

Matrizen- und zustandsraumreduzierende Verfahren zur Leistungsbewertung großer stochastischer Petrinetze

vorgelegt von
Dipl.-Inform. Jörn Freiheit

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr. rer. nat. Klaus Obermayer

Berichter: Prof. Dr.-Ing. Günter Hommel

Berichter: Prof. Dr.-Ing. Reinhard German

Tag der wissenschaftlichen Aussprache: 5. Juli 2002

Berlin 2002
D 83

Technische Universität Berlin
Institut für Technische Informatik



**Matrizen- und zustandsraumreduzierende
Verfahren zur Leistungsbewertung
großer stochastischer Petrinetze**

Jörn Freiheit

Zusammenfassung

Viele qualitative und quantitative Analysemethoden von Petrinetzmodellen basieren auf der Berechnung aller Modellzustände. Für sehr große Modelle ist eine Berechnung wichtiger Eigenschaften aufgrund ihrer zu großen Zustandsräume oft unmöglich. Diese Restriktion ist als das Problem der Zustandsraumexplosion bekannt. Sehr große Zustandsräume ziehen darüber hinaus die Behandlung sehr großer Matrizen nach sich. Während der numerischen Analyse muß nach der Ermittlung aller Zustände eines Petrinetzmodells eine quadratische Matrix erstellt werden, deren Dimension von der Anzahl der ermittelten Zustände ist.

Diese Arbeit untersucht matrizen- und zustandsraumreduzierende Verfahren zur Überwindung des Problems der Zustandsraumexplosion. Durch die in dieser Arbeit beschriebenen Verfahren sind deutlich größere Modelle als bisher analysierbar. Die Möglichkeiten und Grenzen dieser Verfahren werden untersucht und erläutert. Anhand von Beispielen werden die beschriebenen Verfahren zur Reduzierung von Matrizen und Zustandsräumen vorgestellt.

Kern der Arbeit ist eine neue Aggregierungsmethode. Unter Aggregierung wird dabei der Prozeß verstanden, ein großes, komplexes Modell in ein kleineres, weniger komplexes Modell zu überführen. Bedingung einer geeigneten Aggregierung ist die Bewahrung wichtiger Eigenschaften des Originalmodells. Zu diesen Eigenschaften gehören wichtige strukturelle Merkmale eines Modells. In dieser Arbeit werden solche strukturellen Merkmale erläutert und Verfahren vorgestellt, die deren effiziente Ermittlung ermöglichen. Es wird gezeigt, daß die in dieser Arbeit präsentierte Aggregierungsmethode die für die numerische Analyse wichtigen strukturellen Eigenschaften bewahrt.

Eine Bewahrung zeitbehafteter Eigenschaften von Modellen bei deren Aggregierung wurde in bisherigen Untersuchungen kaum beachtet. In dieser Arbeit wird ein Graphen basiertes Verfahren vorgestellt, mit welchem häufig auftretende Strukturen in stochastischen Petrinetzen zu einer zeitbehafteten Transition mit einer markierungsabhängigen Schaltzeit aggregiert werden. Die dadurch erzeugten aggregierten Modelle mit einem deutlich reduzierten Zustandsraum weisen für wichtige Leistungsmaße ein ähnliches, in manchen Fällen gar gleiches Verhalten auf, wie das Originalmodell.

Andere bekannte Verfahren zur Angleichung des Zeitverhaltens des aggregierten Modells an das Originalmodell basieren dagegen auf aufwendigen iterativen numerischen Analysen. Durch die Vermeidung eines solchen Iterationsverfahrens läßt sich die in dieser Arbeit vorgestellte Aggregierungsmethode in bekannte approximative Analyseverfahren integrieren. Die Anwendung einer solchen Integration wird demonstriert. Abschließend wird anhand einer Reihe von Beispielen die Praktikabilität der hier vorgeschlagenen Aggregierungsmethode experimentell untersucht.

Abstract

The generation of state spaces for discrete-state models, such as stochastic Petri nets, is a necessary step for many types of studies, like the numerical analysis of the underlying Markov chain. For large stochastic Petri net models, as they typically arise from real systems, the enormous state spaces make their analysis tedious, if not impossible. This is known as the state space explosion problem. Moreover, large state spaces entail the manipulation of large matrices. The dimension of the quadratic matrix, required for the numerical analysis, corresponds to the size of the state space.

In order to cope with related problems, this thesis investigates methods which reduce the sizes of the involved state spaces and matrices. The proposed techniques allow to analyze significantly larger models as compared to conventional approaches. Along the theoretical elaboration various examples illustrate the applicability and limits of the introduced procedures.

The heart of this thesis is a new aggregation method. The process of aggregation transforms a large complex model into a smaller simpler one. A suitable aggregation method has to preserve important characteristics of the original model. For a consistent model in this sense, these characteristics must include both structural and time-dependent properties.

Several structural properties are introduced in this thesis, together with procedures to compute them efficiently. A proof for the consistency of the aggregation method with respect to the structural properties is given.

Preservation of time-dependent characteristics during the aggregation has received only little attention in research activities. This thesis presents novel, graph-based techniques to aggregate frequently occurring Petri net structures into a single marking-dependent timed transition. The resulting models with drastically reduced state spaces deliver approximate – in many cases even exact – values for important measures like mean values, throughputs, etc.

In general, this approach does not require iterations as often used in other aggregation/decomposition methods. These methods usually adjust different aggregated versions of the original model based on selected measures in an iterative and expensive manner. In fact, the specific feature of the novel techniques allows to integrate them into known iterative methods. The resulting acceleration is demonstrated in an application example. Furthermore, a series of experiments plumbs the practicability of the aggregation techniques presented in this thesis.

Danksagung

Die vorliegende Arbeit wäre ohne die Hilfe einer Vielzahl von Personen aus meinem beruflichen und privaten Umfeld nicht dieselbe. Ihnen möchte ich an dieser Stelle für die mir zu Teil gewordene Unterstützung danken.

Zunächst danke ich Prof. Günter Hommel für die wertvollen Hinweise und Anregungen während der Entstehung dieser Arbeit. Als Betreuer während meiner Forschungstätigkeit an seinem Lehrstuhl stand er mir jederzeit als wertvoller Ansprech- und Diskussionspartner zur Verfügung. Ein herzlicher Dank gilt Prof. Reinhard German, den ich als kollegialen Partner kennen und schätzen lernte. Seine Hinweise erbrachten neue Forschungsaspekte, die die Qualität der Arbeit weiter verbessern halfen. Ganz besonders möchte ich meinen langjährigen Büronachbarn Armin Zimmermann und Armin Heindl danken. Ohne Armin Zimmermann gäbe es diese Arbeit nicht. Er hat mich am Anfang meiner Forschungstätigkeit auf die richtigen Pfade geleitet und war mir stets freundschaftlicher Helfer und Motivator. Armin Heindl bin ich für die Beantwortung meiner ständigen und oftmals nervigen Fragen dankbar. Vielen Aspekten dieser Arbeit fehlte ohne Armins Hilfe die Eleganz. Ich danke Juliane, Dietmar, Marcus und Alexander. Ohne Alex gäbe es keine neues Programm zur Berechnung von Stellen-Invarianten in TimeNET und TimeNET hätte nicht die von mir während dieser Arbeit dringend benötigte Stabilität. Ich danke Maxim, Johannes, Felix, Witold und Xirui für wichtige Erkenntnisse, Implementierungen und anregende Diskussionen.

Der speziellste Dank gilt Daniela, Mats, Finn und meinen Eltern.

Inhaltsverzeichnis

1	Einleitung	i
1.1	Ziel der Arbeit	ii
1.2	Aufbau der Arbeit	iii
2	Modellierung mit Petrinetzen	1
2.1	Stellen, Transitionen und Marken	1
2.2	Formale Beschreibung von Petrinetzen	4
2.3	Abräum- und hemmende Kanten	5
2.4	Farbige Petrinetze	8
2.4.1	Spezielle farbige Petrinetze	10
2.5	Stochastische Petrinetze	15
2.5.1	Zeitkontinuierliche Markowketten	16
2.5.2	Ein Beispielmodell	19
3	Eigenschaften von Petrinetzmodellen	25
3.0.3	Ein komplexeres Beispiel	25
3.0.4	Lebendigkeit, Beschränktheit und Reversibilität	27
3.0.5	Stellen-Invarianten, Fallen und Siphons	28
3.1	Lineare Algebra für Petrinetze und deren Eigenschaften	30
3.1.1	Anwendung am Beispiel	30
3.1.2	Markierungsgleichung	32
3.1.3	Notwendig aber nicht hinreichend	33
3.1.4	Basisalgorithmus zur Berechnung von Stellen-Invarianten	34
3.2	Effizienter Algorithmus zur Berechnung von Stellen-Invarianten	36
3.2.1	Anwendung am Beispiel	37

3.2.2	Die doppelte Beschreibungsmethode	40
3.2.3	Doppelte Beschreibungsmethode für die Berechnung von Fallen und Siphons	42
3.2.4	Begriffsklärungen	43
4	Matrixreduzierende Verfahren	45
4.1	Kronecker-Algebra	45
4.1.1	Ein einführendes Beispiel	46
4.1.2	Allgemeine Anwendung der Kronecker-Operatoren	49
4.1.3	Zugriff auf einzelne Elemente der Generatormatrix des Gesamtsystems	50
4.1.4	Falsche Zustände und weitere Nachteile	52
4.1.5	Integration der Kronecker-Algebra basierten Analyse in TimeNET	53
4.2	Überblick über weitere matrixreduzierende Verfahren	55
4.2.1	Multi-Terminal- und Matrixdiagramme	55
5	Zustandsraumreduzierende Verfahren	61
5.1	MIMO-Graphen basierte Aggregierung	62
5.1.1	Ein einführendes Beispiel	63
5.1.2	Formelle Beschreibung farbloser MIMO-Graphen	65
5.1.3	Transformation eines farblosen Petrinetzes in einen MIMO- Graphen	67
5.1.4	Aggregierung des farblosen MIMO-Graphen	68
5.1.5	Löschen von SISO-Elementen in MIMO-Graphen	71
5.1.6	Rücktransformation des aggregierten MIMO-Graphen in ein Petrinetz	71
5.2	Konsistenz der MIMO-Graphen basierten Aggregierung	72
5.2.1	Konsistenz beim Löschen von SISO-Elementen	72
5.2.2	Konsistenz der Verschmelzung von Transitionen	74
5.2.3	Konsistenz der Verschmelzung von Stellen	77
5.2.4	Vergleich mit bekannten Reduktionsregeln	80
5.3	MIMO-Graphen basierte Aggregierung spezieller farbiger Petrinetze	89
5.4	Teilmodelle: Zerlegung und Aggregierung	91

5.4.1	Zerlegung großer Modelle in Teilmodelle	92
5.4.2	Aggregierung von Teilmodellen	93
5.5	Berechnung der Raten verschmolzener Transitionen	95
5.5.1	Syntaktische Beschreibung der Schaltraten	96
5.5.2	Semantik der Operatoren zur Berechnung der Schaltraten .	103
5.5.3	Ursachen des Approximationsfehlers	118
5.6	Überblick über weitere zustandsraumreduzierende Verfahren . . .	121
6	Anwendungen	125
6.1	Anwendungsbeispiele	126
6.2	Eine iterative numerische Analysemethode	141
6.2.1	Zerlegung und Aggregierung des Beispielmodells	142
6.2.2	Iterative Analyse des Beispielmodells	144
7	Zusammenfassung	151
7.1	Ausblick	153

Kapitel 1

Einleitung

Diese Arbeit handelt von Petrinetzen. Es wird gezeigt, daß Petrinetze ein geeignetes Hilfsmittel sind, um sehr komplexe verteilte Systeme zu modellieren. Neben der Eignung zur Modellierung zeichnen sich Petrinetze dadurch aus, daß sie viele mathematisch fundierte Verfahren zur Analyse von modellierten Systemen bereitstellen. Dadurch können sowohl qualitative als auch quantitative Aussagen über die modellierten Systeme getroffen werden.

Petrinetze beschreiben die *lokalen* Zustände innerhalb eines Systems. Die wichtigsten Verfahren zur Petrinetz basierten Analyse treffen jedoch Aussagen über die *globalen* Zustände des modellierten Systems. Die globalen Zustände lassen sich aus einer Kombination der lokalen Zustände unter Berücksichtigung des dynamischen Verhaltens der Petrinetze berechnen. Bei Erweiterungen der Petrinetzmodelle wächst jedoch die Anzahl der globalen Zustände im Vergleich zur Anzahl der lokalen Zustände exponentiell. Petrinetzmodelle realer Systeme sind oft zu groß, um analysiert werden zu können. Dieses Problem ist als *Problem der Zustandsraumexplosion* bekannt. Ein Gegenstand dieser Arbeit ist, Verfahren vorzustellen, die das Problem der Zustandsraumexplosion überwinden helfen. Mit diesen Verfahren sind deutlich größere Systeme analysierbar als mit den bisher bekannten Standardverfahren.

Einen wichtigen Aspekt bei der Modellierung technischer Systeme stellt das Zeitverhalten der modellierten Systeme dar. Petrinetze lassen sich geeignet erweitern, um den Zeitaspekt zu modellieren. Die bekannteste Klasse von zeiterweiterten Petrinetzen ist die der *stochastischen Petrinetze*. Neben Zeitaspekten sind in stochastischen Petrinetzen auch Wahrscheinlichkeiten des dynamischen Verhaltens der Systeme modellierbar. Die Analyse des Zeitverhaltens der modellierten Systeme wird unter dem Begriff *numerische Analyse* zusammengefaßt. Die numerische Analyse basiert jedoch ebenfalls auf der Gesamtmenge der globalen Zustände der Systeme. Darüber hinaus muß für die numerische Analyse gar eine quadratische

Matrix von der Dimension der Anzahl der globalen Zustände erstellt werden. Deshalb werden neben den Verfahren zur Reduzierung der Anzahl der globalen Zustände in dieser Arbeit Verfahren zur Reduzierung und der geeigneten Verwaltung großer Matrizen vorgestellt.

1.1 Ziel der Arbeit

In dieser Arbeit werden matrizen- und zustandsraumreduzierende Verfahren vorgestellt. Detailliert wird die bekannte matrixreduzierende Methode auf Basis der *Kronecker-Algebra* beschrieben. Weitere bekannte matrixreduzierende Verfahren, die insbesondere auf *Multi-Terminal binären Entscheidungsdiagrammen* und *Matrixdiagrammen* basieren, werden erläutert.

Kern der Arbeit ist die sogenannte *MIMO-Graphen basierte Aggregation*. Als Aggregation wird der Prozeß verstanden, der ein großes, komplexes Modell in ein kleineres Modell transformiert. Ziel der Aggregation ist es, durch die Analyse des kleineren Modells Aussagen über das wegen seiner Größe nicht analysierbare Originalmodell treffen zu können. Die MIMO-Graphen basierte Aggregation erlaubt eine automatische Aggregation eines Originalmodells in ein analysierbares Petrinetzmodell. Die Konsistenz der MIMO-Graphen basierten Aggregation bezüglich der Bewahrung der Lebendigkeits- und Beschränktheitseigenschaft des Originalmodells wird gezeigt. Das bedeutet, daß ein lebendiges und beschränktes Originalmodell auch stets in ein lebendiges und beschränktes Modell aggregiert wird. Diese Bewahrung wichtiger struktureller Eigenschaft ist eine wichtige Voraussetzung zur geeigneten Aggregation stochastischer Modelle.

