller" aktivierten Transitionen angibt. Folglich gilt für das Aktivierungsprädikat der Transition t_1 ³²⁾:

$$\text{AKT}(t_1, M_r)$$

$$:\Leftrightarrow \text{Aktivierungsanzahl} = \#\{t_n : t_n \in T \wedge \text{AKT}(t_n, M_r)\}$$

Betrachtet wird die Ausgangsmarkierung M_0 , unter der sich auf der Stelle s_1 eine Markenkopie befindet. Die Ausprägung ihres Attributs "aktivierungsanzahl" beträgt "Null". Fraglich ist, ob die Transition unter der Ausgangsmarkierung M_0 aktiviert ist. Um diese Frage zu beantworten, wird folgende Fallunterscheidung vorgenommen:

- Angenommen, Transition t_1 ist aktiviert: Dann beträgt die Anzahl aller aktivierten Transitionen "Eins". Dem entspricht aber nicht die Attributausprägung "Null" der Zählermarkenkopie. Folglich ist die Aktivierungsbedingung der Transition t_1 nicht erfüllt. Also kann die Transition t_1 nicht aktiviert sein. Dies widerspricht der voranstehenden Annahme. Also ist der erste Fall infolge Inkonsistenz ausgeschlossen.

- Angenommen, Transition t_1 ist nicht aktiviert: Dann beträgt die Anzahl aller aktivierten Transitionen "Null". Damit stimmt die Attributausprägung "Null" der Zählermarkenkopie überein. Daher ist die Aktivierungsbedingung der Transition t_1 erfüllt. Also ist die Transition t_1 aktiviert. Dies läuft der voranstehenden Annahme zuwider. Also ist auch der zweite Fall infolge Selbstwidersprüchlichkeit unzulässig.

Eine Transition kann nur entweder aktiviert oder aber nicht aktiviert sein: tertium non datur. Beide Fälle erweisen sich aber als inkonsistent. Diese Antinomie kann im Rahmen einer "einfachen" prädikatenlogischen Formalisierung nicht aufgelöst werden³³⁾. Dennoch handelt es sich um keinen Mangel von Prädikatenlogik oder Synthetischen Netzen. Vielmehr legt die Ausdrucksmächtigkeit der Prädikatenlogik erst das Antinomiepotential von reflexiven Definitionen offen³⁴⁾.

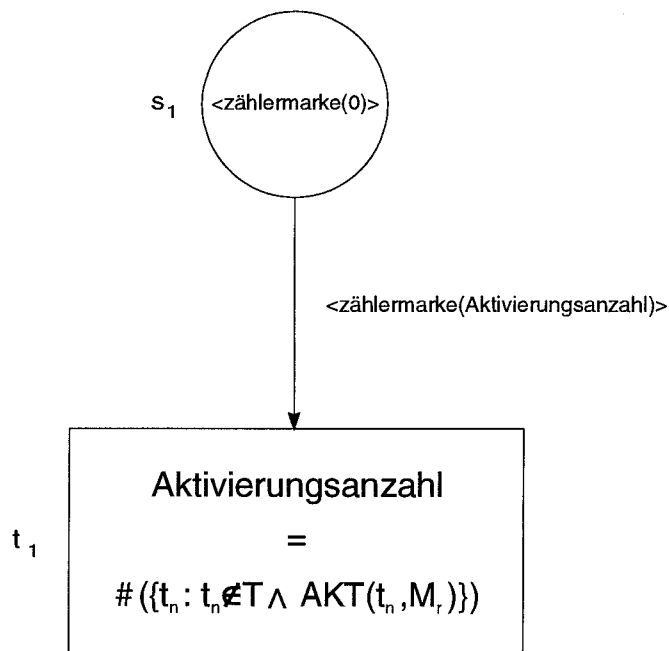


Abb. 104: Netz zur Veranschaulichung einer prädikatenlogischen Antinomie

Anmerkungen zum Kapitel:

1) Dies trifft auf operationale Netzmodelle nicht zu. Denn für die zugrundeliegenden Produktionsregelsysteme fehlt der Nachweis ihrer prädikatenlogischen Vollständigkeit und Korrektheit. Diese beiden Eigenschaften wurden bereits als Minimalanforderungen vorgestellt. Sie sollen von allen Inferenzkonzepten erfüllen werden, die dazu dienen, die logischen Konsequenzen prädikatenlogischer Objektmodellierungen zu untersuchen.

2) Bereits KOWALSKI (1979a), S. 435, hat eine Verwandtschaft angedeutet, die zwischen dem Petrinetz-Konzept und einem speziellen prädikatenlogischen Inferenzkonzept besteht. Er bezog sich auf die bereits früher erwähnte Konnektionsmethode, als er feststellte: "The arrow notation can be used to control the behavior of a connection graph theorem-prover ... Connection graphs controlled in such a manner are similar to Petri nets ...". Allerdings wird dieser vage Hinweis hier nicht weiter verfolgt, weil die Konnektionsmethode in dieser Arbeit keine Berücksichtigung findet.

Das Petrinetz-Konzept böte hier einen vielversprechenden Ausgangspunkt, um das mitunter diskreditierte "formallogische" Denken um netzorientierte Erkenntnisraster zu bereichern. Dabei würde der sonst oftmals beschrittene Weg vermieden, mit "vernetztem Denken" lediglich natürlichsprachliche Alltagsargumentationen wohlwollend zu umschreiben.

3) Ausführliche Erörterungen der Möglichkeiten, deklarative Netzmodelle zur Analyse oder Unterstützung von aussagen- und prädikatenlogischen Inferenzprozessen heranzuziehen, finden sich bei ZISMAN (1978a), S. 57ff., insbesondere S. 61ff.; AZEMA (1984), S. 510ff.; MAINZ (1984), S. 11ff.; GIORDANA (1985), S. 3ff., insbesondere S. 8ff.; LAUTENBACH (1985b), S. 4ff.; FIDELAK (1986a), S. 19ff. u. 108ff.; FIDELAK (1986b), S. 32ff.; MURATA,TA. (1988b), S. 481ff., insbesondere S. 486ff.; VAGIN (1988), S. 98 u. 100ff.; ZELEWSKI (1989c), S. 10 u. 12ff., insbesondere 31ff.; ZELEWSKI (1989e), S. 69ff., insbesondere 75ff.

4) Statt dessen interessiert für die Modellierung flexibler Fertigungssysteme primär das Ausdrucksvermögen operationaler Netzmodelle. Ihre Bevorzugung für die Modellierung von Realproblemen wurde bereits angesprochen.

5) Strenggenommen gilt diese Äquivalenz nur, wenn alle Stellen s_m des deklarativen Netzmodells unbeschränkte Markkapazitäten $KAP_m = \omega$ besitzen. Auf diese Prämisse unbeschränkter Markkapazitäten wird bei ZELEWSKI (1989c), S. 31, explizit hingewiesen. Bei MURATA,TA. (1988b), S. 483f., wird sie indirekt dadurch deutlich, daß dort Petrinetze konstruiert werden, ohne jemals auf Markkapazitäten der Stellen Bezug zu nehmen.

Die Prämisse unbeschränkter Markkapazitäten ist erforderlich, weil endliche Markkapazitäten dazu führen können, daß sich deklarative Netzmodelle und die Formelsysteme der zugrundeliegenden prädikatenlogischen Objektmodelle nicht mehr streng äquivalent verhalten. Dies läßt sich anhand einer einfachen Überlegung verdeutlichen: Die Prädikatenlogik ist durch das Phänomen der Semi-Entscheidbarkeit gekennzeichnet. Die Semi-Entscheidbarkeit äußert sich darin, daß es unmöglich ist zu garantieren, jede Formel, deren Allgemeingültigkeit zwar behauptet wird, die aber tatsächlich nicht allgemeingültig ist, innerhalb endlicher Ressourcenschranken als nicht allgemeingültig zu erkennen. Dieser Garantieverlust kann aber erst dann eintreten, wenn die logischen Inferenzräume unendlich groß sind. Für jedes deklarative Netzmodell mit beschränkten Markkapazitäten ist aber der logische Inferenzraum endlich. Denn die Gesamtheit aller logisch zulässigen Inferenzen wird für ein solches Netzmodell durch dessen Erreichbarkeitsgraphen bestimmt. Der Erreichbarkeitsgraph jedes Netzes mit beschränkten Markkapazitäten ist aber endlich. Folglich kann das prädikatenlogische Phänomen der Semi-Entscheidbarkeit durch Netzmodelle mit beschränkten Markkapazitäten nicht reproduziert werden. Daher ist es ausgeschlossen, daß solche Netzmodelle mit beliebigen prädikatenlogischen Formelsystemen äquivalent sind.

Die Äquivalenz besteht nur in dem Ausmaß, wie sich beschränkte Markkapazitäten auf das Schaltverhalten eines Netzmodells nicht so auswirken, daß sie prädikatenlogisch zulässige Inferenzen im Netzmodell unterbinden. Unter welchen Bedingungen solche Inferenzausschlüsse in deklarativen Netzmodellen mit beschränkten Markkapazitäten auftreten können, liegt jedoch außerhalb des Erkenntnisinteresses dieser Arbeit. Wesentlich ist aber die Feststellung, daß die hier erfolgende Festlegung auf Netzmodelle mit beschränkten Markkapazitäten eine universelle Äquivalenz deklarativer Netzmodelle verhindert. Dies unterstreicht nochmals die bereits geäußerte Präferenz zugunsten operationaler Netzmodelle. Die Entscheidung, sich auf beschränkte Markkapazitäten festzulegen, folgt im wesentlichen aus der leichteren Handhabbarkeit von endlichen Erreichbarkeitsgraphen. Hinzu kommt der Umstand, daß die intendierte Netzimplementierung ohnehin nur endliche Markenanzahlen auf den Stellen eines Netzmodells zuläßt, sofern mit den konventionellen numerischen Ausdrucksmitteln von Programmiersprachen gearbeitet wird.

6) Dabei wird der Einfachheit halber vorausgesetzt, daß die Formel "p" in der Formelmenge noch nicht als Element enthalten ist. Andernfalls wird auf die Erläuterungen verwiesen, die für den Fall $p \in FM$ bereits an früherer Stelle erfolgten.

7) Die Beweisbarkeitsbehauptung der Subjugatformel " $FM \rightarrow p$ " und die Erfüllbarkeitsbehauptung der Konjugatformel " $FM \wedge (\neg p)$ " stellen kontradiktorische Gegenteile dar. Daher bedeutet die Inkonsistenz (Unerfüllbarkeit) der Konjugatformel eine Widerlegung der Erfüllbarkeitsbehauptung und somit einen indirekten Beweis der kontradiktorischen Subjugatformel.

8) Vgl. LAUTENBACH (1985b), S. 24 i.V.m. S. 19ff.; FIDELAK (1986a), S. 19; FIDELAK (1986b), S. 38; MURATA,TA. (1988b), S. 489f.; ZELEWSKI (1989c), S. 31ff., 43ff. u. 74ff.; ZELEWSKI (1989e), S. 74ff., 79, 82 u. 85ff.

9) Strenggenommen erfordert das Netztheorem LAUTENBACH's eine Reihe von Präzisierungen. Vgl. dazu die Quellen aus der voranstehenden Anmerkung. Dabei handelt es sich um folgende Voraussetzungen:

- Das Netzmodell ist endlich, enthält keine 1-Schleifen und keine beschränkten Markenkapazitäten. Diese Prämissen hat der Verf. in ZELEWSKI (1989c), S. 29ff., näher behandelt.
- Die T-Invariante muß ein semi-positiver ganzzahliger Vektor sein. Nur dann kann sie als ein nicht-trivialer Schaltvektor von Transitionen interpretiert werden. Die Nichtnegativität des Vektors stellt sicher, daß seine Deutung als Schaltvektor zulässig ist. Die Verschärfung, es müsse sich um einen semi-positiven Vektor handeln, schließt den trivialen Nullvektor aus.
- Der Schaltvektor muß eine Schaltfolge repräsentieren, die unter der markenfreien Ausgangsmarkierung auch tatsächlich aktiviert ist. Dies ist keineswegs zwangsläufig der Fall.
- Die T-Invariante zeichnet ein charakteristisches Subnetz aus, das keine nicht-triviale S-Invariante enthält.

Vgl. zu den begrifflichen Grundlagen der Invariantenanalyse, die hier als bekannt vorausgesetzt wurden, die Quellen aus der voranstehenden Anmerkung zu LAUTENBACH's Netztheorem. Vgl. des weiteren die Erläuterungen zur Invariantenanalyse in dieser Arbeit.

10) Die markenfreie Ausgangsmarkierung wird auf triviale Weise immer durch denjenigen "Schaltvektor" $\underline{it}=\underline{0}$ "reproduziert", in dem überhaupt keine Transition des Netzmodells geschaltet wird. Er entspricht der Nullschaltfolge $SF_0=()$. Daher bleibt die Nullschaltfolge außer acht, wenn ein Netzmodell auf der Grundlage von LAUTENBACH's Netztheorem untersucht wird. Gleiches gilt für das ähnliche Netztheorem von MURATA und ZHANG.

11) Vgl. MURATA,TA. (1988b), S. 488 i.V.m. S. 486ff.

12) Da die Schaltfolge zumindest die Zieltransition enthalten muß, kann es sich niemals um die Nullschaltfolge handeln.

13) Bei MURATA,TA. (1988b), S. 488, findet sich eine abweichende Formulierung der Konklusion: Falls das Antezedens des Erreichbarkeitsbezogenen Netztheorems erfüllt ist, dann läßt sich aus der zugehörigen, invertierten Schaltfolge eine Inferenzkette rekonstruieren, die mit der Negatformel " $\neg p$ " beginnt und daraus schließlich die Leerklausel " \emptyset " ableitet. Dies wird von MURATA,TA. (1988b), S. 488, bewiesen. Die Ableitung der Leerklausel entspricht der Inkonsistenz des Formelsystems, das der Ableitung zugrundeliegt. Folglich besagt das Netztheorem ebenso, daß die Konjugatformel " $FM \wedge (\neg p)$ " inkonsistent ist; q.e.d.

14) Vgl. MURATA,TA. (1988b), S. 489. Allerdings übersehen MURATA und ZHANG dabei zwei zusätzliche Anforderungen: Erstens muß das Netzmodell ein reines Netz darstellen, damit die Invariantenanalyse überhaupt korrekt angewendet werden kann. Zweitens muß die Schaltfolge, welche die markenfreie Ausgangsmarkierung als T-Invariante reproduzieren soll, unter dieser Ausgangsmarkierung auch tatsächlich aktiviert sein. Diese Argumentationslücken lassen sich jedoch ohne Schwierigkeiten im Sinne der Präzisierungen von LAUTENBACH's Netztheorem schließen, die in einer der voranstehenden Anmerkungen angesprochen wurden.

15) Ein einfaches, besonders anschauliches Beispiel für dieses Analysepotential findet sich bei MURATA,TA. (1988b), S. 484ff.

16) Ein vierter befruchtender Ansatz wird hier nicht weiter verfolgt, da sich derzeit noch keine direkte praktische Nutzung abzeichnet. Es handelt sich jedoch um einen theoretisch interessanten Aspekt: Ausgangspunkt ist wiederum das Erreichbarkeitsbezogene Netztheorem von MURATA und ZHANG. Die Schaltfolgen in einem Netzmodell, die mit seiner Hilfe auf Inferenzprozesse in einem prädikatenlogischen Objektmodell zurückgeführt werden, bedeuten einen Wechsel des Repräsentationskonzepts für die Behandlung des zugrundeliegenden Inferenzproblems. Denn die prädikatenlogischen Inferenzprozesse, die im Objektmodell das kombinierte Unifizierungs- und Resolutionskonzept realisieren, beruhen auf dem problemreduzierenden Repräsentationskonzept (problem-reduction representation). Dabei wird das Gesamtproblem - der Versuch, eine Formel als logische Konsequenz einer Formelmenge nachzuweisen - so lange in einfachere Teilprobleme (Unifizierungs- und Resolutionsaufgaben) dekomponiert, bis sich aus trivialen Lösungen der einfachsten Teilprobleme die gesuchte Lösung des Gesamtproblems gewinnen läßt. Die Schaltfolgen im Netzmodell basieren dagegen auf dem Repräsentationskonzept der Zustandsräume (state-space representation). Dort wird das Gesamtproblem, die gewünschte Schaltfolge aufzufinden, als eine zielgerichtete Transformation von Systemzuständen dargestellt: Es gilt, ein problemrepräsentierendes System von einem Ausgangszustand (Ausgangsmarkierung des Netzmodells) aus so lange durch Anwenden von zustandstransformierenden Operationen (Schaltakte von Transitionen) zu verändern, bis ein erwünschter Zielzustand (das Wiedererreichen der Ausgangsmarkierung nach dem Schalten der Zieltransition) eintritt. Ob sich allerdings aus dem Übergang vom problemreduzierenden zum zustandsraumbezogenen Repräsentationskonzept, der durch den Wechsel von Inferenzprozessen im Objektmodell zu Schaltfolgen im Netzmodell bewirkt wird, konkrete Erkenntnisse gewinnen lassen, steht derzeit noch offen.

Vgl. zur Gegenüberstellung dieser beiden Repräsentationskonzepte für prädikatenlogische Inferenzprozesse einerseits und Schaltfolgen in Netzen andererseits MURATA,TA. (1988b), S. 486. Vgl. des weiteren zu detaillierteren Erörterungen der beiden angesprochenen Repräsentationskonzepte ZELEWSKI (1986a), S. 236ff. (Zustandsraumkonzept) u. 258ff. (Reduktionskonzept).

17) Vgl. ZELEWSKI (1986a), S. 415ff., 872ff. u. 1128; KLEE,H. (1989), S. 82ff.

18) Auch aus der T-Invariante, die das Netztheorem von LAUTENBACH liefert, kann eine solche Schaltfolge rekonstruiert werden. Daher gelten die nachfolgenden Ausführungen auch dafür. Allerdings liefert LAUTENBACH's Netztheorem keine konkrete Anleitung, wie die Rekonstruktion dieser Schaltfolge erfolgen kann. Das Netztheorem von MURATA und ZHANG bezieht sich dagegen direkt auf die interessierende Schaltfolge. Hinzu kommen die oben ange deuteten Schwierigkeiten, T-Invarianten in Höheren Netzen überhaupt festzustellen. Aus beiden Gründen be schränkt sich der Verf. auf das fruchtbarere Netztheorem von MURATA und ZHANG.

19) Vgl. MURATA,TA. (1988b), S. 488 (sowie das Beispiel auf S. 487).

20) Das kombinierte Unifizierungs- und Resolutionskonzept stellt eine indirekte Beweistechnik dar. Das wurde be reits detailliert erläutert.

21) Die Markenfreiheit dieser Ausgangsmarkierung kann intuitiv als Pendant zur Leerklauselel "Ø" aufgefaßt werden.

22) Vgl. dazu das Inferenzbeispiel bei MURATA,TA. (1988b), S. 484ff., insbesondere Fig. 3 auf S. 487 i.V.m. Fig. 1 auf S. 485.

23) Vgl. MURATA,TA. (1988b), S. 484.

24) Dieser Aspekt klingt bei MURATA,TA. (1988b), S. 494ff., insbesondere S. 496 ("The net model ... can be used for guiding the search for a solution."), an. Vgl. dazu auch den Hinweis von KOWALSKI (1979a), S. 435, der bereits in einer früheren Anmerkung zitiert wurde. Dort wurde die konzeptionelle Nähe zwischen Petrinetzen und Konnek tionsgraphen angesprochen, die bei einer speziellen prädikatenlogischen Inferenzmethode den Verlauf von Infe renzprozessen wiedergeben.

25) Vgl. hinsichtlich der Befruchtung der Prädikatenlogik MURATA,TA. (1988b), S. 484.

26) MURATA und ZHANG haben sich bereits ausführlich mit der Parallelisierung von prädikatenlogischen Inferenz prozessen auf der Basis von Prädikat/Transition-Netzen befaßt; vgl. MURATA,TA. (1988b), S. 481, 484 u. 490ff. Die beiden Autoren schätzen die Effizienzsteigerungen, die sich durch Netzmodelle für parallele Inferenzprozesse er zielen lassen, grundsätzlich positiv ein ("The net model offers ... improvement of efficiency through parallel proces sing", S. 495), weisen aber auch auf derzeit noch bestehende Einschränkungen hin (S. 496).

Vgl. zu weiteren Ansätzen, die Petrinetze für die Unterstützung nebenläufiger Inferenzprozesse anregen, VAGIN (1988), S. 98 u. 105.

27) Wegen der früher thematisierten Schwierigkeiten der Invariantenanalyse für Höhere Netze schätzt der Verf. die Realisierungschancen solcher numerischen Beiträge allerdings sehr gering ein. Bisher haben sich Invariantenanaly sen nur auf arithmetischer Basis bewährt, die gerade das Ausdrucksvermögen von Stelle/Transition-Netzen abdeckt. Dagegen läßt sich derzeit kaum vorstellen, wie die wesentlich komplexeren Höheren Netze, die den Formulierungs reichum der Prädikatenlogik erst erschließen, mit Instrumenten der Invariantenanalyse effizient bewältigt werden könnten. Insbesondere erscheint fraglich, ob die primär numerische Ausrichtung der Invariantenanalyse überhaupt geeignet ist, die nicht-numerischen Ausdruckspotentiale der Prädikatenlogik abzudecken. Daher erstaunt es, daß sogar von MURATA,TA. (1988b), S. 488ff., die Invariantenanalyse mit der Absicht untersucht wird, die Ausführung prädikatenlogischer Inferenzprozesse zu unterstützen. Allerdings werden dort auch zwei wesentliche Vorbehalte gegenüber der Eignung der Invariantenanalyse für diesen Zweck geäußert (S. 489f.).

28) Als herausragendes Beispiel solcher Programmiersprachen wird in dieser Arbeit die Sprache PROLOG ausführlicher behandelt.

29) Die Restriktion, lediglich HORN-Klauseln zuzulassen, wurde für die Programmiersprache PROLOG bereits dar gelegt. Die begrenzte Ausdrucksfähigkeit solcher HORN-Klauseln wurde ebenso schon thematisiert.

30) Genau genommen liegt eine imprädikative Definition vor. Bei ihr wird ein Begriff durch Bezugnahme auf eine Gesamtheit definiert, die ihrerseits das Definiendum als notwendigen Bestandteil umfaßt. Das Definiendum stellt hier der Begriff der Transitionsaktivierung dar. Die definierende Gesamtheit besteht aus der Menge aller aktivierten Transitionen. Auf die Entstehung von logischen Antinomien durch imprädikative Begriffsdefinitionen wurde schon in einer früheren Anmerkung hingewiesen.

31) Das Netz lehnt sich an das Zählernetz an, das an früherer Stelle präsentiert wurde.

32) Der Ausdruck "Aktivierungsanzahl" ist eine Variable der Sorte "aktivierungsanzahl".

33) Statt dessen müßte zu einer logischen Typentheorie übergegangen werden, um die Antinomien imprädikativer Begriffsdefinitionen zu überwinden. Vgl. die voranstehende Anmerkung zum imprädikativen Definitionscharakter sowie die frühere Anmerkung zur logischen Typentheorie.

34) Eine ähnliche Auswirkung, die reflexive Konstrukte in Netzen hervorrufen können, stellt FERSTL (1979), S. 107f., vor. Er zeigt, daß die Modellierung eines speziellen ökonomischen Gleichungssystems durch ein Petrinetz in einen unerwünschten *Deadlock* führt. Dabei enthält das Gleichungssystem eine Gleichung, in der eine ökonomische Größe in reflexiver Weise auf beiden Gleichungsseiten verwendet wird.

Literaturverzeichnis zu Teilband 5.2

Vorbemerkungen:

- ❑ Jedes Werk wird durch die Angabe eines Referenztitels (1. Zeile) und durch seine bibliographischen Angaben (folgende Zeilen) aufgeführt. In den Quellenangaben dieser Arbeit wird immer auf den Referenztitel Bezug genommen.
- ❑ Die Referenztitel bestehen nur aus den Autorennachnamen und den Erscheinungsjahren, solange hierdurch eine eindeutige Identifizierung der jeweils zugehörigen Werke möglich ist. Andernfalls dienen zusätzliche - abgekürzte - Autorenvornamen oder alphabetische Zusätze zu den Erscheinungsjahren der eindeutigen Identifizierung.
- ❑ Um eine einheitliche Quellenangabe in allen Bänden des Projekts PEMOPS zu gewährleisten, bezieht sich die eindeutige Identifizierung durch Autorenvornamen und alphabetische Zusätze zu den Erscheinungsjahren auf den Gesamtkorpus aller verarbeiteten Quellen. Daher kann es dazu kommen, daß innerhalb eines Bandes Lücken klaffen. Sie resultieren daraus, daß die scheinbar fehlenden Quellen im Gesamtkorpus zwar enthalten sind, aber im jeweils betroffenen Band nicht verwendet wurden.
- ❑ Die Titel fremdsprachlicher Werke werden grundsätzlich in der Notation des Originals wiedergegeben. Allerdings gelten drei Ausnahmen:
 - Titel, die sich nicht mit dem deutschsprachigen Alphabet ausdrücken lassen, werden in ihrer lautsprachlichen Umschreibung durch das deutschsprachige Alphabet wiedergegeben. Dies gilt insbesondere für Werke mit chinesischen oder kyrillischen Schriftzeichen.
 - Falls die Titel im Original durchgängig mit Großbuchstaben dargestellt werden, erfolgt hier eine Notation in der jeweils sprachspezifischen Groß-/Kleinschreibung von Titeln. Dies trifft vor allem auf anglophone Werke zu, in deren Titeln die jeweils sinnbestimmenden Worte durch Großbuchstaben eingeleitet werden.
 - Accents und andere diakritische Zeichenbestandteile, die nicht im deutschsprachigen Alphabet enthalten sind, werden grundsätzlich ausgelassen.
- ❑ In das Literaturverzeichnis wurden alle Quellen aufgenommen, auf die in den Anmerkungen zum laufenden Text verwiesen wurde.
- ❑ Weitere Publikationen, die sich auf die Thematik des Petrinetz-Konzepts beziehen, aber in den vorgenannten Quellen nicht angesprochen wurden, finden sich im Band 10 des Projekts PEMOPS zur Petrinetz-Literatur.
- ❑ Die Literaturlauswertung wurde 1992 abgeschlossen (vgl. das Vorwort in Band 1).

Abel,D. (1986a)

Abel,D.; Rake,H.: Simulation von Komplexen Steuerungssystemen mit Petri-Netzen; in: Geril, P.; Vansteenkiste,G.C.; Kerckhoffs,E.J.H. (Hrsg.): Proceedings of the 2nd European Simulation Congress, 1986 in Antwerpen, Ghent 1986, S. 57-62.

Abel,D. (1990)

Abel,D.: Petri-Netze für Ingenieure - Modellbildung und Analyse diskret gesteuerter Systeme, Berlin - Heidelberg - New York 1990.

Ackoff (1970)

Ackoff,R.L.; Sasieni,M.W.: Operations Research - Grundzüge der Operationsforschung, Stuttgart 1970.

Agerwala (1973)

Agerwala,T.; Flynn,M.: Comments on Capabilities, Limitations and "Correctness" of Petri Nets; in: Lipovski G.J.; Szygenda,S.A. (Hrsg.): Proceedings of the 1st Symposium on Computer Architecture (ACM), New York 1973, S. 81-86.

Agerwala (1978a)

Agerwala,T.: Some Applications of Petri Nets; in: Tranter,W.H. (Hrsg.): Proceedings of the National Electronics Conference, Vol. 32, 16.-18.-10.1978 in Chicago, Oak Brook 1978, S. 149-154.

Agerwala (1978b)

Agerwala,T.; Choed-Amphai,Y.-C.: A Synthesis Rule for Concurrent Systems; in: o.V.: Proceedings of the 15th Design Automation Conference, im Juni 1978 in Las Vegas, New York 1978, S. 305-311.

Agerwala (1979)

Agerwala,T.: Putting Petri Nets to Work; in: Computer, Vol. 12 (1979), No. 12, S. 85-94.

Aho (1974)

Aho,A.V.; Hopcroft,J.E.; Ullman,J.D.: The Design and Analysis of Computer Algorithms, Reading - Menlo Park - London ... 1974.

Alagic (1986)

Alagic,S.: Relational Database Technology, New York - Berlin - Heidelberg ... 1986.

Aldinger (1985a)

Aldinger,L.: Leitstandunterstützte kurzfristige Fertigungssteuerung bei Einzel- und Kleinserienfertigung, Berlin - Heidelberg - New York - Tokyo 1985.

Amamiya (1982)

Amamiya,M.; Hakozaki,K.; Yokoi,T.; Fasaoka,A.; Tanaka,Y.: New Architecture for Knowledge Base Mechanisms; in: Moto-oka,T. (Hrsg.): Fifth Generation Computer Systems, Proceedings of the International Conference on Fifth Generation Computer Systems, 19.-11.10.1981 in Tokyo, Amsterdam - New York - Oxford 1982, S. 179-188.

Amamiya (1984)

Amamiya,M.; Hasegawa,R.: Dataflow Computing and Eager and Lazy Evaluations; in: New Generation Computing, Vol. 2 (1984), S. 105-129.

Amin (1976)

Amin,A.T.; Murata,T.: A Characterization of Live and Safe Markings of a Directed Graph; in: o.V.: Proceedings of the Annual Princeton Conference on Information Sciences and Systems, 31.03.-2.04.1976 in Baltimore, Princeton 1976, S. 295-299.

Andre (1980)

Andre,C.; Diaz,M.; Girault,C.; Sifakis,J.: Survey of French Research and Applications Based on Petri Nets, in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 321-345.

Anton (1989)

Anton, M.D.; Bretschneider, M.: Formulas, processes, and Petri-Nets applied to the specification and verification of a HDLC protocol; in: Diaz, J.; Orejas, F. (Hrsg.): TAPSOFT'89, Proceedings of the International Joint Conference on Theory and Practice of Software Development, 13.-17.03.1989 in Barcelona, Volume 2: Advanced Seminar on Foundations of Innovative Software Development II and Colloquium on Current Issues in Programming Languages (CCIPL), Lecture Notes in Computer Science 352, Berlin - Heidelberg - New York ... 1989, S. 140-154.

Araki (1976b)

Araki, T.; Taniguchi, K.; Kasami, T.: Some NP-Complete Problems for Bounded Petri Nets; in: The Transactions of the IECE of Japan, Vol. E-69 (1976), S. 7-8.

Araki (1976c)

Araki, T.; Kasami, T.: Solvability of Decision Problems Related to the Deadlock Prevention and Avoidance of Petri Nets / Decision Problems on the Strong Connectivity of Reachability Set; in: Papers of Technical Group on Automation and Languages of the IECE of Japan, Vol. AL-76-2, S. 7-16.

Araki (1977b)

Araki, T.; Kasami, T.: Decidable Problems on the Strong Connectivity of Petri Nets Reachability Sets; in: Theoretical Computer Science, Vol. 4 (1977), S. 99-119.

Arlabosse (1988)

Arlabosse, F.; Jean-Bart, B.; Porte, N.; de Ravinel, B.: An Efficient Problem Solving Architecture Using ATMS - Tested on a Non-Toy Case Study; in: AI Communications, Vol. 1 (1988), S. 6-15.

Aronson (1986)

Aronson, J.E.; Chen, B.D.: A Forward Network Simplex Algorithm for Solving Multiperiod Network Flow Problems; in: Naval Research Logistics Quarterly, Vol. 33 (1986), S. 445-467.

Augin (1978a)

Augin, M.; Boeri, F.; Andre, C.: New Design Using PLAs And Petri Nets; in: o.V.: MECO'78 - Measurement and Control International Symposium, Athen 1978, S. 864-869.

Augin (1980)

Augin, M.; Boeri, F.; Andre, C.: Systematic Method of Realization of Interpreted Petri Nets; in: Digital Processes, Vol. 6 (1980), No. 1, S. 55-68.

Ayache (1979a)

Ayache, J.M.; Diaz, M.; Valette, R.: Methode de specification de la commande dans les systemes de commutation electronique; in: o.V.: Proceedings of the International Switching Symposium ISS'79, Paris 1979, S. 1049-1056.

Azema (1976b)

Azema, P.; Valette, R.; Diaz, M.: Petri Nets As a Common Tool For Design Verification and Hardware Simulation; in: o.V.: Proceedings of the 13th Annual Design Automation Conference (IEEE), in San Francisco, New York 1976, S. 109-116.

Azema (1978)

Azema, P.; Ayache, J.M.; Berthomieu, B.: Design and Verification of Communication Procedures: A Bottom-Up Approach; in: o.V.: Proceedings of the 3rd International Conference on Software Engineering, New York 1978, S. 168-174.

Azema (1980)

Azema, P.; Berthomieu, B.; Decitre, P.: The Design and Validation by Petri Nets of a Mechanism for the Invocation of Remote Servers; in: Lavington, S. (Hrsg.): Information Processing 80, Proceedings of the IFIP Congress 80, 6.-9.10.1980 in Tokyo und 14.-17.10.1980 in Melbourne, Amsterdam - New York - Oxford 1980, S. 599-604.

Azema (1984)

Azema, P.; Juanole, G.; Sanchis, E.; Montbernard, M.: Specification and Verification of Distributed Systems Using PROLOG Interpreted Petri Nets; in: o.V.: Proceedings of the IEEE Software Engineering Conference 1984, New York 1984, S. 510-518.

Bachem (1980)

Bachem, A.: Komplexitätstheorie im Operations Research; in: Zeitschrift für Betriebswirtschaft, 50. Jg. (1980), S. 812-844.

Back-Hock (1991a)

Back-Hock,A.: Perspektiven für die DV-Unterstützung des Controlling; in: Controlling, 3. Jg. (1991), Heft 2, S. 94-99.

Baer,J. (1973a)

Baer,J.L.: A Survey of Some Theoretical Aspects of Multiprocessing; in: Computing Surveys, Vol. 5 (1973), S. 31-80.

Baer,J. (1973b)

Baer,J.L.: Modeling for Parallel Computation: A Case Study; in: o.V.: Proceedings of the Saga-more Computer Conference on Parallel Processing, New York 1973, S. 13-22.

Baer,J. (1977)

Baer,J.-L.; Ellis,C.S.: Model, Design, and Evaluation of a Compiler for a Parallel Processing Environment; in: IEEE Transactions on Software Engineering, Vol. SE-3 (1977), S. 394-405.

Baer,J. (1982)

Baer,J.-L.: Techniques to exploit parallelism; in: Evans,D.J. (Hrsg.): Parallel processing systems, Cambridge - London - New York ... 1982, S. 75-99.

Baker,A. (1988)

Baker,A.D.: Complete Manufacturing Control Using a Contract Net: A Simulation Study; in: o.V.: 1988 International Conference on Computer Integrated Manufacturing, 23.-25.05.1988 in Troy, Washington 1988, S. 100-109.

Baker,H. (1972)

Baker,H.: Petri Nets and Languages, Computation Structures Group Memo 68, Project MAC am Massachusetts Institute of Technology, Cambridge (Massachusetts) 1972.

Bamberg (1989)

Bamberg,G.; Coenenberg,A.G.: Betriebswirtschaftliche Entscheidungslehre, 5. Aufl., München 1989.

Banerji (1980)

Banerji,R.B.: Artificial Intelligence - A Theoretical Approach, New York - Oxford 1980.

Barr,A. (1981)

Barr,A.; Feigenbaum,E.A. (Hrsg.): The Handbook of Artificial Intelligence, Vol. 1, Stanford - Los Altos 1981.

Barr,R. (1989)

Barr,R.S.; Christiansen,M.G.: A Parallel Auction Algorithm: A Case Study in the Use of Parallel Object-Oriented Programming; in: Sharda,R.; Golden,B.L.; Wasil,E.; Balci,O.; Stewart,W. (Hrsg.): Impacts of Recent Computer Advances on Operations Research, New York - Amsterdam - London 1989, S. 23-32.

Barzilai (1979)

Barzilai,Z.; Yoeli,M.; Strasbourger,E.: On Structured Parallel Control Schema, Technical Report No. 151, Department of Computer Science, Technion - Israel Institute of Technology, Haifa 1979.

Bathelt (1982)

Bathelt,P.: Vergleich von Synchronisationsmechanismen, Diplomarbeit, Universität Erlangen, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik) an der Universität Erlangen-Nürnberg, Bd. 15, Nr. 3, Erlangen 1982.

Battiston (1988)

Battiston,E.; De Cindio,F.; Mauri,G.: OBJSA Nets: a class of high-level nets having objects as domains; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1988, Lecture Notes in Computer Science 340, Berlin - Heidelberg - New York ... 1988, S. 20-43.

Bauer,F. (1981)

Bauer,F.L.; Wössner,H.: Algorithmische Sprache und Programmentwicklung, Berlin - Heidelberg - New York 1981.

Baumers (1977)

Baumers,B.: Zur Reduktion und graphischen Manipulation von Petri-Netzen, Diplomarbeit, Abteilung Informatik, Universität Dortmund, Dortmund 1977.

Baumgarten (1978)

Baumgarten,B.; Prinoth,R.: Einige Begriffe und Ergebnisse aus der Theorie der Petri-Netze, Teil I, Interner Bericht IFV am Institut für Datenfernverarbeitung der Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Darmstadt 1978.

Becker,J. (1978)

Becker,J.: Stand und Tendenzen der Weiterentwicklung von DV-gestützten Produktionsplanungs- und Produktionssteuerungssystemen; in: Ellinger,T.; Wildemann,H. (Hrsg.): Praktische Fälle zur Produktionssteuerung, Wiesbaden 1978, S. 499-518.

Beckmann,M. (1968)

Beckmann,M.J.: Dynamic Programming of Economic Decisions, Berlin - Heidelberg - New York 1968.

Beckstein (1988a)

Beckstein,C.: Zur Logik der Logik-Programmierung - Ein konstruktiver Ansatz -, Dissertation, Universität Erlangen-Nürnberg, Erlangen 1988. (Auch erschienen als: Informatik-Fachberichte 199, Berlin - Heidelberg - New York ... 1988.)

Behrendt (1988)

Behrendt,R.; Müller,J.: Expertensysteme bringen einige Vorteile mit sich - Qualitative Vorteile für Planung und Disposition; in: DVZ, Ausgabe vom 1.10.1988, Sonderbeilage Luftfracht, S. 32-33.

Bekhi (1989)

Bekhi,N.; Tavares,S.E.: An Integrated Approach To Protocol Design; in: o.V.: Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 01.-02.06.1989 in Victoria, New York 1989, S. 244-248.

Bellman (1954)

Bellman,R.: The Theory of Dynamic Programming; in: Bulletin of the American Mathematical Society, Vol. 60 (1954), S. 503-515.

Bellman (1957a)

Bellman,R.: Dynamic Programming, Princeton - New York 1957.

Bellman (1957b)

Bellman,R.: Dynamic Programming and the Numerical Solution of Variational Problems; in: Operations Research, Vol. 5 (1957), S. 277-288.

Bellman (1967)

Bellman,R.: Dynamische Programmierung und selbstanpassende Regelprozesse, München - Wien 1967.

Benito (1978)

Benito,F.; Gröflin,H.: Optimierungsprobleme mit nicht polynomial begrenzten Algorithmen; in: Liebling,T.M.; Rössler,M. (Hrsg.): Kombinatorische Entscheidungsprobleme: Methoden und Anwendungen, Lecture Notes in Economics and Mathematical Systems 153, Berlin - Heidelberg - New York 1978, S. 81-121.

Bermond (1983)

Bermond,J.C.; Memmi,G.: A Graph Theoretical Characterization of Minimal Deadlocks; in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. (1983), S. 31-42.

Bernstein (1973)

Bernstein,P.A.: Description Problems in the Modeling of Asynchronous Computer Systems, Master of Science-Thesis 1972, zugleich: Technical Report No. 48, Department of Computer Science, University of Toronto, Toronto 1973.

Berthelot (1976)

Berthelot,G.; Roucairol,G.: Reduction of Petri Nets; in: Mazurkiewicz,A. (Hrsg.): Mathematical Foundations of Computer Science 1976, Proceedings, 5th Symposium, 6.-10.09.1976 in Gdansk, Lecture Notes in Computer Science 45, Berlin - Heidelberg - New York 1976, S. 202-209.

Berthelot (1977)

Berthelot,G.: Cheking Liveness of Petri-Nets; in Feilmeier,M. (Hrsg.): Parallel Computers - Parallel Mathematics, Proceedings of the IMACS (AICA)-GI-Symposium, 14.-16.03.1977 in München, Amsterdam - New York - Oxford 1977, S. 217-220. (Anmk. des Verf.: Druckfehler in "Cheking"; gemeint ist wohl: "Checking".)

Berthelot (1980a)

Berthelot,G.; Roucairol,G.; Valk,R.: Reductions of Nets and Parallel Programs; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 277-290.

Berthelot (1982b)

Berthelot,G.; Terrat,R.: Modeling and Proofs of a Data Transfer Protocol by Predicate/Transition Nets; in: Girault,C.; Reisig,W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 251-257.

Berthelot (1982c)

Berthelot,G.; Terrat,R.: Protocoles de Transfert d'Informations de Connexion et de Deconnexion, Extrait de G.W. Brams "Reseaux de Petri, Theorie et Pratique", Institut de Programmation, Université Pierre et Marie Curie Paris, Paris 1982.

Berthelot (1982d)

Berthelot,G.; Terrat,R.: Petri Nets Theory for the Correctness of Protocols; in: van de Riet,R.P.; Litwin,W. (Hrsg.): Distributed Data Sharing Systems, Proceedings of the 2nd Seminar on Distributed Data Sharing Systems, im Juni 1982, Amsterdam 1982, S. 23-43.

Berthelot (1983a)

Berthelot,G.; Memmi,G.; Reisig,W.: A Control Structure for Sequential Processes Synchronized by Buffers; in: o.V.: Papers Presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 43-58.

Berthelot (1985a)

Berthelot,G.: Analyse de processus paralleles par transformation de reseaux de Petri: Application a un protocole de reseau; in: Technique et Science Informatique, Vol. 4 (1985), No. 1, S. 73-82.

Berthelot (1985b)

Berthelot,G.: Transformations de reseau de Petri; in: Technique et Science Informatiques, Vol. 4 (1985), No. 1, S. 91-101.

Berthelot (1986)

Berthelot,G.: Checking properties of nets using transformations; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1985, Lecture Notes in Computer Science 222, Berlin - Heidelberg - New York - Tokyo 1986, S. 19-40.

Berthelot (1990)

Berthelot,G.; Finkel,A.; Johnen,C.; Petrucci,L.: A generic example for testing performance of reachability and covering graphs construction algorithms; in: Petri Net Newsletter (Gesellschaft für Informatik: Special Interest Group on Petri Nets and related System Models), No. 35 (1990), S. 6-7.

Best,E. (1974)

Best,E.: Beiträge zur Petrinetz-Theorie, Diplomarbeit am Institut für Informatik, Universität Karlsruhe, Karlsruhe 1974.

Best,E. (1975a)

Best,E.; Schmid,A.: Systems of Open Paths in Petri Nets; in: Becvar,J. (Hrsg.): Mathematical Foundations of Computer Science 1975, 4th Symposium, 1.-5.09.1975 in Marianske Lazne, Lecture Notes in Computer Science 32, Berlin - Heidelberg - New York 1975, S. 186-193.

Best,E. (1980a)

Best,E.: Atomicity of Activities; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 225-250.

Best,E. (1980c)

Best,E.: Adequacy of Path Programs; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 291-305.

Best,E. (1982d)

Best,E.: Adequacy Properties of Path Programs; in: Theoretical Computer Science, Vol. 18 (1982), S. 149-171.

Bibel (1982a)

Bibel,W.: Automated Theorem Proving, Braunschweig - Wiesbaden 1982.

Bibel (1989)

Bibel,W.; del Cerro,L.F.; Fronhöfer,B.; Herzig,A.: Plan Generation by Linear Proofs: On Semantics; in: Metzging,D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 49-62.

Bic (1984)

Bic,L.: A Data-Driven Model for Parallel Interpretation of Logic Programs; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 517-523.

Billington (1981a)

Billington,J.; Symons,F.J.W.: Modelling and Analysis of Communication Protocols - Part I; in: o.V.: Proceedings IREECON International, Melbourne - Sydney 1981, S. 18-20.

Bland (1981)

Bland,R.G.; Goldfarb,D.; Todd,M.J.: The Ellipsoid Method: A Survey; in: Operations Research, Vol. 29 (1981), S. 1039-1091.

Blohm (1988)

Blohm,H.; Lüder,K.: Investition - Schwachstellen im Investitionsbereich des Industriebetriebes und Wege zu ihrer Beseitigung, 6. Aufl., München 1988.

Boden (1977)

Boden,M.: Artificial Intelligence and Natural Man, Hassocks - New York 1977.

Boley (1981)

Boley,H.: AI Languages and AI Machines: An Overview; in: Siekamnn,J.H. (Hrsg.): GWAI-81, (5th) German Workshop on Artificial Intelligence, 26.-31.01.1981 in Bad Honnef, Informatik-Fachberichte 47, Berlin - Heidelberg - New York 1981, S. 136-155.

Borgwardt (1982)

Borgwardt,K.-H.: The Average Number of Pivot Steps Required by the Simplex-Method is Polynomial; in: Zeitschrift für Operations Research, Vol. 26 (1982), S. 157-177.

Borgwardt (1985)

Borgwardt,K.H.: Der durchschnittliche Rechenaufwand beim Simplexverfahren; in: Ohse,D., Esprester,A.C.; Küpper,H.-U.; Stähly,P.; Steckhan,H. (Hrsg.): Operations Research Proceedings 1984, DGOR - Vorträge der 13. Jahrestagung, 12.-14.09.1984 in Sankt Gallen, Berlin - Heidelberg - New York ... 1985, S. 647-660.

Boussin (1978)

Boussin,J.L.: Synthesis and Analysis of Logic Automation Systems; in: o.V.: Proceedings of the 7th Triennial World Congress of the International Federation of Automatic Control, im Juni 1978 in Helsinki, Oxford - New York 1978, S. 1527-1535.

Bradley,S. (1977)

Bradley,S.P.; Hax,A.C.; Magnanti,T.L.: Applied Mathematical Programming, Reading - Menlo Park - London ... 1977.

Bretschneider (1980a)

Bretschneider,G.: Petrinetze - Ein leicht verständliches Verfahren zum Systementwurf, München 1980.

Brewka (1989)

Brewka, G.: Nonmonotonic Logics - A Brief Overview; in: AI Communications, Vol. 2 (1989), S. 88-97.

Brinch Hansen (1972)

Brinch Hansen, P.: A Comparison of Two Synchronizing Concepts; in: Acta Informatica, Vol. 1 (1972), S. 190-199.

Brown, A. (1987)

Brown, A.L.; Gaucas, D.E.; Benanav, D.: An Algebraic Foundation for Truth Maintenance; in: o.V.: IJCAI 87 - Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 23.-28.08.1987 in Mailand, o.O. (Los Altos) 1987, Vol. 2, S. 973-980.

Brown, C.A. (1981)

Brown, C.A.; Purdom, P.W.: How to Search Efficiently; in: Drinan, A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, S. 588-594.

Brucker (1975)

Brucker, P.; Lenstra, J.K.; Rinnooy Kan, A.H.G.: Complexity of Machine Scheduling Problems, Preprint BW 43/75, stichting mathematisch centrum, afdeling mathematische beslistkunde, Amsterdam 1975.

Brucker (1976)

Brucker, P.: Die Komplexität von Scheduling Problemen; in: Kohlas, J.; Seifert, O.; Stähly, P.; Zimmermann, H.-J. (Hrsg.): Proceedings in Operations Research 5, Vorträge der Jahrestagung 1975 - DGOR, Würzburg - Wien 1976, S. 357-368.

Brucker (1979)

Brucker, P.: NP-Complete Operations Research Problems and Approximation Algorithms; in: Zeitschrift für Operations Research, Bd. 23 (1979), S. 73-94.

Brucker (1981)

Brucker, P.: Scheduling, Wiesbaden 1981.

Bruderer (1978)

Bruderer, H.E.: Simulation von Entscheidungsprozessen - Künstliche Intelligenz, Institut für linguistische Datenverarbeitung, Münsingen 1978.

Brüggemann (1991)

Brüggemann, U.: Integriertes Bedienfeld - ein neuer Weg der Informationsverarbeitung an der Werkzeugmaschine; in: Zeitschrift für wirtschaftliche Fertigung und Automatisierung, 86. Jg. (1991), S. 226-229.

Bruno, J. (1971)

Bruno, J.; Altman, S.M.: A Theory of Asynchronous Control Networks; in: IEEE Transactions on Computers, Vol. C-20 (1971), S. 629-638.

Buchanan (1982)

Buchanan, B.G.; Duda, R.O.: Principles of Rule-Based Expert Systems, Report No. STAN-CS-82-926, Department of Computer Science, Stanford University, Stanford 1982.

Budinas (1989)

Budinas, B.L.: Decidability of the Petri net Reachability Problem; in: Automation and Remote Control, Vol. 49 (1989), S. 1393-1422.

Bungers (1984)

Bungers, D.; Gebhardt, F.: Suchverfahren; in: Bungers, D.; Müller, B.S.; Raulefs, P. (Hrsg.): Expertensysteme - Beiträge aus einem Workshop zur Ausarbeitung des State-of-the-Art, Bd. 1: Grundlagen der Künstlichen Intelligenz, Arbeitspapiere der GMD 140, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1984, S. 1-33.

Burkhard (1982a)

Burkhard, H.-D.: On Fairness in Petri Nets; in: o.V.: Berichte vom 20. Mathematischen Kolloquium in Rostock, o.O. 1982, S. 85-96.

Burks (1963)

Burks, A.W.; Goldstine, H.H.; von Neumann, J.: Preliminary Discussion of the Logical Design of an Electronic Computing Instrument (2. Aufl. eines Berichts am Institute for Advanced Study aus dem Jahr 1947); in: Taub, A.H. (Hrsg.): John von Neumann - Collected Works, Vol. V: Design of Computers, Theory of Automata and Numerical Analysis, Oxford - London - New York ... 1963, S. 34-79. (Auch erschienen in: Pylyshyn, Z.W. (Hrsg.): Perspectives in Computer Revolution, Englewood Cliffs 1970, S. 37-46.)

Byrn (1974)

Byrn, W.H.: Sequential Processes, Deadlocks, and Semaphore Primitives, Dissertation, Department of Applied Mathematics, Harvard University, Cambridge (Massachusetts) 1974.

Cardoza (1976)

Cardoza, E.; Lipton, R.; Meyer, A.R.: Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report; in: o.V.: Conference Record of The Eighth Annual ACM Symposium on Theory of Computing, Papers presented at the Symposium, 3.-5.05.1976 in Hershey, New York 1976, S. 50-54.

Carstensen (1982)

Carstensen, H.: Fairness bei Petrinetzen mit unendlichem Verhalten, Bericht Nr. 93 am Fachbereich Informatik, Universität Hamburg, Hamburg 1982.

Cavarroc (1974)

Cavarroc, J.C.; Blanchard, M.; Gillon, J.: An Approach to the Modular Design of Industrial Switching Systems; in: o.V.: Discrete Systems, Papers from an International Symposium, 30.09.-4.10.1974 in Riga, o.O. 1974, S. 93-102.

Cerf (1972)

Cerf, V.G.: Multiprocessors, Semaphores, and a Graph Model of Computation, Dissertation, Computer Science Department, University of California, Los Angeles 1972.

Chandy (1984)

Chandy, K.M.; Misra, J.: The Drinking Philosophers Problem; in: ACM Transactions on Programming Languages and Systems, Vol. 6 (1984), S. 632-646.

Chin, H. (1989)

Chin, H.L.; Cooper, G.F.: Bayesian Belief Network Inference Using Simulation; in: Kanal, L.N.; Levitt, T.S.; Lemmer, J.F. (Hrsg.): Uncertainty in Artificial Intelligence 3, Amsterdam - New York - Oxford ... 1989, S. 129-147.

Christaller (1982)

Christaller, T.: Konsistenzüberprüfungen bei sich verändernden Wissensbasen; in: Wahlster, W. (Hrsg.): GWAI-82, 6th German Workshop on Artificial Intelligence, 27.09.-01.10.1982 in Bad Honnef, Informatik-Fachberichte 58, Berlin - Heidelberg - New York 1982, S. 63-71.

Chrzastowski-Wachtel (1983b)

Chrzastowski-Wachtel, P.: Puzzle No 3; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 13 (1983), S. 22. (Anmk. des Verf.: Druckfehler in "Chrzastowski"; gemeint ist wohl: "Chrzastowski".)

Clarkson (1960)

Clarkson, G.P.E.; Simon, H.A.: Simulation of Individual and Group Behavior; in: American Economic Review, Vol. 50 (1960), No. 5, S. 920-932.

Clausen, J. (1986)

Clausen, J.; Krarup, J.: Combinatorial Optimization: Challenges and Trends; in: Streitferdt, L.; Hauptmann, H.; Marusev, A.W.; Ohse, D.; Pape, U. (Hrsg.): Operations Research Proceedings 1985, DGOR - Vorträge der 14. Jahrestagung, Berlin - Heidelberg - New York ... 1986, S. 24-46.

Clausen, M. (1989)

Clausen, M.; Fortenbacher, A.: Efficient Solution of Linear Diophantine Equations; in: Journal of Symbolic Computation, Vol. 8 (1989), S. 201-216.

Clocksinn (1990)

Clocksinn, W.F.; Mellish, C.S.: Programmieren in Prolog, (Übersetzung der 3. Aufl. 1987), Berlin - Heidelberg - New York ... 1990.

Coffman (1971)

Coffman,E.G.; Elphick,M.J.; Shoshani,A.: System Deadlocks; in: Computing Surveys, Vol. 3 (1971), S. 67-78.

Coffman (1973)

Coffman,E.G.; Denning,P.J.: Operating Systems Theory, Enlewood Cliffs 1973.

Coffman (1982)

Coffman,E.G.; Frederickson,G.N.; Lueker,G.S.: Probabilistic Analysis of the LPT Processor Scheduling Heuristic; in: Dempster,M.A.H.; Lenstra,J.K.; Rinnooy Kan,A.H.G. (Hrsg.): Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6.-17.07.1981 in Durham, Dordrecht - Boston - London 1982, S. 319-331.

Cohen,E. (1975)

Cohen,E.S.: A Semantic Model For Parallel Systems With Scheduling; in: o.V.: Conference Record of the 2nd ACM Symposium on Principles of Programming Languages, 20.-22.01.1975 in Palo Alto, Palo Alto 1975, S. 87-94.

Cohen,P. (1982)

Cohen,P.R.; Feigenbaum,E.A. (Hrsg.): The Handbook of Artificial Intelligence, Vol. III, London - Los Altos 1982.

Colom (1989b)

Colom,J.M.; Silva,M.: Convex Geometry and Semiflows in P/T Nets. A Comparative Study of Algorithms for Computation of Minimal P-Semiflows; in: o.V.: 10th International Conference on Application and Theory of Petri Nets, 28.-30.06.1989 in Bonn, o.O. 1989, S. 74-95.

Commoner (1971a)

Commoner,F.; Holt,A.W.; Even,S.; Pnueli,A.: Marked Directed Graphs; in: Journal of Computer and System Sciences, Vol. 5 (1971), S. 511-523.

Conry (1979a)

Conry,S.E.; Jump,J.R.: On Functional Equivalences in a Model for Parallel Computation; in: Information and Control, Vol. 41 (1979), S. 247-274.

Cook,S. (1971)

Cook,S.A.: The Complexity of Theorem-Proving Procedures; in: o.V.: Conference Record of the 3rd Annual ACM Symposium on Theory of Computing, 1971 in Urbana, o.O. (New York) 1971, S. 151-158.

Cook,S. (1983)

Cook,S.A.: An Overview of Computational Complexity; in: Communications of the ACM, Vol. 26 (1983), 401-408.

Coopriider (1976)

Coopriider,L.W.: Petri Nets and the Representation of Standard Synchronizations, Department of Computer Science, Carnegie-Mellon University, Pittsburgh 1976.

Cordes (1988)

Cordes,R.; Kruse,R.; Langendörfer,H.; Rust,H.: Prolog - Eine methodische Einführung, Braunschweig - Wiesbaden 1988.

Cotronis (1977)

Cotronis,J.Y.; Lauer,P.E.: Verification of Concurrent Systems of Processes; in: Morlet,E.; Ribbens,D. (Hrsg.): Proceedings of the International Computing Symposium 1977, 4.-7.04.1977 in Liege, Amsterdam - New York - Oxford 1977, S. 197-207.

Cox,L. (1978)

Cox,L.A.: Predicting Concurrent Computer System Performance Using Petri-Net Models; in: o.V.: Proceedings of the ACM Annual National Conference, 4.-6.12.1978, Washington 1978, S. 901-906.

Crespi-Reghizzi (1975)

Crespi-Reghizzi,S.; Mandrioli,D.: A Decidability Theorem for a Class of Vector-Addition Systems; in: Information Processing Letters, Vol. 3 (1975), S. 78-80.

Crespi-Reghizzi (1977)

Crespi-Reghizzi,S.; Mabdrili,D.: Petri Nets and Szilard Languages; in: Information and Control, Vol. 33 (1977), S. 177-192.

Cunningham,W. (1976)

Cunningham,W.H.: A Network Simplex Method; in: Mathematical Programming, Vol. 11 (1976), S. 105-116.

Czaja (1980)

Czaja,L.: Deadlock and Fairness in Parallel Schemas: A Set-Theoretic Characterization and Decision Problems; in: Information Processing Letters, Vol. 10 (1980), S. 234-239.

Dadda (1976c)

Dadda,L.: Programming methods for simulated Petri networks as control tool; in: Rivista di informatica, Vol. VI (1976), No. 1/2, S. 49-64.

Dantzig (1966)

Dantzig,G.B.: Lineare Programmierung und Erweiterungen, Berlin - Heidelberg - New York 1966.

Dantzig (1979)

Dantzig,G.B.: Comments on Khachiyna's Algorithm for Linear Programming, Technical Report SOL 79-22, Systems Development Laboratory, Stanford University, Stanford 1979.

Davis,R. (1981)

Davis,R.; Smith,R.G.: Negotiation as a Metaphor for Distributed Problem Solving, A.I. Memo No. 624, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1981.

Davis,R. (1982a)

Davis,R.; Lenat,D.B.: Knowledge-Based Systems in Artificial Intelligence, New York - St. Louis - San Francisco ... 1982.

Davis,R. (1983)

Davis,R.; Smith,R.G.: Negotiation as a Metaphor for Distributed Problem Solving; in: Artificial Intelligence, Vol. 20 (1983), S. 63-109. (Anmk. des Verf.: geringfügig überarbeitete Version von Davis,R. (1981).)

de Champeaux (1975)

de Champeaux,D.; Sint,L.: An Improved Bi-Directional Heuristic Search Algorithm; in: o.V.: Advance Papers of the Fourth International Joint Conference on Artificial Intelligence (IJCAI-75), 03.-08.09.1975 in Tbilisi, o.O. (Cambridge/Massachusetts) 1975, Vol. 1, S. 309-314.

De Cindio (1982)

De Cindio,F.; De Michelis,G.; Pomello,L.; Simone,S.: Superposed Automata Nets; in: Girault,C.; Reisig,W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 269-279.

De Cindio (1983a)

De Cindio,F.; De Michelis,G.; Pomello,L.; Simone,C.: Equivalence Notions for Concurrent Systems; in: Pagnoni,A.; Rozenberg,G. (Hrsg.): Applications and Theory of Petri Nets, Selected Papers from the 3rd European Workshop on Applications and Theory of Petri Nets, 27.-30.09.1982 in Varenna, Informatik-Fachberichte 66, Berlin - Heidelberg - New York - Tokyo 1983, S. 29-39.