Eine Besonderheit der MIMO-Graphen basierten Aggregation ist die strukturbasierte Behandlung zeitbehafteter Transitionen bei der Aggregation. Wird ein großes Petrinetzmodell in ein kleineres aggregiert, so ändert sich dabei das Zeitverhalten des Modells. Ähnlich der Bewahrung struktureller Eigenschaften wäre eine Bewahrung des Zeitverhaltens eines stochastischen Modells bei der Aggregation wünschenswert. Im Rahmen dieser Arbeit wird erstmals eine strukturbasierte Angleichung des Zeitverhaltens des aggregierten Modells an das entsprechende Originalmodell vorgeschlagen. Als Leistungsmaß für die Angleichung des zeitlichen Verhaltens der Modelle wurde der Durchsatz gewählt. Die MIMO-Graphen basierte Aggregation ist ein approximatives Verfahren. Das heißt, daß der Durchsatz des aggregierten Modells dem des Originalmodells angenähert wird. Für einige der in dieser Arbeit diskutierten Modelle bleibt der Durchsatz nach der Aggregation gar exakt. Erstmals ermöglicht die hier vorgestellte MIMO-Graphen basierte Aggregation die Annäherung des Zeitverhaltens bereits während des Prozesses der Aggregation. Dabei werden Strukturinformationen ausgewertet, die eine Beschreibung des Zeitverhaltens von aggregierten

gierten Transitionen ermöglichen.

Anhand von Beispielen und Anwendungen werden die Möglichkeiten und Einschränkungen der MIMO-Graphen basierten Aggregation demonstriert. Es wird auf die Problematik eingegangen, daß nur Transitionen mit einer negativ-exponentiellen Verteilungsfunktion zugeordneten Schaltzeit behandelbar sind, daß eine beliebige Struktur noch nicht genügend auswertbar und eine Abschätzung des Approximationsfehlers nicht möglich ist.

1.2 Aufbau der Arbeit

Diese Arbeit ist wie folgt gegliedert. In Kapitel 2 werden Petrinetze eingeführt. Es werden die Grundbegriffe der Petrinetztheorie vorgestellt. Anhand von Beispielen wird die Modellierung mit Petrinetzen erläutert. Es wird gezeigt, daß für die Modellierung großer Systeme die Verwendung *farbiger* Petrinetze Vorteile bietet. Farbige Petrinetze werden erläutert und eine Klasse *spezieller* farbiger Petrinetze vorgestellt, die insbesondere die Modellierung von Fertigungssystemen erleichtern. Am Ende des Kapitels 2 werden die Konzepte stochastischer Petrinetze erläutert.

Im Kapitel 3 werden die analysierbaren qualitativen und quantitativen Eigenschaften von Petrinetzmodellen beschrieben. Es werden die grundsätzlichen Ideen zur Berechnung dieser Eigenschaften vorgestellt. Darüber hinaus wird ein neues Verfahren zur Berechnung wichtiger struktureller Eigenschaften von Petrinetzmodellen auf der Basis von Methoden der linearen Algebra und der konvexen Geometrie präsentiert. Mit diesem Verfahren wird die sehr effiziente Berechnung von *Stellen-Invarianten*, *Fallen* und *Siphons* ermöglicht. Zusätzlich läßt sich dieses Verfahren zur Abschätzung der Erreichbarkeit bestimmter Zustände und der Größe des Zustandsraumes einsetzen.

Kapitel 4 beschreibt matrixreduzierende Verfahren. Zentraler Inhalt dieses Kapitels ist ein matrixreduzierendes Verfahren auf der Basis der Kronecker-Algebra. Anhand von Beispielen wird die Anwendung von Kronecker-algebraischen Methoden erläutert und bisher wenig dokumentierte Details für die Implementierung dieser Methoden vorgestellt. Abschließend werden jüngste Arbeiten aus der Literatur erläutert, in denen Matrixdiagramme verwendet werden, um eine effizientere Speicherung großer Matrizen zu ermöglichen.

Kapitel 5 stellt Methoden zur Überwindung des Problems der Zustandsraumexplosion vor. Darin wird eine neu entwickelte Methode zur Aggregation komplexer Modelle vorgestellt. Die Anwendung und Effizienz dieser hier vorgestellten sogenannten *MIMO-Graphen basierten* Aggregation für die qualitative und quantitative Analyse von Petrinetzmodellen wird anhand von Beispielen verdeutlicht. Diese Arbeit stellt erstmals Formeln zur Aggregation komple-

xer Strukturen stochastischer Petrinetze vor. Mit diesen Formeln ist es möglich, komplexere Strukturen derart zu aggregieren, daß das aggregierte Modell ein dem Originalmodell äquivalentes Zeitverhalten besitzt. Außerdem wird bewiesen, daß die MIMO-Graphen basierte Aggregation wichtige der in Kapitel 3 vorgestellten Eigenschaften bewahrt. Das bedeutet, daß diese Eigenschaften sowohl das originale als auch das aggregierte Modell besitzen. Den Abschluß des Kapitels 5 bildet wiederum die Präsentation aus der Literatur bekannter Verfahren zur Reduzierung des Zustandsraumes.

Kapitel 6 präsentiert die Anwendungen einiger der in dieser Arbeit vorgestellten Verfahren an verschiedenen Modellen. Dabei liegt der Schwerpunkt auf der Anwendung der MIMO-Graphen basierten Aggregation. Dadurch, daß das Zeitverhalten aggregierter Modelle bereits während der Aggregation an das der Originalmodelle angeglichen wird, ist die MIMO-Graphen basierte Aggregation in bekannte iterative Approximationsverfahren integrierbar. Anhand eines realistischen Modells wird diese Integration in eine iterative Analysemethode demonstriert. Für die werkzeugunterstützte Analyse von Petrinetzen wird TimeNET verwendet. TimeNET ist ein Modellierungs- und Analysewerkzeug für stochastische Petrinetze. Sowohl das in Kapitel 3 beschriebene neue Verfahren zur Berechnung struktureller Eigenschaften der Modelle als auch die Kronecker-basierte Analyse und die MIMO-Graphen basierte Aggregation wurden im Rahmen dieser Arbeit in TimeNET integriert.

Kapitel 2

Modellierung mit Petrinetzen

Gegenstand der Informatik sind Berechnungsvorschriften, auch Algorithmen genannt. Einen intuitiven Berechnungsbegriff gibt es bereits, seit erste mathematische Operatoren, wie die Addition und Subtraktion und deren Anwendungen beschrieben wurden. Jedoch gehen die ersten formalen Definitionen des Berechenbarkeitsbegriffes erst auf das Jahr 1936 zurück. Der bekannteste Algorithmusbegriff stammt von A.M. Turing, der die sogenannte *Turingmaschine* entwickelte. Seine Intention war, eine formal beschreibbare Maschine anzugeben, mit der sich jeder beliebige intuitive Berechnungsprozeß beschreiben läßt. Die (nicht beweisbare) Churchsche These behauptet Turings Erfolg. C.A. Petri griff diese Idee auf und beschäftigte sich mit der Frage der Skalierbarkeit der Turingmaschine. Er erkannte, daß sich die Turingmaschine nicht als Modell für beliebig komplexe Algorithmen bzw. Algorithmen angewendet auf beliebig komplexe Systeme eignet, da ihre Konfiguration per Definition stets eine „Momentaufnahme“ des Gesamtsystems ist, auch wenn diese Momentaufnahme durch die Größe des Zustandes praktisch unmöglich ist. Obwohl bspw. dem Zählen der Weltbevölkerung ein sehr simpler Algorithmus zugrunde liegt, ist es praktisch unmöglich, eine exakte Bestandsaufnahme zu leisten. Er kam zu dem Schluß, daß die Welt sich zwar durch ihre Zustände und Zustandsübergänge beschreiben läßt, ein globaler Zustand oft jedoch nur theoretisch existiert. Tatsächlich aber ist die Welt verteilt!

2.1 Stellen, Transitionen und Marken

Petrinetze sind eine sehr einfache Modellierungsmethode. Die fundamentale Modellierungsidee besteht in der Beschreibung *lokaler Zustände* und *lokaler Aktionen* und ihrer gegenseitigen Beeinflussung. Die Einfachheit der Modellierung wird

unterstützt durch eine grafische Veranschaulichung mit wenigen grafischen Elementen. Lokale Zustände werden mit Kreisen, lokale Aktionen mit Rechtecken dargestellt. Zur Darstellung ihrer Beziehungen untereinander werden gerichtete Kanten benutzt. Lokale Aktionen können nur dann stattfinden, wenn eine bestimmte Menge von lokalen Zuständen erreicht sind.

Abbildung 2.1 zeigt das Petrinetzmodell eines Nachfrage-Angebot-Beispiels.

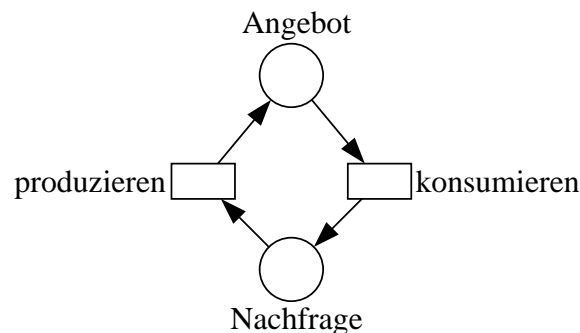


Abbildung 2.1: Ein Nachfrage-Angebot-Beispiel

Das modellierte System zeigt durch die Richtung der Kanten, daß auf Nachfrage durch produzieren reagiert wird und daß bei vorhandenem Angebot konsumiert werden kann. Die Transitionen produzieren und konsumieren stellen lokale Aktionen dar, während Angebot und Nachfrage lokale Zustände repräsentieren.

Marken

Das Petrinetzmodell ist jedoch noch nicht vollständig. Es lassen sich noch keine Aussagen über den „derzeitigen“ lokalen Zustand des modellierten Systems treffen, d.h. es fehlen Informationen darüber, ob gerade der Angebot- oder der Nachfragezustand erreicht ist. Dazu werden sogenannte *Marken* eingeführt. Eine Marke wird grafisch als schwarzer Punkt in einer Stelle dargestellt. Liegt eine Marke auf einer Stelle, so ist diese Stelle *markiert*. Eine markierte Stelle beschreibt, daß der von der Stelle repräsentierte lokale Zustand erfüllt ist. In Abbildung 2.2 ist die Stelle Nachfrage markiert.

Schalten von Transitionen

In dem Zustand Nachfrage kann nur produziert, nicht aber konsumiert werden, da es kein Angebot gibt. In einem Petrinetz kann nur dann eine Transition schalten, wenn alle Stellen in ihrem *Vorbereich* markiert sind. Der Vorbereich einer Transition sind alle Stellen, von denen aus eine Kante zur Transition verläuft.

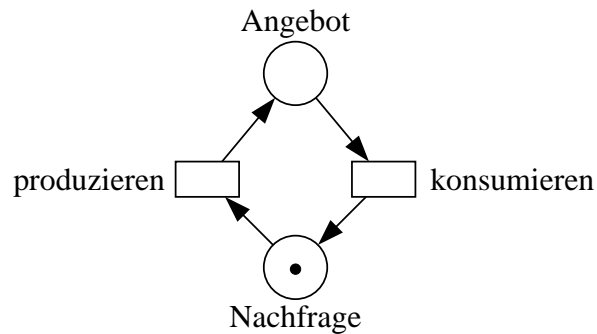


Abbildung 2.2: Petrinetzmodell des Nachfrage-Angebot-Beispiels

Der *Nachbereich* einer Transition dagegen sind alle die Stellen, zu denen von der Transition aus eine Kante verläuft. Der *Vorbereich* der Transition *produzieren* ist die mit *Nachfrage* beschriftete Stelle, der *Nachbereich* dieser Transition ist die mit *Angebot* beschriftete Stelle. Beim Schalten einer Transition werden aus allen Stellen des Vorbereiches eine Marke abgezogen und alle Stellen des Nachbereiches werden markiert.

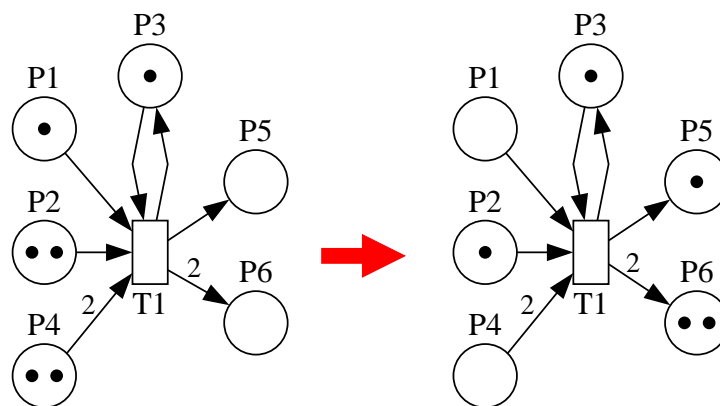


Abbildung 2.3: Schalten einer Transition

Abbildung 2.3 veranschaulicht die Schaltsemantik einer Transition. Die linke Seite der Abbildung zeigt den Zustand des Petrinetzes vor dem Schalten der Transition $T1$. Die Stellen $P1$, $P2$, $P3$ und $P4$ sind der Vorbereich, während $P3$, $P5$, $P6$ den Nachbereich der Transition $T1$ bilden. Alle Stellen des Vorbereiches sind markiert. In den Stellen $P1$ und $P3$ befinden sich eine, in $P2$ und $P4$ jeweils zwei Marken. An der Kante von $P4$ zu $T1$ steht eine 2. Sie bezeichnet die *Kardinalität* der Kante. Diese Kardinalität besagt, daß die Transition $T1$ beim Schalten zwei Marken aus der Stelle $P4$ entnimmt. Unbeschriftete Kanten haben die Kardinalität eins. Die rechte Seite des Bildes gibt den Zustand des Netzes nach Schalten

der Transition T1 an. Auf alle Stellen im Nachbereich werden durch das Schalten soviele Marken gelegt, wie es die Kardinalität der die Transition und die Stelle verbindenden Kante aussagt. Die Stellen P1 und P4 sind nach dem Schalten unmarkiert, P2 enthält noch eine Marke, da nur eine der beiden Marken durch das Schalten abgezogen wurde. Die Stelle P3 enthält eine Marke, da sie sowohl im Vor- als auch im Nachbereich der Transition liegt. Auf die Stelle P6 wurden entsprechend der Kantenkardinalität zwei Marken gelegt.

2.2 Formale Beschreibung von Petrinetzen

In diesem Abschnitt werden Petrinetze formal definiert. Zur Einführung in die Theorie von Petrinetzen sei auf [115, 123, 104] verwiesen.

2.1 Definition (Petrinetz)

Ein Petrinetz ist als ein 5-Tupel $N = (P, T, F, K, m_0)$ beschreibbar, wobei die Stellen P und die Transitionen T endliche und disjunkte Mengen, die Menge der Kanten $F \subseteq (P \times T) \cup (T \times P)$, die Kardinalitäten K eine Abbildung $F \rightarrow \mathbb{N} \setminus 0$ und die initiale Markierung m_0 eine Abbildung $P \rightarrow \mathbb{N} \cup 0$ sind.

Für $f \notin F$ sei $K(f) = 0$ definiert. Für jedes $x \in (P \cup T)$ sei $\bullet x = \{y \mid (y, x) \in F\}$ der *Vorbereich* sowie $x^\bullet = \{y \mid (x, y) \in F\}$ der *Nachbereich*.

Eine Stelle mit mehreren Transitionen im Nachbereich wird *Konfliktstelle* genannt. Die Transitionen im Nachbereich einer Konfliktstelle stehen zueinander in *Konflikt*. Transitionen, deren Vorbereich derart markiert sind, daß sie schalten könnten, besitzen *Konzession*. Jede Transition, die unter einer Markierung Konzession besitzt, kann schalten. Wenn zwei zueinander in Konflikt stehende Transitionen unter einer Markierung Konzession besitzen, kann oft nur eine der beiden schalten, während dadurch die andere Transition ihre Konzession verliert.

2.2 Definition (Schaltregel)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. Eine Transition $t \in T$ hat Konzession unter einer Markierung m genau dann, wenn für alle $p \in \bullet t$ mit $p \in P, t \in T$ gilt $m(p) \geq K((p, t))$. Hat eine Transition Konzession, kann sie schalten. Ihr Schalten führt zu einer Markierung m' , wobei $m'(p) = m(p) - K((p, t)) + K((t, p))$ für alle $p \in P$ gilt. Dieses Schalten schreiben wir als $m \xrightarrow{t} m'$.

Für eine Schaltsequenz $m_1 \xrightarrow{t_1} m_2, m_2 \xrightarrow{t_2} m_3, \dots, m_n \xrightarrow{t_n} m_{n+1}$ schreiben wir $m_1 \xrightarrow{t_1 \dots t_n} m_{n+1}$. Die Existenz einer solchen Schaltsequenz notieren wir durch $m \xrightarrow{*} m'$.

2.3 Definition (Erreichbarkeitsgraph)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. Die Menge der von der Markierung m aus erreichbaren Markierungen m' wird definiert als $R(m) = \{m' | m \xrightarrow{*} m'\}$. Der gerichtete Graph $RG = (V, E)$ wird Erreichbarkeitsgraph genannt, wenn $V = R(m_0) \cup m_0$ und $E = (m, m')$ für alle $m, m' \in V$, wenn es eine Transition $t \in T$ gibt mit $m \xrightarrow{t} m'$.

Abbildung 2.4 zeigt den Erreichbarkeitsgraphen des Nachfrage-Angebot-Beispiels (siehe Abbildung 2.2). Die Ellipsen stehen für die Zustände, die hier wie die entsprechenden Stellen beschriftet sind. Die Kanten des Erreichbarkeitsgraphen beschreiben die Zustandsübergänge und sind für das Beispiel entsprechend den Transitionen beschriftet.

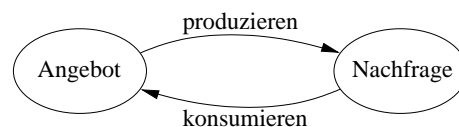


Abbildung 2.4: Erreichbarkeitsgraph des Nachfrage-Angebot-Beispiels

2.3 Abräum- und hemmende Kanten

Die Anwendung von hemmenden Kanten bei der Modellierung mit Petrinetzen ist weit verbreitet. Hemmende Kanten sind grafisch daran erkennbar, daß sie anstelle eines Pfeiles durch einen Kreis am Ende der Kante dargestellt sind. Sind eine Stelle und eine Transition durch eine hemmende Kante miteinander verbunden, so kann die Transition nur dann Konzession besitzen, wenn die Stelle unmarkiert ist. Es gibt auch mit einer Kardinalität n beschriftete hemmende Kanten. Die damit verbundene Transition besitzt nur dann Konzession, wenn die entsprechende Stelle mit weniger als n Marken belegt ist. Als Argument für den Einsatz hemmender Kanten wird immer wieder vorgebracht, daß Petrinetze dadurch Turingmächtigkeit erlangen. Dieses Argument ist richtig, für den praktischen Gebrauch von Petrinetzen jedoch nutzlos. Vielmehr soll an dieser Stelle über den Schritt hin zur Turingmächtigkeit diskutiert werden. Abbildung 2.5 zeigt auf der rechten Seite ein Petrinetz mit einer hemmenden Kante. Transition $T3$ kann nur dann schalten, wenn die Stelle $P2$ unmarkiert ist.

Die linke Seite der Abbildung zeigt ein äquivalentes Petrinetz, das ohne hemmende Kanten dieselben Abläufe erzeugt, wie das Modell auf der rechten Seite der Abbildung. Diese Äquivalenz wurde durch eine sogenannte Komplementärstelle $P3$ erreicht, die mit denselben Transitionen wie die Stelle $P2$ verbunden ist, nur

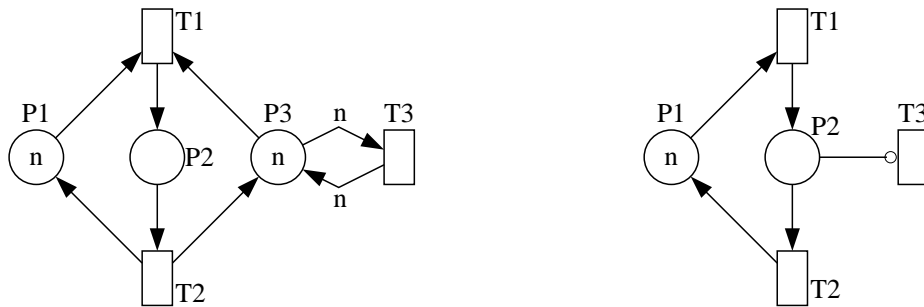


Abbildung 2.5: Vergleich Komplementärstelle – hemmende Kante

daß deren Kantenrichtungen denen der mit der Stelle P2 verbundenen entgegengesetzt sind. Auch in diesem Modell kann die Transition T3 nur dann schalten, wenn die Stelle P2 unmarkiert ist. Für alle Petrinetze, in denen in keiner erreichbaren Markierung eine beliebige Stelle mit mehr als einer bestimmten Anzahl von Marken belegt ist, können hemmende Kanten durch den Einsatz von Komplementärstellen wie in Abbildung 2.5 dargestellt, ersetzt werden. Das bedeutet auch, daß die Turingmächtigkeit erst durch den Einsatz von hemmenden Kanten bei unbeschränkten Modellen erfüllt wird. Die linke Seite auf Abbildung 2.6 zeigt ein solches unbeschränktes Modell.

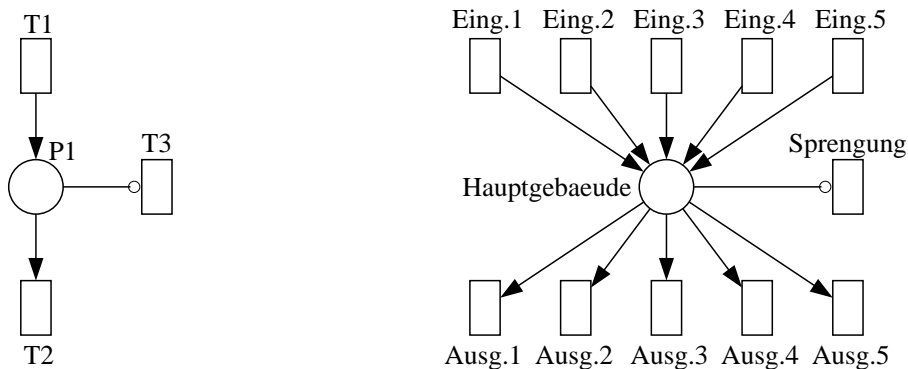


Abbildung 2.6: Unbeschränktes Modell mit hemmender Kante und möglicher Interpretation

Die darin verwendete hemmende Kante ist nicht durch Komplementärstellen zu ersetzen, da die Anzahl der Marken auf P1 unbeschränkt wachsen kann. Jede Komplementärstelle hätte aber eine Beschränkung dieser Anzahl der Marken und somit des Zustandsraumes zur Folge. Die rechte Seite in Abbildung 2.5 zeigt mittels der Beschriftung der Transitionen und Stellen eine mögliche Interpretation dieses Modells. Zur besseren Veranschaulichung wurden noch einige Transitionen hinzugefügt, ohne den Sinn des Modells zu verändern. Dieses Modell

ist unbestritten ein korrektes Modell für eine eventuelle Sprengung des Hauptgebäudes, welche, wenn überhaupt, selbstverständlich nur dann erfolgen sollte, wenn das Gebäude leer ist. Trotzdem ist das Modell unrealistisch, denn ohne Sperrung der Eingänge ist eine Kontrolle der Leere des Hauptgebäudes nicht möglich. Jedes diesem Modell entsprechende Programm läßt sich nur mittels einer durch einmaligen Lesezugriff überprüfbaren Speicherstruktur realisieren. Selbst ein unbegrenzt wachsender Stack oder eine Liste müssen geeignet verwaltet werden, um zu erkennen, wann sie leer sind. In dieser Arbeit werden aus den soeben genannten Gründen keine hemmenden Kanten verwendet.

Ebenso wie hemmende Kanten sind Abräumkanten leicht für die Modellierung zu verwenden und ihre Semantik einfach in einem Programm zu realisieren. Sind eine Stelle und eine Transition über eine Abräumkante miteinander verbunden, so führt das Schalten der Transition zum „Entleeren“ der Stelle. Das bedeutet, daß alle sich in der Stelle befindlichen Marken durch das Schalten einer Transition abgezogen werden. Hierbei muß grundlegend zwischen der Weiterverwendung der Marken und deren „Vernichten“ unterschieden werden. Das Vernichten ist unproblematisch und praktisch relevant. Abbildung 2.7 zeigt zwei Petrinetzmodelle mit Abräumkanten.

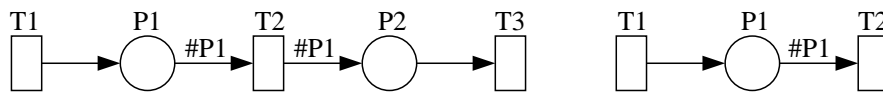


Abbildung 2.7: Abräumkanten mit Weiterverwendung und Vernichtung

Das Modell auf der linken Seite zeigt die Verwendung von Abräumkanten mit Weiterverwendung der Marken. Die Beschriftung #P1 der Kanten steht für das markierungsabhängige Schalten der Transition T2. Sie entnimmt beim Schalten der Stelle P1 alle in der Stelle befindlichen Marken und schaltet diese in die Stelle P2. In einer Implementierung dieses Modells muß wiederum eine Speicherstruktur verwendet werden, die einen einmaligen Zugriff auf alle Marken erlaubt. Wären die Marken Dateien auf einer Festplatte, die auf eine andere Festplatte bewegt werden sollen, müßten die Dateien entweder einzeln und nacheinander auf die andere Platte bewegt werden oder vorher in eine große Datei gepackt, um diese zu bewegen. Keine dieser beiden Alternativen entspricht dem Modell. Die rechte Seite in Abbildung 2.7 zeigt das Vernichten aller Marken aus der Stelle P2. Dies entspricht beispielsweise dem Löschen einer Festplatte.

Zusammenfassend läßt sich sagen, daß beim Modellieren unter Verwendung hemmender und Abräumkanten stets darauf geachtet werden muß, ob das reale System einen tatsächlich atomaren Vorgang, der durch das Schalten der betreffenden Transition im Modell ausgedrückt wird, verwirklicht. In den meisten Fällen sind die Stellen, auf die mittels hemmender oder Abräumkanten zugegriffen werden, kritische Bereiche, die eine komplexere Verwaltung erfordern [129, 116].

2.4 Farbige Petrinetze

In den bisher betrachteten Modellen sind die Marken nicht unterscheidbar. Jede Marke wird grafisch als Punkt dargestellt. Diese Petrinetzmodelle werden *farblos* genannt. Diese Nicht-Unterscheidbarkeit von Marken führt bei komplexeren Systemen zu sehr großen und unübersichtlichen Modellen. Abbildung 2.8 zeigt das Modell eines sehr kleinen Ausschnittes einer Fertigungsanlage für Autos.

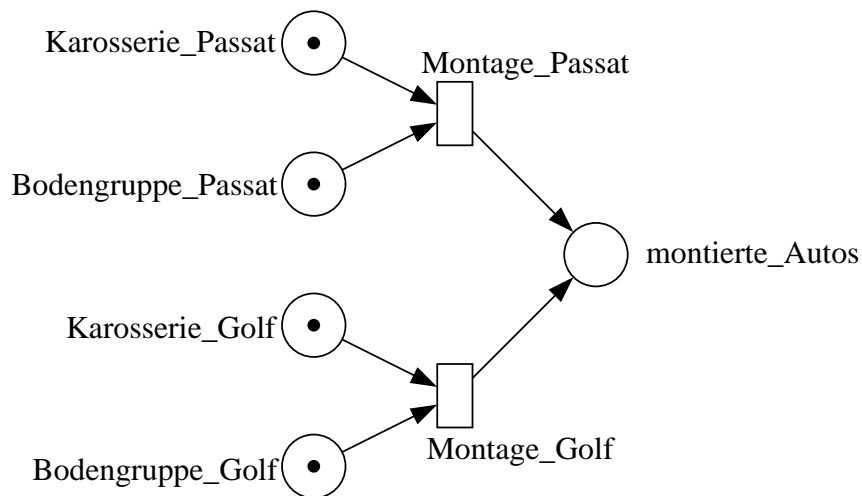


Abbildung 2.8: Farbloses Petrinetzmodell – Ausschnitt einer Fertigungsanlage für Autos

Diese Modellierung birgt zwei Nachteile: Zum einen ist offensichtlich, daß die Modellierung einer Montage vieler verschiedener Typen ein sehr großes Modell erzeugt, da die Unterscheidung der verschiedenen Bauteile und deren Montage durch eigene Stellen und Transitionen erfolgen muß. Zum anderen ist das Modell kein genaues Abbild der Realität, da die durch das Modell suggerierte örtliche Trennung der Lager der Bauteile und deren Montage in der Realität nicht existiert. Vielmehr werden in modernen Fertigungsanlagen an demselben Ort verschiedene Teile montiert. Realistischer ist somit das in Abbildung 2.9 dargestellte Modell desselben Beispiels eines Ausschnittes einer Fertigungsanlage für Autos.

In den Stellen *Lager_Karosserie* und *Lager_Bodengruppe* befinden sich jeweils zwei Marken *P* und *G*. Die Kanten von diesen Stellen zur Transition *Montage* sind mit $\langle x \rangle$ und $\langle y \rangle$ beschriftet. Dabei sind *x* und *y* Variablen, die mit den genannten Marken belegt werden können. Für die Transition *Montage* existieren somit die in Abbildung 2.10 dargestellten Schaltmöglichkeiten.

Diese Tabelle wird auch die der Transition *Montage* zugeordnete *Übergangstabelle* genannt. Die zweite und vierte Spalte der gezeigten Übergangstabelle sind gleich, da die den Spalten entsprechenden Kanten mit derselben Variablenbezeichnung beschriftet sind.