Dederichs (1989b)

Dederichs,F.; Weber,R.: Safety and Liveness from a Methodological Point of View, Bericht MIP-8918, Fakultät für Mathematik und Informatik, Universität Passau, Passau 1989.

Deering (1984)

Deering,M.F.: Hardware and Software Architectures for Efficient AI; in: o.V.: Proceedings of the National Conference on Artificial Intelligence, AAAI-84, 06.-10.08.1984 in Austin, Los Altos 1983 (Anmk. des Verf.: Druckfehler in "1983"; gemeint ist wohl: "1984"), S. 73-78.

de Kleer (1984)

de Kleer, J.: Choices Without Backtracking; in: o.V.: Proceedings of the National Conference on Artificial Intelligence, AAAI-84, 06.10.1984 in Austin, Los Altos 1983, S. 79-85. (Anmk. des Verf.: Druckfehler in "1983"; gemeint ist wohl: "1984".)

de Kleer (1986a)

de Kleer, J.: An Assumption-based TMS; in: Artificial Intelligence, Vol. 28 (1986), S. 127-162.

de Kleer (1986b)

de Kleer, J.: Extending the ATMS; in: Artificial Intelligence, Vol. 28 (1986), S. 163-196.

de Kleer (1986c)

de Kleer, J.: Problem Solving with the ATMS; in: Artificial Intelligence, Vol. 28 (1986), S. 197-224.

de Kleer (1986d)

de Kleer, J.; Williams, B.C.: Back to Backtracking: Controlling the ATMS; in: proceedings aaai-86, fifth national conference on artificial intelligence, 11.-15.08.1986 in Philadelphia, Los Altos 1986, Vol. 2, S. 910-917.

Delfmann (1989a)

Delfmann, W.: Die Planung "robuster" Distributionsstrukturen bei Ungewißheit über die Nachfrageentwicklung im Zeitablauf; in: Hax, H.; Kern, W.; Schröder, H.-H. (Hrsg.): Zeitaspekte betriebswirtschaftlicher Theorie und Praxis, 50. Wissenschaftliche Jahrestagung des Verbandes der Hochschullehrer für Betriebswirtschaft e.V., 24.-28.05.1988 in Köln, Stuttgart 1989, S. 215-229.

Delgrande (1988)

Delgrande, J.P.: An Approach to Default Reasoning Based on a First-Order Conditional Logic: Revised Report; in: Artificial Intelligence, Vol. 36 (1988), S. 63-90.

Dennis, J. (1974b)

Dennis, J.B.; Misunas, D.P.: A Computer Architecture for Highly Parallel Signal Processing; in: o.V.: 1974 National Computer Conference, 06.-10.05. 1974 in Chicago, AFIPS Conference Proceedings, Vol. 43, Montvale 1974, S. 402-409.

Dennis, J. (1975)

Dennis, J.B.; Misunas, D.P.: A Preliminary Architecture for a Basis Data Flow Processor; in: o.V.: Proceedings of the 2nd International Symposium on Computer Architecture, 20.-22.01.1975 in Houston, New York 1975, S. 126-132.

Dennis, J. (1980)

Dennis, J.B.: Data Flow Supercomputers; in: Computer, Vol. 13 (1980), S. 48-56.

Derigs (1985)

Derigs, U.: Neuere Ansätze in der Linearen Programmierung - Motivation, Konzepte und Verfahren; in: Streitferdt, L.; Hauptmann, H.; Marusev, A.W.; Ohse, D.; Pape, U. (Hrsg.): Operations Research Proceedings 1985, DGOR - Vorträge der 14. Jahrestagung, Berlin - Heidelberg - New York ... 1985, S. 47-58.

Devillers (1978)

Devillers, R.E.; Lauer, P.E.: A General Mechanism for Avoiding Starvation with Distributed Control; in: Information Processing Letters, Vol. 7 (1978), S. 156-158.

Devy (1979)

Devy, M.; Diaz, M.: Multilevel Description and Validation of the Control in Communication Systems; in: o.V.: Proceedings of the 1st International Conference on Distributed Computing Systems, 1.-5.10.1979 in Huntsville, New York 1979, S. 43-50.

Diel (1984)

Diel, H.: Concurrent Data Access Architecture; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 373-382.

Dijkstra (1972)

Dijkstra, E.W.: Hierarchical Ordering of Sequential Processes; in: Hoare, C.A.R.; Perrott, R.H. (Hrsg.): Operating Systems Techniques, Proceedings of a Seminar, 30.08.-03.09.1971 in Belfast, London - New York 1972, S. 72-93.

Dilger (1979)

Dilger, W.: Theorembeweiser als Bestandteil der Problemlösungskomponente; in: Kolvenbach, M.; Lötscher, A.; Lutz, H.D. (Hrsg.): Künstliche Intelligenz und natürliche Sprache - Sprachverstehen und Problemlösen mit dem Computer, Bd. 42 der Forschungsberichte des Instituts für deutsche Sprache in Mannheim, Tübingen 1979, S. 265-288.

Dilger (1983)

Dilger, W.; Janson, A.: Unifikationsgraphen für intelligentes Backtracking in Deduktionssystemen; in: Neumann, B. (Hrsg.): GWAI-83, 7th German Workshop on Artificial Intelligence, 19.-23.09.1983 in Dassel/Solling, Informatik-Fachberichte 76, Berlin - Heidelberg - New York ... 1983, S. 189-196.

Dinkelbach (1982)

Dinkelbach, W.: Entscheidungsmodelle, Berlin - New York 1982.

Diruf (1983)

Diruf, G.: Strategisch-logistische Müllentsorgungs-Planung mit einem lernorientierten Modellsystem; in: Bühler, W.; Fleischmann, B.; Schuster, K.P.; Streitferdt, L.; Zander, H. (Hrsg.): Operations Research Proceedings 1982, DGOR - Vorträge der 11. Jahrestagung, 22.-24.09.1982 in Frankfurt, Berlin - Heidelberg - New York 1983, S. 237-247.

Diruf (1984)

Diruf, G.: Modell- und computergestützte Gestaltung physischer Distributionssysteme; in: Albach, H. (Schriftleitung): Unternehmensführung und Logistik, Ergänzungsheft 2/84 zu: Zeitschrift für Betriebswirtschaft, Wiesbaden 1984, S. 114-130.

Dittrich, G. (1989b)

Dittrich, G.; Evertz-Jägers, B.: Der Kanal-Instanz-Netz Editor KINED - Ein Tool zur Unterstützung einer methodischen Systemmodellierung mit Hilfe von hierarchisch dargestellten Kanal-Instanz-Netzen, Forschungsbericht Nr. 308, Fachbereich Informatik, Universität Dortmund, Dortmund o.J. (1989).

Döpp (1983a)

Döpp, K.: Zum Hackschen Wohlformungssatz für Free-Choice-Petrinetze; in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 19 (1983), S. 3-15.

Dörfler (1973)

Dörfler, W.; Mühlbacher, J.: Graphentheorie für Informatiker, Berlin - New York 1973.

Dörner, D. (1979)

Dörner, D.: Problemlösen als Informationsverarbeitung, 2. Aufl., Stuttgart - Berlin - Köln ... 1979.

Dörnhöfer (1991)

Dörnhöfer, K.: Graphische Modellierung betriebswirtschaftlicher Strukturen; in: Biethahn, J.; Hummeltenberg, W.; Schmidt, B. (Hrsg.): Simulation als betriebliche Entscheidungshilfe, Band 2, Berlin - Heidelberg - New York ... 1991, S. 125-136.

Döttling (1981)

Döttling, W.: Flexible Fertigungssysteme - Steuerung und Überwachung des Fertigungsablaufs, Berlin - Heidelberg - New York 1981.

Domschke (1990)

Domschke, W.; Drexler, A.: Einführung in Operations Research, Berlin - Heidelberg - New York ... 1990.

Doyle (1977a)

Doyle, J.: Truth Maintenance Systems for Problem Solving, Master of Science-Thesis und Technical Report No. 419, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1977.

Doyle (1977b)

Doyle, J.: Truth Maintenance Systems for Problem Solving; in: o.V.: 5th International Joint Conference on Artificial Intelligence, IJCAI-77, Proceedings of the Conference, 22.-25.08.1977 in Cambridge (Massachusetts), o.O. (Pittsburgh) 1977, Vol. 1, S. 247.

Doyle (1979a)

Doyle, J.: A Glimpse of Truth Maintenance; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 232-237. (Anmk. des Verf.: inhaltlich nicht identisch mit Doyle (1979b).)

Doyle (1979b)

Doyle, J.: A Glimpse of Truth Maintenance; in: Winston, P.H.; Brown, R.H. (Hrsg.): Artificial Intelligence: An MIT Perspective, Vol. 1: Expert Problem Solving - Natural Language Understanding - Intelligent Computer Coaches - Representation and Learning, Cambridge (Massachusetts) - London 1979, S. 117-135. (Anmk. des Verf.: inhaltlich nicht identisch mit Doyle (1979a).)

Doyle (1980)

Doyle, J.; London, P.: A Selected Descriptor-Indexed Bibliography to the Literature on Belief Revision; in: SIGART Newsletter, No. 71 (1980), S. 7-23.

Doyle (1983)

Doyle, J.: The Ins and Outs of Reason Maintenance; in: Bundy, A. (Hrsg.): IJCAI-83, Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Los Altos 1983, Vol. 1, S. 349-351.

Doyle (1985)

Doyle, J.: Reasoned Assumptions and Pareto Optimality; in: o.V.: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-85, 19.-23.08.1985 in Los Angeles, o.O. (Los Altos) 1985, Vol. 1, S. 87-90.

Dressler (1987)

Dressler, O.: Erweiterungen des Basic ATMS; in: Morik, K. (Hrsg.): GWAI-87, 11th German Workshop on Artificial Intelligence, 28.09.-02.10.1987 in Geseke, Proceedings, Informatik-Fachberichte 152, Berlin - Heidelberg - New York ... 1987, S. 185-194.

Dressler (1988a)

Dressler, O.: Assumption-based Truth Maintenance; in: Stoyan, H. (Hrsg.): Begründungsverwaltung - Beiträge zu einem Workshop über Reason Maintenance, am 09.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 162, Berlin - Heidelberg - New York ... 1988, S. 63-85.

Dressler (1989a)

Dressler, O.; Farquhar, A.: Problem Solver Control Over the ATMS; in: Metzger, D. (Hrsg.): GWAI-89, 13th German Workshop on Artificial Intelligence, 18.-22.09.1989 in Eringerfeld, Proceedings, Informatik-Fachberichte 216, Berlin - Heidelberg - New York ... 1989, S. 17-26.

Dressler (1989b)

Dressler, O.; Freitag, H.: Truth Maintenance Systeme; in: Künstliche Intelligenz: Forschung, Entwicklung, Erfahrungen., 3. Jg. (1989), Heft 2, S. 13-19.

Dycke (1988)

Dycke, A.: Zur Ableitung von Managementregeln aus kontrolltheoretischen Unternehmensmodellen, Dissertation, Universität Passau, Bergisch Gladbach - Köln 1988.

Ebert, J. (1981)

Ebert, J.; Perl, J.: Reachability Homomorphisms on Nets; in: Noltemeier, H. (Hrsg.): Graphtheoretic Concepts in Computer Science, Proceedings of the International Workshop WG 80, 15.-18.06.1980 in Bad Honnef, Lecture Notes in Computer Science 100, Berlin - Heidelberg - New York 1981, S. 326-334.

Ecker (1977)

Ecker, K.: Organisation von parallelen Prozessen - Theorie deterministischer Schedules, Mannheim - Wien - Zürich 1977.

Eggert (1978)

Eggert, H.: Eine Anwendung von Petri-Netzen für eine partielle Prozeßbeschreibung und deren Abbildung auf Echtzeitelemente von PEARL (Die Automatisierung eines technischen Prozesses zur Bierherstellung mit einer Prozeßdatenverarbeitungsanlage), Dissertation, Technische Universität Berlin 1978, zugleich: Bericht KfK-PDV 166, Kernforschungszentrum Karlsruhe GmbH, Karlsruhe 1978.

Eiselt (1977)

Eiselt,H.A.; v. Frajer,H.: Operations Research Handbook - Standard Algorithms and Methods with Examples, Berlin - New York 1977.

Eisenführ (1978)

Eisenführ,F.: Die Wissenschaft vom vernünftigen Handeln; in: Die Betriebswirtschaft, 38. Jg. (1978), S. 435-448.

Eisinger (1980)

Eisinger,N.; Siekmann,J.H.; Smolka,G.; Unvericht,E.; Walther,C.: Das Karlsruher Beweissystem; in: Wilhelm,R. (Hrsg.): GI - 10. Jahrestagung, 30.09.-02.10.1980 in Saarbrücken, Informatik-Fachberichte 33, Berlin - Heidelberg - New York 1980, S. 400-412.

Ellinger (1990a)

Ellinger,T.: Operations Research - Eine Einführung, 3. Aufl., Berlin - Heidelberg - New York ... 1990.

Ellis,C.A. (1977)

Ellis,C.A.: Consistency and Correctness of Duplicate Database Systems; in: o.V.: Proceedings of Sixth ACM Symposium on Operating System Principles, im November 1977, New York 1977, S. 67-84.

Ellis,C.J. (1979)

Ellis,C.J.: The Design and Evaluation of Algorithms for Parallel Processing, Dissertation an der University of Washington, Washington 1979.

Erickson (1989)

Erickson,H.H.; Mulvey,J.M.; Vanmarcke,E.H.: Integrating Expert Systems and Mathematical Programming: An Example in Infrastructure Management; in: Glover,F.; Greenberg,H.J. (Hrsg.): Linkages with Artificial Intelligence, zugleich: Annals of Operations Research, Vol. 21 (1989), Basel 1989, S. 275-299.

Eschenbacher (1991)

Eschenbacher,P.: Formulierung transactions-orientierter Modelle mit der systemtheoretischen Beschreibungssprache SIMPLEX-MDL; in: Biethahn,J.; Hummeltenberg,W.; Schmidt,B. (Hrsg.): Simulation als betriebliche Entscheidungshilfe, Band 2, Berlin - Heidelberg - New York ... 1991, S. 221-235.

Eversheim (1981)

Eversheim,W.: Organisation in der Produktionstechnik, Bd. 4: Fertigung und Montage, Düsseldorf 1981.

Fahlman (1974)

Fahlman,S.E.: A Planning System for Robot Construction Tasks; in: Artificial Intelligence, Vol. 5 (1974), S. 1-49.

Fahrion (1989)

Fahrion,R.: Wirtschaftsinformatik - Grundlagen und Anwendungen, Heidelberg 1989.

Fandel (1972)

Fandel,G.: Optimale Entscheidung bei mehrfacher Zielsetzung, Lecture Notes in Economics and Mathematical Systems 76, Berlin - Heidelberg - New York 1972.

Fandel (1991a)

Fandel,G.: Produktion I - Produktions- und Kostentheorie, 3. Aufl., Berlin - Heidelberg - New York ... 1991.

Fandel (1991b)

Fandel,G.: Bedeutung der ökonomischen Theorie für das betriebswirtschaftliche Gebiet der Produktion; in: Ordelleide,D.; Rudolph,B.; Büselmann,E. (Hrsg.): Betriebswirtschaftslehre und Ökonomische Theorie, 52. Wissenschaftliche Jahrestagung des Verbandes der Hochschullehrer für Betriebswirtschaft e.V., 05.-09.06.1990 in Frankfurt, Stuttgart 1991, S. 227-254.

Fantao (1972)

Fantao,R.O.: A Concurrent Algorithm For Avoiding Deadlocks in Multiprocess Multiple Resource Systems; in: Operating Systems Review - ACM - SIGOPS, Vol. 6 (1972), S. 72-79.

Favrel (1987)

Favrel, J.; Oh, G.R.; Baptiste, P.; Lee-Kwang, H.: Hierarchical Control Structure of FMS by Petri Net Method; in: Yoshikawa, H.; Burbidge, J.L. (Hrsg.): New Technologies for Production Management Systems, Proceedings of the IFIP TC 5/WG 5.7 Working Conference on New Technologies for Production Management Systems, 01.-03.10.1986 in Tokyo, Amsterdam - New York - Oxford ... 1987, S. 271-279.

Fehling (1990a)

Fehling, R.: Hierarchische Petrinetze - Idee und grundlegende Struktur, Forschungsbericht Nr. 344, Fachbereich Informatik, Universität Dortmund, Dortmund 1990.

Fehling (1990b)

Fehling, R.; Zelewski, S.: Schriftwechsel zur Thematik "Schaltverhalten von Petrilab", Dortmund - Köln 1990.

Feichtinger (1982a)

Feichtinger, G.: Anwendungen des Maximumprinzips im Operations Research, Teil 1; in: Operations Research-Spektrum, Bd. 4 (1982), S. 171-190.

Feigenbaum (1963)

Feigenbaum, E.A.; Feldman, J. (Hrsg.): Computers and Thought, New York - San Francisco - Toronto ... 1963.

Ferstl (1979)

Ferstl, O.K.: Konstruktion und Analyse von Simulationsmodellen, Dissertation, Universität Regensburg 1979, Königstein 1979.

Fidelak (1986a)

Fidelak, M.: Wissensdarstellung und -verarbeitung auf der Basis von Petri-Netzen, Diplomarbeit am Fachbereich Informatik/Universität Bonn, Bonn 1986. (Auch veröffentlicht als: Arbeitspapiere der GMD Nr. 225, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1986.)

Fidelak (1986b)

Fidelak, M.: Petri-Netze - Eine formale Sprache zur Wissensrepräsentation; in: Rundbrief des Fachausschusses 1.2 Künstliche Intelligenz & Mustererkennung in der Gesellschaft für Informatik, Nr. 43 (1986), S. 32-38.

Fidelak (1988b)

Fidelak, M.; Lischka, C.; Voß, H.: Repräsentation der Dynamik technisch-physikalischer Systeme; in: Hoschka, P. (Hrsg.): Forschungsgruppe Expertensysteme - Aus der Arbeit der Forschungsgruppe Expertensysteme, Arbeitspapiere der GMD 337, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1988, 5. Beitrag.

Finkel (1982)

Finkel, A.; Memmi, G.: FIFO Nets: A New Model of Parallel Computation; in: Cremers, A.B.; Kriegel, H.P. (Hrsg.): Theoretical Computer Science, 6th GI-Conference, 5.-7.01.1983 in Dortmund, Lecture Notes in Computer Science 145, Berlin - Heidelberg - New York 1982, S. 111-121.

Finkel (1987b)

Finkel, A.: A Generalization of the Procedure of Karp and Miller to Well Structured Transition Systems; in: Ottmann, T. (Hrsg.): Automata, Languages and Programming, 14th International Colloquium, 13.-17.07.1987 in Karlsruhe, Proceedings, Lecture Notes in Computer Science 267, Berlin - Heidelberg - New York ... 1987, S. 499-508.

Fisher, M. (1982)

Fisher, M.L.: Worst-Case Analysis of Heuristic Algorithms for Scheduling and Picking; in: Dempster, M.A.H.; Lenstra, J.K.; Rinnooy Kan, A.H.G. (Hrsg.): Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6.-17.07.1981 in Durham, Dordrecht - Boston - London 1982, S. 15-34.

Fleig (1991)

Fleig, J.: Optimale Reihenfolgeplanung - mit Bonner Segen; in: Arbeitsvorbereitung, 28. Jg. (1991), S. 97-98.

Florin (1982)

Florin,G.; Natkin,S.: Evaluation Based upon Stochastic Petri Nets of the Maximum Throughput of a Full Duplex Protocol; in: Girault,C.; Reisig,W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 280-288.

Forgy (1984)

Forgy,C.; Gupta,A.; Newell,A.; Wedig,R.: Initial Assessment of Architectures for Production Systems; in: o.V.: Proceedings of the National Conference on Artificial Intelligence, AAAI-84, 06.-10.08.1984 in Austin, Los Altos 1983 (Anmk. des Verf.: Druckfehler in "1983"; gemeint ist wohl: "1984"), S. 116-120.

Forrer (1990)

Forrer,M.G.: Expertenwissen an den Maschinenbediener weitergeben - Graphikfähiges Expertensystem und Bedienerführung; in: Zeitschrift für wirtschaftliche Fertigung und Automatisierung, 85. Jg. (1990), S. CA111-CA114.

Fox,B. (1987a)

Fox,B.R.: The Implementation of Opportunistic Scheduling; in: Hertzberger,L.O.; Groen,F.C.A. (Hrsg.): Intelligent Autonomous Systems, An International Conference, 8.-11.12.1986 in Amsterdam, Amsterdam - New York - Oxford - Tokyo 1987, S. 231-240.

Franke,G. (1988)

Franke,G.; Hax,H.: Finanzwirtschaft des Unternehmens und Kapitalmarkt, Berlin - Heidelberg - New York ... 1988.

Fraunhofer-Institut für Systemtechnik (1982)

Fraunhofer-Institut für Systemtechnik und Innovationsforschung (ISI); Institut für Arbeitsmarkt- und Berufsforschung (IAB); Institut für Werkzeugmaschinen- und Fertigungstechnik (IWF): Der Einsatz flexibler Fertigungssysteme - Technische, einführungsorganisatorische, wirtschaftliche und arbeitsplatzbezogene Aspekte, Forschungsbericht KfK-PFT 41, Kernforschungszentrum Karlsruhe GmbH, Karlsruhe 1982.

Freedman (1988b)

Freedman,P.; Malowany,A.: SAGE: A Decision Support System for the Sequencing of Operations within a Robotic Workcell; in: Decision Support Systems, Vol. 4 (1988), S. 329-343.

Freitag (1988a)

Freitag,H.; Reinfrank,M.: A Non-Monotonic Deduction System Based on (A)TMS, TEX-B Memo 29-88, Frankfurt - Karlsruhe - Sankt Augustin ... 1988.

Frenkel (1986)

Frenkel,K.A.: Piecing Together Complexity; in: Communications of the ACM, Vol. 29 (1986), S. 110-111.

Frese (1991)

Frese,E.: Organisationstheorie - Stand und Aussagen aus betriebswirtschaftlicher Sicht, Wiesbaden 1991.

Früchtenicht (1984)

Früchtenicht,H.-W.; Heger,D.; Steusloff,H.: Wege zur Verbesserung von Effizienz und Zugang bei wissensverarbeitenden Systemen; in: FhG-Berichte, o.Jg. (1984), Nr. 2, S. 51-55.

Fuchi (1983)

Fuchi,K.: The Direction the FGCS Project Will Take; in: New Generation Computing, Vol. 1 (1983), S. 3-9.

Gabbay (1980)

Gabbay,D.; Pnueli,A.; Shelah,S.; Stavi,J.: On the Temporal Analysis of Fairness; in: o.V.: Conference Record of 7th ACM Symposium on Principles of Programming Languages, im Januar 1980, New York 1980, S. 163-173.

Gacs (1981)

Gacs,P.; Lovasz,L.: Khachiyan's Algorithm for Linear Programming; in: Mathematical Programming Studies, Vol. 14 (1981), S. 61-68.

Garey (1978)

Garey, M.R.; Johnson, D.S.: "Strong" NP-Completeness Results: Motivation, Examples, and Implications; in: *Journal of the Association for Computing Machinery*, Vol. 25 (1978), S. 499-508.

Garey (1979)

Garey, M.R.; Johnson, D.S.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*, San Francisco 1979.

Garratt (1990)

Garratt, P.W.: Multiplan: a 'what if' planning tool for system development projects; in: Malouppas, A. (Hrsg.): *Management Technology - control tools for the 90s*, London 1990, S. 26-38.

Gaschnig (1977)

Gaschnig, J.: Exactly How Good Are Heuristics?: Toward a Realistic Predictive Theory of Best-First Search; in: o.V.: *5th International Joint Conference on Artificial Intelligence, IJCAI-77, Proceedings of the Conference, 22.-25.08.1977 in Cambridge (Massachusetts), o.O. (Pittsburgh) 1977*, Vol. 1, S. 434-441.

Genrich (1971b)

Genrich, H.J.: Das Zollstationenproblem, *Interne Mitteilung 71-10-13*, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1971.

Genrich (1973a)

Genrich, H.J.; Lautenbach, K.: Synchronisationsgraphen; in: *Acta Informatica*, Vol. 2 (1973), S. 143-161.

Genrich (1978a)

Genrich, H.J.; Lautenbach, K.: Facts in Place/Transition Nets; in: Winkowski, J. (Hrsg.): *Mathematical Foundation of Computer Science 1978, Proceedings on the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64*, Berlin - Heidelberg - New York 1978, S. 213-231.

Genrich (1980b)

Genrich, H.J.; Stankiewicz-Wiechno, E.: A Dictionary of Some Basic Notions of Net Theory; in: Brauer, W. (Hrsg.): *Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84*, Berlin - Heidelberg - New York 1980, S. 519-531.

Geoffrion (1980)

Geoffrion, A.M.; Powers, R.F.: Facility Location Analysis is Just the Right Beginning (If You Do it Right); in: *Interfaces*, Vol. 10 (1980), No. 2, S. 22-30.

Georgeff (1981)

Georgeff, M.P.: Search Methods Using Heuristic Strategies; in: Drinan, A. (Hrsg.): *Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981*, Vol. 1, S. 563-568.

Gevarter (1982)

Gevarter, W.B.: An Overview of Expert Systems; in: o.V.: *Proceedings of the International Conference on Cybernetics and Society, 28.-30.10.1982 in Seattle, New York 1982*, S. 156-160.

Ghosh, S. (1983)

Ghosh, S.; Chaudhuri, A.: Solution to puzzle No. 3 by Piotr Chrzystowaski-Wachtel published in Newsletter 13, February 1983. (p. 22); in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 15 (1983), S. 22.

Giloi (1978)

Giloi, W.K.: Grundlagen, Operationsprinzipien und Strukturen von innovativen Rechnerarchitekturen; in: Schindler, S.; Giloi, W.K. (Hrsg.): *GI - 8. Jahrestagung, 1978 in Berlin, Informatik-Fachberichte 16*, Berlin - Heidelberg - New York 1978, S. 274-307.

Giloi (1981)

Giloi, W.K.: *Rechnerarchitektur*, Berlin - Heidelberg - New York 1981.

Giloi (1982)

Giloi, W.K.: *Rechnerarchitektur - heute und morgen*; in: Nehmer, J. (Hrsg.): *GI- 12. Jahrestagung, 05.-07.10.1982 in Kaiserslautern, Proceedings, Informatik-Fachberichte 57*, Berlin - Heidelberg - New York 1982, S. 1-29.

Giloi (1984)

Giloi, W.K.: Die Entwicklung der Rechnerarchitektur von der von-Neumann-Maschine bis zu den Rechnern der "fünften Generation"; in: Elektronische Rechenanlagen, 26. Jg. (1984), S. 55-70.

Giordana (1985)

Giordana, A.; Saitta, L.: Modeling Production Rules by Means of Predicate Transition Networks; in: Information Sciences, Vol. 35 (1985), S. 1-41.

Girault (1980)

Girault, C.: Proof of Protocols in the Case of Failures, Institut de Programmation, Universite Pierre et Marie Curie Paris, Paris 1980.

Girault (1982c)

Girault, C.: Exemple de Modelisation par Reseaux de Petri Coherence des Accés a un Fichier, D.E.A. de Systemes Informatiques, Institut de Programmation, Universite Pierre et Marie Curie, Paris 1982.

Glover (1985)

Glover, F.; Klingman, D.D.; Phillips, N.V.; Schneider, R.F.: New Polynomial Shortest Path Algorithms and their Computational Attributes; in: Management Science, Vol. 31 (1985), S. 1106-1128.

Glover (1990)

Glover, F.; Klingman, D.; Phillips, N.: Netform Modeling and Applications; in: Interfaces, Vol. 20 (1990), No. 4, S. 7-27.

Godbersen (1978)

Godbersen, H.P.; Meyer, B.E.: Function Nets as a Tool for the Simulation of Information Systems, Interner CIS-Bericht 3/78, Institut für Angewandte Informatik, Technische Universität Berlin, Berlin 1978. (Erschienen auch in: o.V.: Proceedings of the Summer Computer Simulation Conference, 24.-26.07.1978 in Newport Beach, o.O. o.J., S. 46-53.)

Godbersen (1979)

Godbersen, H.P.: Funktionsnetze - Ein Ansatz zur Beschreibung, Analyse und Simulation von sozio-technischen Systemen; in: Mayr, H.C.; Meyer, B.E. (Hrsg.): Formale Modelle für Informationssysteme, GI-Fachtagung, 24.-26.05.1979 in Tutzing, Berlin - Heidelberg - New York 1979, S. 246-265.

Godin (1978)

Godin, V.B.: Interactive Scheduling: Historical Survey and State of the Art; in: AIIE Transactions, Vol. 10 (1978), S. 331-337.

Götze, U. (1991)

Götze, U.: Szenario-Technik in der strategischen Unternehmensplanung, Dissertation, Universität Göttingen 1990, Wiesbaden 1991.

Goldberg (1984a)

Goldberg, A.; Pohl, I.: Is Complexity Theory of Use to AI?; in: Elithorn, A.; Banerji, R. (Hrsg.): Artificial and Human Intelligence, Edited Review Papers presented at the International NATO Symposium on Artificial and Human Intelligence, im Oktober 1981 in Lyon, Amsterdam - New York - Oxford 1984, S. 43-55.

Goldfarb (1979)

Goldfarb, D.; Sit, W.Y.: Worst Case Behavior of the Steepest Edge Simplex Method; in: Discrete Applied Mathematics, Vol. 1 (1979), S. 277-285.

Goldfarb (1990)

Goldfarb, D.; Hao, J.; Kai, S.-R.: Efficient Shortest Path Simplex Algorithms; in: Operations Research, Vol. 38 (1990), S. 624-628.