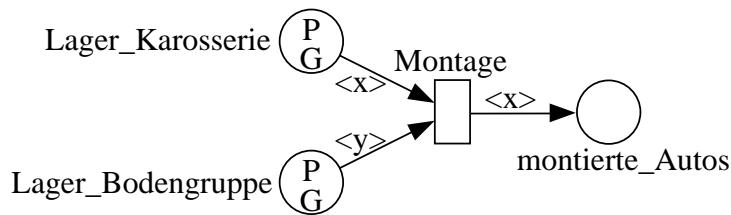


Abbildung 2.9: Farbiges Petrinetzmodell – Ausschnitt einer Fertigungsanlage für Autos

Nr.	die von der Stelle Lager_Karosserie abgezogene Marke	die von der Stelle Lager_Bodengruppe abgezogene Marke	die auf die Stelle montierte_Autos bewegte Marke
1.	P	P	P
2.	P	G	P
3.	G	P	G
4.	G	G	G

Abbildung 2.10: Übergangstabelle der Transition Montage

Die in Abbildung 2.10 dargestellte Übergangstabelle der Transition Montage enthält jedoch noch Übergänge, die nicht gewünscht sind, da jeweils die Bodengruppe und die Karosserie desselben Typs montiert werden sollen. Diese Forderung läßt sich durch eine der Transition Montage zugeordnete Schaltbedingung $[x=y]$ realisieren. Die Transition Montage ist dann nur noch unter den Übergängen 1. und 4. aus der in Abbildung 2.10 dargestellten Übergangstabelle schaltfähig.

Zu farbigen Petrinetzen gehört stets eine Festlegung der verwendeten Typen sowie eine Typdeklaration der verwendeten Variablen. Darüber hinaus wird jeder Stelle ein Typ zugeordnet, so daß diese Stelle nur Marken des zugeordneten Typs enthalten kann. Auf diese zusätzliche textuelle Beschreibung eines Modells soll hier nicht weiter eingegangen werden. Sattdessen wird auf [78, 129] verwiesen.

In den in [77, 78] vorgeschlagenen farbigen Petrinetzen werden die möglichen Übergänge, wie beschrieben, entsprechend den Variablenbezeichnungen und den Schaltbedingungen der Transitionen ermittelt. Wie bereits an einem sehr kleinen Beispiel verdeutlicht, kann dabei die Anzahl der möglichen Übergänge sehr groß sein, wenn sich mehrere Marken in den Stellen des Vorbereichs einer Transition befinden. Die verbotenen Übergänge durch Schaltbedingungen auszuschließen, kann für viele Modelle komplexer sein, als explizit die erlaubten Übergänge anzugeben. Darüber hinaus kann von dem Modellierer oft nicht erwartet werden, alle unerlaubten Übergänge erkennen und ausschließen zu müssen. Vielmehr sind stattdessen während der Modellierung ausschließlich die erwünschten

Übergänge bekannt und von Bedeutung. Diesen Anforderungen entsprechend wurden für die Modellierung von Fertigungssystemen spezielle farbige Petrinetze entwickelt [132, 138]. Diese sind Gegenstand des folgenden Abschnittes.

2.4.1 Spezielle farbige Petrinetze

Die in [132, 138, 135] vorgeschlagenen speziellen farbigen Petrinetze für die Modellierung von Fertigungssystemen zeichnen sich durch eine Trennung des sogenannten *Strukturmodells* und der diesem Strukturmodell assoziierten *Arbeitsplänen* aus. Wie später erläutert wird, legen die Arbeitspläne die Übergangstabellen der im Strukturplan modellierten Transitionen fest. In [135, 137, 133] wird die Anwendung spezieller farbiger Petrinetze zur Modellierung von Fertigungssystemen unter der Nutzung des Modellierungs- und Analysewerkzeuges TimeNET detailliert erläutert. Jüngere Arbeiten [47, 48] zeigen die Anwendbarkeit spezieller farbiger Petrinetze auch im Bereich der Modellierung von *workflow management* Systemen.

Spezielle farbige Petrinetze lassen sich wie auch die farbigen Petrinetze aus [78] hierarchisch verfeinern. Diese hierarchische Verfeinerung dient der Erhöhung der Übersichtlichkeit der Modelle. Wird auf der höheren Hierarchiestufe bspw. eine Maschine durch eine sogenannte *Vertretertransition* repräsentiert, wird diese Vertretertransition in der nächstniedrigeren Hierarchiestufe detailliert verfeinert. Vertretertransitionen sind somit ein grafischer Platzhalter für ein feineres Teilmodell. Sie besitzen keine der für Transitionen üblichen Schaltsemantiken. Hierarchisierung unter Bewahrung einer Schaltsemantik der einem Teilmodell übergeordneten Transition ist in [16, 17] und Verfeinerungen von Stellen ist in [89] untersucht worden, ohne einen neuen Standard der Hierarchisierung zu etablieren.

Strukturmodell

Anhand eines Beispiels werden im folgenden die Modellierungstechniken bei der Verwendung spezieller farbiger Petrinetze näher erläutert. Ein erster wichtiger Aspekt der Modellierung mit speziellen farbigen Petrinetzen ist die bereits erwähnte Trennung von Strukturmodell und den diesem Strukturmodell assoziierten Arbeitsplänen. Dieser Abschnitt beschreibt zunächst die Modellierung des Strukturmodells.

Abbildung 2.11 zeigt die oberste Ebene des Strukturmodells einer Fischertechnik-Modellfabrik [117, 135]. Das Strukturmodell beschreibt die Ressourcen und ihre arbeitsplanunabhängigen Eigenschaften.

Bei der Modellierung des Strukturmodells wird nach dem Prinzip verfahren, Aufenthaltsorte von Werkstücken mit Stellen und Maschinen sowie Transport-

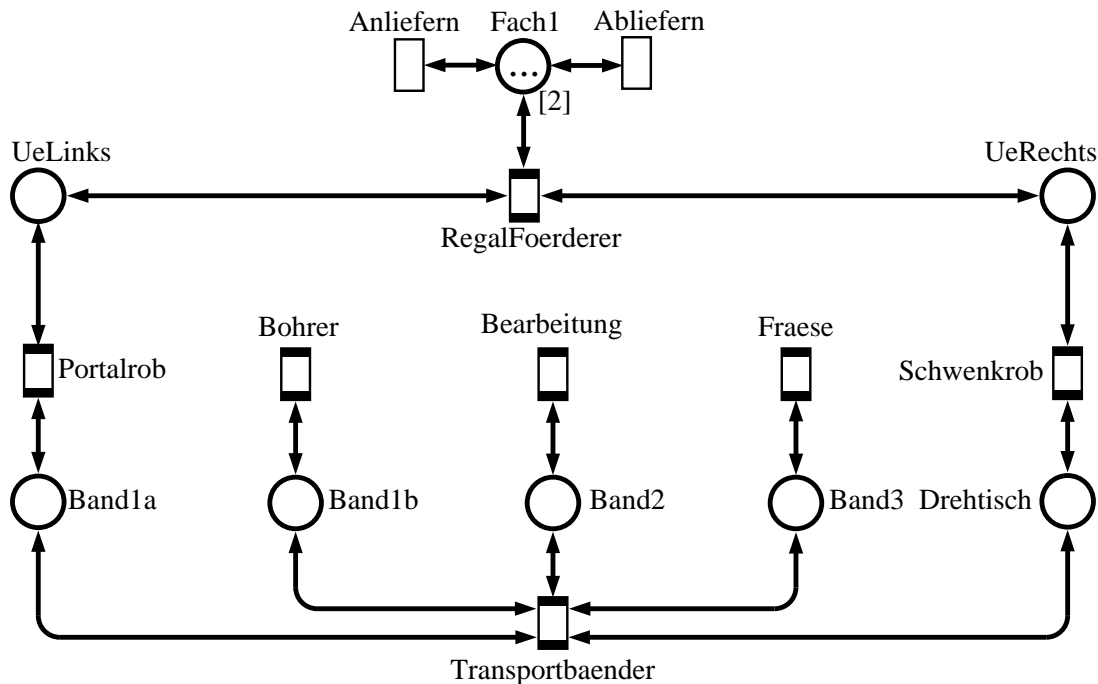


Abbildung 2.11: Strukturmodell der höchsten Hierarchiestufe

einrichtungen mit Transitionen abzubilden. Jede Stelle im gezeigten Modell entspricht einem möglichen Aufenthaltsort eines Werkstücks. An Stellen stehende Zahlen in eckigen Klammern geben Kapazitäten an. Ändert sich durch die Bearbeitung eines Werkstückes nur der Bearbeitungszustand, nicht jedoch der Ort, wird dies durch das Abziehen der „alten“ Marke und gleichzeitiges Ablegen der Marke mit dem neuen Bearbeitungszustand in derselben Stelle beim Schalten einer Transition nachgebildet (siehe dazu die Beschreibung von Arbeitsplänen in Abschnitt 2.4.1). Deshalb sind häufig Pfeile zwischen Stellen und Transitionen in beiden Richtungen vorhanden, die der Einfachheit halber übereinander gezeichnet wurden und wie ein Doppelpfeil aussehen. Kanten stehen entweder für Transportwege zwischen den Funktionseinheiten oder Zustandsveränderungen der Marken in der damit verbundenen Stelle.

Die Transitionen Anliefern und Abliefern modellieren den Austausch von angelieferten bzw. fertiggestellten Werkstückes mit der Außenwelt, der über das Hochregallager stattfindet. Die initiale Markierung beschreibt zwei leere Paletten (2 Palette.leer) in der Stelle Fach1. Aus Platzgründen ist die Darstellung der beiden Marken nur durch die drei Punkte dargestellt. Transportvorgänge zwischen den Maschinen sind immer nur durch das Zusammenspiel zweier Transportbänder zu erreichen. Aus diesem Grund wird das Verhalten aller drei Transportbänder gemeinsam in einer Vertretertransition Transportbaender zusammengefaßt. Die restlichen Transitionen beschreiben jeweils eine Maschine oder eine unabhängige Transporteinrichtung. Jede Vertretertransition

wird in einem separaten Untermodell detailliert beschrieben. Ein solches Untermodell ist in Abbildung 2.12 dargestellt.

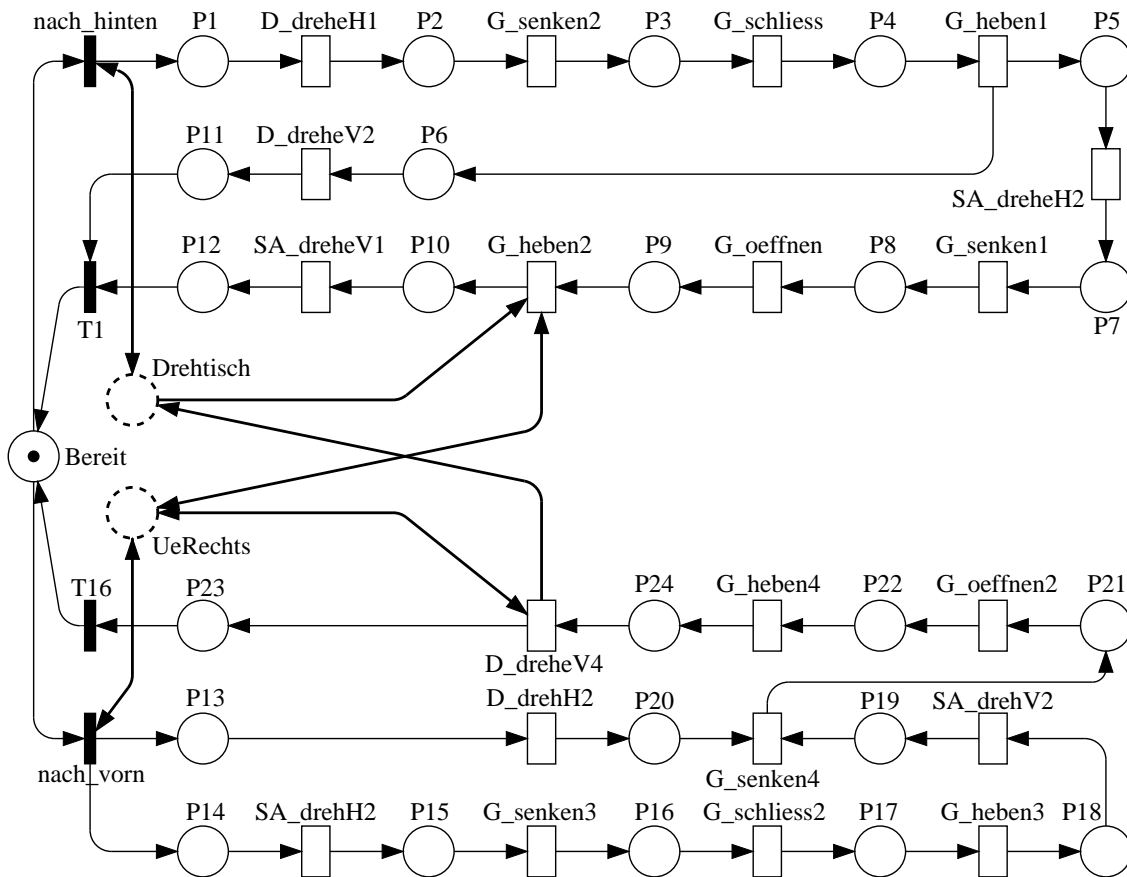


Abbildung 2.12: Verfeinertes Modell des Schwenkarmroboters

Abbildung 2.12 zeigt das verfeinerte Modell des Schwenkarmroboters. Die dicken gestrichelten Stellen stehen für die Verbindungen zur höheren Hierarchiestufe. Diese Stellen finden sich bereits in dem Modell der höheren Hierarchiestufe. Um die Interaktion des Untermodells mit der Umgebung zu verdeutlichen, sind sie ebenfalls im Untermodell dargestellt.

Die dicken Kanten beschreiben die Wege der in diesem Modell Werkstücken entsprechenden farbigen Marken. Dick gezeichnete Stellen können nur solche farbigen Marken enthalten. Darüber hinaus gibt es auch dünn gezeichnete Stellen und Kanten. Sie dienen der Beschreibung gemeinsamer Ressourcen im System. Im Beispiel des Schwenkarmroboters werden bspw. die Zustände des Roboters und dessen Position beschrieben. Zur dynamischen Beschreibung der Zustände und Positionen solcher Ressourcen werden sogenannte *elementare* Marken verwendet, die den Marken in farblosen Petrinetzen entsprechen.

Neben dem im vorangegangenen Abschnitt beschriebenen Strukturmodell wer-

den zur Beschreibung eines Produktionsprozesses die Arbeitspläne aller zu fertigenden Werkstücke benötigt.

Arbeitspläne

Für die Modellierung von Arbeitsplänen werden die gleichen farbigen Petrinetze eingesetzt, so daß nur ein Beschreibungsmittel notwendig ist. Für eines der beiden mit der Modellfertigungsanlage produzierten Werkstücke — hier A genannt — ist das entsprechende Modell in Abbildung 2.13 dargestellt. Wie das Strukturmodell ist es entsprechend hierarchisch aufgebaut.

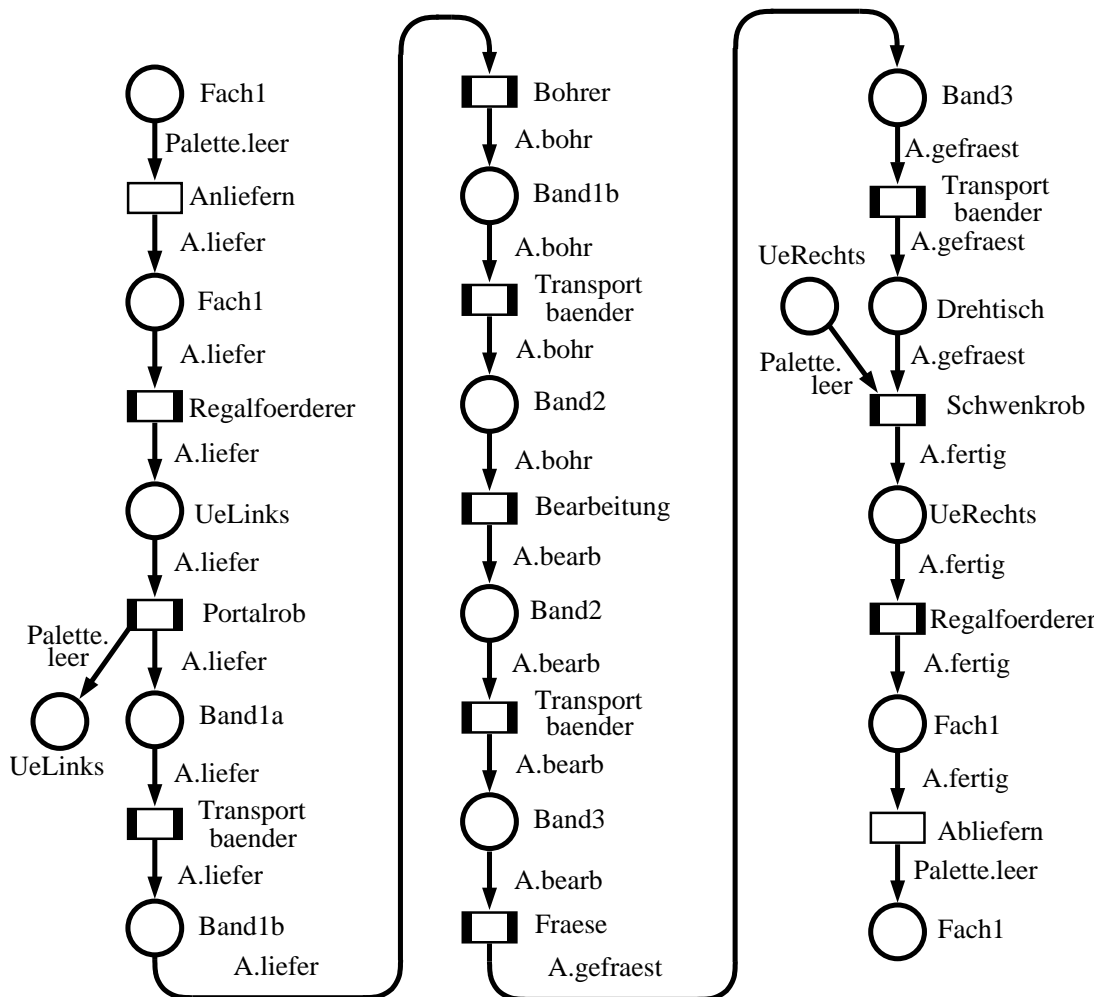


Abbildung 2.13: Arbeitsplanmodell für ein Werkstück auf der obersten Hierarchiestufe

Die Modelle der Arbeitspläne beschreiben den bei der Fertigung eines Produktes zu durchlaufenden Pfad durch das Strukturmodell. Sie werden aufgrund ihrer

engen Beziehung zum Strukturmodell als zu diesem *assoziierte Petrinetze* bezeichnet. Ein solches Netz enthält nur Stellen, Transitionen und Kanten, zu denen es eine Entsprechung im Strukturmodell gibt. Dies entspricht der Forderung, in einem Arbeitsplan nur tatsächlich im Fertigungssystem verfügbare Betriebsmittel einzusetzen. Diese Zuordnung beginnt mit der höchsten Hierarchisierungsstufe und setzt sich auf allen untergeordneten Hierarchisierungsstufen fort. Sie ist durch die Verwendung identischer Namen in den assoziierten Seiten festgelegt. Markennamen beschreiben durch einen Punkt getrennt sowohl den Werkstücktyp (z.B. A) als auch den Zustand des Werkstückes (z.B. fertig).

An den Kanten im Arbeitsplanmodell ist der Typ der transportierten oder bearbeiteten Werkstücke verzeichnet. Für die Transition *Anliefern* wird somit in Abbildung 2.13 links oben folgender Übergang festgelegt: Von der Stelle *Fach1* wird die Marke *Palette.leer* abgezogen und eine Marke *A.liefer* auf dieselbe Stelle bewegt. Arbeitspläne legen somit explizit die Übergänge einer Transition fest. Darüber hinaus wird der Modellierer von der expliziten Verwaltung gemeinsamer Ressourcen entlastet. Betrachten wir dazu nochmals das detaillierte Modell des Schwenkarmroboters aus Abbildung 2.12. Abbildung 2.14 zeigt die festgelegten Übergänge des Schwenkarmroboters für die Paletten und das Werkstück A.

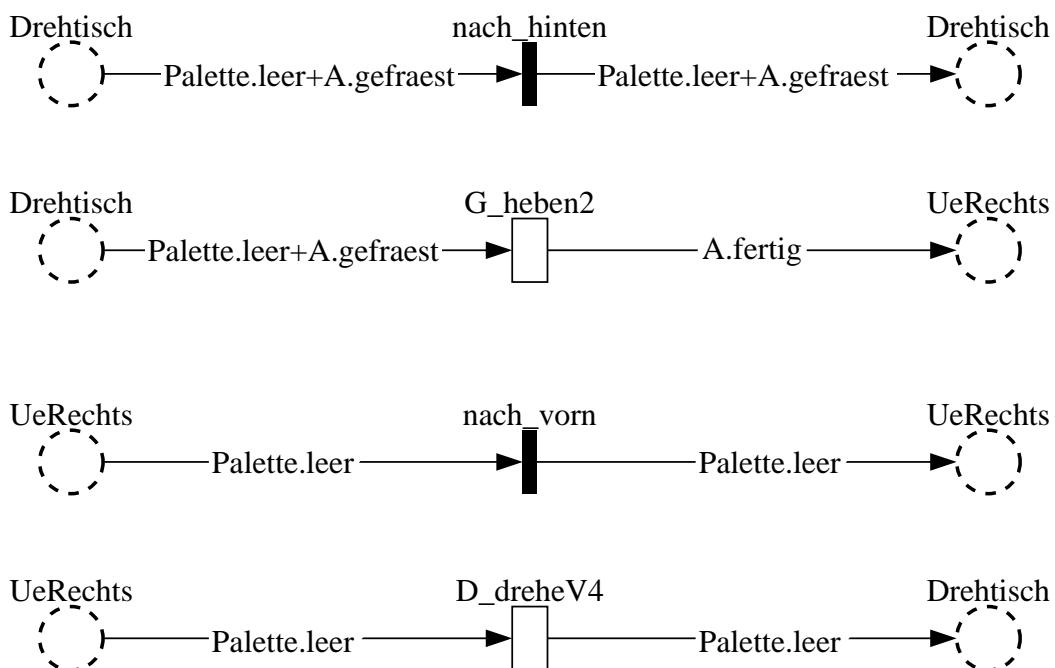


Abbildung 2.14: Detailliertes Arbeitsplanmodell für den Schwenkarmroboter

Der diesem Modell assoziierte Arbeitsplan enthält nur die die dicken Kanten betreffenden Übergänge. Das bedeutet, daß zum einen nur die Übergänge der Transitionen *nach_hinten*, *G_Heben2*, *nach_vorn* und *D_DreheV4* und dabei nur

die die Stellen `Drehtisch` und `UeRechts` betreffenden Teile der Übergänge beschrieben werden müssen. Die zusätzlichen Bedingungen für die Zustände und Positionen der Ressourcen sind bereits durch das Strukturmodell genügend beschrieben. Die Transition `G_Heben2` kann nur schalten, wenn sich der Schwenkarmroboter in der durch die Stelle `P9` repräsentierten Position befindet.

Fertigungsalternativen wurden im Beispiel nicht benötigt, können aber im Modell problemlos abgebildet werden. Dabei ist es möglich, zur Abbildung bestimmter Materialflußverteilungen bzw. Transportstrategien einen bestimmten Prozentsatz der Werkstücke für eine Alternative vorzusehen oder das Betreten von Alternativen vom aktuellen Zustand des Fertigungssystems abhängig zu gestalten.

Anhand des Strukturmodells ist der gezeigte Arbeitsplan klar nachzuvollziehen, denn er beschreibt lediglich einen Pfad durch das Strukturmodell. Das Werkstück `A` wird durch die Bohrmaschine, die Bearbeitungsstation und die Fräse bearbeitet. Im Modell kann anhand der Bearbeitungsschritte und Aufenthaltsorte leicht zwischen Transporten und Bearbeitungen unterschieden werden.

Zu Beginn und am Ende des Arbeitsplanes muß das Werkstück `A` auf einer Palette transportiert werden. Dies wird nachgebildet, indem beim Schalten der Transition `Anliefern` eine Montageoperation mit der Palette vorgenommen wird. Da dies gleichzeitig den Eintritt des angelieferten Werkstücks in das Modell darstellt, wird keine Marke für `A` verbraucht. Nach dem Entladen der Palette durch den Portalroboter ist sie jedoch wieder verfügbar, so daß bei Transition `Portalrob` eine Demontage modelliert wird. Die auf der Stelle `UeLinks` abgelegte Marke mit der Bezeichnung `Palette.leer` wird im Arbeitsplan von `A` nicht weiter betrachtet. Auch für die leeren Paletten ist ein eigenes Arbeitsplanmodell vorhanden, in dem beispielsweise die Transportstrategie des Regalförderers für leere Paletten abgebildet wird.

2.5 Stochastische Petrinetze

Viele Systeme sind ohne die Betrachtung der Zeit nur unvollständig spezifiziert. Für wichtige Anwendungsklassen wie Fertigungs- und Kommunikationssysteme sind Leistungs- und Zuverlässigkeitsbewertungen zentraler Inhalt der Untersuchungen. Die meisten dieser Systeme zeichnen sich darüber hinaus durch eine stochastische Spezifikation der Dynamik aus. So unterliegt bspw. der Ausfall einer Maschine oder auch der Zugriff auf das Internet keiner festen Zeit, sondern vielmehr einer gewissen Wahrscheinlichkeit. Um solche Systeme trotzdem adäquat mit Petrinetzen modellieren zu können, wurden sogenannte *stochastische* Petrinetze entwickelt. In stochastischen Petrinetzen wird jeder Transition eine Schaltzeit zugeordnet. Stochastische Petrinetze wurden unabhängig voneinander in den Dissertationen von Symons [125], Natkin [106] und Molloy [101]

vorgeschlagen. In [102] wurde gezeigt, daß der Erreichbarkeitsgraph eines beschränkten stochastischen Petrinetzes isomorph zu einer endlichen *zeitkontinuierlichen Markowkette* ist.

2.5.1 Zeitkontinuierliche Markowketten

Die *Markoweigenschaft* besagt, daß der zukünftige zeitliche Verlauf unabhängig von der Vergangenheit ist: $P\{X \leq x + x_0 | X > x\} = P\{X \leq x_0\}$. Auf Zustands- und Zustandsübergangssicht bezogen, bedeutet diese Eigenschaft, daß die verbleibende Aufenthaltszeit in einem Zustand unabhängig von der bisherigen Aufenthaltszeit in diesem Zustand ist. Nur wenn die Aufenthaltszeit in zeitkontinuierlichen Systemen einer exponentiell verteilten Zufallsvariablen entspricht, gilt diese Markoweigenschaft. Die negativ exponentielle Dichtefunktion ist wie folgt definiert:

$$f_X(x) = \begin{cases} \lambda \cdot e^{-\lambda x} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Die Exponentialverteilung ist eindeutig durch die Rate λ beschrieben, deren reziproker Wert λ^{-1} dem Erwartungswert der Zufallsvariablen entspricht. Diese Rate wird in stochastischen Petrinetzen Transitionen attribuiert, denen eine negativ exponentielle Schaltzeitverteilung zugeordnet ist. Diese Transitionen werden auch vereinfachend *exponentielle* Transitionen genannt und die der Transition zugeordnete Rate *Schaltrate*.

Eine zeitkontinuierliche Markowkette kann entweder durch einen gerichteten Zustandsgraphen oder durch eine Zustandsübergangsmatrix \mathbf{G} beschrieben werden. Setzt man für die Diagonalelemente der Zustandsübergangsmatrix die negative Summe der (stets nichtnegativen) Zeileneinträge ein, so erhält man die Generatormatrix \mathbf{Q} . Mit Hilfe der Generatormatrix läßt sich für jeden Zustand z der Markowkette und somit für jeden Zustand des durch die Markowkette repräsentierten Systems die stationäre Aufenthaltswahrscheinlichkeit π_z dafür berechnen, daß sich das System in diesem Zustand befindet. Der Wahrscheinlichkeitsvektor π über die Menge aller Zustände Z errechnet sich aus

$$\mathbf{0} = \pi \mathbf{Q} \quad ; \quad \pi \mathbf{e}^T = 1$$

wobei \mathbf{e} der Einheitsvektor sei.

Die numerische Analyse eines stochastischen Petrinetzmodells im eingeschwungenen Zustand besteht somit aus drei Schritten. Im ersten Schritt wird der Erreichbarkeitsgraph des Petrinetzmodells erstellt und den Zustandsübergängen

werden die Schaltraten der entsprechenden Transitionen zugeordnet. In einem zweiten Schritt wird aus diesem Erreichbarkeitsgraphen die Generatormatrix \mathbf{Q} erzeugt. Der letzte Schritt löst das Gleichungssystem $\mathbf{0} = \pi\mathbf{Q}; \pi\mathbf{e}^T = 1$. Die Größe des Erreichbarkeitsgraphen und der Generatormatrix sind die stärksten Restriktionen bei der Analyse stochastischer Petrinetzmodelle. Die Kapitel 4 und 5 beschreiben Vorschläge zur Überwindung dieser Restriktionen.

Leistungsmaße

Aus den Zustandswahrscheinlichkeiten lassen sich wichtige Leistungsparameter der modellierten Systeme berechnen. Bspw. entsprechen Puffer- oder Lagerfüllstände dem Erwartungswert der Anzahl von Marken auf den die Puffer oder Lager modellierenden Stellen. Der Erwartungswert der Anzahl der Marken auf einer Stelle berechnet sich aus einer gewichteten Summe der Zustandswahrscheinlichkeiten. Das Gewicht entspricht der Anzahl der Marken auf der Stelle im bewerteten Zustand. Allgemeiner lassen sich die Berechnungen von Leistungsparametern mittels der stationären Zustandswahrscheinlichkeit π_z durch folgende Formeln ausdrücken: Seien $\text{expr}^{\mathbb{B}} : Z \rightarrow \mathbb{B}$ ein zustandsabhängiger¹ boolescher und $\text{expr}^{\mathbb{R}} : Z \rightarrow \mathbb{R}$ ein zustandsabhängiger reellwertiger Ausdruck. Die Wahrscheinlichkeit eines zustandsabhängigen booleschen Ausdrucks $\text{expr}^{\mathbb{B}}$ wird berechnet aus² $P[\text{expr}^{\mathbb{B}}] = \sum_{z:\text{expr}^{\mathbb{B}}(z)} \pi_z$. Der Erwartungswert eines reellwertigen Ausdrucks $\text{expr}^{\mathbb{R}}$ berechnet sich aus $E[\text{expr}^{\mathbb{R}}] = \sum_Z \text{expr}^{\mathbb{R}}(z) \cdot \pi_z$. Diese Formeln lassen sich erweitern, um bedingte Wahrscheinlichkeiten und bedingte Erwartungswerte zu berechnen [132].