Gomez, P. (1978)

Gomez, P.: Die kybernetische Gestaltung des Operations Managements - Eine Systemmethodik zur Entwicklung anpassungsfähiger Organisationsstrukturen, Bern - Stuttgart 1978.

Gonzalez, T. (1978)

Gonzalez, T.: Preemptive Scheduling of Uniform Processor Systems; in: Journal of the Association for Computing Machinery, Vol. 25 (1978), S. 92-101.

Goodwin (1982)

Goodwin, J.W.: An Improved Algorithm for Non-monotonic Dependency Net Update, Research Report LITH-MAT-R-82-23, Software Systems Research Center, Linköping Institute of Technology, Linköping 1982.

Gosh (1977)

Gosh, S.: Structured Petri Nets, Bericht Nr. 49/77, Abteilung Informatik, Universität Dortmund, Dortmund 1977.

Gostelow (1971)

Gostelow, K.P.: Flow of Control, Resource Allocation, and the Proper Termination of Programs, Dissertation an der University of California, Los Angeles 1971.

Gostelow (1972)

Gostelow, K.; Cerf, V.G.; Volansky, S.: Proper Termination of Flow-of-Control in Programs Involving Concurrent Processes; in: SIGPLAN-Notices, No. 7 ((1972), S. 15-27.

Gostelow (1975)

Gostelow, K.P.: Computation Modules and Petri Nets; in: o.V.: Proceedings of the 3rd IEEE-ACM Milwaukee Symposium on Automatic Computation and Control, New York 1975, S. 345-353.

Gottschalk, W. (1977)

Gottschalk, W.: Petri-Netze in der Eisenbahnsignaltechnik; in: Siemens-Zeitschrift, 51. Jg. (1977), S. 876-879.

Grabowski, J. (1978)

Grabowski, J.: On the Analysis of Switching Circuits by Means of Petri Nets; in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 14 (1978), S. 611-617.

Grabowski, J. (1980b)

Grabowski, J.: The Decidability of Persistence for Vector Addition Systems; in: Information Processing Letters, Vol. 11 (1980), S. 20-23.

Grabowski, J. (1980c)

Grabowski, J.: Lineare Methoden in der Theorie der Vektoradditionssysteme (III), Bericht PSF 1297, Sektion Mathematik, Humboldt-Universität Berlin (Ost), Berlin (Ost) 1980.

Grabowski, J. (1981b)

Grabowski, J.: An Algorithm to Identify Slices, with Applications to Vector Replacement Systems; in: Gecseg, F. (Hrsg.): Fundamentals of Computation Theory, Proceedings of the 1981 International FCT-Conference, 24.-28.08.1981 in Szeged, Lecture Notes in Computer Science 117, Berlin - Heidelberg - New York 1981, S. 425-432.

Graham, N. (1983)

Graham, N.: Künstliche Intelligenz - Wie Sie Ihren Computer zum Denken bringen, Sprendlingen 1983.

Griese, W. (1979)

Griese, W.: Lebendigkeit in NSK-Petri-Netzen, TUM-Info-7906, Institut für Informatik, Technische Universität München, München 1979.

Griese, W. (1980)

Griese, W.: Liveness in NSC-Petri Nets; in: Pape, U. (Hrsg.): Discrete Structures and Algorithms, Proceedings of the Workshop WG 79, 5th Conference on Graphtheoretic Concepts in Computer Science, 21.-23.06.1979 in Berlin, München - Wien 1980, S. 255-264.

Grill (1982)

Grill, E.: Design-Review by Petri-Nets; in: Girault, C.; Reisig, W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 38-44.

Grötschel (1981)

Grötschel, M.; Lovasz, L.; Schrijver, A.: The Ellipsoid Method and its Consequences in Combinatorial Optimization; in: Combinatorica, Vol. 1 (1981), S. 169-197.

Grzesik (1987)

Grzesik,R.: Strategische Gesamtplanung - Ein Modellansatz auf der Basis der Erfahrungskurvenhypothese, Dissertation, Universität Köln, Köln 1987.

Günther,H. (1986)

Günther,H.O.: The Design of an Hierarchical Model for Production Planning and Scheduling; in: Axsäter,S., Schneeweiss,C.; Silver,E. (Hrsg.): Multi-Stage Production Planning and Inventory Control, Lecture Notes in Economics and Mathematical Systems 266, Berlin - Heidelberg - New York - Tokyo 1986, S. 227-260.

Gupta,S. (1968)

Gupta,S.K.; Rosenhead,J.: Robustness in Sequential Investment Decisions; in: Management Science, Vol. 15 (1968), S. B-18 - B-29.

Haag (1988)

Haag,A.; Zetzsche,F.; Zinser,G.: Die Behandlung von Alternativen in der Planung: Erfahrungen mit ATMS-basierten Expertensystemarchitekturen; in: Hertzberg,J.; Günter,A. (Hrsg.): Beiträge zum 2. Workshop Planen und Konfigurieren, Arbeitspapiere der GMD 310, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1988, S. 113-132.

Haas (1987)

Haas,P.J.; Shedler,G.S.: Stochastic Petri Nets with Simultaneous Transition Firings; in: o.V.: International Workshop on Petri Nets and Performance Models, PNPM87, 24.-26.08.1987 in Madison, Washington 1987, S. 24-32.

Habel (1983)

Habel,C.: Logische Systeme und Repräsentationsprobleme; in: Neumann,B. (Hrsg.): GWAI-83, 7th German Workshop on Artificial Intelligence, 19.-23.09.1983 in Dassel/Solling, Informatik-Fachberichte 76, Berlin - Heidelberg - New York ... 1983, S. 118-142.

Habermann (1969)

Habermann,A.N.: Prevention of System Deadlocks; in: Communications of the ACM, Vol. 12 (1969), S. 373-477 u. 385.

Hack,M. (1972)

Hack,M.H.: Analysis of Production Schemata be Petri Nets, Master of Science Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Technical Report TR-94, Project MAC, Cambridge (Massachusetts) 1972.

Hack,M. (1974a)

Hack,M.: Extended State-Machine Allocatable Nets (ESMA), an extension of free choice Petri net results, Computation Structures Group Memo 78-1, Project MAC, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1974.

Hack,M. (1974b)

Hack,M.: The Recursive Equivalence of the Reachability Problem And the Liveness Problem For Petri Nets And Vector Addition Systems; in: o.V.: Proceedings of the 15th Annual Symposium on Switching and Automata Theory, 14.-16.10.1974, New York 1974, S. 156-164.

Hack,M. (1975a)

Hack,M.H.T.: Decidability Questions For Petri Nets, Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1975.

Hack,M. (1975b)

Hack,M.: Petri Net Languages, Computation Structures Group Memo 124, Project MAC, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1975.

Hackmann (1981)

Hackmann,o.Vn. (W.K.); Buchholz,o.Vn.: Petrinetze zur Kommunikation; in: Online, 19. Jg. (1981), Heft 5, S. 372.

Hackmann (1982)

Hackmann,W.K.: Systementwicklung mit Petrinetzen PET/PEM/PES: Ein Überblick, Bericht ZT ZTI SDF 1 der Siemens AG, München 1982.

Hämmäinen (1987)

Hämmäinen,H.; Eloranta,E.; Milin,A.-M.: Intelligent Mail - Tool for Decentralization of Production Management; in: Yoshikawa,H.; Burbidge,J.L. (Hrsg.): New Technologies for Production Management Systems, Proceedings of the IFIP TC 5/WG 5./ Working Conference on New Technologies for Production Management Systems, 01.-03.10.1986 in Tokyo, Amsterdam - New York - Oxford ... 1987, S. 283-294.

Hässig (1979)

Hässig,K.: Graphentheoretische Methoden des Operations Research, Stuttgart 1979.

Häussler (1976)

Häussler,A.: Polynomial beschränkte nichtdeterministische Turingmaschinen und die Vollständigkeit des aussagenlogischen Erfüllungsproblems; in: Specker,E.; Strassen,V. (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 20-35.

Hagstrom (1988)

Hagstrom,J.N.: Computational Complexity of PERT Problems; in: Networks, Vol. 18 (1988), S. 139-147.

Hahn,D. (1990)

Hahn,D.; Laßmann,G.: Produktionswirtschaft - Controlling industrieller Produktion, Band 1: Grundlagen, Führung und Organisation, Produkte und Produktprogramm, Material und Dienstleistungen, 2. Aufl., Heidelberg 1990.

Hall,P. (1971)

Hall,P.A.V.: Branch-and-Bound and Beyond; in: o.V.: Second International Joint Conference on Artificial Intelligence (IJCAI-71), 01.-03.09.1971 in London, Portsmouth 1971, S. 641-650.

Hamacher,H. (1980a)

Hamacher,H.: Maximal Algebraic Flows: Algorithms and Examples; in: Pape,U. (Hrsg.): Discrete Structures and Algorithms, Proceedings of the Workshop WG 79, 5th Conference on Graphtheoretic Concepts in Computer Science, 21.-23.06.1979 in Berlin, München - Wien 1980, S. 153-166.

Hamacher,H. (1980b)

Hamacher,H.: Algebraic Flows in Regular Matroids; in: Discrete Applied Mathematics, Vol. 2 (1980), S. 27-38.

Han (1978a)

Han,Y.W.: Performance Evaluation of a Digital System Using a Petri Net-Like Approach; in: Tranter,W.H. (Hrsg.): Proceedings of the National Electronics Conference, Vol. 32, 16.-18.10.1978 in Chicago, Oak Brook 1978, S. 166-172.

Han (1979)

Han,Y.-W.: Petri Nets for Distributed Digital System Modeling and Evaluation; in: o.V.: Proceedings of the 12th Hawaii International Conference on System Sciences, North Hollywood 1979, S. 270-279.

Hansen,P. (1980)

Hansen,P.: An $O(m \log D)$ Algorithm for Shortest Paths; in: Discrete Applied Mathematics, Vol. 2 (1980), S. 151-153.

Hanssmann (1978a)

Hanssmann,F.: Einführung in die Systemforschung - Methodik der modellgestützten Entscheidungsvorbereitung, München - Wien 1978. (Anmk. des Verf.: partiell über Hanssmann (1987) hinausgehend.)

Hanssmann (1987)

Hanssmann,F.: Einführung in die Systemforschung - Methodik der modellgestützten Entscheidungsvorbereitung, 3. Aufl., München 1987.

Hanssmann (1989)

Hanssmann,F.: Robuste Planung; in: Szyperski,N.; Winand,U. (Hrsg.): Handwörterbuch der Planung, Stuttgart 1989, Sp. 1758-1764.

Hanssmann (1990)

Hanssmann,F.: Quantitative Betriebswirtschaftslehre - Lehrbuch der modellgestützten Unternehmensplanung, 3. Aufl., München - Wien 1990.

Haralick (1979)

Haralick,R.M.; Elliott,G.L.: Increasing Tree Search Efficiency for Constraint Satisfaction Problems; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 356-364.

Hart,P. (1968)

Hart,P.E.; Nilsson,N.J.; Raphael,B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths; in: IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-4 (1968), S. 100-107.

Hart,P. (1982)

Hart,P.E.: Directions for AI in the Eighties; in: SIGART Newsletter, No. 79 (1982), S. 11-16.

Hartley (1984)

Hartley,J.: FMS at work, Kempston/Bedford - Amsterdam - New York 1984.

Hartmanis (1976)

Hartmanis,J.; Berman,L.: On Isomorphisms and Density of NP and Other Complete Sets; in: o.V.: Conference Record of The Eighth Annual ACM Symposium on Theory of Computing, 03.-05.05.1976 in Hershey, New York 1976, S. 30-40.

Hastings (1978)

Hastings,N.A.J.; Mello,J.M.C.: Decision Networks, Chichester - New York - Brisbane ... 1978.

Hauk (1973)

Hauk,W.: Einplanung von Produktionsaufträgen nach Prioritätsregeln - Eine Untersuchung von Prioritätsregeln mit Hilfe der Simulation, Berlin - Köln - Frankfurt 1973.

Haun (1990)

Haun,P.: Einsatz von Arbeitsplatzrechnern zur betriebswirtschaftlichen Planung und Entscheidungsunterstützung; in: Kostenrechnungspraxis, o.Jg. (1990), Sonderheft 1/90, S. 14-20.

Havender (1968)

Havender,J.W.: Avoiding deadlock in multitasking systems; in: IBM Systems Journal, Vol. 7 (1968), S. 74-84.

Hax,H. (1972)

Hax,H.; Laux,H.: Flexible Planung - Verfahrensregeln und Entscheidungsmodelle für die Planung bei Ungewißheit; in: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, 24. Jg. (1972), S. 318-340.

Hax,H. (1974)

Hax,H.: Entscheidungsmodelle in der Unternehmung - Einführung in Operations Research, Reinbek 1974.

Hax,H. (1985)

Hax,H.: Investitionstheorie, 5. Aufl., Würzburg - Wien 1985.

Heibey (1977)

Heibey,H.W.; Lutterbeck,B.; Töpel,M.: Auswirkungen der elektronischen Datenverarbeitung in Organisationen, Forschungsbericht DV 77-01 Datenverarbeitung, Institut für Informatik der Universität Hamburg, Eggenstein-Leopoldshafen 1977.

Heimerdinger (1978)

Heimerdinger,W.L.: A Petri Net Approach to System Level Fault Tolerance Analysis; in: Tranter,W.H. (Hrsg.): Proceedings of the National Electronics Conference, Vol. 32, 16.-18.10.1978 in Chicago, Oak Brook 1978, S. 161-165.

Heinemann (1980)

Heinemann,B.: Teilklassen der selbst-modifizierenden Netze, Bericht Nr. 69 FBI-HH-B-69/80, Fachbereich Informatik, Universität Hamburg, Hamburg 1980.

Heinhold (1989)

Heinhold,M.: Simultane Unternehmensplanungsmodelle - ein Irrweg?; in: Die Betriebswirtschaft, 49. Jg. (1989), S. 689-708.

Helberg (1987)

Helberg,P.: PPS als CIM-Baustein - Gestaltung der Produktionsplanung und -steuerung für die computerintegrierte Produktion, Berlin 1987.

Herden (1990)

Herden,W.; Hein,H.-W. (Hrsg.): Kurzlexikon Wissensbasierte Systeme, München - Wien 1990.

Herman (1985)

Herman,T.; Chandy,K.M.: On Distributed Search; in: Information Processing Letters, Vol. 21 (1985), S. 129-133.

Hermes (1961)

Hermes,H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit - Einführung in die Theorie der rekursiven Funktionen, Berlin - Göttingen - Heidelberg 1961.

Herrtwich (1989b)

Herrtwich,R.G.; Hommel,G.: Kooperation und Konkurrenz - Nebenläufige, verteilte und echtzeitabhängige Programmsysteme, Berlin - Heidelberg - New York ... 1989.

Hertz (1964)

Hertz,D.B.: Risk Analysis in Capital Investment; in: Harvard Business Review, Vol. 42 (1964), No. 1, S. 95-106.

Hertzberg (1985)

Hertzberg,J.: Über Künstliche Intelligenz und die reale Welt, KI-Bericht Nr. 2, Institut für Informatik, Universität Bonn, Bonn 1985.

Herzog,O. (1973)

Herzog,O.: Petrinetze als Modelle für parallele Prozesse, im Auftrag der Firma ORGALOGIC/Köln erstellter Bericht, o.O. 1973.

Herzog,O. (1976a)

Herzog,O.: Kontrollstruktur-Netze: eine Darstellung paralleler Programme zur Deadlock-Analyse; in: Noltemeier,H. (Hrsg.): Graphen, Algorithmen, Datenstrukturen - Ergebnisse des Workshops WG 76, 2. Fachtagung über Graphentheoretische Konzepte der Informatik, 16.-18.06.1976 in Göttingen, München - Wien 1976, S. 297-314.

Herzog,O. (1976b)

Herzog,O.; Yoeli,M.: Control Nets for Asynchronous Systems, Part I, Technical Report No. 74, Department of Computer Science, Technion - Israel Institute of Technology, Haifa 1976.

Herzog,O. (1976c)

Herzog,O.: Zur Analyse der Kontrollstruktur von parallelen Programmen mit Hilfe von Petri-Netzen, Dissertation an der Universität Dortmund, Bericht Nr. 24/76, Abteilung Informatik, Universität Dortmund, Dortmund 1976.

Herzog,O. (1977)

Herzog,O.: Automatic Deadlock Analysis of Parallel Programs; in: Morlet,E.; Ribbens,D. (Hrsg.): Proceedings of the International Computing Symposium 1977, 4.-7.04.1977 in Liege, Amsterdam - New York - Oxford 1977, S. 209-216.

Herzog,O. (1979)

Herzog,O.: Static Analysis of Concurrent Processes for Dynamic Properties Using Petri Nets; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 66-90.

Hespos (1965)

Hespos,R.F.; Strassman,P.A.: Stochastic Decision Trees for the Analysis of Investment Decisions; in: Management Science, Vol. 11 (1965), S. B-244 - B-259.

Heß-Kinzer (1979)

Heß-Kinzer,D.: Termingrobplanung; in: Kern,W. (Hrsg.): Handwörterbuch der Produktionswirtschaft, Stuttgart 1979, Sp. 1979-1992.

Hillis (1986b)

Hillis,W.D.: The Connection Machine, Cambridge (Massachusetts) - London 1986.

Hilty (1987)

Hilty,L.: Konzepte zur Unterstützung der Modellbildung und Simulation - am Beispiel der Modellbildung auf dem Umweltsektor; in: Jaeschke,A.; Page,B. (Hrsg.): Kolloquium "Informatikanwendungen im Umweltbereich", 30.09.-01.10.1986 in Karlsruhe, Bericht KfK 4223, Kernforschungszentrum Karlsruhe GmbH, Karlsruhe 1987, S. 213-234.

Hintz (1987)

Hintz,G.-W.: Ein wissensbasiertes System zur Produktionsplanung und -steuerung für flexible Fertigungssysteme, Dissertation, Universität Aachen, Düsseldorf 1987.

Holt,A. (1970a)

Holt,A.; Commoner,F.: Events and Conditions; in: Dennis,J.B. (Hrsg.): Record of the Project MAC, Conference on Concurrent Systems and Parallel Computation, 2.-5.06.1970 in Woods Hole, New York 1970, S. 3-52.

Holt,A. (1970b)

Holt,A.W.; Commoner,F.: Events and Conditions: An Approach to the Description and Analysis of Dynamic Systems, Information System Theory Project, Research Report, Applied Data Research, Inc., Wakefield (auch: Princeton) 1970.

Holt,A. (1975d)

Holt,A.W.: Communication Mechanics; in: Massachusetts Computer Associates, Inc. (Hrsg.): Second Semi-Annual Technical Report (1.12.1973-31.03.1975) for the Project "Development of Theoretical Foundations for Description and Analysis of Discrete Information Systems", Wakefield 1975, S. 156-176.

Holt,R. (1971a)

Holt,R.C.: On Deadlocks in Computer Systems, Dissertation, Cornell University, Ithaca 1971.

Holt,R. (1971b)

Holt,R.C.: Comments on Prevention of Deadlocks; in: Communications of the ACM, Vol. 14 (1971), S. 36-38.

Holt,R. (1972b)

Holt,R.C.: Some Deadlock Properties of Computer Systems; in: Operating Systems Review - ACM - SIGOPS, Vol. 6 (1972), S. 64-71.

Holzmann (1987)

Holzmann,G.J.: On Limits and Possibilities of Automated Protocol Analysis; in: Rudin,H.; West,C.H. (Hrsg.): Protocol Specification, Testing, and Verification, VII, Proceedings of the IFIP WG 6.1 International Workshop on Protocol Specification, Testing, and Verification, 05.-08.05.1987 in Zürich, Amsterdam 1987, S. 339-344.

Holzmann (1991)

Holzmann,G.J.: Design and Validation of Computer Protocols, London - Sydney - Toronto ... 1991.

Hooker (1986)

Hooker,J.N.: Karmarkar's Linear Programming Algorithm; in: Interfaces, Vol. 16 (1986), No. 4, S. 75-90.

Hopcroft (1974)

Hopcroft,J.E.: Complexity of Computer Computations; in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of the IFIP Congress 1974, 05.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 620-626.

Hormann,D. (1973)

Hormann,D.: Betrieb rechnergesteuerter Fertigungssysteme, Dissertation, Technische Hochschule Aachen, Aachen 1973.

Horn,W. (1984)

Horn,W.: Methoden der Artificial Intelligence; in: Retti,J.; Bibel,W.; Buchberger,B.; Buchberger,E.; Horn,W.; Kobsa,A.; Steinacker,I.; Trappl,R.; Trost,H.: Artificial Intelligence - Eine Einführung, Stuttgart 1984, S. 27-45.

Howell (1987c)

Howell,R.R.; Rosier,L.E.: Recent Results on the Complexity of Problems Related to Petri Nets; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1987, Lecture Notes in Computer Science 266, Berlin - Heidelberg - New York ... 1987, S. 45-72.

Hruschka (1980a)

Hruschka,P.; Kappatsch,A.: Formale Semantik von Realzeitsprachen mit Netz-attribuierten Grammatiken; in: Elektronische Rechenanlagen, 22. Jg. (1980), S. 267-275.

Huber,A. (1986)

Huber,A.: Wissensbasierte Echtzeit-Steuerung in CIM; in: CIM Management, 2. Jg. (1986), Nr. 4, S. 94-97.

Huber,O. (1983)

Huber,O.: Zur Psychologie des Problemlösens; in: Schauer,H.; Tauber,H.J. (Hrsg.): Psychologie des Programmierens, München - Wien 1983, S. 59-78.

Huen (1975)

Huen,W.H.; Siewiorek,D.P.: Intermodule Protocol for Register Transfer Level Modules: Representation and Analytic Tools; in: o.V.: Symposium on Computer Architecture 2, 20.-22.01.1975 in Houston, zugleich: Computer Architecture News 3/4, New York 1975, S. 56-62.

Huet (1978)

Huet,G.: An Algorithm to Generate the Basis of Solutions to Homogeneous Linear Diophantine Equations; in: Information Processing Letters, Vol. 7 (1978), S. 144-147.

Hummel,S. (1970)

Hummel,S.: Wirklichkeitsnahe Kostenerfassung - Neue Erkenntnisse für eine eindeutige Kostenermittlung, Berlin 1970.

Hura (1981b)

Hura,G.S.; Singh,H.; Nanda,N.K.: A Petri net approach to the evaluation of the complexity of a program; in: International Journal of Electronics, Vol. 51 (1981), S. 79-85.

Hura (1982c)

Hura,G.S.: Petri Net as a Modeling Tool; in: Microelectronics and Reliability, Vol. 22 (1982), S. 433-439.

Hura (1982d)

Hura,G.S.: Enumeration of Success Paths in a Graph; in: Microelectronics and Reliability, Vol. 22 (1982), S. 1033-1034.

Hwang,S. (1984)

Hwang,S.-L.; Barfield,W.; Chang,T.-C.; Salvendy,G.: Integration of humans and computers in the operation and control of flexible manufacturing systems; in: International Journal of Production Research, Vol. 22 (1984), S. 841-856.

Ibaraki (1987)

Ibaraki,T.: Enumerative Approaches to Combinatorial Optimization - Part I (S. 1-342) und Part II (S. 343-602), Basel 1987 (zugleich: Annals of Operations Research, Vol. 10 (1987), No. 1-4: Part I / Vol. 11 (1987), No. 1-4: Part II.)

Imai (1979)

Imai,M.; Yoshida,Y.; Fukumura,T.: A Parallel Searching Scheme for Multiprocessor Systems and its Application to Combinatorial Problems; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 416-418.

IntelliCorp (1989)

IntelliCorp GmbH: Knowledge Engineering Environment - Eine Einführung, München 1989.

Itzinger (1976)

Itzinger,O.: Methoden der maschinellen Intelligenz, München - Wien 1976.

Jaffe (1977)

Jaffe,J.M.: Semilinear Sets and Applications, Master of Science Thesis am Massachusetts Institute of Technology, Cambridge (Massachusetts) 1977.

Janich (1974)

Janich,P.; Kambartel,F.; Mittelstraß,J.: Wissenschaftstheorie als Wissenschaftskritik, Frankfurt 1974.

Janicki (1978)

Janicki,R.: Synthesis of Concurrent Schemes; in: Winkowski,J. (Hrsg.): Mathematical Foundations of Computer Science 1978, Proceedings of the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64, Berlin - Heidelberg - New York 1978, S. 298-307.

Janicki (1981a)

Janicki,R.: On the Design of Concurrent Systems; in: o.V.: Proceedings of the 2nd International Conference on Distributed Computing Systems, 8-10.04.1981 in Paris, Piscataway 1981, S. 455-466.

Jantzen (1980a)

Jantzen,M.; Valk,R.: Formal Properties of Place/Transition Nets; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 165-212.

Jantzen (1983a)

Jantzen,M.: The Large Markings Problem; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 14 (1983), S. 24-25.

Johnson,D. (1981)

Johnson,D.S.: The NP-Completeness Column: An Ongoing Guide; in: Journal of Algorithms, Vol. 2 (1981), S. 393-405.

Johnson,D. (1982)

Johnson,D.T.; Schbert,L.K.: A Planning Control Strategy that Allows for the Cost of Planning; in: Trappl,R. (Hrsg.): Cybernetics and Systems Research, Proceedings of the Sixth European Meeting on Cybernetics and Systems Research, 13.-16.04.1982 in Wien, Amsterdam - New York - Oxford 1982, S. 965-971.

Johnson,R.R. (1972)

Johnson,R.R.: Some Steps Toward An Information System Performance Theory; in: o.V.: Proceedings of the 1st USA-Japan Computer Conference, Montvale 1972, S. 161-172.

Johnsonbaugh (1979)

Johnsonbaugh,R.; Kao,M.C.; Murata,T.: Additional Transformations of Live and Safe Marked Graphs; in: o.V.: Proceedings of 17th Annual Allerton Conference on Communication, Control, and Computing, 1979 in Monticello, Urbana 1979, S. 387-396.

Jones,N. (1977)

Jones,N.D.; Landweber,L.H.; Lien,Y.E.: Complexity of some problems in Petri Nets; in: Theoretical Computer Science, Vol. 4 (1977), S. 277-299.

Jump (1973)

Jump,J.R.; Thiagarajan,P.S.: On the Equivalence of Asynchronous Control Structures; in: SIAM Journal of Computing, Vol. 2 (1973), S. 67-87.

Jump (1975)

Jump,J.R.; Thiagarajan,P.S.: On the Interconnection of Asynchronous Control Structures; in: Journal of the Association for Computing Machinery, Vol. 22 (1975), S. 596-612.

Kamien (1981)

Kamien,M.J.; Schwartz,N.L.: Dynamic Optimization - The Calculus of Variations and Optimal Control in Economics and Management, New York - Oxford 1981.

Kanal (1981)

Kanal,L.; Kupar,V.: Branch & Bound Formulation for Sequential and Parallel Game Tree Searching: Preliminary Results; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 569-571.

Karmarkar,N. (1984)

Karmarkar,N.: A New Polynomial-Time Algorithm for Linear Programming; in: Combinatorica, Vol. 4 (1984), S. 373-395.

Karp (1967b)

Karp,R.M.; Miller,R.E.: Parallel Program Schemata: A Mathematical Model for Parallel Computation; in: o.V.: Symposium on 8th Annual Switching and Automata Theory. Conference Record, im Oktober 1967 in Austin, o.O. (New York) 1967, S. 55-61.

Karp (1969)

Karp,R.M.; Miller,R.E.: Parallel Program Schemata; in: Journal of Computer and System Sciences, Vol. 3 (1969), S. 147-195.

Karp (1972)

Karp,R.M.: Reducibility Among Combinatorial Problems; in: Miller,R.E.; Thatcher,J.W. (Hrsg.): Complexity of Computer Computations, Proceedings of a Symposium on the Complexity of Computer Computations, 20.-22.03.1972 in New York, New York - London 1972, S. 85-103.

Karp (1975a)

Karp,R.M.: The Fast Approximate Solution of Hard Combinatorial Problems; in: o.V.: Proceedings of the 6th South-Eastern Conference on Combinatorics, Graph Theory, and Computing, Winnipeg 1975, S. 15-31.

Karp (1975b)

Karp,R.M.: On the Computational Complexity of Combinatorial Problems; in: Networks, Vol. 5 (1975), S. 45-68.

Karp (1976)

Karp,R.M.: The Probabilistic Analysis of Some Combinatorial Search Algorithms; in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 1-19.

Karp (1985)

Karp,R.M.; Lenstra,J.K.; McDiarmid,C.J.H.; Rinnoy Kan,A.H.G.: Probabilistic Analysis; in: O'hEigeartaigh,M.; Lenstra,J.K.; Rinnooy Kan,A.H.G. (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane ... 1985, S. 52-88.

Karp (1986)

Karp,R.M.: Combinatorics, Complexity, and Randomness; in: Communications of the ACM, Vol. 29 (1986), S. 98-109.

Kauffmann (1987)

Kauffmann,H.; Grumbach,A.: MULTILOG: MULTIPLE worlds in LOGIC Programming; in: du Boulay,B.; Hogg,D.; Steels,L. (Hrsg.): Advances in Artificial Intelligence - II, Seventh European Conference on Artificial Intelligence, ECAI-86, 20.-25.07.1986 in Brighton, Amsterdam - New York - Oxford ... 1987, S. 233-247.

Kazmaier (1984)

Kazmaier,E.: Ablaufplanung im Dialog - Alternative oder Ergänzung zur Optimierung; in: Steckhan,H.; Bühler,W.; Jäger,K.E.; Schneeweiß,C.; Schwarze,J. (Hrsg.): Operations Research Proceedings 1983, DGOR - Papers of the 12th Annual Meeting, Berlin - Heidelberg - New York - Tokyo 1984, S. 163-168.

Keller,R. (1972a)

Keller,R.M.: Vector Replacement Systems: A Formalism for Modeling Asynchronous Systems, Technical Report 117, Department of Electrical Engineering, Computer Sciences Laboratories, Princeton University, Princeton 1972.