Zeitlose Transitionen

Zeitlose Transitionen [3, 2] wurden in [4] für die Modellierung mit stochastischen Petrinetzen eingeführt. Sie werden als schwarze Balken dargestellt und hauptsächlich zur Beschreibung zweier Systemkonstrukte verwendet. Zum einen wird mit zeitlosen Transitionen der Eintritt in einen beschränkten oder kritischen Bereich modelliert. Zum Anderen lassen sich Alternativen mit zeitlosen Transitionen geeignet modellieren.

Abbildung 2.15 zeigt zwei Modelle zur Erläuterung der Anwendung zeitloser Transition für die Modellierung beschränkter und kritischer Bereiche. In Abbil-

¹Im Zusammenhang mit stochastischen Petrinetzen werden die Ausdrücke in der Regel markierungsabhängig definiert. Durch den Isomorphismus der Markierungen mit den Zuständen einer Markowkette definieren wir hier aus Gründen der Vereinfachung die Ausdrücke jedoch zustandsabhängig.

²Der Ausdruck $z : \text{expr}^{\mathbb{B}}(z)$ steht für alle Zustände $z \in Z$ für die $\text{expr}^{\mathbb{B}}(z) = \text{wahr}$ gilt

Abbildung 2.15 befinden sich in dem auf der linken Seite gezeigten Modell auf der Stelle P2 nie mehr als k Marken.

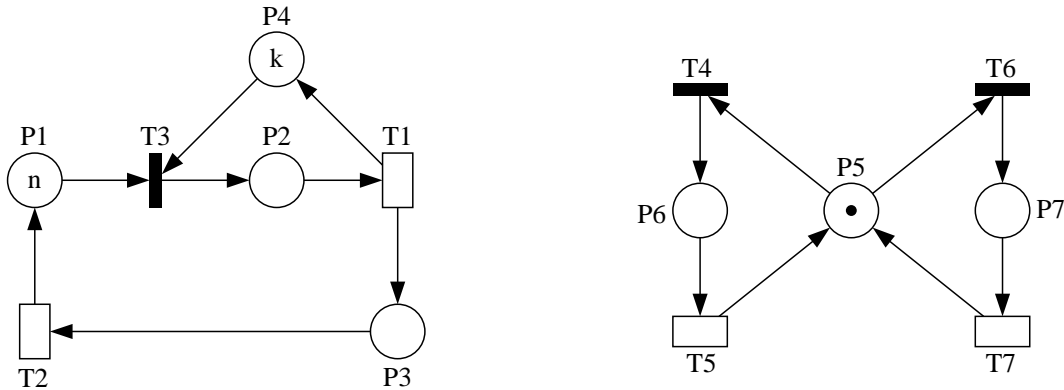


Abbildung 2.15: Verwendung zeitloser Transitionen zur Modellierung eines beschränkten und kritischen Bereichs

Zur Modellierung von Fertigungssystemen werden dadurch bspw. Lagerkapazitäten modelliert. Bei der Verwendung spezieller farbiger Petrinetze wird für solche Lagerkapazitäten eine Beschriftung der das Lager modellierenden Stelle der Form $[k]$ verwendet. Das entspricht jedoch nur einer übersichtlicheren Sicht auf die hier dargestellte Konstruktion mit einer zeitlosen Transition und einer Komplementärstelle.

Die rechte Seite in Abbildung 2.15 zeigt ein Modell zur Verwaltung zweier kritischer Bereiche. Die Markierung der Stellen P6 und P7 ist wechselseitig ausgeschlossen. Die Verteilung des durch die Markierung der Stelle P5 modellierten Schlüssels wird durch die zeitlosen Transitionen T4 und T6 geregelt.

Das auf der rechten Seite in Abbildung 2.15 dargestellte Modell verweist bereits auf das zweite wichtige mit zeitlosen Transitionen modellierbare Systemkonstrukt, die *Alternative*. Abbildung 2.16 zeigt ein eine Alternative beschreibendes Modell.

Wenn bspw. die Transitionen T3 und T4 Maschinen modellieren, so weisen die zeitlosen Transitionen dem durch die Marke modelliertem Werkstück stochastisch die bearbeitende Maschine zu. Durch Wahrscheinlichkeitsangaben an den zeitlosen Transitionen läßt sich bestimmen, in welchem Verhältnis die Maschinen T3 und T4 als bearbeitende Maschinen gewählt werden sollen. Sind den zeitlosen Transitionen T1 die Schaltwahrscheinlichkeit w_1 und T2 die Schaltwahrscheinlichkeit w_2 zugeordnet, so schaltet T1 unter der dargestellten initialen Markierung mit der Wahrscheinlichkeit $\frac{w_1}{w_1+w_2}$ während T2 mit der Wahrscheinlichkeit $\frac{w_2}{w_1+w_2}$ schaltet.

Besitzen eine zeitlose und eine zeitbehaftete Transition nebenläufig Konzession, so schaltet stets die zeitlose Transition zuerst. In einem Zustand, in dem eine

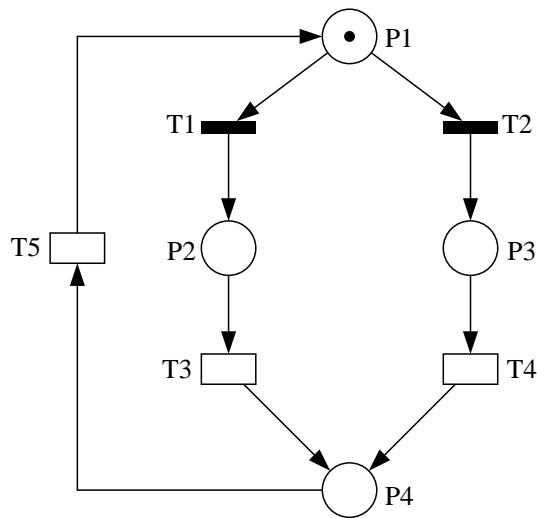


Abbildung 2.16: Verwendung zeitloser Transitionen zur Modellierung einer Alternative

zeitlose Transition Konzession besitzt, vergeht keine Zeit. Diese Zustände werden *verschwindende* Zustände genannt. Die relative Aufenthaltswahrscheinlichkeit in verschwundenen Zuständen ist gleich Null. Zustände, in denen nur zeitbehaftete Transitionen Konzession besitzen, heißen dagegen *andauernde* Zustände. Während der Erreichbarkeitsgraph eines Petrinetzes mit zeitlosen Transitionen sowohl andauernde als auch verschwindene Zustände enthalten kann, sind für die Ermittlung der mittleren Aufenthaltswahrscheinlichkeit der Zustände nur die andauernden Zustände von Bedeutung. Der Erreichbarkeitsgraph eines stochastischen Petrinetzes mit ausschließlich andauernden Zuständen heißt *reduzierter* Erreichbarkeitsgraph. In [7, 12] wurde ein Verfahren zur Erstellung des reduzierten Erreichbarkeitsgraphen ohne den Zwischenschritt der Erstellung des Erreichbarkeitsgraphen mit verschwindenden Zuständen vorgestellt. Die Grundidee dabei ist, den aus dem Schalten zeitloser Transitionen entstehenden Pfaden Wahrscheinlichkeiten zuzuordnen. Diese Pfadwahrscheinlichkeiten werden mit der letzten Schaltrate multipliziert und führen in den jeweiligen Pfadendzustand, wenn die letzte Schaltrate exponentiell war [92].

Im folgenden Abschnitt wird die Modellierung mit stochastischen Petrinetzen näher erläutert.

2.5.2 Ein Beispielmodell

Zur Einführung der Modellierung mit stochastischen Petrinetzen wird ein aus der Literatur [35, 131, 85] bekanntes farbloses Petrinetzmodell verwendet. Abbildung 2.17 zeigt das Petrinetzmodell eines Fertigungssystems.

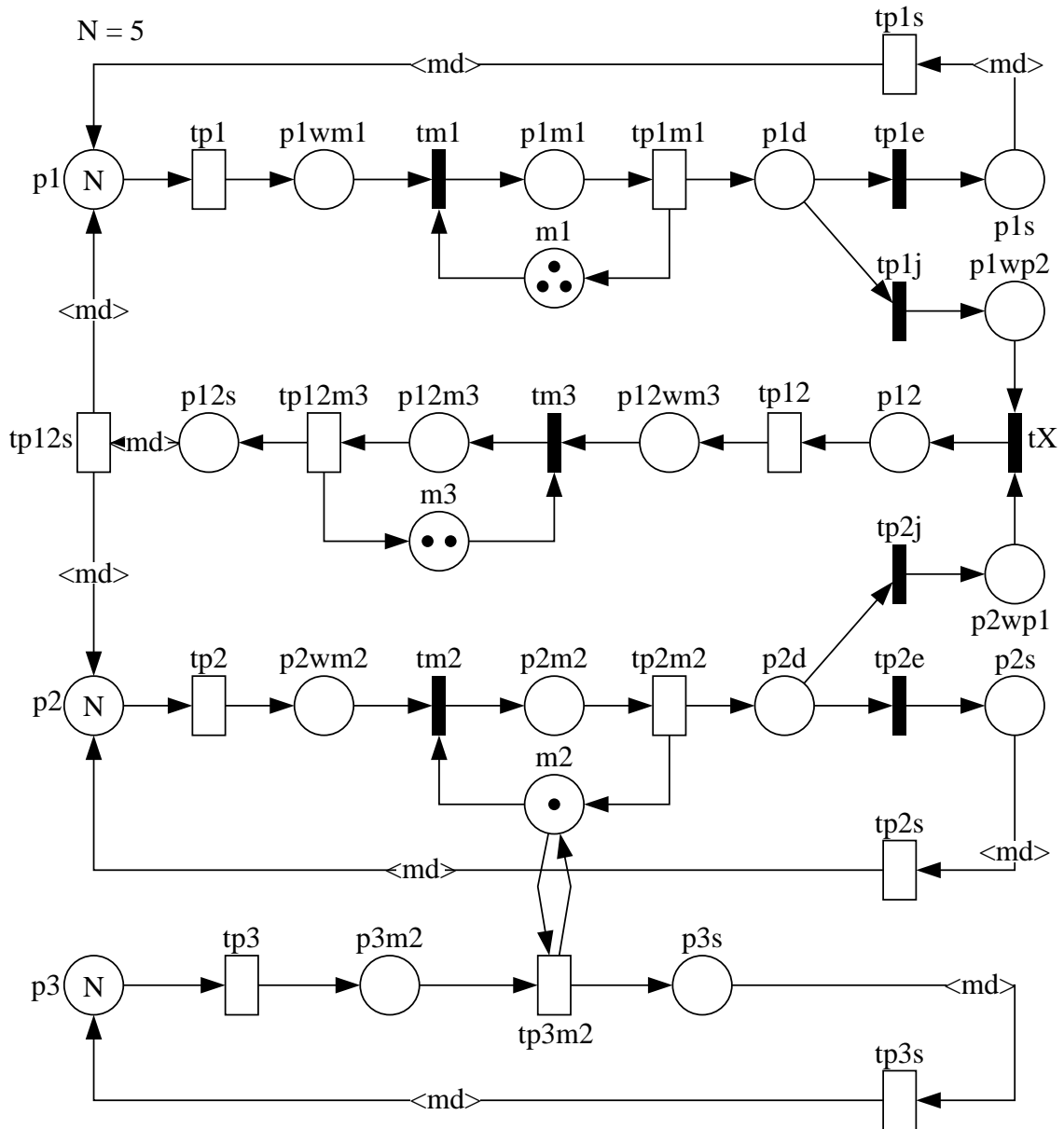


Abbildung 2.17: Farbloses stochastisches Petrinetzmodell eines Fertigungssystems

Das modellierte System besteht aus drei verschiedenen Typen von Maschinen: M_1 , M_2 und M_3 . Vom Typ M_1 gibt es drei Maschinen, modelliert durch drei Marken auf m_1 . Diese Maschinen bearbeiten (Transition t_{p1m1}) Werkstücke vom Typ P_1 . Es gibt eine Maschine vom Typ M_2 , modelliert durch eine Marke auf m_2 . Diese Maschine bearbeitet (Transition t_{p2m2}) hauptsächlich Werkstücke vom Typ P_2 . Wenn jedoch keine Werkstücke vom Typ P_2 zur Bearbeitung verfügbar sind, werden von dieser Maschine auch Werkstücke vom Typ P_3 bearbeitet (Transition t_{p3m2}). Zwei Maschinen (m_3) vom Typ M_3 setzen jeweils ein Werkstück vom Typ P_1 und P_2 zusammen (t_{p12m3}), so daß ein neuer Werkstücktyp P_{12} entsteht. Fertige Werkstücke der Typen P_1 , P_2 , P_3 und P_{12} verlassen das System über die Transitionen t_{p1s} , t_{p2s} , t_{p3s} bzw. t_{p12s} . Ebenso viele Werkstücke werden als „roh“ zur Bearbeitung zur Verfügung gestellt. Die Zustände der Werkstücke, wie „roh“ oder „fertig“, sind für farblose Petrinetze rein virtuell und nicht dem Petrinetz direkt abzulesen. Dieser Nachteil führt zur Klasse der farbigen Petrinetze, die bereits im Abschnitt 2.4 beschrieben sind.

Den mit $\langle md \rangle$ beschrifteten Kanten sind markierungsabhängige Kardinalitäten zugeordnet. Wenn z.B. t_{p12s} schaltet, werden alle in p_{12s} befindlichen Marken abgezogen und in dieser Anzahl den Stellen p_1 und p_2 zugeordnet (siehe Abschnitt 2.3). Alle durch einen schwarzen Balken dargestellte Transitionen sind zeitlose Transitionen. Der Zeitpunkt ihrer Aktivierung und ihres Schaltens bzw. ihrer Deaktivierung ist derselbe. Die allen anderen Transitionen zugeordnete Schaltzeitverteilung ist negativ exponentiell. In TimeNET wird einer Transition nicht die Rate der Verteilungsfunktion, sondern ihr reziproker Wert als Ausdruck für die Schaltzeit zugeordnet. Tabelle 2.18 zeigt die den zeitbehafteten Transitionen des Beispiels zugeordneten Verzögerungszeiten in der bei TimeNET zu verwendenden Syntax. Im unteren Teil der Tabelle sind die Wahrscheinlichkeiten des Schaltens der in Konflikt stehenden zeitlosen Transitionen festgelegt.

Von jedem Werkstücktyp gibt es jeweils N Teile. Das Modell läßt sich unter Verwendung von TimeNET auf einem aktuellen PC mit 256MB Hauptspeicher nur bis einschließlich $N = 6$ numerisch analysieren. Ab $N = 7$ ist der Zustandsraum des Modells mit 1.639.440 andauernden Zuständen bereits zu groß. Tabelle 2.19 zeigt die Anzahl der andauernden Zustände Z^{tim} und die durch TimeNET errechneten jeweiligen Resultate der im folgenden in TimeNET-Syntax definierten und ebenfalls [35] entnommenen „Produktivitätsfunktion“

$$\Psi = 400 * E\{\#p_1\} + 600 * E\{\#p_2\} + 100 * E\{\#p_3\} + 1100 * E\{\#p_{12}\}$$

der stationären Analyse.