Keller,R. (1975a)

Keller,R.M.: Generalized Petri Nets as Models for System Verification, Technical Report 202, Department of Electrical Engineering, Computer Science Laboratory, Princeton University, Princeton 1975.

Keller,R. (1976)

Keller,R.M.: Formal Verification of Parallel Programs; in: Communications of the ACM, Vol. 19 (1976), S. 371-384.

Keramidis (1979)

Keramidis,S.; Grote,W.: Beiträge zur Lösung des Verklemmungsproblems in prioritätsfreien Betriebsmittelmaschinen und Petri-Netzen, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik), Universität Erlangen-Nürnberg, Bd. 12, Nr. 9, Erlangen 1979.

Keramidis (1980a)

Keramidis,S.: Ein Kalkül zur Konstruktion deadlockfreier Systeme; in: Pape,U. (Hrsg.): Discrete Structures and Algorithms, Proceedings of the Workshop WG 79, 5th Conference on Graphtheoretic Concepts in Computer Science, 21.-23.06.1979 in Berlin, München - Wien 1980, S. 265-268.

Kern,W. (1974)

Kern,W.: Investitionsrechnung, Stuttgart 1974.

Kern,W. (1977)

Kern,W.; Schröder,H.-H.: Forschung und Entwicklung in der Unternehmung, Reinbek 1977.

Kern,W. (1987)

Kern,W.: Operations Research - Einführung und Überblick, 6. Aufl., Stuttgart 1987.

Kern,W. (1990a)

Kern,W.: Industrielle Produktionswirtschaft, 4. Aufl., Stuttgart 1990.

Khachiyan (1979)

Khachiyan,L.G.: A Polynomial Algorithm in Linear Programming; in: Soviet Mathematics Doklady, Vol. 20 (1979), No. 1, S. 191-194. (Anmk. des Verf.: Die Schreibweise "Khachiyan" lehnt sich an die überwiegend übliche Transskription des kyrillischen Originals an. In der o.a. Quelle wird die seltenere Transskription "Hacijan" verwendet.)

Kießler (o.J.)

Kießler,G.: Petrinetz-Entwurfstechnologie, Bestandteil der Technologie komplexer Systeme (Basisinformation), AP-Bericht Nr. 34, Siemens AG, o.O. (München) o.J.

Kindervater (1985)

Kindervater,G.A.P.; Lenstra,J.K.: Parallel Algorithms; in: O'hEigeartaigh,M.; Lenstra,J.K.; Rinnooy Kan,A.H.G. (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane ... 1985, S. 106-128.

Kirsch (1977b)

Kirsch,W.: Zweiter Band - Informationsverarbeitungstheorie des Entscheidungsverhaltens; in: Kirsch,W.: Einführung in die Theorie der Entscheidungsprozesse, 2. Aufl. der Bände I bis III, Habilitationsschrift (unter dem Titel "Entscheidungen und Entscheidungsprämissen in der Unternehmungsorganisation"), Universität München, Wiesbaden 1977.

Kistner (1981c)

Kistner,K.-P.: Einflüsse der Unternehmensforschung auf die betriebswirtschaftliche Theorie; in: Brockhoff,K.; Krelle,W. (Hrsg.): Unternehmensplanung - Referate eines Kolloquiums an der Fernuniversität Hagen, 11.-12.07.1981 in Hagen, Berlin - Heidelberg - New York 1981, S. 63-80.

Klee,H. (1989)

Klee,H.W.: Zur Akzeptanz von Expertensystemen - Eine empirische Analyse der Relevanz und Angemessenheit der Erklärungskomponente, Dissertation, Universität Köln, Bergisch Gladbach - Köln 1989.

Klee,V. (1972)

Klee,V.; Minty,G.J.: How Good Is the Simplex Algorithm?; in: Shisha,O. (Hrsg.): Symposium on Inequalities III, New York 1972, S. 159-175.

Kleene (1952)

Kleene,S.C.: Introduction to Metamathematics, Amsterdam - Groningen 1952.

Kleiner,F. (1991)

Kleiner,F.: Kostenrechnung bei flexibler Automatisierung, Dissertation, Universität Stuttgart, München 1991.

Knobloch (1990)

Knobloch,T.: Simultane Anpassung der Produktion, Dissertation, Universität Köln 1989, Wiesbaden 1990.

Knödel (1969)

Knödel,W.: Graphentheoretische Methoden und ihre Anwendungen, Berlin - Heidelberg - New York 1969.

Knoop (1986)

Knoop,J.: Online-Kostenrechnung für die CIM-Planung - Prozeßorientierte Kostenrechnung zur Ablaufplanung flexibler Fertigungssysteme, Berlin 1986.

Knoop (1987)

Knoop,J.: Prozeßorientierte Kostenrechnung - Ein Instrument zur Planung flexibler Fertigungssysteme; in: Kostenrechnungspraxis, 31. Jg. (1987), Heft 2, S. 47-58.

Koch,M. (1990)

Koch,M.; Loseries,F.; Tran,T.: Integration von Animations- und Simulationswerkzeugen in den Design-Prozeß; in: Reuter,A. (Hrsg.): GI - 20. Jahrestagung II, Informatik auf dem Weg zum Anwender, 08.-12.10.1990 in Stuttgart, Proceedings, Informatik-Fachberichte 258, Berlin - Heidelberg - New York ... 1990, S. 549-558.

Kochan,D. (1986)

Kochan,D. (Hrsg. u. Autor); Merchant,o.Vn.; Kozar,o.Vn.; Schaller,J.; Hutchinson,G.K.; Olling,o.Vn.; Semenov,o.Vn.; Klimov,W.; Spur,G.; Krause,F.L.; Pistorius,E.; Crestin,J.P. (Koauto-ren): CAM Developments in Computer-Integrated Manufacturing, Berlin - Heidelberg - New York - Tokyo 1986.

Kohler,W. (1976)

Kohler,W.H.; Steiglitz,K.: Enumerative and Iterative Computational Approaches; in: Coffman, E.G. (Hrsg.) Computer and Job-Shop Scheduling Theory, New York - London - Sydney - Toronto 1976, S. 229-287.

Kolata (1984)

Kolata,G.: A Fast Way to Solve Hard Problems; in: Science, Vol. 225 (1984), S. 1379-1380.

Konolige (1980)

Konolige,K.; Nilsson,N.J.: Multiple-Agent Planning Systems; in: o.V.: Proceedings of the First Annual National Conference on Artificial Intelligence, AAAI-80, 18.-21.08.1980 in Stanford, o.O. (Menlo Park) 1980, S. 138-142.

Kornfeld (1981a)

Kornfeld,W.A.: The Use of Parallelism to Implement a Heuristic Search; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 575-580.

Kosaraju (1982)

Kosaraju,S.R.: Decidability of Reachability in Vector Addition Systems; in: o.V.: Conference Record of the 14th Annual ACM symposium on Theory of Computing, o.O. (New York) 1982, S. 267-281.

Kotov (1978)

Kotov,V.E.: An Algebra for Parallelism Based on Petri Nets; in: Winkowski,J. (Hrsg.): Mathematical Foundations of Computer Science 1978, Proceedings of the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64, Berlin - Heidelberg - New York 1978, S. 39-55.

Kowalski (1979a)

Kowalski,R.: Algorithm = Logic + Control; in: Communications of the ACM, Vol. 22 (1979), S. 424-436.

Kowalski (1983a)

Kowalski,R.: Logic for Problem Solving, 2. und zugleich 3. Aufl., New York - Amsterdam - Oxford 1983.

Krämer (1981)

Krämer,B.; Schmidt,H.W.: Der Entwurf nebenläufiger Systeme mit Handlungs-Entscheidungs-Netzen; in: Brauer,W. (Hrsg.): GI - 11. Jahrestagung in Verbindung mit: Third Conference of the European Co-operation in Informatics (ECI), Proceedings, 20.-23.10.1981 in München, Informatik-Fachberichte 50, Berlin - Heidelberg -New York 1981, S. 460-471.

Krallmann (1989b)

Krallmann,H.; Siebert,V.: Der CIM-Leitstand - Integration wissensbasierter Komponenten in das Konzept der elektronischen Fertigungssteuerung; in: o.V.: Mit Technologie die Zukunft bewältigen - Software im Maschinen- und Anlagenbau - Fallbeispiele für das Management, Tagung, 01.-02.06.1989 in Frankfurt, Band 9, Frankfurt 1989, 9. Beitrag.

Krause,F. (1985)

Krause,F.-L.: Veränderung der Konstruktionstätigkeit durch CAD-Systeme; in: Zeitschrift für wirtschaftliche Fertigung, 80. Jg. (1985), S. 60-66.

Kreimeier (1988)

Kreimeier,D.: Neue Systeme für werkstatorientierte Programmierverfahren; in: Werkstattstechnik, 78. Jg. (1988), S. 393-396.

Krelle (1968)

Krelle,W.: Präferenz- und Entscheidungstheorie, Tübingen 1968.

Kreowski (1981)

Kreowski,H.-J.: A Comparison Between Petri-Nets and Graph Grammars; in: Noltmeier,H. (Hrsg.): Graphtheoretic Concepts in Computer Science, Proceedings of the International Workshop WG 80, 15.-18.06.1980 in Bad Honnef, Lecture Notes in Computer Science 100, Berlin - Heidelberg - New York 1981, S. 306-317.

Krieg,B. (1977)

Krieg,B.: Petrinetze und Zustandsgraphen, Bericht Nr. 29, Institut für Informatik, Universität Hamburg, Hamburg 1977.

Kruschwitz (1990)

Kruschwitz,L.: Investitionsrechnung, 4. Aufl., Berlin - New York 1990.

Krycha (1972)

Krycha,K.-T.: Methoden der Ablaufplanung, Frankfurt - Zürich 1972.

Küpper,Wi. (1975)

Küpper,Wi.; Lüder,K.; Streitferdt,L.: Netzplantechnik, Würzburg - Wien 1975.

Kwan (1977a)

Kwan,C.L.; Michel,C.; Le Beux,P.: Logical Systems Design USING PLAs and Petri Nets - Programmable Hardwired Systems; in: Gilchrist,B. (Hrsg.): Information Processing 77, Proceedings of IFIP Congress 77, 8.-12.08.1977 in Toronto, Amsterdam - New York - Oxford 1977, S. 607-611.

Kwong (1977)

Kwong,Y.S.: On Reduction of Asynchronous Systems; in: Theoretical Computer Science, Vol. 5 (1977), S. 25-50.

Kwong (1979)

Kwong,Y.S.: On the Absence of Livelocks in Parallel Programs; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 172-190.

Lambert,J. (1987)

Lambert,J.L.: Consequences of the Decidability of the Reachability Problem for Petri Nets; in: o.V.: Papers presented at the 8th European Workshop on Application and Theory of Petri Nets, 24.-26.06.1987 in Zaragoza, o.O. 1987, S. 451-470.

Landweber (1978)

Landweber,L.H.; Robertson,E.L.: Properties of Conflict-Free and Persistent Petri Nets; in: Journal of the Association for Computing Machinery, Vol. 25 (1978), S. 352-364.

Langley (1984)

Langley,P.; Carbonell,J.G.: Approaches to Machine Learning; in: Journal of the American Society for Information Science, Vol. 35 (1984), S. 306-316.

Laßmann (1968)

Laßmann,G.: Die Kosten- und Erlösrechnung als Instrument der Planung und Kontrolle in Industriebetrieben, Düsseldorf 1968.

Laßmann (1980b)

Laßmann,G.: Neue Aufgaben der Kosten- und Erlösrechnung aus der Sicht der Unternehmensführung; in: Hahn,D. (Hrsg.): Führungsprobleme industrieller Unternehmungen, Festschrift für Friedrich Thomee zum 60. Geburtstag, Berlin - New York 1980, S. 327-347.

Latombe (1979)

Latombe,J.-C.: Failure Processing in a System for Designing Complex Assemblies; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 508-515.

Lauer (1975a)

Lauer,P.E.; Campbell,R.H.: Formal Semantics of a Class of High-Level Primitives for Coordinating Concurrent Processes; in: Acta Informatica, Vol. 5 (1975), S. 297-332.

Lauer (1975b)

Lauer,P.E.; Campbell,R.H.: A Description of Path Expressions by Petri Nets; in: o.V.: Conference Record of the Second ACM Symposium on Principles of Programming Languages, Papers Presented at the Symposium, 20.-22.01.1975 in Palo Alto, Palo Alto 1975, S. 95-105.

Lauer (1978a)

Lauer,P.E.; Best,E.; Shields,M.W.: On the Problem of Achieving Adequacy of Concurrent Programs; in: Neuhold,E.J. (Hrsg.): Formal Description of Programming Concepts, Proceedings of the IFIP Working Conference on Formal Description of Programming Concepts, 1.-5.08.1977 in Saint Andrews, Amsterdam 1978, S. 301-336.

Lauer (1979)

Lauer,P.E.; Torrigiani,P.R.; Shields,M.W.: COSY - A System Specification Language Based on Paths and Processes; in: Acta Informatica, Vol. 12 (1979), S. 109-158.

Lauer (1980)

Lauer,P.E.; Shields,M.W.; Best,E.: Design and analysis of highly parallel and distributed systems; in: Bjorner,D. (Hrsg.): Abstract Software Specifications, 1979 Copenhagen Winter School, Proceedings, 22.01.-2.02.1979 in Kopenhagen, Lecture Notes in Computer Science 86, Berlin - Heidelberg - New York 1980, S. 451-503.

Laurent (1984)

Laurent,J.-P.: Control structures in expert systems; in: Technology and Science of Informatics, Vol. 3 (1984), No. 3, S. 147-162.

Lautenbach (1973)

Lautenbach,K.: Exakte Bedingungen der Lebendigkeit für eine Klasse von Petri-Netzen, Dissertation Bonn 1973, Bericht Nr. 82, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1973.

Lautenbach (1974a)

Lautenbach,K.; Schmid,H.A.: Use of Petri Nets for Proving Correctness of Concurrent Process Systems; in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of the IFIP Congress 1974, 5.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 187-191.

Lautenbach (1975a)

Lautenbach,K.: Liveness in Petri Nets, Interner Bericht ISF-75-02.1, Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1975.

Lautenbach (1979a)

Lautenbach,K.: Wegsysteme in Petri-Netzen; in: Petri,C.A. (Hrsg.): Ansätze zur Organisationstheorie Rechnergestützter Informationssysteme, Bericht Nr. 111, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, München - Wien 1979, S. 93-106.

Lautenbach (1985b)

Lautenbach,K.: On Logical and Linear Dependencies, Arbeitspapiere der GMD 147, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1985.

Lautenbach (1987a)

Lautenbach, K.: Linear Algebraic Techniques for Place/Transition Nets; in: Brauer, W.; Reisig, W.; Rozenberg, G. (Hrsg.): Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, 8.-19.09.1986 in Bad Honnef, Lecture Notes in Computer Science 254, Berlin - Heidelberg - New York ... 1987, S. 142-167.

Laux (1971)

Laux, H.: Flexible Investitionsplanung - Einführung in die Theorie der sequentiellen Entscheidungen bei Unsicherheit, Habilitationsschrift, Universität des Saarlandes 1971, Opladen 1971.

Laux (1991)

Laux, H.: Entscheidungstheorie I - Grundlagen, 2. Aufl., Berlin - Heidelberg - New York ... 1991.

Lawler (1976)

Lawler, E.L.: Sequencing to Minimize the Weighted Number of Tardy Jobs; in: Revue Francaise d'Automatique, Informatique, Recherche Operationelle - Informatique, Supplement au Vol. 10 (1976), No. 5, S. 27-33.

Lenat (1982)

Lenat, D.B.: The Nature of Heuristics; in: Artificial Intelligence, Vol. 19 (1982), S. 189-249.

Lenstra (1977)

Lenstra, J.K.; Rinnooy Kan, A.H.G.; Brucker, P.: Complexity of Machine Scheduling Problems; in: Hammer, P.L.; Johnson, E.L.; Korte, B.H.; Nemhauser, G.L. (Hrsg.): Annals of Discrete Mathematics, Vol. 1 (1977), Studies in Integer Programming, Amsterdam - New York - Oxford 1977, S. 343-362.

Lenstra (1979)

Lenstra, J.K.; Rinnooy Kan, A.H.G.: Computational Complexity of Discrete Optimization Problems; in: Hammer, P.L.; Johnson, E.L.; Korte, B.H. (Hrsg.): Discrete Optimization I, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications, im August 1977 in Banff und Vancouver; zugleich: Annals of Discrete Mathematics, Vol. 4 (1979), Amsterdam - New York - Oxford 1979, S. 121-140.

Lenstra (1982)

Lenstra, J.K.; Rinnooy Kan, A.H.G.; van Emde Boas, P.: An appraisal of computational complexity for operations researchers; in: European Journal of Operational Research, Vol. 11 (1982), S. 201-210.

Lien (1976a)

Lien, Y.E.: A Note on Transition Systems; in: Information Sciences, Vol. 10 (1976), S. 347-362.

Liggesmeyer (1990)

Liggesmeyer, P.: Modultest und Modulverifikation - State of the art, Mannheim - Wien - Zürich 1990.

Lipton (1976)

Lipton, R.: The Reachability Problem Requires Exponential Space, Research Report No. 62, Department of Computer Science, Yale University, New Haven 1976.

Lockemann (1975)

Lockemann, P.C.: Information Systems: A Survey by Examples; in: Mühlbacher, J. (Hrsg.): GI - 5. Jahrestagung, 8.-10.10.1975 in Dortmund, Lecture Notes in Computer Science 34, Berlin - Heidelberg - New York 1975, S. 3-34.

Lockemann (1978)

Lockemann, P.C.; Mayr, H.C.: Rechnergestützte Informationssysteme, Berlin - Heidelberg - New York 1978.

Löhr (1974)

Löhr, K.-P.: Datenflußgesteuerte Koppelung nichtsequentieller Systeme, Dissertation, Technische Universität Berlin, Berlin 1974.

Löwe (1987)

Löwe, C.R.; Dembski, M.; Fieten, R.; Bauer, C.O.; Behrens, G.; Cichy, U.; Pip, H.; Schwarze, K.: Risiko und Innovation - Fallstudien zur Handhabung von Risiken bei Innovationsvorhaben in Klein- und Mittelbetrieben, Köln 1987.

London (1978)

London, P.E.: Dependency Networks as a Representation for Modelling in General Problem Solvers, Dissertation, University of Maryland, Ann Arbor 1978. (Anmk. des Verf.: auch erschienen als: Report TR-691, Department of Computer Science, University of Maryland.)

Loui (1983)

Loui, R.P.: Optimal Paths in Graphs with Stochastic or Multidimensional Weights; in: Communications of the ACM, Vol. 26 (1983), S. 670-676.

Lowas (1986b)

Lowas, A.: Computer-Aided Design of Multiprocessor Control Systems; in: o.V.: Proceedings of the 25th IEEE Conference on Decision and Control, im Dezember 1986 in Athen, Vol. 3, New York 1986, S. 1692-1696.

Lukaszewicz (1985)

Lukaszewicz, W.: Nonmonotonic Logic for Default Theories; in: O'Shea, T. (Hrsg.): Advances in Artificial Intelligence, Proceedings of the Sixth European Conference on Artificial Intelligence, ECAI-84, 05.-07.09.1984 in Pisa, Amsterdam - New York - Oxford 1985, S. 403-412.

Mädler (1990)

Mädler, F.; Gust, H.: Über ein Meta-Prinzip zur Explikation von Kontrollwissen; in: Marburger, H. (Hrsg.): GWAI-90, 14th German Workshop on Artificial Intelligence, 10.-14.09.1990 in Eringerfeld, Proceedings, Informatik-Fachberichte 251, Berlin - Heidelberg - New York ... 1990, S. 85-94.

Magee (1964a)

Magee, J.F.: Decision Trees for Decision Making; in: Harvard Business Review, Vol. 42 (1964), No. 4, S. 126-138.

Magee (1964b)

Magee, J.F.: HOW TO USE Decision Trees in Capital Investment; in: Harvard Business Review, Vol. 42 (1964), No. 5, S. 79-96.

Maier, U. (1980)

Maier, U.: Arbeitsgangterminierung mit variabel strukturierten Arbeitsplänen - Ein Beitrag zur Fertigungssteuerung flexibler Fertigungssysteme, Berlin - Heidelberg - New York 1980.

Maier-Rothe (1985)

Maier-Rothe, C.: Wettbewerbsvorteile durch höhere Produktivität und Flexibilität - Strategien für Computer-Integrated Manufacturing; in: Arthur D. Little International (Hrsg.): Management im Zeitalter der strategischen Führung, Wiesbaden 1985, S. 123-161.

Mainz (1984)

Mainz, U.: Netztheoretische Repräsentation prädikatenlogischer Begriffe und Methoden, Diplomarbeit am Institut für Informatik, Universität Bonn, Bonn 1984.

Manna (1979)

Manna, Z.: The Modal Logic of Programs; in: Maurer, H.A. (Hrsg.): Automata, Languages and Programming, Sixth Colloquium, 16.-20.07.1979 in Graz, Lecture Notes in Computer Science 71, Berlin - Heidelberg - New York 1979, S. 385-409.

Marcus (1984)

Marcus, R.: An Application of Artificial Intelligence to Operations Research; in: Communications of the ACM, Vol. 27 (1984), S. 1044-1047.

Marin (1975)

Marin, J.: Sur le test en ligne des machines sequentielles realisees a partir des reseaux de Petri, Dissertation, Universität Nizza, Nizza 1975.

Martelli (1975)

Martelli, A.; Montanari, U.: From Dynamic Programming to Search Algorithms with Functional Costs; in: o.V.: Advance Papers of the Fourth International Conference on Artificial Intelligence (IJCAI-75), 03.-08.09.1975 in Tbilisi, o.O. (Cambridge/Massachusetts) 1975, Vol. 1, S. 345-350.

Martin, R.K. (1990)

Martin, R.K.; Rardin, R.L.; Campbell, B.A.: Polyhedral Charakterization of Discrete Dynamic Programming; in: Operations Research, Vol. 38 (1990), S. 127-138.

Massachusetts Computer Associates (1974b)

Massachusetts Computer Associates, Inc. (Hrsg.): Final Report for the Project "Development of Theoretical Foundations for Description and Analysis of Discrete Information Systems", Volume I - Semantics, Wakefield 1974.

Massachusetts Computer Associates (1974c)

Massachusetts Computer Associates, Inc. (Hrsg.): Final Report for the Project "Development of Theoretical Foundations for Description and Analysis of Discrete Information Systems, Volume II - Mathematica, Wakefield 1974.

Mayr,E. (1975)

Mayr,E.: Einige Sätze über Umformungen und verklemmungsfreie Führbarkeit bei bewerteten Petri-Netzen, Diplomarbeit am Institut für Informatik, Technische Universität München, München 1975.

Mayr,E. (1980a)

Mayr,E.W.: Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Probleme, Dissertation an der Technischen Universität München, München 1980.

Mayr,E. (1981b)

Mayr,E.W.: An Algorithm for the General Petri Net Reachability Problem; in: o.V.: Conference Record of the 13th Annual ACM Symposium on Theory of Computing, 1981 in Milwaukee, o.O. (New York) 1981, S. 238-246.

Mayr,E. (1981c)

Mayr,E.: Persistence of Vector Replacement Systems is Decidable; in: Acta Informatica, Vol. 15 (1981), S. 309-318.

Mayr,E. (1984)

Mayr,E.W.: An Algorithm for the General Petri Net Reachability Problem; in: SIAM Journal on Computing, Vol. 13 (1984), S. 441-460.

McDermott (1979)

McDermott,D.; Doyle,J.: An Introduction to Non-Monotonic Logic; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 1, S. 562-567.

McDermott (1980)

McDermott,D.; Doyle,J.: Non-Monotonic Logic; in: Artificial Intelligence, Vol. 13 (1980), S. 41-72.

Mehlhorn (1977)

Mehlhorn,K.: Effiziente Algorithmen, Stuttgart 1977.

Mehlhorn (1984)

Mehlhorn,K.: Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness, Berlin - Heidelberg - New York ... 1984.

Meißner (1978)

Meißner,J.-D.: Heuristische Programmierung, Wiesbaden 1978.

Mekly (1980)

Mekly,L.J.; Yau,S.S.: Software Design Representation Using Abstract Process Networks; in: IEEE Transactions on Software Engineering, Vol. SE-6 (1980), S. 420-435.

Memmi (1978a)

Memmi,G.: Fuites dans les Reseaux de Petri; in: Revue Francaise d'Automatique, Informatique et Recherche Operationelle - Informatique theoretique, Vol. 12 (1978), S. 125-144.

Memmi (1978c)

Memmi,G.: Applications of the Semiflow Notion to the Boundedness and Liveness Problems in the Petri Nets Theory; in: o.V.: Proceedings of the Annual Princeton Conference on Information Sciences and Systems, Vol. 12, Baltimore 1978, S. 505-509.

Memmi (1979)

Memmi,G.: Notion de Duality et de Symmetrie dans les Reseaux de Petri; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 91-108.

Memmi (1980)

Memmi,G.; Roucairol,G.: Linear Algebra in Net Theory; in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 213-224.

Memmi (1981a)

Memmi,G.: Analyse des Reseaux de Petri par l'Algebre Lineaire, D.E.A. de Systemes Informatiques, Institut de Programmation, Universite Pierre et Marie Curie Paris, Paris 1981.

Memmi (1982)

Memmi,G.: Leakage Notion; in: Girault,C.; Reisig,W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 171-177.

Memmi (1983b)

Memmi,G.: A Graph Theoretical Property for Minimal Deadlock; in: Pagnoni,A.; Rozenberg,G. (Hrsg.): Applications and Theory of Petri Nets, Selected Papers from the 3rd European Workshop on Applications and Theory of Petri Nets, 27.-30.09.1982 in Varenna, Informatik-Fachberichte 66, Berlin - Heidelberg - New York - Tokyo 1983, S. 221-228.

Merlin,P. (1976a)

Merlin,P.M.: A Methodology for the Design and Implementation of Communication Protocols; in: IEEE Transactions on Communications, Vol. COM-24 (1976), S. 614-621.

Merlin,P. (1978)

Merlin,P.M.; Randell,B.: State Restoration in Distributed Systems; in: o.V.: Proceedings of the 8th Annual International Symposium on Fault-Tolerant Computing - Digest of Papers (IEEE), Toulouse 1978, S. 129-134.

Mero (1981)

Mero,L.: Some Remarks on Heuristic Search Algorithms; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 572-574.

Mertens (1983)

Mertens,P.; Allgeyer,K.: Künstliche Intelligenz in der Betriebswirtschaft; in: Zeitschrift für Betriebswirtschaft, 53. Jg. (1983), S. 686-709.

Mertins (1985a)

Mertins,K.: Steuerung rechnergeführter Fertigungssysteme, München - Wien 1985.

Mertins (1985b)

Mertins,K.: Entwicklungsstand flexibler Fertigungssysteme - Linien-, Netz- und Zellenstrukturen; in: Zeitschrift für wirtschaftliche Fertigung, 80. Jg. (1985), S. 249-264.

Meyer zu Selhausen (1980b)

Meyer zu Selhausen,H.: Wege zu einer verhaltensorientierten Methodik der Modellbildung und Modell-Implementierung, Teil II: Der scenario-bezogene OR-Prozeß; in: IHS-Journal, Vol. 4 (1980), S. 131-145.

Michel,M. (1978)

Michel,(M.)C.; Kwan,C.L.: Hardwired logic and microprocessors; in: Euromicro Journal, Vol. 4 (1978), S. 167-180.

Migliore (1990)

Migliore,M.; Ayala,G.F.; Fornili,S.L.: Modeling of Neuronal Systems on Transputer Networks; in: Eckmiller,R.; Hartmann,G.; Hauske,G. (Hrsg.): Parallel Processing in Neural Systems and Computers, International Conference on Parallel Processing in Neural Systems and Computers, 19.-21.03.1990 in Düsseldorf, Amsterdam - New York - Oxford ... 1990, S. 291-294.

Miller,D.P. (1988)

Miller,D.P.: A Task and Resource Scheduling System for Automated Planning; in: Annals of Operations Research, Vol. 12 (1988), S. 169-198.

Minker (1983)

Minker,J.; Asper,C.; Cao,D.; Chakravarthy,U.S.; Csoeke-Poeckh,A.; Kasif,S.; Kohli,M.; Piazza,R.; Wang,D.: Parallel Problem Solving in ZMOB; in: o.V.: Proceedings Trends & Applications 1983 - Automating Intelligent Behavior - Applications and Frontiers, Silver Spring 1983, S. 142-146.

Mittermeir (1974)

Mittermeir,R.: Modularer Aufbau von Petrinetzen, Dipolmarbeit, Institut für Informationstechnik, Technische Hochschule Wien, Wien 1974.

Moalla (1976b)

Moalla,M.: L'Approche Fonctionnelle dans la Verification des Systemes Informatiques. Proposition d'un Ensemble de Methodologies, Dissertation am Institut National Polytechnique de Grenoble, Grenoble 1976.

Moalla (1977)

Moalla,M.; Sifakis,J.: A Design Methodology for Complex Logical Systems; in: o.V.: Proceedings of the 2nd IFAC International Symposium on Discrete Systems, im März 1977 in Dresden, o.O. 1977, S. 89-101.

Moalla (1978a)

Moalla,M.; Pulou,J.; Sifakis,J.: Reseaux de Petri Synchronises; in: Revue Francaise d'Automatique, Informatique et Recherche Operationelle - Automatique/Systems Analysis and Control, Vol. 12 (1978), S. 103-130.

Moeller (1981)

Moeller,G.L.; Digman,L.A.: Operations Planning with VERT; in: Operations Research, Vol. 29 (1981), S. 676-697.