Allgemeine Schaltzeitverteilungen

Neben Transitionen, denen eine negativ exponentiell verteilte Schaltzeit zugeordnet ist, gibt es auch Transitionen mit deterministischer oder allgemeinerer Schalt-

Transition	Schaltzeit
tp1s	60
tp1	IF (#p1>0):(1/#p1) ELSE 1;
tp1m1	IF (#p1m1>0):(4/#p1m1) ELSE 1;
tp12s	60
tp12	IF (#p12>0):(1/#p12) ELSE 1;
tp12m3	IF (#p12m3>0):(1/#p12m3) ELSE 1;
tp2	IF (#p2>0):(1/#p2) ELSE 1;
tp2m2	6
tp2s	60
tp3	IF (#p3>0):(1/#p3) ELSE 1;
tp3m2	2
tp3s	60
Transitionen	Wahrscheinlichkeiten
tp1e und tp1j	0,8 und 0,2
tp2e und tp2j	0,6 und 0,4

Abbildung 2.18: Schaltzeiten und Schaltwahrscheinlichkeiten des Fertigungssystembeispiels

N	Z^{tim}	Ψ
1	54	13,874112
2	810	29,157447
3	6 520	44,447930
4	35 910	59,556565
5	152 712	74,385576
6	537 768	88,873708

Abbildung 2.19: Anzahl der andauernden Zustände und Resultate für variierende N des Fertigungssystembeispiels

zeitverteilung. Diese Schaltzeitverteilungen sind aus der Menge der *expolynomialen* Verteilungen [64, 63, 65]. Expolynomiale Funktionen sind stückweise aus Kombinationen von Polynomen und Exponentialfunktionen zusammengesetzt. Betrachten wir eine Zerlegung der Zeitachse $[0, \infty)$ in $L + 1$ Intervalle, dann ist eine expolynomiale Verteilungsfunktion $F(t)$ durch

$$F_l(t) = \sum_{m=0}^M \sum_{n=0}^N a_{lmn} t^m e^{-\lambda_n t}$$

mit $t_l \leq t < t_{l+1}$ für alle $0 \leq l \leq L$, $a_{lmn} \in \mathbb{R}$ und $\lambda_n \in [0, \infty)$ definiert.

Die deterministische Verteilungsfunktion läßt sich bspw. durch folgende Parameter beschreiben:

$$L = 1 \quad \begin{array}{l} l = 0 : M = N = 0 \quad \lambda = 0 \quad a_{000} = 0 \\ l = 1 : M = N = 0 \quad \lambda = 0 \quad a_{100} = 1 \end{array}$$

Weitere gebräuchliche Verteilungen, wie die Dreiecksverteilung oder die negativ exponentielle sind Spezialfälle expolynomialer Verteilungen (Parameterbelegung siehe z.B. [69]). Stochastische Petrinetze mit Transitionen, denen expolynomiale Schaltzeitverteilungen zugeordnet sind, werden als *semi-regenerative* stochastische Petrinetze bezeichnet, da der diesen Modellen zugrunde liegende stochastische Prozeß von diesem Typ ist [30]. Semi-regenerative stochastische Petrinetze sind numerisch analysierbar, wenn in jedem erreichbaren Zustand höchstens eine nichtexponentielle Transition Konzession besitzt. Unter dieser Voraussetzung kann für jeden sogenannten *Regenerationszeitpunkt* des semi-regenerativen Prozesses eine *eingebettete Markowkette* gebildet werden. Regenerationszeitpunkte sind die Zeitpunkte des stochastischen Prozesses, in denen der Prozeß gedächtnislos, seine zukünftige Entwicklung also nur vom gegenwärtigen Zustand abhängig ist (siehe Markoweigenschaft in Abschnitt 2.5.1).

Die Zustandsübergänge U werden demnach unterteilt in exponentielle U^{exp} und nichtexponentielle U^{gen} . Die im reduzierten Erreichbarkeitsgraphen beschriebenen andauernden Zustände Z werden analog unterteilt in die Menge der Zustände Z^{exp} , in denen nur exponentielle Zustandsübergänge, und die Menge Z^{gen} , in denen auch ein nichtexponentieller Zustandsübergang möglich ist. Die Menge $Z_u^{\text{gen}} \subset Z^{\text{gen}}$ beschreibt die andauernden Zustände, in denen der nichtexponentielle Zustandsübergang $u \in U^{\text{gen}}$ möglich ist. Erfolgt zwischen den Regenerationszeitpunkten ein nichtexponentieller Zustandsübergang u aus U^{gen} , so wird für die initialen Zustände des zugeordneten Prozesses eine zugeordnete Markowkette zur Ermittlung der Übergangswahrscheinlichkeiten Ω_u und der relativen Aufenthaltsdauern Ψ_u der eingebetten Markowkette wie folgt berechnet [64]:

$$\Omega_u = \int_0^{\infty} e^{\mathbf{Q}_u t} d u^{\text{delay}}(t)$$

$$\Psi_u = \int_0^{\infty} e^{\mathbf{Q}_u t} (1 - u^{\text{delay}}(t)) dt$$

Dabei beschreibt \mathbf{Q}_u die Matrix der nebenläufig zu u aus den Zuständen Z_u^{gen} heraus schaltfähigen exponentiellen Zustandsübergänge. Unter Verwendung der Matrizen Ω_u und Ψ_u für alle Zustände Z_u^{gen} , in denen der nichtexponentielle Zustandsübergang u möglich ist, werden die Zustandsübergangswahrscheinlichkeiten \mathbf{P} sowie Konversionsfaktoren \mathbf{C} aller Zustandsübergänge der eingebetteten Markowkette berechnet:

$$\forall z_i \in Z^{\text{exp}} : \quad \mathbf{P}_{z_i, z_j} = \begin{cases} 0 & \text{falls } i = j \\ \frac{\lambda_{z_i, z_j}}{\lambda_{z_i}} & \text{sonst} \end{cases}$$

$$\mathbf{C}_{z_i, z_j} = \begin{cases} \frac{1}{\lambda_{z_i}} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

$$\forall u \in U^{\text{gen}} : \quad \mathbf{P}_{Z_u} = \Omega_u \Delta_u$$

$$\mathbf{C}_{Z_u, z_i} = \begin{cases} \Psi_u & \text{falls } z_i \in Z_u \\ 0 & \text{sonst} \end{cases}$$

Dabei wird in $\Delta_u = \delta_{Z_u, z_i}$ die Wahrscheinlichkeit dafür angegeben, daß nach dem Schalten des nichtexponentiellen Zustandsüberganges u durch einen Pfad zeitloser Übergänge der andauernde Zustand z_i erreicht wird. λ_{z_i} steht für die Summe aller aus z_i ausgehenden Raten exponentieller Zustandsübergänge.

Mit Hilfe der Zustandsübergangswahrscheinlichkeiten der eingebetteten Markowkette kann für den eingeschwungenen Zustand der eingebetteten Markowkette der Vektor der mittleren Aufenthaltswahrscheinlichkeiten γ durch Lösen des folgenden linearen Gleichungssystems berechnet werden:

$$\gamma \cdot (\mathbf{P} - \mathbf{E}) = \mathbf{0} \quad , \quad \gamma \cdot \mathbf{e} = 1$$

Dabei sind \mathbf{E} die Einheitsmatrix und \mathbf{e} der Einheitsvektor mit der dem Eigenvektor γ entsprechenden Dimension.

Der Vektor der relativen Aufenthaltswahrscheinlichkeiten π aller andauernden Zustände des durch das semi-regenerative stochastische Petrinetz beschriebenen Modells im eingeschwungenen Zustand berechnet sich aus

$$\gamma' = \gamma \cdot \mathbf{C} \quad , \quad \pi = \frac{1}{\gamma' \cdot \mathbf{e}} \cdot \gamma'$$

Mit Hilfe des π -Vektors können, wie in Abschnitt 2.5.1 beschrieben, relevante Leistungsmaße berechnet werden. Einen geeigneten Überblick über die Analysemethoden für stochastische Petrinetze, in denen Transitionen allgemeine Schaltzeitverteilungen zugeordnet sind bietet [64]. In dieser Arbeit werden im folgenden jedoch nur stochastische Petrinetze mit zeitlosen und exponentiellen Transitionen betrachtet.

Kapitel 3

Eigenschaften von Petrinetzmodellen

Viele Fragen über Eigenschaften modellierter Systeme können bereits ohne eine Betrachtung des Zeitverhaltens der Modelle beantwortet werden. Mit Petrinetzmodellen von Fertigungssystemen kann z.B. untersucht werden, ob tatsächlich die gewünschten Werkstücke produziert werden oder ob das System in unerwünschte, nicht behebbare Zustände eintritt. Darüber hinaus können Aussagen über die Anzahl von Marken auf einer Menge von Stellen des Modells getroffen werden. Von Bedeutung sind dabei Aussagen über eine stets konstante Anzahl von Marken oder Aussagen derart, daß einmal leere Stellen für immer leer bleiben. Es läßt sich untersuchen, ob bestimmte Transitionen nie oder nach Erreichen bestimmter Zustände nie mehr schalten. Zusätzlich ist analysierbar, ob das modellierte System eine nur endliche Menge von Zuständen beschreibt. Eigenschaften dieser Art werden in diesem Kapitel anhand von Beispielen eingeführt und näher erläutert. Darüber hinaus werden Methoden vorgestellt, die eine Berechnung dieser Eigenschaften ermöglichen. Insbesondere wird ein effizienter Algorithmus zur Berechnung sogenannter *Stellen-Invarianten* vorgestellt. Die Steigerung der Effizienz dieses Algorithmus gegenüber bekannten Verfahren (z.B. [96, 39]) ergibt sich zum einen aus einer verbesserten Methode [68] zur Ermittlung ganzzahliger als auch einer verbesserten Methode [62] zur Ermittlung natürlichzahliger Lösungen für diese Stellen-Invarianten. Dieses neue, effizientere Verfahren wird in Abschnitt 3.2 vorgestellt.

3.0.3 Ein komplexeres Beispiel

Das Nachfrage-Angebot-Beispiel aus Abbildung 2.2 ist in mehrerer Hinsicht sehr einfach. Es besteht nur aus je zwei Stellen und Transitionen, wobei jedes dieser Elemente genau eine Eingangs- und genau eine Ausgangskante besitzt. Bereits weniger spezielle Petrinetze, in denen alle Transitionen (Stellen) genau eine

Eingangs- und genau eine Ausgangskante besitzen, werden *Zustandsmaschinen* (*markierte Graphen*) genannt. Darüber hinaus gibt es keinen Zustand des Modells, in dem sich innerhalb einer Stelle mehr als eine Marke befindet. Man sagt, das Modell ist *1-beschränkt*. Ein strukturell komplexeres Modell einer Fertigungsanlage ist in Abbildung 3.1 dargestellt.

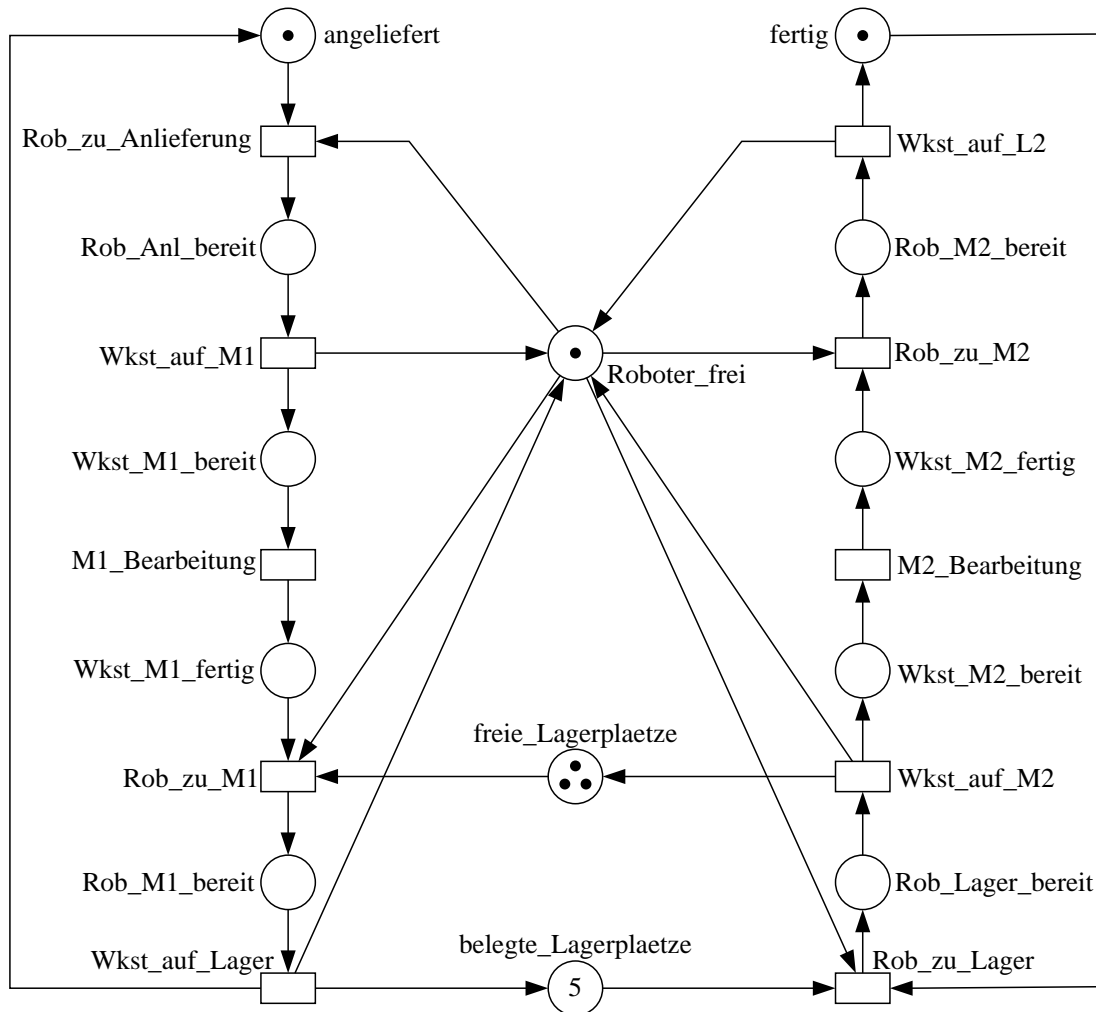


Abbildung 3.1: Petrinetzmodell einer Fertigungsanlage

Initial ist ein unbearbeitetes Werkstück angeliefert (Stelle *angeliefert* markiert). Ein angeliefertes Werkstück kann vom Roboter zur Maschine M1 transportiert werden. Dazu fährt der Roboter zu Lager 1 (Transition *Rob_zu_Anlieferung*). Das Werkstück wird zur Maschine M1 gebracht (*Wkst_auf_M1*) und dort bearbeitet (*M1_Bearbeitung*). Wenn das Werkstück von der Maschine M1 bearbeitet ist (*Wkst_M1_fertig*), fährt der Roboter zur Maschine (*Rob_zu_M1*) und legt das Werkstück in das Lager (*Wkst_auf_Lager*). Jedes Werkstück belegt einen der

freien Plätze im Lager. Insgesamt stehen acht Lagerplätze zur Verfügung; initial sind davon fünf belegt und drei frei. Die rechte Seite der Modelldarstellung beschreibt die Entnahme vom Lager, den Transport durch den Roboter zur Maschine M2 und die Bearbeitung durch diese Maschine. Abschließend wird das fertige Werkstück zur Ablieferung transportiert. Die Lagerplätze und der Roboter sind sogenannte *gemeinsame Ressourcen* innerhalb des modellierten Fertigungssystems. Die Verwaltung gemeinsamer Ressourcen stellt einen der Vorteile bei der Modellierung mit Petrinetzen gegenüber anderen bekannten Modellierungsverfahren dar. Der Modellierer modelliert die gemeinsamen Ressourcen durch Stellen und führt von dort aus Kanten zu Transitionen ein, die die Benutzung gemeinsamer Ressourcen modellieren. Das Freigeben gemeinsamer Ressourcen wird durch Rückkanten an die entsprechende Stelle modelliert. Die dynamische Verwaltung der Ressourcen wird durch die Schaltregeln der Transitionen definiert und ist nicht mehr Aufgabe des Modellierers. Ist z.B. der Roboter bereit, um ein angeliefertes Werkstück zur Maschine 1 zu transportieren (*Rob_An1_bereit*), so sind die Transitionen *Rob_zu_M1*, *Rob_zu_Lager* und *Rob_zu_M2* deaktiviert.