Montel (1983a)

Montel,B.; Grissault,D.; Le Mer,E.; Robert,C.; Sivet,A.; Azema,P.; Bachmann,S.; Berthomieu, B.; Diaz,M.; Pradin,B.: OVIDE, a Software Package for the Validation of Systems Represented by Petri Net Based Models; in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 292-308.

Moore,L. (1976)

Moore,L.J.; Clayton,E.R.: GERT Modeling and Simulation: Fundamentals and Applications, New York 1976.

Moore,R. (1985)

Moore,R.C.: Semantical Considerations on Nonmonotonic Logic; in: Artificial Intelligence, Vol. 25 (1985), S. 75-94.

Morgan (1986)

Morgan,E.T.; Razouk,R.R.: Computer-Aided Analysis of Concurrent Systems; in: Diaz,M. (Hrsg.): Protocol Specification, Testing, and Verification, V, Proceedings of the IFIP WG 6.1 5th International Workshop, 10.-13.06.1985 in Toulouse-Moissac, Amsterdam 1986, S. 49-58.

Morgan (1987)

Morgan,E.T.; Razouk,R.R.: Interactive State-Space Analysis of Concurrent Systems; in: IEEE Transactions on Software Engineering, Vol. SE-13 (1987), S. 1080-1091.

Morris,P. (1986)

Morris,P.H.; Nado,R.A.: Representing Actions with an Assumption-Based Truth Maintenance System; in: o.V.: proceedings aaai-86, fifth national conference on artificial intelligence, 11.-15.08.1986 in Philadelphia, Los Altos 1986, Vol. 1, S. 13-17.

Morris,P. (1988)

Morris,P.H.: The Anomalous Extension Problem in Default Reasoning; in: Artificial Intelligence, Vol. 35 (1988), S. 383-399.

Mosner (1991)

Mosner,H.-M.; Heeg,G.: Visuelle interaktive Simulation und Modellierung mit dem objekt-orientierten Programmiersystem Smalltalk-80; in: Biethahn,J.; Hummeltenberg,W.; Schmidt,B. (Hrsg.): Simulation als betriebliche Entscheidungshilfe, Band 2, Berlin - Heidelberg - New York ... 1991, S. 137-145.

Moto-oka (1984)

Moto-oka,T.; Tanaka,H.; Aida,H.; Hirata,K.; Maruyama,T.: The Architecture of a Parallel Inference Engine -PIE-; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 479-488.

Müller,A. (1987)

Müller,A.: Produktionsplanung und Pufferbildung bei Werkstattfertigung, Dissertation unter dem Titel "Der Pufferbedarf im Rahmen der kurzfristigen Produktionsplanung bei Werkstattfertigung", Technische Hochschule Aachen 1986, Wiesbaden 1987.

Müller,B. (1981)

Müller,B.: Einsatz von Petri-Netzen in der Software-Entwicklung; in: Fandel,G.; Fischer,D.; Pfohl,H.-C.; Schuster,K.-P.; Schwarze,J. (Hrsg.): Operations Research Proceedings 1980, DGOR - Vorträge der Jahrestagung, Berlin - Heidelberg - New York 1981, S. 199-206.

Müller,H. (1980b)

Müller,H.: Decidability of Reachability in Persistent Vector Replacement Systems; in: Dembinski,P. (Hrsg.): Mathematical Foundations of Computer Science 1980, Proceedings of the 9th Symposium, 1.-5.09.1980 in Rydzyna, Lecture Notes in Computer Science 88, Berlin - Heidelberg - New York 1980, S. 426-438.

Müller,H. (1982c)

Müller,H.: Filling a gap in Kosaraju's proof for the decidability of the reachability problem in VAS; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 12 (1982), S. 4-10.

Müller,H. (1983b)

Müller,H.: The Reachability Problem for VAS; in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 4-5. (Erweiterte Fassung: Müller,H. (1985a).)

Müller,H. (1983c)

Müller,H.: On Kosaraju's Proof of the Decidability of the Reachability Problem for Vector Addition Systems; in: Priese,L. (Hrsg.): Report on the 1st GTI-Workshop, 10.-16.10.1982 in Paderborn, Bericht Nr. 13, Reihe Theoretische Informatik, Universität Paderborn, Paderborn 1983, S. 174-183.

Müller,H. (1983e)

Müller,H.: Vortrag, gehalten am 27.09.1983 in Toulouse anlässlich: 4th European Workshop on Applications and Theory of Petri Nets.

Müller,H. (1985a)

Müller,H.: The Reachability Problem for VAS; in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1984, Lecture Notes in Computer Science 188, Berlin - Heidelberg - New York - Tokyo 1985, S. 376-391.

Müller,H. (1985b)

Müller,H.: Weak Petri Net Computers for Ackermann Functions; in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 21 (1985), S. 236-246.

Müller-Merbach (1973)

Müller-Merbach,H.: Operations Research - Methoden und Modelle der Optimalplanung, 3. Aufl., München 1973.

Müller-Merbach (1976a)

Müller-Merbach,H.: Morphologie heuristischer Verfahren; in: Zeitschrift für Operations Research, Bd. 20 (1976), S. 69-87.

Müller-Merbach (1978)

Müller-Merbach,H.; Golling,H.-J.: Die Rolle von Wahrscheinlichkeitsverteilungen in Entscheidungsprozessen; in: Helmstädter,E. (Hrsg.): Neuere Entwicklungen in den Wirtschaftswissenschaften, Verhandlungen auf der Arbeitstagung der Gesellschaft für Wirtschafts- und Sozialwissenschaften - Verein für Socialpolitik, 19.-21.09.1977 in Münster, Berlin 1978, S. 413-430.

Muller,H. (1987)

Muller(-Malek),H.; de Samblanckx,S.; Matthys,D.: The expert system approach and the flexibility-complexity problem in scheduling production systems; in: International Journal of Production Research, Vol. 25 (1987), S. 1659-1670.

Murata,Ta. (1976a)

Murata,Ta.; Shah,T.: On Liveness, Deadlocks, and Reachability of E-Nets; in: o.V.: Proceedings of the 14th Allerton Conference on Circuit and System Theory, 1976 in Moticello, Urbana 1976, S. 597-605.

Murata,Ta. (1976c)

Murata,Ta.; Church,R.W.; Amin,A.T.: Matrix Equations for Petri Nets and Marked Graphs; in: o.V.: Conference Record of the 9th Annual Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove (1976), S. 36-41.

Murata,Ta. (1977a)

Murata,Ta.: State Equation, Controllability, and Maximal Matchings of Petri Nets; in: IEEE Transactions on Automatic Control, Vol. AC-22 (1977), S. 412-416.

Murata,Ta. (1977b)

Murata,Ta.: Circuit Theoretic Analysis and Synthesis of Marked Graphs; in: IEEE Transactions on Circuit and Systems, Vol. CAS-24 (1977), S. 400-405.

Murata,Ta. (1977c)

Murata,Ta.: petri nets, marked graphs, and circuit-system theory, a recent CAS application, Reprinted from: Circuits and Systems, Vol. 11 (1977), No. 3, S. 2-12.

Murata,Ta. (1979a)

Murata,Ta.: Synthesis of Marked Graph Computation Models for Prescribed Resources and Performance; in: o.V.: Conference Record of the 12th Asilomar Conference on Circuits, Systems, and Computers, im Jahr 1978, o.O. (New York) 1979, S. 807-812. (Inhaltlich identisch mit Murata (1980).)

Murata,Ta. (1980a)

Murata,Ta.: Synthesis of Decision-Free Concurrent Systems for Prescribed Resources and Performance; in: IEEE Transactions on Software Engineering, Vol. SE-6 (1980), S. 525-530. (Inhaltlich identisch mit Murata (1979a).)

Murata,Ta. (1987a)

Murata,Ta.; Peterka,G.: Application of Colored Petri Net T-Invariants to Logic Programming; in: o.V.: Proceedings of the Fifth International Conference on Systems Engineering, 1987 in Dayton, New York 1987, S. 381-384.

Murata,Ta. (1988b)

Murata,Ta.; Zhang,D.: A Predicate-Transition Net Model for Parallel Interpretation of Logic Programs; in: IEEE Transactions on Software Engineering, Vol. SE-14 (1988), S. 481-497.

Nakagawa (1979)

Nakagawa,S.; Sakai,T.: A Parallel Tree Search Method; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 2, S. 628-632.

Nardi (1987)

Nardi,B.A.; Paulson,E.A.: Multiple Worlds with Truth Maintenance in AI Applications; in: du Boulay,B.; Hogg,D.; Steels,L. (Hrsg.): Advances in Artificial Intelligence - II, Seventh European Conference on Artificial Intelligence, ECAI-86, 20.-25.07.1986 in Brighton, Amsterdam - New York - Oxford ... 1987, S. 563-572.

Neumann,K. (1968)

Neumann,K.: Dynamische Optimierung und PONTRJAGINSches Maximumprinzip; in: Unternehmensforschung, 12. Jg. (1968), S. 55-70.

Neumann, K. (1975b)

Neumann, K.: Operations Research Verfahren, Band III: Graphentheorie, Netzplantechnik, München - Wien 1975.

Neumann, K. (1977)

Neumann, K.: Operations Research Verfahren, Band II: Dynamische Optimierung, Lagerhaltung, Simulation, Warteschlangen, München - Wien 1977.

Neumann, K. (1989a)

Neumann, K.: Graphen und Netzwerke; in: Gal, T. (Hrsg.): Grundlagen des Operations Research, (Band) 2: Graphen und Netzwerke - Netzplantechnik, Transportprobleme - Ganzzahlige Optimierung, 2. Aufl., Berlin - Heidelberg - New York ... 1989, S. 1-164.

Neumann, K. (1990)

Neumann, K.: Stochastic Project Networks - Temporal Analysis, Scheduling and Cost Minimization, Lecture Notes in Economics and Mathematical Systems 344, Berlin - Heidelberg - New York ... 1990.

Newell (1965)

Newell, A.: Limitations of the Current Stock of Ideas about Problem Solving; in: Kent, A.; Taubee, D.E. (Hrsg.): Electronic Information Handling, Washington - London 1965, S. 195-208.

Newell (1969)

Newell, A.: Heuristic Programming: Ill-Structured Problems; in: Aronofsky, J.S. (Hrsg.): Progress in Operations Research, Vol. III: Relationship between Operations Research and the Computer, New York - London - Sydney ... 1969, S. 361-414.

Newell (1982b)

Newell, A.; Simon, H.A.: Computer Science as Empirical Enquiring: Symbols and Search; in: Haugeland, J. (Hrsg.): Mid Design - Philosophy, Psychology, Artificial Intelligence, 2. Druck, Cambridge (Massachusetts) - London 1982, S. 35-66.

Niederhausen (1988)

Niederhausen, H.-P.: Realzeit-Expertensysteme zur dynamischen Reihenfolgeplanung für flexible Fertigungszellen; in: Mertens, P.; Wiendahl, H.-P.; Wildemann, H. (Hrsg.): CIM-Komponenten zur Planung und Steuerung - Expertensysteme in der Produktion, München 1988, S. 399-404.

Niehuis (1986)

Niehuis, S.; Victor, F.: Modellierung und Simulation von Pr/T-Netzen in Prolog, Arbeitspapiere der GMD 231, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1986.

Niemann (1983)

Niemann, H.: Control Strategies in Image and Speech Understanding; in: Neumann, B. (Hrsg.): GWAI-83 - 7th German Workshop on Artificial Intelligence, 19.-23.09.1983 in Dassel/Solling, Informatik-Fachberichte 76, Berlin - Heidelberg - New York ... 1983, S. 31-49.

Niemann (1984)

Niemann, H.: Mustererkennung - eine einführende Übersicht; in: Handbuch der Modernen Datenverarbeitung, 21. Jg. (1984), Heft 115, S. 3-22.

Nieß (1980)

Nieß, P.S.: Kapazitätsabgleich bei flexiblen Fertigungssystemen, Berlin - Heidelberg - New York 1980.

Nieters (1978)

Nieters, H.-H.: Darstellung einiger Aspekte von Programmiersprachen mit Transitions- und Factnetzen, Diplomarbeit, Mathematisches Institut, Universität Bonn, Bonn 1978.

Nievergelt (1984)

Nievergelt, J.: Die n-te Generation; in: Informatik-Spektrum, Bd. 7 (1984), S. 237-242.

Nilsson, N. (1980a)

Nilsson, N.J.: Principles of Artificial Intelligence, Palo Alto 1980.

Noe (1971)

Noe, J.D.: A Petri Net Model of The CDC 6400; in: o.V.: Proceedings of the ACM/SIGOPS Workshop on System Performance Evaluation, Cambridge (Massachusetts) 1971, S. 362-378.

Noe (1972)

Noe, J.D.; Nutt, G.J.: Validation of a tree-driven CDC 6400 simulation; in: o.V.: Spring Joint Computer Conference, 16.-18.05.1972 in Atlantic City, AFIPS Conference Proceedings, Vol. 40 (1972), Montvale 1972, S. 749-757.

Noe (1973a)

Noe, J.D.; Nutt, G.J.: Macro E-Nets for Representation of Parallel Systems; in: IEEE Transactions on Computers, Vol. C-22 (1973), S. 718-727.

Nyhuis, P. (1989)

Nyhuis, P.: Optimale Fertigungslosgrößen - eine Funktion des Bestandes?; in: Industrie-Anzeiger, 111. Jg. (1989), Nr. 76, S. 40-41.

O'hEigeartaigh (1985)

O'hEigeartaigh, M.; Lenstra, J.K.; Rinnooy Kan, A.H.G. (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane ... 1985.

o.V. (1982a)

o.V.: PES - Petrinetz-Entwicklungssystem, Analysator, Benutzerhandbuch, Version 3.0, o.O. (München) 1982.

o.V. (1983a)

o.V.: SYSECA - TEMPS REEL - OVIDE, Outil de Validation de Reseaux de Petri, Saint Cloud o.J. (1983).

Pacini (1983)

Pacini, G.; Turini, F.: Demonizing Production Systems; in: Bundy, A. (Hrsg.): IJCAI-83, Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Los Altos 1983, Vol. 2, S. 862-866.

Padberg (1980)

Padberg, M.W.; Rao, M.R.: The Russian Method for Linear Inequalities and Linear Optimization, Report, New York University Graduate School of Business Administration, New York 1980.

Pagnoni (1990)

Pagnoni, A.: Project Engineering - Computer-Oriented Planning and Operational Decision Making, Berlin - Heidelberg - New York ... 1990.

Pakas-Skewes (1979)

Pakas-Skewes, E.: A Design Methodology for Digital Systems Using Petri Nets, Dissertation, University of Texas, Austin 1979.

Pan (1986)

Pan, V.; Reif, J.: Efficient Parallel Linear Programming; in: Operations Research Letters, Vol. 5 (1986), No. 3, S. 127-135.

Parker, R. (1982a)

Parker, R.G.; Rardin, R.L.: An Overview of Complexity Theory in Discrete Optimizations: Part I. Concepts; in: AIIE Transactions, Vol. 14 (1982), S. 3-10.

Parker, R. (1982b)

Parker, R.G.; Rardin, R.L.: An Overview of Complexity Theory in Discrete Optimization: Part II. Results and Implications; in: AIIE Transactions, Vol. 14 (1982), S. 83-89.

Pascoletti (1985)

Pascoletti, K.-H.: Diophantische Systeme und Lösungsmethoden zur Bestimmung aller Invarianten in Petri-Netzen, Dissertation, Universität Bonn, Bonn 1985. (Auch erschienen als: Berichte der GMD 160, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, München - Wien 1986.)

Passino (1988a)

Passino, K.M.; Antsaklis, P.J.: Artificial Intelligence Planning Problems in a Petri Net Framework; in: o.V.: Proceedings of the 1988 American Control Conference, Piscataway 1988, Vol. 1, S. 626-631.

Patil (1971)

Patil, S.S.: Limitations and Capabilities of Dijkstra's Semaphore Primitives for Coordination among Processes, Computation Structures Group Memo No. 57, Project MAC, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1971.

Patil (1972a)

Patil,S.S.: Circuit Implementation of Petri Nets, Computation Structures Group Memo No. 73, Project MAC, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1972.

Patil (1973)

Patil,S.S.; Dennis,J.B.: The Description and Realization of Digital Systems; in: Revue Francaise d'Automatique, Informatique, et Recherche Operationelle - Automatique, Vol. 7 (1973), S. 55-69.

Patil (1974)

Patil,S.S.: Cellular Arrays for Asynchronous Computation; in: o.V.: Conference Record: 7th Annual Workshop on Microprogramming, Papers presented at the Workshop, im September/Oktober 1974 in Palo Alto, New York 1974, S. 178-185.

Paul,W. (1978)

Paul,W.J.: Komplexitätstheorie, Stuttgart 1978.

Pearl (1981)

Pearl,J.: Heuristic Search Theory: Survey of Recent Results; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 554-562.

Pearl (1983)

Pearl,J.: Knowledge Versus Search: a Quantitative Analysis Using A*; in: Artificial Intelligence, Vol. 20 (1983), S. 1-14.

Pearl (1987)

Pearl,J.: Evidential Reasoning Using Stochastic Simulation of Causal Models; in: Artificial Intelligence, Vol. 32 (1987), S. 245-257.

Perl (1980)

Perl,J.: Übertragung des Markierungsbegriffs auf vergrößerte Petri-Netze, Osnabrücker Schriften zur Mathematik, Reihe M: Mathematische Manuskripte, Heft 3, Osnabrück 1980.

Pernards (1977)

Pernards,P.: Simulation einer Büro-Rechenanlage - Konzept, Modell, Ergebnis -; in: Spies,P.P. (Hrsg.): Modelle für Rechensysteme, Workshop der Gesellschaft für Informatik, 31.03.-1.04.1977 in Bonn, Berlin - Heidelberg - New York 1977, S. 211-225.

Perridon (1991)

Perridon,L.; Steiner,M.: Finanzwirtschaft der Unternehmung, 6. Aufl., München 1991.

Peterson,J. (1973a)

Peterson,J.L.: Modelling of Parallel Systems, Dissertation am Department of Electrical Engineering, Stanford University, Stanford 1973.

Peterson,J. (1976)

Peterson,J.L.: Computation Sequence Sets; in: Journal of Computer and System Sciences, Vol. 13 (1976), S. 1-24.

Peterson,J. (1977)

Peterson,J.L.: Petri Nets; in: Computing Surveys, Vol. 9 (1977), S. 223-252.

Peterson,J. (1978a)

Peterson,J.L.: An Introduction to Petri Nets; in: Tranter,W.H. (Hrsg.): Proceedings of the National Electronics Conference, Vol. 32, 16.-18.10.1978 in Chicago, Oak Brook, S. 144-148.

Peterson,J. (1981)

Peterson,J.L.: Petri Net Theory and the Modeling of Systems, Englewood Cliffs 1981.

Petri,C. (1976b)

Petri,C.A.: Interpretations of Net Theory, Interner Bericht 75-07, Paper präsentiert anlässlich: International Symposium on Mathematical Foundations of Computer Science MFCS'75, 1.-5.09.1975 in Marianske Lazne, 2. Aufl., Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1976.

Petri,C. (1979d)

Petri,C.A.: Concurrency as a Basis of Systems Thinking; in: Jensen,K.; Mayoh,B.H.; Moller,K. K. (Hrsg.): Proceedings of the 5th Scandinavian Logic Symposium, 17.-19.01.1979 in Aalborg, Aalborg 1979, S. 143-162. (Auch erschienen als: Internal Report ISF-78-06, Paper präsentiert anlässlich: 7th International Symposium on Mathematical Foundations of Computer Science MFCS'78, 4.-8.09.1978 in Zakopane, 2. Aufl., Insitut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1979.)

Petrie (1989)

Petrie,C.J.: Reason Maintenance in Expert Systems; in: Künstliche Intelligenz: Forschung, Entwicklung, Erfahrungen., 3. Jg. (1989), Heft 2, S. 54-60.

Pfohl (1981)

Pfohl,H.-C.; Braun,G.E.: Entscheidungstheorie - Normative und deskriptive Grundlagen des Entscheidens, Landsberg 1981.

Pnueli (1977)

Pnueli,A.: The Temporal Logic of Programs; in: o.V.: Proceedings of the 18th Annual Symposium on Foundations of Computer Science, Providence 1977, S. 46-57.

Pnueli (1979)

Pnueli,A.: The Temporal Semantics of Concurrent Programs; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 1-20.

Pohl (1970)

Pohl,I.: Heuristic Search Viewed as Path Finding in a Graph; in: Artificial Intelligence, Vol. 1 (1970), S. 193-204.

Pohl (1971)

Pohl,I.: Bi-directional Search; in: Meltzer,B.; Michie,D. (Hrsg.): Machine Intelligence 6, Edinburgh 1971, S. 127-140.

Pohl (1977)

Pohl,I.: Practical and Theoretical Considerations in Heuristic Search Algorithms; in: Elcock,E. W.; Micie,D. (Hrsg.): Machine Intelligence 8: Machine Representations of Knowledge, New York - Chichester - Brisbane ... 1977, S. 55-72.

Polya (1949)

Polya,G.: Schule des Denkens - Vom Lösen mathematischer Probleme, Bern 1949.

Pomberger (1987)

Pomberger,G.; Wallmüller,E.: Ada und Modula-2 - ein Vergleich; in: Informatik-Spektrum, Bd. 10 (1987), S. 181-191.

Poole,D. (1988)

Poole,D.: A Logical Framework for Default Reasoning; in: Artificial Intelligence, Vol. 36 (1988), S. 27-47.

Pospelov,G. (1984)

Pospelov,G.S.: Artificial Intelligence as a Basis for a New Information Technology; in: Ponomaryov,V.M. (Hrsg.): Artificial Intelligence, Proceedings of the IFAC Symposium, 04.-06.10. 1983 in Leningrad, Oxford - New York - Toronto ... 1984, S. 1-14.

Pressmar (1982)

Pressmar,D.B.: Zur Akzeptanz von computergestützten Planungssystemen; in: Krallmann,H. (Hrsg.): Unternehmensplanung und -steuerung in den 80er Jahren - Eine Herausforderung an die Informatik, Anwendersgespräch, 24.-25.11.1981 in Hamburg, Berlin - Heidelberg - New York 1982, S. 324-348.

Priese (1979)

Priese,L.: Asynchrone, Modulare Netze: Petri-Netze, Normierte Netze, APA-Netze, Habilitation an der Universität Dortmund, Forschungsbericht Nr. 94, Abteilung Informatik, Univertät Dortmund, Dortmund 1979.

Priese (1983)

Priese,L.: Automata and Concurrency; in: Theoretical Computer Science, Vol. 25 (1983), S. 221-265.

Prinoth (1977)

Prinoth,R.: Eigenschaften färbbarer Petri-Netze; in: Tzschach,H.; Waldschmidt,H.; Walter,H.K.-G. (Hrsg.): Theoretical Computer Science, 3rd GI Conference, 28.-30.03.1977 in Darmstadt, Lecture Notes in Computer Science 48, Berlin - Heidelberg - New York 1977, S. 306-318.

Prinoth (1978a)

Prinoth,R.: Starke Färbbarkeit in Petri-Netzen, Dissertation, Technische Hochschule Darmstadt, Bericht Nr. 117, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, München - Wien 1978.

Prinoth (1978b)

Prinoth,R.: Notwendige und hinreichende Bedingungen für die Existenz lebendiger und sicherer Markierungen einer Klasse von Petri-Netzen, Interner Bericht, Institut für Datenfernverarbeitung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Darmstadt 1978.

Prinoth (1979a)

Prinoth,R.: Färbbarkeitskonzepte und Systemmodellierung, Interner Bericht, Institut für Datenfernverarbeitung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Darmstadt 1979.

Pritsker (1977)

Pritsker,A.A.B.: Modeling and Analysis Using Q-GERT Networks, New York - London - Sydney - Toronto 1977.

PROLOG (o.J.)

o.V.: Turbo Prolog Owner's Handbook, o.O. o.J.

Puppe (1987)

Puppe,F.: Diagnostisches Problemlösen mit Expertensystemen, überarbeitete Version der Dissertation "Assoziatives diagnostisches Problemlösen mit dem Expertensystem-Shell MED2", Universität Kaiserslautern 1986, Informatik-Fachberichte 148, Berlin - Heidelberg - New York ... 1987.

Pylyshyn (1970)

Pylyshyn,Z.W. (Hrsg.): Perspectives in Computer Revolution, Englewood Cliffs 1970.

Rabin (1976)

Rabin,M.O.: Probabilistic Algorithm; in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 21-39.

Raiffa (1973)

Raiffa,L.: Einführung in die Entscheidungstheorie, München - Wien 1973.

Rajurkar (1987)

Rajurkar,K.P.; Navelkar,V.V.; Kamelian,Y.: Application of Multi-Objective Optimization Techniques to Manufacturing Systems; in: o.V.: Proceedings of the 1987 International Conference on Systems, Man and Cybernetics, Vol. 2, New York 1987, S. 513-516.

Ramamoorthy (1980)

Ramamoorthy,C.V.; Ho,G.S.: Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets; in: IEEE Transactions on Software Engineering, Vol. SE-6 (1980), S. 440-449.

Raman (1989b)

Raman,N.: Real-Time Scheduling Problems in a General Flexible Manufacturing System, Dissertation, The University of Michigan 1988, Ann Arbor 1989.

Rammig (1978)

Rammig,F.J.: Überlegungen zur Kontrollstruktur einer Computer-Hardware-Beschreibungssprache, Dissertation, zugleich: Forschungsbericht Nr. 62, Abteilung Informatik, Universität Dortmund, Dortmund 1978.

Ramöller (1976)

Ramöller,R.: Implementierung eines erweiterten Petrinetzmodells zur Simulation diskreter Systeme, Diplomarbeit an der Fakultät für Informatik, Universität Karlsruhe, Karlsruhe 1976.

Raphael (1976)

Raphael,B.: The Thinking Computer - Mind Inside Matter, San Francisco 1976.

Rardin (1982)

Rardin,R.L.; Lin,B.W.: Test Problems for Computational Experiments -- Issues and Techniques; in: Mulvey,J.M. (Hrsg.): Evaluating Mathematical Programming Techniques, Proceedings of a Conference, 05.-06.01.1981 in Boulder, Lecture Notes in Economics and Mathematical Systems 199, Berlin - Heidelberg - New York 1982, S. 8-15.

Raubold (1972)

Raubold,E.; Ullrich,G.: Graphenmodelle zur Beschreibung der Wechselwirkung zwischen asynchronen Prozessen; in: o.V.: NTG-GI-Fachtagung: Rechner- und Betriebssysteme - Analyse, Simulation und Entwurf, 12.-14.04.1972 in Darmstadt, Berlin 1972, S. 207-216.

Razouk (1985c)

Razouk,R.R.; Morgan,E.T.: The P-NUT System: An Environment for Modeling and Analyzing Concurrent Systems, Technical Report No. 85-10, Department of Information and Computer Science, University of California, Irvine 1985.

Reinfrank (1985b)

Reinfrank,M.: An Introduction to Non-Monotonic Reasoning, MEMO-SEKI-85-02, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern 1985.

Reinfrank (1986)

Reinfrank,M.; Beetz,M.; Freitag,H.; Klug,J.: KAPRI - A Rule-Based Non-Monotonic Inference Engine with an Integrated Reason Maintenance System, SEKI Report SR-86-03, Fachbereich Informatik, Universität Karlsruhe, Karlsruhe 1986. (Anmk. des Verf.: abweichende Schreibweisen auf Cover und Titelblatt; hier letztes zugrundegelegt.)

Reinfrank (1988a)

Reinfrank,M.: Reason Maintenance Systems; in: Stoyan,H. (Hrsg.): Begründungsverwaltung - Beiträge zu einem Workshop über Reason Maintenance, am 09.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 162, Berlin - Heidelberg - New York ... 1988, S. 1-26.

Reinfrank (1989b)

Reinfrank,M.: Begründungsverwaltung (Reason/Truth Maintenance); in: Künstliche Intelligenz: Forschung, Entwicklung, Erfahrungen., 3. Jg. (1989), Heft 2, S. 20.

Reisig (1979d)

Reisig,W.: Untersuchungen einer Klasse kooperierender sequentieller Prozessoren mit Hilfe von Petri-Netzen, Dissertation, Technische Hochschule Aachen, Aachen 1979.

Reisig (1980a)

Reisig,W.: On Solving Conflicts in Petri Nets; in: Pape,U. (Hrsg.): Discrete Structures and Algorithms, Proceedings of the Workshop WG 79, 5th Conference on Graphtheoretic Concepts in Computer Science, 21.-23.06.1979 in Berlin, München - Wien 1980, S. 241-253.

Reisig (1982a)

Reisig,W.: Deterministic Buffer Synchronization of Sequential Processes; in: Acta Informatica, Vol. 18 (1982), S. 117-134.

Reisig (1983c)

Reisig,W.: Cover Picture Story; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 15 (1983), S. 3-4 u. Titelblatt zu Newsletter 15.

Reisig (1985b)

Reisig,W.: Petri Nets: An Introduction, Berlin - Heidelberg - New York - Tokyo 1985. (Anmk. des Verf.: englische Übersetzung der deutschen Originalfassung (1. Aufl.: 1982); deren 2. Aufl. hier als Reisig (1986a).)

Reisig (1986a)

Reisig,W.: Petrinetze - Eine Einführung, 2. Aufl., Berlin - Heidelberg - New York Tokyo 1986. (Anmk. des Verf.: englische Übersetzung der 1. Aufl. (1982) erschienen als Reisig (1985b).)

Reisig (1987a)

Reisig,W.: Place/Transition Systems; in: Brauer,W.; Reisig,W.; Rozenberg,G. (Hrsg.): Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, 8.-19.09.1986 in Bad Honnef, Lecture Notes in Computer Science 254, Berlin - Heidelberg - New York ... 1987, S. 117-141.

Reisig (1989a)

Reisig,W.: Petri Nets and Abstract Data Types, Bericht TUM-I8904, Institut für Informatik, Technische Universität München, München 1989.

Reisig (1989b)

Reisig,W.: Cover Picture Story - The Container Crane System; in: Petri Net Newsletter (Gesellschaft für Informatik: Special Interest Group on Petri Nets and related System Models), No. 34 (1989), S. 3-9.

Reiter (1978a)

Reiter,R.: On Reasoning by Default; in: Waltz,D. (Hrsg.): Proceedings of TINLAP-2, Workshop on Theoretical Issues in Natural Language Processing - 2, im Juli 1978 in Chicago, o.O. (New York) 1978, S. 210-218.

Reiter (1980)

Reiter,R.: A Logic for Default Reasoning; in: Artificial Intelligence, Vol. 13 (1980), S. 81-132.

Reiter (1981)

Reiter,R.; Criscuolo,G.: On Interacting Defaults; in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, o.O. (Menlo Park) 1981, Vol. 1, S. 270-276.

Reitman (1964)

Reitman,W.R.: Heuristic decision procedures, open constraints, and the structure of ill-defined problems; in: Shelly,M.W.; Bryan,G. (Hrsg.): Human Judgements and Optimability, New York - London - Sydney 1964, S. 282-315.

Retti (1984)

Retti,J.: Knowledge Engineerings und Expertensysteme; in: Retti,J.; Bibel,W.; Buchberger,B.; Buchberger,E.; Horn,W.; Kobsa,A.; Steinacker,I.; Trappl,R.; Trost,H.: Artificial Intelligence - Eine Einführung, Stuttgart 1984, S. 73-98.

Riebel (1959)

Riebel,P.: Das Rechnen mit Einzelkosten und Deckungsbeiträgen; in: Zeitschrift für handelswissenschaftliche Forschung, 11. Jg. (1959), S. 213-238.

Riebel (1964a)

Riebel,P.: Der Aufbau der Grundrechnung im System des Rechnens mit relativen Einzelkosten und Deckungsbeiträgen; in: Aufwand und Ertrag, 10. Jg. (1964), S. 84-87.

Riebel (1964b)

Riebel,P.: Durchführung und Auswertung der Grundrechnung im System des Rechnens mit relativen Einzelkosten und Deckungsbeiträgen; in: Aufwand und Ertrag, 10. Jg. (1964), S. 117-120 u. 142-146.

Riebel (1979a)

Riebel,P.: Zum Konzept einer zweckneutralen Grundrechnung; in: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, 31. Jg. (1979), S. 785-798.

Riebel (1979b)

Riebel,P.: Gestaltungsprobleme einer zweckneutralen Grundrechnung; in: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, 31. Jg. (1979), S. 863-893.

Riebel (1981b)

Riebel,P.; Sinzig,W.: Zur Realisierung der Einzelkosten- und Deckungsbeitragsrechnung mit einer relationalen Datenbank; in: Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung, 33. Jg. (1981), S. 457-489.

Riebel (1982)

Riebel,P.; Sinzig,W.: Einsatzmöglichkeiten relationaler Datenbanken zur Unterstützung einer entscheidungsorientierten Kosten-, Erlös- und Deckungsbeitragsrechnung; in: Stahlknecht,P. (Hrsg.): EDV-Systeme im Finanz- und Rechnungswesen, Anwendergespräch, 08.-09.06.1982 in Osnabrück, Berlin - Heidelberg - New York 1982, S. 93-125.

Riebel (1987)

Riebel,P.: Überlegungen zur Integration von Unternehmensplanung und Unternehmensrechnung; in: Zeitschrift für Betriebswirtschaft, 57. Jg. (1987), S. 1154-1168.

Riebel (1989a)

Riebel,P.: Wirtschaftsdynamik, Unternehmensführung und Unternehmensrechnung; in: Zeitschrift für Betriebswirtschaft, 59. Jg. (1989), S. 247-259.

Riebel (1990)

Riebel,P.: Einzelkosten- und Deckungsbeitragsrechnung, Grundfragen einer markt- und entscheidungsorientierten Unternehmensrechnung, 6. Aufl., Wiesbaden 1990.

Rivett (1980)

Rivett,P.: Model Building for Decision Analysis, Chichester - New York - Brisbane ... 1980.

Rolin (1982)

Rolin,P.: Using Petri-Nets in Measurement of a Distributed Data Base System; in: Girault,C.; Reisig,W. (Hrsg.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, 23.-26.09.1980 in Strasbourg bzw. 28.-30.09.1981 in Bad Honnef, Informatik-Fachberichte 52, Berlin - Heidelberg - New York 1982, S. 11-22.

Rose,Ha. (1989)

Rose,H.: Computergestützte Störungsbewältigung beim Durchlauf von Produktionsaufträgen unter besonderer Berücksichtigung wissensbasierter Elemente, Dissertation, Universität Erlangen-Nürnberg, Erlangen - Nürnberg 1989.

Rosenberg,O. (1990)

Rosenberg,O.; Förster,U.: Belastungsorientierte Produktionslogistiksteuerung; in: Reuter,A. (Hrsg.): GI - 20. Jahrestagung II, Informatik auf dem Weg zum Anwender, 08.-12.10.1990 in Stuttgart, Proceedings, Informatik-Fachberichte 257, Berlin - Heidelberg - New York ... 1990, S. 6-15.

Rosenfield (1985)

Rosenfield,D.B.; Copacino,W.C.; Payne,E.C.: Logistics Planning and Evaluation Using "What-If" Simulation; in: Journal of Business Logistics, Vol. 6 (1985), No. 2, S. 89-109.

Rosenhead (1972)

Rosenhead,J.; Elton,M.; Gupta,S.K.: Robustness and Optimality as Criteria for Strategic Decisions; in: Operational Research Quarterly, Vol. 23 (1972), S. 413-431.

Rosenhead (1980a)

Rosenhead,J.: Planning Under Uncertainty: I. The Inflexibility of Methodologies; in: The Journal of the Operational Research Society, Vol. 31 (1980), S. 209-216.

Rosenhead (1980b)

Rosenhead,J.: Planning Under Uncertainty: II. A Methodology for Robustness Analysis; in: The Journal of the Operational Research Society, Vol. 31 (1980), S. 331-341.

Rosenstengel (1979)

Rosenstengel,B.; Winand,U.: Grundlagen eines interaktiven Modells zur ad hoc-Improvisation von Flugplänen, Arbeitsbericht Nr. 24, Seminar für Allgemeine Betriebswirtschaftslehre und Betriebswirtschaftliche Planung, Universität Köln, Köln 1979.

Rosenstengel (1982)

Rosenstengel,B.; Winand,U.: Petri-Netze - Eine anwendungsorientierte Einführung, Braunschweig - Wiesbaden 1982.

Rosenstengel (1991)

Rosenstengel,B.; Winand,U.: Petri-Netze - Eine anwendungsorientierte Einführung, 4. Aufl., Braunschweig - Wiesbaden 1991. (Anmk. des Verf.: erheblich veränderte und überarbeitete Fassung von Rosenstengel (1982).)

Roski (1984)

Roski,R.: Das Maximumprinzip von Pontrjagin; in: Wirtschaftswissenschaftliches Studium, 13. Jg. (1984), S. 515-520.

Roucairol (1982b)

Roucairol,G.: Proprietes des Reseaux de Petri, D.E.A. de Systems Informatiques, Institut de Programmation, Universite Pierre et Marie Curie Paris, Paris 1982.

Rusca (1988)

Rusca,C.V.: The IRMA Project: Towards a Robust Parallel Logic Programming Environment in Robotics; in: Bunke,H.O. (Hrsg.): Artificial Intelligence in Manufacturing, Assembly and Robotics, München - Wien 1988, S. 113-128.

Sacerdote (1977)

Sacerdote,G.S.; Tenney,R.L.: The Decidability of the Reachability Problem for Vector Addition Systems; in: o.V.: Conference Record of the 9th Annual ACM Symposium on Theory of Computing, 2.-4.05.1977 in Boulder, New York 1977, S. 61-76.

Sacerdoti (1974)

Sacerdoti,E.D.: Planning in a Hierarchy of Abstraction; in: Artificial Intelligence, Vol. 5 (1974), S. 115-135.

Sacerdoti (1979)

Sacerdoti,E.D.: Problem Solving Tactics; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 2, S. 1077-1085.

Sahni (1974)

Sahni,S.; Gonzalez,T.: P-Complete Problems and Approximate Solutions; in: o.V.: Proceedings of the 15th IEEE Annual Symposium on Switching and Automata Theory, New York 1974, S. 28-32.

Sakai (1984)

Sakai,H.; Iwata,K.; Kamiya,S.; Abe,M.; Tanaka,A.; Shibayama,S.; Murakami,K.: Design and Implementation of Relational Database Engine; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 419-426.

Sammet (1986)

Sammet,J.E.: Why Ada Is Not Just Another Programming Language; in: Communications of the ACM, Vol. 29 (1986), S. 722-732.

Sammet (1987)

Sammet,J.E.: Warum ADA nicht nur eine weitere Programmiersprache ist; in: Informationstechnik, 29. Jg. (1987), S. 191-201.

Sandring (1982)

Sandring,S.; Starke,P.H.: A Note on Liveness in Generalized Petri nets; in: Annales Societatis Mathematicae Polonae, Series 4: Fundamenta Informaticae, Vol. 5 (1982), S. 217-232.

Savage (1976)

Savage,J.E.: The Complexity of Computing, New York - London - Sydney ... 1976.

Schannath (1991)

Schannath,H.: Polynomiale und streng polynomiale Algorithmen für Netzwerkfluß-Probleme, Dissertation, Universität Passau, Wiesbaden 1991.

Scheer (1983b)

Scheer,A.-W.: Stand und Trends der computergestützten Produktionsplanung und -steuerung (PPS) in der Bundesrepublik Deutschland; in: Zeitschrift für Betriebswirtschaft, 53. Jg. (1983), S. 138-155.

Scheer (1990c)

Scheer,A.-W.: EDV-orientierte Betriebswirtschaftslehre - Grundlagen für ein effizientes Informationsmanagement, 4. Aufl., Berlin - Heidelberg - New York ... 1990.

Scheer (1991d)

Scheer,A.-W.: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, Berlin - Heidelberg - New York ... 1991.

Scheschonk (1977)

Scheschonk,G.: Eine einführende Zusammenfassung der Petri-Netz-Theorie, Bericht Nr. 77/22, Fachbereich 20 (Informatik), Technische Universität Berlin, Berlin 1977.

Schickhoff (1991)

Schickhoff,W.: Computer-Leitstand - Fortsetzung der Fertigungssteuerung mit anderen Mitteln; in: Arbeitsvorbereitung, 28. Jg. (1991), S. 99-100.

Schiemenz (1982)

Schiemenz, B.: Betriebskybernetik - Aspekte des betrieblichen Managements, Stuttgart 1982.

Schiffers (1977)

Schiffers, M.: Behandlung eines Synchronisationsproblems mit gefärbten Petri-Netzen, Diplomarbeit, Mathematisches Institut, Universität Bonn, Bonn 1977.

Schmalenbach (1948)

Schmalenbach, E.: Pretiale Wirtschaftslenkung, Bd. 2: Pretiale Lenkung des Betriebes, Bremen-Horn 1948.

Schmalenbach (1963)

Schmalenbach, E.: Kostenrechnung und Preispolitik, 8. Aufl., Köln - Opladen 1963.

Schmid, H. (1978)

Schmid, H.A.; Best, E.: A Step Towards a Solution of the Liveness Problem in Petri Nets; Technical Report No. 114, Computing Laboratory, University of Newcastle upon Tyne, Newcastle upon Tyne 1978.

Schmitz, P. (1985)

Schmitz, P.; Seibt, D.: Einführung in die anwendungsorientierte Informatik, Bd. 1: Systemtechnische Grundlagen, 3. Aufl., München 1985.

Schneeweiß, C. (1989b)

Schneeweiß, C.: Der Zeitaspekt in der Planung; in: Hax, H.; Kern, W.; Schröder, H.-H. (Hrsg.): Zeitaspekte betriebswirtschaftlicher Theorie und Praxis, 50. Wissenschaftliche Jahrestagung des Verbandes der Hochschullehrer für Betriebswirtschaft e.V., 24.-28.05.1988 in Köln, Stuttgart 1989, S. 3-19.

Schnorr (1974)

Schnorr, C.P.: Rekursive Funktionen und ihre Komplexität, Stuttgart 1974.

Schnupp (1989)

Schnupp, P.; Nguyen Huu, C.T.; Bernhard, L.W.: Expert Systems Lab Course, Berlin - Heidelberg - New York ... 1989.

Schönlein (1981)

Schönlein, A.: Der Algorithmus von Khachian; in: Angewandte Informatik, 23. Jg. (1981), S. 115-121.

Schönlein (1986)

Schönlein, A.: Der Algorithmus von Karmarkar - Idee, Realisation, Beispiel und numerische Erfahrungen; in: Angewandte Informatik, 28. Jg. (1986), S. 344-353.

Schrader (1982)

Schrader, R.: Ellipsoid Methods; in: Korte, B. (Hrsg.): Modern Applied Mathematics, Optimization and Operations Research, Amsterdam - New York - Oxford 1982, S. 265-311.

Schrader (1983)

Schrader, R.: The Ellipsoid Method and Its Implications; in: Operations Research-Spektrum, Bd. 5 (1983), S. 1-13.

Schroff (1974)

Schroff, R.: Vermeidung von totalen Verklemmungen in bewerteten Petrinetzen, Dissertation, Technische Universität München, München 1974.

Schumacher (1975)

Schumacher, F.: Simulation of Transport and Storage of Spent Nuclear Fuel; in: o.V.: Proceedings of the International Symposium on Course SIMULATION'75, im Juni 1975 in Zürich, o.O. 1975, S. 408-413.

Schumacher (1978)

Schumacher, F.: Beschreibung und Auswertung diskreter dynamischer Systeme, Bericht KfK 2635, Institut für Datenverarbeitung in der Technik, Kernforschungszentrum Karlsruhe GmbH, Projekt Wiederaufarbeitung und Abfallbehandlung, Karlsruhe 1978.

Schurz (1982)

Schurz, G.: Ein logisch-pragmatisches Modell von deduktiv-nomologischer Erklärung (Systematisierung); in: Erkenntnis, Vol. 17 (1982), S. 321-347.

Schuster,P. (1976)

Schuster,P.: Probleme, die zum Erfüllungsproblem der Aussagenlogik polynomial äquivalent sind; in: Specker,E.; Strassen,V. (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 36-48.

Schwärtzel (1986)

Schwärtzel,H.; Schlemm,H.; Wehrum,P.: Chancen und Herausforderungen von Ada; in: Informatik-Spektrum, Bd. 9 (1986), S. 349-355.

Seelbach (1979)

Seelbach,H.: Ablaufplanung bei Einzel- und Serienproduktion; in: Kern,W. (Hrsg.): Handwörterbuch der Produktionswirtschaft, Stuttgart 1979, Sp. 12-28.

Seiche (1990)

Seiche,W.: Control Synthesis Based on a Graph-Theoretical Petri Net Analysis; in: Petri Net Newsletter (Gesellschaft für Informatik: Special Interest Group on Petri Nets and related System Models), No. 35 (1990), S. 16-31.

Seliger (1983)

Seliger,G.: Wirtschaftliche Planung automatisierter Fertigungssysteme, München - Wien 1983.

Selman (1986)

Selman,A.L. (Hrsg.): Structure in Complexity Theory, Proceedings of the Conference of Structure in Complexity Theory, 02.-05.06.1986 in Berkeley, Berlin - Heidelberg - New York ... 1986.

Shapiro,E. (1984)

Shapiro,E.: Systolic Programming: A Paradigm of Parallel Processing; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 458-470.

Shapiro,R. (1971)

Shapiro,R.M.; Saint,M.: A New Approach to Optimization of Sequencing Decisions; in: Halpern,M.J.; Shaw,C.J. (Hrsg.): Annual Review in Automatic Programming, Vol. 6 (1970), Oxford - New York - Toronto - Sydney - Braunschweig 1971, S. 257-288.

Shapiro,R. (1975b)

Shapiro,R.M.: System Modeling with Net Structures; in: Massachusetts Computer Associates, Inc. (Hrsg.): Second Semi-Annual Technical Report (1.12.1973-31.03.1975) for the Project "Development of Theoretical Foundations for Description and Analysis of Discrete Information Systems", Wakefield 1975, S. 95-147.

Shields (1979a)

Shields,M.W.: Adequate Path Expressions; in: Kahn,G. (Hrsg.): Semantics of Concurrent Computation, Proceedings of the International Symposium, 2.-4.07.1979 in Evian, Lecture Notes in Computer Science 70, Berlin - Heidelberg - New York 1979, S. 249-265.

Shields (1979b)

Shields,M.W.; Lauer,P.E.: On the abstract specification and formal analysis of synchronization properties of concurrent systems; in: Blum,E.K.; Paul,M.; Takasu,S. (Hrsg.): Mathematical Studies of Information Processing, Proceedings of the International Conference, 23.-26.08.1978 in Kyoto, Lecture Notes in Computer Science 75, Berlin - Heidelberg - New York 1979, S. 1-32.

Shoshani (1970)

Shoshani,A.; Coffman,E.G.: Detection and Prevention of Deadlocks; in: o.V.: Proceedings of the 4th Annual Princeton Conference on Information Sciences and Systems, 26.-27.03.1970 in Princeton, Princeton 1970, S. 355-360.

Sieben (1990)

Sieben,G.; Schildbach,T.: Betriebswirtschaftliche Entscheidungstheorie, 3. Aufl., Düsseldorf 1990.

Siekmann (1982b)

Siekmann,J.H.: Einführung in die künstliche Intelligenz; in: Bibel,W.; Siekmann,J.H. (Hrsg.): Künstliche Intelligenz - Frühjahrsschule Teisendorf, 15.-24.03.1982 in Teisendorf, Informatik-Fachberichte 59, Berlin - Heidelberg - New York 1982, S. 1-60.

Sifakis (1977a)

Sifakis,J.: Use of Petri Nets for Performance Evaluation; in: Beilner,H.; Gelenbe,E. (Hrsg.): Measuring, Modelling and Evaluating Computer Systems, Amsterdam 1977, S. 75-93.

Sifakis (1978)

Sifakis,J.: Structural Properties of Petri Nets; in: Winkowski,J. (Hrsg.): Mathematical Foundations of Computer Science 1978, Proceedings of the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64, Berlin - Heidelberg - New York 1978, S. 474-483.

Sifakis (1980c)

Sifakis,J.: Deadlocks and Livelocks in Transition Systems; in: Dembinski,P. (Hrsg.): Mathematical Foundations of Computer Science 1980, Proceedings of the 9th Symposium, 1.-5.09.1980 in Rydzyna, Lecture Notes in Computer Science 88, Berlin - Heidelberg - New York 1980, S. 587-600.

Sifakis (1982)

Sifakis,J.: A Unified Approach for Studying the Properties of Transition Systems; in: Theoretical Computer Science, Vol. 18 (1982), S. 227-258.

Silva,M. (1981b)

Silva,M.: Sur le Concept de Macroplace et son Utilisation pour l'Analyse des Reseaux de Petri; in: Revue Francaise d'Automatique, d'Informatique et de Recherche Operationelle - Automatique/Systems Analysis and Control, Vol. 15 (1981), S. 335-345.

Simon,H. (1975)

Simon,H.A.; Kadana,J.B.: Optimal Problem-Solving Search: All-or-None Solutions; in: Artificial Intelligence, Vol. 6 (1975), S. 235-247. (Wiederabgedruckt in: Simon,H.A.: Models of Bounded Rationality, Volume 1: Economic Analysis and Public Policy, Cambridge (Massachusetts) - London 1982, S. 248-260.)

Simon,H. (1976b)

Simon,H.A.; Kadane,J.B.: Problems of Computational Complexity in Artificial Intelligence; in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 281-299.

Simon,H. (1979b)

Simon,H.A.: Artificial Intelligence Research Strategies in the Light of AI Models of Scientific Discovery; in: o.V.: IJCAI-79, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 20.-23.08.1979 in Tokyo, o.O. (Stanford) 1979, Vol. 2, S. 1086-1094.

Simon,H. (1983a)

Simon,H.A.: Search and Reasoning in Problem Solving; in: Pearl,J. (Hrsg.): Search and Heuristics, Reprinted from the Journal of Artificial Intelligence, Volume 21, Numbers 1,2, Amsterdam - New York - Oxford 1983, S. 7-29.

Smale (1983)

Smale,S.: The Problem of the Average Speed of the Simplex Method; in: Bachem,A.; Grötschel, M.; Korte,B. (Hrsg.): Mathematical Programming - The State of the Art - Bonn 1982, Berlin - Heidelberg - New York ... 1983, S. 530-539.

Sowa (1984)

Sowa,J.F.: Conceptual Structures: Information Processing in Mind and Machine, Reading - Menlo Park - London ... 1984.

Spur (1980)

Spur,G.; Albrecht,R.; Armbruster,N.; Badur,K.; Göhren,H.; Junike,W.; Kruska,J.; Mattle,H.-P.; Popken,W.; Prehn,W.; Rall,K.; Rittinghausen,H.; Seliger,G.; Sinnig,H.; Viehweger,B.: Realisierung eines modularen, flexiblen Fertigungssystem mit automatischer Informationsverarbeitung, Forschungsbericht KfK-PDV 195, Kernforschungszentrum Karlsruhe GmbH, Karlsruhe 1980.

Spur (1981a)

Spur,G.; Albrecht,R.; Rittinghausen,H.: Strategien zur Online-Fertigungsoptimierung; in: Zeitschrift für wirtschaftliche Fertigung, 76. Jg. (1981), S. 114-118.

Stachowiak (1973)

Stachowiak,H.: Allgemeine Modelltheorie, Wien - New York 1973.

Stadtler (1983a)

Stadtler,H.: Interaktive Lösung schlecht-strukturierter Entscheidungsprobleme - Methoden und Erkenntnisse bei der Stauung von Chemikalientankern, Dissertation, Universität Hamburg, München 1983.

Stahlknecht (1982)

Stahlknecht,P.: Der Einzug der Mathematik in die Software-Entwicklung; in: Angewandte Informatik, 24. Jg. (1982), S. 115-125.

Stallman (1977)

Stallman,R.M.; Sussman,G.J.: Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis; in: Artificial Intelligence, Vol. 9 (1977), S. 135-196.

Stallman (1979)

Stallman,R.M.; Sussman,G.S.: Problem Solving About Electrical Circuits; in: Winston,P.H.; Brown,R.H. (Hrsg.): Artificial Intelligence: An MIT Perspective, Vol 1: Expert Problem Solving - Natural Language Understanding - Intelligent Computer Coaches - Representation and Learning, Cambridge (Massachusetts) - London 1979, S. 33-91.

Starke (1979b)

Starke,P.H.: Semilinearity and Petri Nets; in: Budach,L. (Hrsg.): Fundamentals of Computation Theory - FCT'79, Proceedings of the Conference on Algebraic, Arithmetic, and Categorical Methods in Computation Theory, 17.-21.09.1979 in Berlin (Ost), Berlin (Ost), 1979, S. 423-429.

Starke (1980)

Starke,P.H.: Petri-Netze - Grundlagen, Anwendungen, Theorie, Berlin (Ost) 1980.

Staudinger (1990)

Staudinger,R.: The Design of an Industrial LP Model; in: Operations Research-Spektrum, Bd. 12 (1990), S. 173-179.

Stede (1983)

Stede,M.: Einführung in die Künstliche Intelligenz, Bd. 1: Methodische Grundlagen, Sprenglingen 1983.

Steel (1987)

Steel,S.: On Trying to Do Dependency-Directed Backtracking by Searching Transformed State Spaces (and Failing); in: Hallam,J.; Mellish,C. (Hrsg.): Advances in Artificial Intelligence, Proceedings of the 1987 AISB Conference, 06.-10.04.1987 in Edinburgh, Chichester - New York - Brisbane ... 1987, S. 207-221.

Stegmüller (1973a)

Stegmüller,W.: Unvollständigkeit und Unentscheidbarkeit - Die metamathematischen Resultate von Gödel, Church, Kleene, Rosser und ihre erkenntnistheoretische Bedeutung, 3. Aufl., Wien - New York 1973.

Stepan (1977)

Stepan,A.: Die Anwendung der Kontrolltheorie auf betriebswirtschaftliche Problemstellungen - Mit einer Einführung in die Kontrolltheorie, Meisenheim 1977.

Stepan (1988)

Stepan,A.; Fischer,E.O.: Betriebswirtschaftliche Optimierung - Einführung in die quantitative Betriebswirtschaftslehre, München - Wien 1988.

Stetter (1988)

Stetter,F.: Grundbegriffe der Theoretischen Informatik, Berlin - Heidelberg - New York ... 1988.

Stoyan (1988a)

Stoyan,H. (Hrsg.): Begründungsverwaltung - Beiträge zu einem Workshop über Reason Maintenance, am 09.10.1986 in Berlin, Proceedings, Informatik-Fachberichte 162, Berlin - Heidelberg - New York ... 1988.

Streim (1975)

Streim,H.: Heuristische Lösungsverfahren - Versuch einer Begriffsklärung; in: Zeitschrift für Operations Research, Bd. 19 (1975), S. 143-162.

Streppel (1982)

Streppel,H.: Ein Programmsystem zur interaktiven Erzeugung und Analyse von Petrinetzen, Diplomarbeit am Institut für Informatik, Universität Bonn, Bonn 1982.

Struß (1986)

Struss,P.: Multiple Representation of Structure and Function, Draft (Paper) 4/23/86, München 1986. (Anmk. des Verf.: in anderen Quellen wird der Name des Autors als "Struß" notiert; diese Schreibweise wird hier im Interesse der Kompatibilität mit anderen Referenztiteln übernommen.)

Stute (1978a)

Stute,G.; Storr,A.; Döttling,W.; Schwager,J.; Wörn,H.: Prozeßüberwachung in flexiblen Fertigungssystemen, Forschungsbericht KfK-PDV 148, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, Universität Stuttgart, o.O. (Stuttgart) 1978.

Suraj (1980)

Suraj,Z.: A Resource Allocation Problem; in: Dembinski,P. (Hrsg.): Mathematical Foundations of Computer Science 1980, Proceedings of the 9th Symposium, 1.-5.09.1980 in Rydzyna, Lecture Notes in Computer Science 88, Berlin - Heidelberg - New York 1980, S. 627-642.

Suzuki,I. (1980a)

Suzuki,I.; Murata,T.: A Method for Hierarchically Representing Large Scale Petri Nets; in: o.V.: Papers of the IEEE 1980 International Conference on Circuits and Computers, Vol. 2, Portchester 1980, S. 620-623.

Systems Modeling Corporation (o.J.a)

Systems Modeling Corporation (Hrsg.): The SIMAN Simulation Language, Sewickley o.J.

Systems Modeling Corporation (o.J.b)

Systems Modeling Corporation (Hrsg.): Cinema - simulation animation ... like you've never seen before!, Sewickley o.J.

Szlanco (1978)

Szlanco,J.: Petri nets For Proving Some Correctness Properties of Parallel Programs; in: Smedema,C.H. (Hrsg.): Real Time Programming, Proceedings of the IFAC/IFIP Workshop, 20.-22.06.1977 in Eindhoven, Oxford - New York - Toronto ... 1978, S. 75-83.

Szwarcfiter (1976)

Szwarcfiter,J.L.; Lauer,P.E.: A Search Strategy for the Elementary Cycles of a Directed Graph; in: Bit, Vol. 16 (1976), S. 192-204.

Tanaka,H. (1982)

Tanaka,H.; Amamiya,M.; Tanaka,Y.; Kadowaki,Y.; Yamamoto,M.; Shimada,T.; Sohma,Y.; Takizawa,M.; Ito,N.; Takeuchi,A.; Kitsuregawa,M.; Goto,A.: The Preliminary Research of Data Flow Machine and Data Base Machine as the Basic Architecture of Fifth Generation Computer Systems; in: Moto-oka,T. (Hrsg.): Fifth Generation Computer Systems, Proceedings of the International Conference on Fifth Generation Computer Systems, 19.-22.10.1981 in Tokyo, Amsterdam - New York - Oxford 1982, S. 209-219.

Tardos (1986)

Tardos,E.: A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs; in: Operations Research, Vol. 34 (1986), S. 250-256.

Tarjan (1973)

Tarjan,R.: Enumeration of the Elementary Circuits of a Directed Graph; in: SIAM Journal on Computing, Vol. 2 (1973), S. 211-216.

Taylor,B. (1980a)

Taylor,B.W.; Moore,L.J.: R&D Project Planning With Q-GERT Network Modeling and Simulation; in: Management Science, Vol. 26 (1980), S. 44-59.

Taylor,B. (1982)

Taylor,B.W.; Moore,L.J.: Estimating Job Flow-Times in a Job Shop for Contractually Negotiated Due-Dates; in: The Journal of the Operational Research Society, Vol. 33 (1982), S. 845-854.

Tempelmeier,H. (1989a)

Tempelmeier,H.: Simulation mit SIMAN, 6. Aufl., Institut für Betriebswirtschaftslehre, Fachgebiet Fertigungs- und Materialwirtschaft, Technische Hochschule Darmstadt, Darmstadt 1989.

Thiagarajan (1983a)

Thiagarajan,P.S.: Puzzles and Problems; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 14 (1983), S. 20-23.

Thiagarajan (1983b)

Thiagarajan,P.S.; Voss,K.: In Praise of Free Choice Nets; in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 322-341.

Thieler-Mevissen (1977)

Thieler-Mevissen,G.: The Petri Net Calculus of Predicate Logic, Interner Bericht ISF-76-09, Institut für Informationssystemforschung, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1977.

Thieler-Mevissen (1987a)

Thieler-Mevissen,G.: Existential Quantifiers in Predicate-Fact-Nets; in: Voss,K.; Genrich,H.J.; Rozenberg,G. (Hrsg.): Concurrency and Nets - Advances in Petri Nets, Berlin - Heidelberg - New York ... 1987, S. 533-553.

Tiernan (1970)

Tiernan,J.C.: An Efficient Search Algorithm to Find the Elementary Circuits of a graph; in: Communications of the ACM, Vol. 13 (1970), S. 722-726.

Törn (1981)

Törn,A.A.: Simulation Graphs: A General Tool for Modeling Simulation Design; in: Simulation, Vol. 39 (1981), Heft December, S. 187-194.

Torrigiani (1977)

Torrigiani,P.R.; Lauer,P.E.: An Object Oriented Notation for Path Expression; in: o.V.: Proceedings of the Annual Congress of Associazione Italiana per il Calcolo Automatico, Vol. 3, Pisa 1977, S. 349-371.

Torrigiani (1978)

Torrigiani,P.R.: Synchronic Aspects of Data Types: Construction of a non-algorithmic solution of the Banker's problem; in: Bracchi,G.; Lockemann,P.C. (Hrsg.): Information Systems Methodology, Proceedings, 2nd Conference of the European Cooperation in Informatics, 10.-12.10.1978 in Venedig, Lecture Notes in Computer Science 65, Berlin - Heidelberg - New York 1978, S. 560-583.

Toulotte (1979)

Toulotte,J.M.; Parsy,J.P.: A Method for Decomposing Interpreted Petri Nets and Its Utilization; in: Digital Processes, Vol. 5 (1979), S. 223-234.

Traub (1982)

Traub,J.F.; Wozniakowski,H.: Complexity of Linear Programming; in: Operations Research Letters, Vol. 1 (1982), S. 59-62.

Treleaven (1982)

Treleaven,P.C.; Brownbridge,D.R.; Hopkins,R.P.: Data-Driven and Demand-Driven Computer Architecture; in: Computing Surveys, Vol. 14 (1982), S. 93-143.

Tremolieres (1979)

Tremolieres,R.: De Nouvelles Voies pour la Simulation - Initiation aux Reseaux de Petri; in: Informatique et Gestion, No. 105 (1979), S. 60-63.

Tsichritzis (1971)

Tsichritzis,D.: Modular System Description, Technical Report No. 33, Department of Computer Science, University of Toronto, Toronto 1971.

Tsukiyama (1975)

Tsukiyama,S.; Shirakawa,I.; Ozaki,H.: A Survey: Generating All the Cycles of a Digraph; in: o.V.: 2nd USA-Japan Computer Conference, Proceedings, Tokio - Montvale 1975, S. 92-96.

Twellmann (1979)

Twellmann,W.: Beitrag zur rechnerunterstützten Planung von Fertigungssystemen der Serienfertigung, Dissertation, Universität Hnnover, Hannover 1979.

Ullman (1973)

Ullman, J.D.: Polynomial Complete Scheduling Problems; in: o.V.: Proceedings of the 4th Symposium on Operating System Principles, Yorktown Heights 1973, S. 96-101.

Ullman (1976)

Ullman, J.D.: Complexity of Sequencing Problems; in: Coffman, E.G. (Hrsg.): Computer and Job-Shop Scheduling Theory, New York - London - Sydney ... 1976, S. 139-164.

Ullrich (1976)

Ullrich, G.: Der Entwurf von Steuerstrukturen für parallele Abläufe mit Hilfe von Petri-Netzen, Dissertation an der Universität Hamburg, Hamburg 1976. (Auch veröffentlicht als: Bericht Nr. 36, Fachbereich Informatik, Universität Hamburg, Hamburg 1977.)

Vagin (1988)

Vagin, V.N.; Zakharov, V.N.; Rozenblyum, L.Y.: Logical Inference on Interpreted Petri Nets; in: Soviet Journal of Computer and Systems Sciences, Vol. 26 (1988), No. 3, S. 98-105.

Valette (1977a)

Valette, R.; Courvoisier, M.: Recherche d'un Modele Adapte aux Systemes de Commande de Processus a Evolutions Paralleles; in: Revue Francaise d'Automatique, Informatique et Techerche Operationelle - Automatique/Systems Analysis and Control, Vol. 11 (1977), S. 51-85.

Valette (1977b)

Valette, R.: An Analysis Oriented Description of Parallel Systems Allowing Timing Considerations; in: o.V.: Proceedings of the 2nd IFAC International Symposium on Discrete Systems, im März 1977 in Dresden, o.O. (1977), S. 102-111.

Valette (1978b)

Valette, R.; Diaz, M.: Top-Down Formal Specification and Verification of Parallel Control Systems; in: Digital Processes, Vol. 4 (1978), S. 181-199.

Valette (1979a)

Valette, R.: Analysis of Petri Nets by Stepwise Refinements; in: Journal of Computer and System Sciences, Vol. 18 (1979), S. 35-46.

Valette (1979b)

Valette, R.; Diaz, M.: A Methodology for Easily Provable Implementation of Synchronization Mechanisms; in: Syre, H.J. (Hrsg.): 1st European Conference on Parallel and Distributed Processing, 14.-16.02.1979 in Toulouse, Toulouse 1979, S. 156-162.

Valette (1982c)

Valette, R.: Verification des Specification d'un Automatisme; in: o.V.: Les Methodes Modernes d'Etudes et de Realisation des Automatismes, 2.-3.02.1978 in Gif-sur-Yvette, o.O. 1982, o.S.

Valiant (1978)

Valiant, L.G.: The Complexity of Combinatorial Computations: An Introduction; in: Schindler, S.; Giloi, W.K. (Hrsg.): GI - 8. Jahrestagung, 1978 in Berlin, Informatik - Fachberichte 16, Berlin - Heidelberg - New York 1978, S. 326-337.

Valk (1977a)

Valk, R.; Vidal, G.: On the Rationality of Petri Net Languages; in: Tzschach, H.; Waldschmidt, H.; Walter, H.K.-G. (Hrsg.): Theoretical Computer Science, 3rd GI Conference, 28.-30.03.1977 in Darmstadt, Lecture Notes in Computer Science 48, Berlin - Heidelberg - New York 1977, S. 319-328.

Valk (1981b)

Valk, R.: Generalization of Petri Nets; in: Gruska, J.; Chytil, M.: Mathematical Foundations of Computer Science 1981, Proceedings, 10th Symposium, 31.08.-4.09.1981 in Strbske Pleso, Lecture Notes in Computer Science 118, Berlin - Heidelberg - New York 1981, S. 140-155.

Valk (1983b)

Valk, R.; Jantzen, M.: The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets, Skript, präsentiert am 28.09.1983 in Toulouse anlässlich: 4th European Workshop on Applications and Theory of Petri Nets. (Erweiterte Fassung von Valk (1983c).)

Valk (1983c)

Valk,R.; Jantzen,M.: The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets; in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 342-363. (Kurzfassung von Valk (1983b).)

Valmari (1988a)

Valmari,A.: Error Detection by Reduced Reachability Graph Generation; in: o.V.: Application and Theory of Petri Nets - 9th European Workshop, 22.-24.06.1988 in Venedig, o.O. 1988, Vol. I, S. 95-112.

Valtorta (1983)

Valtorta,M.: A Result on the Computational Complexity of Heuristic Estimates for the A* Algorithm; in: Bundy,A. (Hrsg.): IJCAI-83, Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 08.-12.08.1983 in Karlsruhe, Los Altos 1983, Vol. 2, S. 777-779.

Van Dyke Parunak (1987)

Van Dyke Parunak,H.: Manufacturing Experience with the Contract Net; in: Huhns,M.N. (Hrsg.): Distributed Artificial Intelligence, London - Los Altos 1987, S. 285-310.

Van Emde Boas (1978)

Van Emde Boas,P.: The Connection between Model logic and Algorithmic logics; in: Winowski,J. (Hrsg.): Mathematical Foundations of Computer Science 1978, Proceedings of the 7th Symposium, 4.-8.09.1978 in Zakopane, Lecture Notes in Computer Science 64, Berlin - Heidelberg - New York 1978, S. 1-15.

Van Marcke (1987)

Van Marcke,K.: A Parallel Algorithm for Consistency Maintenance in Knowledge Representation.; in: du Boulay,B.; Hogg,D.; Steels,L. (Hrsg.): Advances in Artificial Intelligence - II, Seventh European Conference on Artificial Intelligence, ECAI-86, 20.-25.07.1986 in Brighton, Amsterdam - New York - Oxford ... 1987, S. 263-275.

Vidal-Naquet (1982b)

Vidal-Naquet,G.; Roucairaol,G.; Berthelot,G.; Memmi,G.; Sifakis,J.; Girault,C.: Extensions et Abreviations des Reseaux de Petri, D.E.A. de Systemes Informatiques, Institut de Programmation, Universite Pierre et Marie Curie Paris, Paris 1982.

Villa (1988c)

Villa,A.: Distributed Architecture for Production Planning and Control in Discrete Manufacturing; in: o.V.: 1988 International Conference on Computer Integrated Manufacturing, 23.-25.05.1988 in Troy, Washington 1988, S. 357-366.

Vincke (1986)

Vincke,P.: Analysis of multicriteria decision aid in Europe; in: European Journal of Operational Research, Vol. 25 (1986), S. 160-168.

von Bechtolsheim (1991)

von Bechtolsheim,M.; SchweichhartK.; Winand,U.: Expertensystem-Werkzeuge - Produkte, Aufbau, Auswahl, Braunschweig 1991.

von Kleist-Retzow (1991)

von Kleist-Retzow,H.; Kreifelts,T.; Kreplin,K.; Lutz,E.; Seuffert,P.; Woetzel,G.; Bauer,D.: Integrierte Post- und Vorgangsbearbeitung; in: Lutze,R.; Kohl,A. (Hrsg.): Wissensbasierte Systeme im Büro - Ergebnisse aus dem WISDOM-Verbundprojekt, München - Wien 1991, S. 231-266.

von Martial (1991a)

von Martial,F.; Victor,F.: Interaktive Planung von Bürovorgängen; in: Lutze,R.; Kohl,A. (Hrsg.): Wissensbasierte Systeme im Büro - Ergebnisse aus dem WISDOM-Verbundprojekt, München - Wien 1991, S. 313-324.

von Neumann (1958)

von Neumann,J.: The Computer and the Brain, New Haven 1958.

von Neumann-Cosel (1983)

von Neumann-Cosel,R.: Verfahren zur Lösung von Problemen mit mehrfacher Zielsetzung, Frankfurt - New York 1983.

Voss,K. (1979)

Voss,K.: Using Predicate/Transition-Nets To Model And Analyze Disrtibuted Database Systems; in: o.V.: Proceedings of COMPSAC - the IEEE Computer Society's 3rd International Computer Software and Applications Conference, Papers, Chicago 1979, S. 801-806.

Voss,K. (1983a)

Voss,K.: Cover Picture Story - Having a home is a privilege; in: Special Interest Group "Petri Nets and Related System Models" (Gesellschaft für Informatik), Newsletter 14 (1983), S. 3-4.

Webber (1983)

Webber,B.L.: Logic and Natural Language; in: Computer, Vol. 16 (1983), S. 43-46.

Weber,H. (1982)

Weber,H.H.: Khachiyani's Algorithmus; in: Zeitschrift für Operations Research, Bd. 26 (1982), S. B/229-B/240.

Weck (1991d)

Weck,M.; Lange,N.: COSMOS 2000, die modulare, offene Steuerungsarchitektur für flexible Fertigungssysteme; in: Pritschow,G.; Spur,G.; Weck,M. (Hrsg.): Leit- und Steuerungstechniken in flexiblen Produktionsanlagen, München - Wien 1991, S. 3-31.

Weck (1991e)

Weck,M.; Lopez,M.: Konfigurierbare Bedienoberflächen im Fertigungsbereich; in: Pritschow, G.; Spur,G.; Weck,M. (Hrsg.): Leit- und Steuerungstechniken in flexiblen Produktionsanlagen, München - Wien 1991, S. 115-130.

Wedde (1973)

Wedde,H.: Lose Kopplung in Informationssystemen; in: Brauer,W. (Hrsg.): GI - 3. Jahrestagung, 8.-10.10.1983 in Hamburg, Lecture Notes in Computer Science 1, Berlin - Heidelberg - New York 1973, S. 372-381.

Wedde (1975)

Wedde,H.: Lose Kopplung von Systemkomponenten, Bericht Nr. 96, Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn, Sankt Augustin 1975.

Wedde (1977)

Wedde,H.; Winkowski,J.: Determing Processes by Violations; in: Gruska,J. (Hrsg.): Mathematical Foundation of Computer Science 1977, Proceedings, 6th Symposium, 5.-9.09.1977 in Tatranska Lomnica, Lecture Notes in Computer Science 53, Berlin - Heidelberg - New York 1977, S. 549-559.

Wedde (1980a)

Wedde,H.: Fundamentals of a theory of infringements; in: Pichler,F.R.; Hanika,F. de P.(Hrsg.): Progress in Cybernetics and Systems Research, Vol. VII, Washington - New York - London 1980, S. 97-104.

Wedde (1980b)

Wedde,H.: Control of Interacting Processes on the Basis of Infringements; in: Systems Science, Vol. 6 (1980), No. 1, S. 5-22. (Überarbeitete und erweiterte Fassung von Wedde (1980a).)

Wedde (1980d)

Wedde,H.: Modelling and Analyzing the Interaction of Partially Independent System Parts; in: Proceedings of the SGSR San Francisco Meeting, Louisville 1980, S. 19-25.

Wedekind (1989c)

Wedekind,H.: Eine logische Analyse des Verhältnisses von Anwendungs- und Datenbanksystemen; in: Härder,T. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, GI/SI-Fachtagung, 01.-03.03.1989 in Zürich, Proceedings, Informatik-Fachberichte 204, Berlin - Heidelberg - New York ... 1989, S. 19-42.

Weizenbaum (1982)

Weizenbaum,J.: Die Macht der Computer und die Ohnmacht der Vernunft, 3. Aufl., Frankfurt 1982.

Wendt (1974a)

Wendt,S.: Petri-Netze und asynchrone Schaltwerke; in: Elektronische Rechenanlagen, 16. Jg. (1974), S. 208-216.

Whitehouse (1982)

Whitehouse, G.E.: Network Optimization; in: Salvendy, G. (Hrsg.): Handbook of Industrial Engineering, New York - Chichester - Brisbane ... 1982, S. 15.5.1-14.5.12.

Wicharz (1983)

Wicharz, R.E.: Die Flexibilität industrieller Produktionsplanung und -steuerung, Dissertation, Universität Köln, Düsseldorf 1983.

Wildemann (1987a)

Wildemann, H.: Investitionsplanung und Wirtschaftlichkeitsrechnung für flexible Fertigungssysteme (FFS), Stuttgart 1987.

Wildemann (1988c)

Wildemann, H.: Methodenintegration in Modularprogrammen zur Realisierung von CIM und JIT; in: Mertens, P.; Wiendahl, H.-P.; Wildemann, H. (Hrsg.): CIM-Komponenten zur Planung und Steuerung - Expertensysteme in der Produktion, München 1988, S. 39-96.

Winand (1980)

Winand, U.; Rosenstengel, B.: Interaktive Improvisation von Flugplänen auf der Basis der Petri-Netztheorie; in: Zeitschrift für Betriebswirtschaft, 50. Jg. (1980), S. 1229-1256.

Windisch (1979)

Windisch, I.: Zur Anwendung von Petri-Graphen beim rechnergestützten Entwurf digitaler Automaten, Dissertation, Technische Universität Dresden, Dresden 1979.

Winslett (1986)

Winslett, M.: Is Belief Revision Harder than You Thought?; in: o.V.: proceedings aaai-86, fifth national conference on artificial intelligence, 11.-15.08.1986 in Philadelphia, Los Altos 1986, S. 421-427.

Winston (1977)

Winston, P.H.: Artificial Intelligence, Reading - Menlo Park - London ... 1977.

Winter, R. (1977)

Winter, R.: An Evaluation Net Model For the Performance Evaluation of a Computer Network; in: Beilner, H.; Gelenbe, E. (Hrsg.): Measuring, Modelling and Evaluation Computer Systems, Proceedings of the Third International Symposium, 3.-5.10.1977 in Bonn/Bad Godesberg, Amsterdam - New York - Oxford 1977, S. 95-113.

Winter, Ro. (1991)

Winter, Ro.: Mehrstufige Produktionsplanung in Abstraktionshierarchien auf der Basis relationaler Informationsstrukturen, Dissertation, Universität Frankfurt 1989, Berlin - Heidelberg - New York ... 1991.

Witte, T. (1979a)

Witte, T.: Heuristisches Planen - Vorgehensweise zur Strukturierung betrieblicher Planungsprobleme, Wiesbaden 1979.

Wittmann (1959)

Wittmann, W.: Unternehmung und unvollkommene Information / Unternehmerische Voraussicht - Ungewißheit und Planung, Köln - Opladen 1959.

Würtz (1990)

Würtz, R.P.; Vorbrüggen, J.C.; von der Malsburg, C.: A Transputer System for the Recognition of Human Faces by Labeled Graph Matching; in: Eckmiller, R.; Hartmann, G., Hauske, G. (Hrsg.): Parallel Processing in Neural Systems and Computers, International Conference on Parallel Processing in Neural Systems and Computers, 19.-21.03.1990 in Düsseldorf, Amsterdam - New York - Oxford ... 1990, S. 37-41.

Wynne (1982)

Wynne, B.: Qualitative Modelling - A Beginning; in: Interfaces, Vol. 12 (1982), No. 2, S. 34-36.

Wysk (1992)

Wysk, R.A.; Joshi, S.; Yang, N.-S.: Scheduling and Control of Flexible Manufacturing Systems - Some Experiences and Observations; in: Fandel, G.; Gullledge, T.; Jones, A. (Hrsg.): New Directions for Operations Research in Manufacturing, Proceedings of a Joint US/German Conference, 30.-31.07.1992 in Gaithersburg, Berlin - Heidelberg - New York ... 1992, S. 216-233.

Yamaguchi, Y. (1984)

Yamaguchi, Y.; Toda, K.; Herath, J.; Yuba, T.: EM-3: A LISP-Based Data-Driven Machine; in: o.V.: Fifth Generation Computer Systems 1984 - Proceedings of the International Conference, 06.-09.11.1984 in Tokyo, Tokyo - Amsterdam 1984, S. 524-532.

Yamasaki (1981)

Yamasaki, H.: On Weak Persistency of Petri Nets; in: Information Processing Letters, Vol. 13 (1981), S. 94-97.

Yoeli (1973)

Yoeli, M.: Petri Nets and Asynchronous Control Networks, Research Report CS-73-07, Department of Applied Analysis and Computer Science, University of Waterloo, Waterloo 1973.

Yoeli (1980a)

Yoeli, M.; Ginzburg, A.: Control Nets for Parallel Processing; in: Lavington, S. (Hrsg.): Information Processing 80, Proceedings of IFIP Congress 80, 6.-9.10.1980 in Tokyo und 14.-17.10.1980 in Melbourne, Amsterdam - New York - Oxford 1980, S. 71-76.

Yu, S. (1978)

Yu, S.H.; Murata, T.: PT-Marked Graphs: A Reduced Model of Petri Nets; in: o.V.: Proceedings of the 16th Allerton Conference on Communication, Control, and Computing, New York 1978, S. 175-183.

Zachariades (1977)

Zachariades, M.: MAS. Realisation d'un langage d'aide a la description et a la conception des systemes logiques, Dissertation, Universität Grenoble, Grenoble 1977.

Zelewski (1986a)

Zelewski, S.: Das Leistungspotential der Künstlichen Intelligenz - Eine informationstechnisch-betriebswirtschaftliche Analyse, Band 1, 2 und 3, Dissertation (unter dem Titel: Das Leistungspotential der Künstlichen Intelligenz - Bestandsaufnahme und Bewertungsansätze aus informationstechnisch-betriebswirtschaftlicher Perspektive unter besonderer Berücksichtigung produktionswirtschaftlicher Aspekte -), Universität Köln 1985, Witterschlick (Bonn) 1986.

Zelewski (1986c)

Zelewski, S.: Netztheoretische Fundierung von parallelen Algorithmen für die Lösung linear-ganzzahliger OR-Modelle, Arbeitsbericht Nr. 12 (2. Aufl. des Arbeitsberichts 9/1986), Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1986.

Zelewski (1988b)

Zelewski, S.: Netztheoretische Fundierung von parallelen Algorithmen zur Bestimmung von Lösungen für linear-ganzzahlige OR-Modelle ohne Extremalziele; in: Angewandte Informatik, 30. Jg. (1988), S. 352-362.

Zelewski (1988c)

Zelewski, S.: Expertensysteme für Prozeßplanung und -steuerung in der Fabrik der Zukunft - Ein Überblick über Konzepte und erste Prototypen, Arbeitsbericht Nr. 22, Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1988.

Zelewski (1989a)

Zelewski, S.: Komplexitätstheorie - als Instrument zur Klassifizierung und Beurteilung von Problemen des Operations Research, Braunschweig - Wiesbaden 1989.

Zelewski (1989c)

Zelewski, S.: Petrinetze für die Konstruktion und Konsistenzanalyse von logisch orientierten Problembeschreibungen, Arbeitsbericht Nr. 28, Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1989.

Zelewski (1989e)

Zelewski, S.: Contributions of Net-Theory to the Modelling of OR-Problems from a Logically Based Point of View; in: Rivista di matematica per le scienze economiche e sociali, Anno 12 (1989), Fascicolo 2, S. 67-92.

Zelewski (1990a)

Zelewski, S.: PPS-Expertensysteme für die Terminfeinplanung und -steuerung - Teil 1: Konzepte; in: Information Management, 5. Jg. (1990), Heft 1, S. 56-65.

Zelewski (1991d)

Zelewski,S.: ATMS-Systeme, Arbeitsbericht Nr. 37, Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft, Universität Köln, Köln 1991.

Zervos (1977)

Zervos,C.R.; Irani,K.B.: Colored Petri Nets: Their Properties and Applications, Disseration an der University of Michigan, Technical Report No. RADC-TR-77-246, Department of Electrical Engineering, University of Michigan, Ann Arbor 1977.

Zima (1976)

Zima,H.: Betriebssysteme - Parallele Prozesse, Mannheim - Wien - Zürich 1976.

Zimmermann,H. (1987a)

Zimmermann,H.-J.: Methoden und Modelle des Operations Research, Braunschweig - Wiesbaden 1987.

Zimmermann,H. (1987b)

Zimmermann,H.-J.: Fuzzy Sets, Decision Making, and Expert Systems, Boston - Dordrecht - Lancaster 1987.

Zionts (1991)

Zionts,S.: Negotiations and MCDM: their Interrelationships; in: Korhonen,P.; Lewandowski,A.; Wallenius,J. (Hrsg.): Multiple Criteria Decision Support, Proceedings of the International Workshop, 07.-11.08.1989 in Helsinki, Lecture Notes in Economics and Mathematical Systems, Berlin - Heidelberg - New York ... 1991, S. 377-386.

Zisman (1978a)

Zisman,M.D.: Use of Production Systems for Modeling Asynchronous, Concurrent Processes; in: Waterman,D.A.; Hayes-Roth,F. (Hrsg.): Pattern-Directed Inference Systems, New York - San Francisco - London 1978, S. 53-68.

Zuse,K. (1969)

Zuse,K.: Rechnender Raum, Braunschweig 1969.

Zuse,K. (1972)

Zuse,K.: Rechnender Raum; in: Nova Acta Leopoldina - Neue Folge, Bd. 37/1 (1972), Nr. 206, S. 129-137.

**Institut für Produktionswirtschaft und Industrielle Informationswirtschaft
der Universität Leipzig**

Verzeichnis der Arbeitsberichte

- Nr. 1: ZELEWSKI, STEPHAN: Das Konzept technologischer Theorietransformationen - eine Analyse aus produktionswirtschaftlicher Perspektive, Leipzig 1994.
- Nr. 2: SIEDENTOPF, JUKKA: Anwendung und Beurteilung heuristischer Verbesserungsverfahren für die Maschinenbelegungsplanung - Ein exemplarischer Vergleich zwischen Neuronalen Netzen, Simulated Annealing und genetischen Algorithmen, Leipzig 1994.
- Nr. 3: ZELEWSKI, STEPHAN: Unternehmenskrisen und Konzepte zu ihrer Bewältigung, Leipzig 1994.
- Nr. 4: SIEDENTOPF, JUKKA: Ein effizienter Scheduling-Algorithmus auf Basis des Threshold Accepting, Leipzig 1995.
- Nr. 5: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 1: Exposition, Leipzig 1995.
- Nr. 6: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 2: Bezugsrahmen, Leipzig 1995.
- Nr. 7: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 3: Einführung in Stelle/Transition-Netze, Leipzig 1995.
- Nr. 8: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 4: Verfeinerungen von Stelle/Transition-Netzen, Leipzig 1995.
- Nr. 9: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 5: Einführung in Synthetische Netze, Teilband 5.1: Darstellung des Kernkonzepts, Leipzig 1995.
- Nr. 10: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 5: Einführung in Synthetische Netze, Teilband 5.2: Auswertungsmöglichkeiten, Leipzig 1995.
- Nr. 11: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 6: Erweiterungen von Synthetischen Netzen, Leipzig 1995.
- Nr. 12: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 7: Fallstudie, Leipzig 1995.
- Nr. 13: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 8: Charakterisierung des Petrinetz-Konzepts, Leipzig 1995.
- Nr. 14: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 9: Beurteilung des Petrinetz-Konzepts, Leipzig 1995.
- Nr. 15: ZELEWSKI, STEPHAN: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Band 10: Petrinetz-Literatur, Leipzig 1995.

Verzeichnis der Arbeitsberichte

- Nr. 16: SIEDENTOPF, JUKKA: An Efficient Scheduling Algorithm Based upon Threshold Accepting, Leipzig 1995.
- Nr. 17: SIEDENTOPF, JUKKA: The Threshold Waving Algorithm for Job Shop Scheduling, Leipzig 1995.
- Nr. 18: ZELEWSKI, STEPHAN: Diskussionspapier zum Text "Zur wirtschaftlichen und sozialen Lage in Deutschland" einer evangelisch-katholischen Arbeitsgruppe, Leipzig 1995.
- Nr. 19: SCHIMMEL, KATRIN; ZELEWSKI, STEPHAN: Untersuchung alternativer Auktionsformen hinsichtlich ihrer Eignung zur Koordination verteilter Agenten auf Elektronischen Märkten, Leipzig 1996.
- Nr. 20: SIEDENTOPF, JUKKA: Feinterminierung unter restriktiven Laufzeitanforderungen - Ein exemplarischer Vergleich lokaler Suchverfahren (Teil I), Leipzig 1996.
- Nr. 21: ZELEWSKI, STEPHAN: Strukturalistische Rekonstruktion von ökologisch induzierten Entwicklungen der produktionswirtschaftlichen Theoriebildung, Leipzig 1996.
- Nr. 22: RÖBLER, HENRIK; SCHIMMEL, KATRIN: Zur Animation und Simulation hierarchischer Petrinetze., Leipzig 1996.
- Nr. 23: RÖBLER, HENRIK; WURCH, MAIK: Implementierung des Modells eines Flexiblen Fertigungssystems, Teilbände 1-3, Leipzig 1996.
- Nr. 24: SCHIMMEL, KATRIN: Abstimmung der Implementierungssoftware INCOME/STAR. Bericht zu Phase 1 des Projekts PEMVEK, Leipzig 1996/ 2. Auflage 1997.
- Nr. 25: WURCH, MAIK: Modellierung eines Flexiblen Fertigungssystems sowie von Produktionsaufträgen. Bericht zu den Phasen 2 und 3 des Projekts PEMVEK, Leipzig 1996.
- Nr. 26: SCHIMMEL, KATRIN: Der Einsatz elektronischer Märkte zur Koordination in Flexiblen Fertigungssystemen, Leipzig 1996.
- Nr. 27: TÖPFER, ANDREAS: Vergleichende Wirtschaftlichkeitsbetrachtung von Windkraftanlagen im Raum Halle/Leipzig - Ergebniszusammenfassung, Leipzig 1996.
- Nr. 28: WURCH, MAIK: Implementierung von Vickrey-Auktionen mit Hilfe von Petrinetzen, Leipzig 1996.
- Nr. 29: WURCH, MAIK: Coordinating Electronic Markets by Auctions, Leipzig 1996.
- Nr. 30: SCHIMMEL, KATRIN; WURCH, MAIK: Simulation eines Koordinations-Moduls in einem Flexiblen Fertigungssystem, Leipzig 1996.
- Nr. 31: RÖBLER, HENRIK: XPNC - Auswahltool für parallele Schaltentscheidungen bei der Simulation von Petrinetzen, Leipzig 1997.
- Nr. 32: ZELEWSKI, STEPHAN: Handelsinformationssysteme - erweiterte Fassung einer Rezension, Leipzig 1997.

Verzeichnis der Arbeitsberichte

- Nr. 33: ZELEWSKI, STEPHAN: Erfahrungen mit Höheren Petrinetzen bei der Modellierung von Prozeßkoordinierungen in komplexen Produktionssystemen. Bericht zu Phase 7 des Projekts PEMVEK, Leipzig 1997. [in Arbeit]
- Nr. 34: ZELEWSKI, STEPHAN: Optimierung in Petrinetz-Modellen - eine Analyse aus betriebswirtschaftlicher Sicht, Leipzig 1997. [in Arbeit]
- Nr. 35: WURCH, MAIK: Simulation von Koordinations-Modulen unter Berücksichtigung strategischen Agentenverhaltens, Leipzig 1997. [in Arbeit]
- Nr. 36: SCHIMMEL, KATRIN: Komponente für Erreichbarkeitsanalysen. Bericht zu Phase 6 des Projekts PEMVEK, Leipzig 1997.
- Nr. 37: WURCH, MAIK: Modellierung der Prozeßkoordinierung. Bericht zu Phase 4 des Projekts PEMVEK, Leipzig 1997. [in Arbeit]
- Nr. 38: BODE, JÜRGEN; FUNG, RICHARD Y.K.: Integrating Cost Considerations in Quality Function Deployment, Leipzig 1997.