3.0.4 Lebendigkeit, Beschränktheit und Reversibilität

Eine Besonderheit bei der Modellierung gemeinsamer Ressourcen wird ebenfalls im in Abbildung 3.1 dargestellten Beispiel deutlich. Eine gleichzeitige Nutzung verschiedener gemeinsamer Ressourcen erfordert eine Vermeidung „unerwünschter“ Zustände. In der Realität ist ein Lagerplatz entweder frei oder besetzt. In dem Modell aus Abbildung 3.1 ist jedoch der Übergang von freien Lagerplätzen zu belegten Lagerplätzen nicht atomar. Die Transition *Rob_zu_M1* kann nur schalten, wenn sich auf der Stelle *freie_Lagerplaetze* mindestens eine Marke befindet, jedoch schaltet erst die Transition *Wkst_auf_Lager* wieder eine Marke auf die Stelle *belegte_Lagerplaetze*. Nehmen wir nun an, daß die Kante nicht von *freie_Lagerplaetze* zu *Rob_zu_M1*, sondern stattdessen zu *Wkst_auf_Lager* verläuft und der Rest des Modells unverändert bleibt. Dann wäre folgendes Verhalten des Systems möglich. Der Roboter fährt zur ersten Maschine (Transition *Rob_zu_M1*) und ist danach bereit, eine bearbeitetes Werkstück in das Lager zu transportieren (Stelle *Rob_M1_bereit* markiert). Wenn aber gar kein freier Platz mehr im Lager zur Verfügung steht (Stelle *freie_Lagerplaetze* unmarkiert) kann nicht nur die Transition *Wkst_auf_Lager* nicht schalten, sondern keine Transition im gesamten Modell. Man sagt, ein *toter Zustand* ist erreicht.

Die gegenteilige Eigenschaft von Modellen mit toten Zuständen sind Modelle, die *frei von toten Zuständen* sind. In solchen Modellen hat in jedem erreichbaren Zustand mindestens eine Transition Konzession und kann schalten. Eine noch strengere Eigenschaft als die Freiheit von toten Zuständen ist die *Lebendigkeit* von

Petrinetzen. Ein Petrinetz ist lebendig, wenn jede Transition des Petrinetzes immer wieder Konzession bekommt und schalten kann. Man sagt dann, daß alle Transitionen dieses Petrinetzes lebendig sind.

3.1 Definition (lebendige, nicht lebendige und tote Transitionen)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. Eine Transition $t \in T$ heißt

- *lebendig* genau dann, wenn aus jeder erreichbaren Markierung m eine Markierung m' erreichbar ist, in der t Konzession besitzt.
- *nicht lebendig* genau dann, wenn es eine erreichbare Markierung m gibt, von der aus ausschließlich Markierungen erreichbar sind, in denen t keine Konzession besitzt.
- *tot* wenn für alle erreichbaren Markierungen m gilt, daß t keine Konzession in m besitzt.

Ein Petrinetz ist *reversibel*, wenn aus jedem erreichbaren Zustand heraus wieder der Anfangszustand erreicht werden kann.

Im vorangegangenen Abschnitt 3.0.3 wurde bereits die *Beschränktheit* des Petrinetzmodells aus Abbildung 2.2 besprochen. Folgend wird dieser Begriff definiert.

3.2 Definition (beschränkt, unbeschränkt)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz und $k \in \mathbb{N}$. Eine Stelle $p \in P$ heißt *beschränkt* (*k-beschränkt*), wenn für alle erreichbaren Markierungen m gilt $m(p) \leq k$. N heißt beschränkt, wenn alle $p \in P$ beschränkt sind, sonst *unbeschränkt*.

Die in diesem Abschnitt definierten Eigenschaften von Petrinetzmodellen sind intensiver Untersuchungsgegenstand in dieser Arbeit. Insbesondere wird die in Kapitel 5 vorgestellte Aggregierungsmethode dahingehend untersucht, ob sie bspw. die Lebendigkeit und Beschränktheit von Petrinetzen bewahrt.

3.0.5 Stellen-Invarianten, Fallen und Siphons

Die im vorhergehenden Abschnitt beschriebene Besonderheit bei der Modellierung gemeinsamer Ressourcen ist bezüglich einer weiteren Eigenschaft von Petrinetzen Gegenstand der folgenden Beschreibung. Der Roboter im in Abbildung 3.1 dargestellten Petrinetzmodell ist entweder frei verfügbar oder zur Anlieferung, am Lager, an Maschine 1 oder an Maschine 2 bereit. Im Modell ist das darin erkennbar, daß stets genau eine der Stellen `Roboter_frei`, `Rob_An1_bereit`, `Rob_M1_bereit`, `Rob_M2_bereit` oder `Rob_Lager_bereit` markiert ist. Formaler ausgedrückt, ist die Summe der Marken auf den genannten Stellen für alle erreichbaren Zustände konstant:

3.3 Definition (Stellen-Invariante)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. Eine (nichtnegative¹) Stellen-Invariante i ist eine Abbildung $i : P \rightarrow \mathbb{N}$, wenn für alle Transitionen $t \in T$ gilt:

$$\sum_{p \in \bullet t} i(p) \cdot K(p, t) = \sum_{p' \in t \bullet} i(p') \cdot K(t, p')$$

und $i(p), i(p') \neq 0$ für endlich viele $p, p' \in P$.

Die am Beispiel erläuterte Eigenschaft für *Stellen-Invarianten* lässt sich dann formal wie folgt beschreiben: Sei i eine Stellen-Invariante, so gilt für jede erreichbare Markierung m eines Petrinetzes $N = (P, T, F, K, m_0)$

$$\sum_{p \in P} i(p) \cdot m(p) = \sum_{p \in P} i(p) \cdot m_0(p).$$

Im folgenden nennen wir auch vereinfachend alle die Stellen selbst eine Stellen-Invariante, die durch eine solche auf eine Zahl größer Null abgebildet werden. Die obige Definition einer Stellen-Invariante lässt sich auch wie folgt interpretieren: Jede Transition, die von einer Stelle innerhalb einer Stellen-Invariante eine Marke abzieht, legt auch stets (mindestens) eine Marke wieder auf eine Stelle dieser Stellen-Invarianten. Analog zu diesem beschriebenen Verhalten von Transitionen bezüglich Stellen-Invarianten lassen sich zwei Teilmengen von Stellen mit verschiedenem dynamischen Verhalten definieren: Als *Fallen* wird eine Menge von Stellen bezeichnet, die einmal markiert², für alle folgenden erreichbaren Zustände markiert bleibt. Im Gegensatz dazu sind *Siphons* Mengen von Stellen, die einmal unmarkiert, für alle folgenden erreichbaren Zustände unmarkiert bleiben.

Stellen-Invarianten, Fallen und Siphons werden sehr häufig benötigt, um auf Eigenschaften von Petrinetzen wie Lebendigkeit oder Freiheit von toten Zuständen zu schließen. Beispielsweise ist ein Petrinetz frei von toten Zuständen, wenn jeder nicht leere Siphon ohne isolierte Stellen eine initial markierte Falle enthält.

Innerhalb dieser Arbeit ist ein neuer Algorithmus zur Berechnung von Stellen-Invarianten, Fallen und Siphons entwickelt und implementiert worden, der im Abschnitt 3.2 vorgestellt wird. Dieser Algorithmus basiert unter anderem auf linearer Algebra.

Die Repräsentation eines Petrinetzes mittels einer Matrix und die Beschreibung der vorgestellten Eigenschaften mittels Gleichungs- und Ungleichungssystemen ist Gegenstand des nächsten Abschnittes.

¹Wir beschränken uns in dieser Arbeit auf nichtnegative Stellen-Invarianten, indem die folgende Abbildung auf die Menge der natürlichen Zahlen erfolgt.

²Wir nennen eine Menge von Stellen markiert (unmarkiert) in einer Markierung m , wenn mindestens eine Stelle aus dieser Menge markiert (unmarkiert) ist.

3.1 Lineare Algebra für Petrinetze und deren Eigenschaften

Die Berechnung struktureller Eigenschaften von Petrinetzmodellen unter ausschließlicher Verwendung der Struktur der Petrinetze ist für alle wichtigen Eigenschaften sehr komplex. Für die Berechnung von Stellen-Invarianten, Fallen und Siphons muß man gar alle möglichen Kombinationen von Stellen probieren und ihre Zugehörigkeit zu einem der drei Konstrukte durch Schalten jeder mit diesen Stellen verbundenen Transitionen testen. Für die Bestimmung der Lebendigkeit, toter Markierungen, Beschränktheit und Erreichbarkeit ist die Erstellung des gesamten Erreichbarkeitsgraphen nötig (zu Verfahren mit reduziertem Erreichbarkeitsgraphen siehe Kapitel 5). Petrinetze lassen sich jedoch auch durch eine sogenannte *Inzidenzmatrix* repräsentieren. Auf Basis dieser Matrix existieren linearalgebraische Methoden zur Berechnung einiger der genannten Eigenschaften. Diese Methoden werden im folgenden vorgestellt.

3.1.1 Anwendung am Beispiel

Wir betrachten wiederum das Modell einer Fertigungsanlage, ändern aber aus Platzgründen die Transitions- und Stellenbeschriftungen (siehe Abbildung 3.2). Dieses Petrinetzmodell läßt sich mittels der in Abbildung 3.3 dargestellten Inzidenzmatrix repräsentieren (alle nicht belegten Einträge sind 0). In einer Inzidenzmatrix N repräsentieren die Zeilen die Stellen und die Spalten die Transitionen eines Petrinetzmodells N . Schaltet eine Transition T durch Abzug von $K(P1, T)$ vielen Marken von der Stelle $P1$ und legt dabei $K(T, P2)$ viele Marken auf eine Stelle $P2$, so enthält die Spalte für T in der Inzidenzmatrix zwei Einträge. In der Zeile von $P1$ lautet der Eintrag $-K(P1, T)$ und in der Zeile von $P2$ lautet der Eintrag $+K(T, P2)$ in der Spalte von T .

Eine Markierung \mathbf{m} wird als Vektor dargestellt, wobei jede Stelle des Petrinetzes auf die Anzahl der in dieser Stelle befindlichen Marken abgebildet ($\mathbf{m} : P \rightarrow \mathbb{N}$) wird. Die initiale Markierung des Petrinetzmodells aus Abbildung 3.2 läßt sich als Spaltenvektor darstellen, deren Komponenten analog der alphanumerisch geordneten Stellenbeschriftungen positioniert sind:

$$\mathbf{m}_0 = \begin{pmatrix} m(P1) \\ m(P2) \\ \vdots \\ m(P9) \\ m(P10) \\ m(P11) \\ m(P12) \\ m(P13) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 3 \\ 5 \end{pmatrix}$$

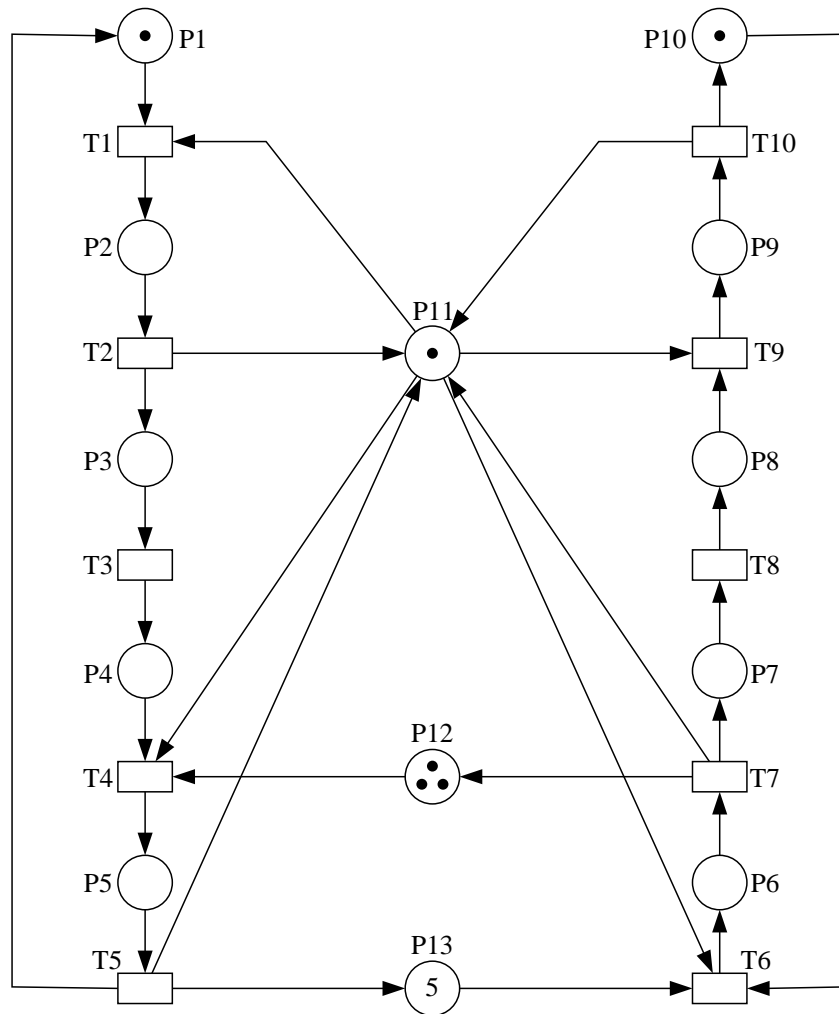


Abbildung 3.2: Petrinetzmodell einer Fertigungsanlage mit geänderten Beschriftungen

Jedes Petrinetz läßt sich somit eindeutig durch seine Inzidenzmatrix und die Anfangsmarkierung beschreiben, mit jedoch einer Ausnahme: Wenn eine Transition mit einer Stelle sowohl durch eine Eingangs- als auch eine Ausgangskante verbunden ist, bezeichnen wir diese Struktur als *Schlinge*. Schlingen sind in der Inzidenzmatrix jedoch nicht sichtbar. Es wird daher in Verfahren, in denen es möglich ist, auf die **PRE**- und **POST**-Matrix eines Petrinetzes zurückgegriffen mit $\mathbf{POST}(t, p) = K(t, p)$ und $\mathbf{PRE}(p, t) = K(p, t)$ für alle Stellen $p \in P$ und alle Transitionen $t \in T$. Für ein Netz N mit der Inzidenzmatrix $\mathbf{N} = \mathbf{POST}_N - \mathbf{PRE}_N$ repräsentieren die positiven und negativen Einträge der Inzidenzmatrix die Einträge der **POST**- und **PRE**-Matrizen vollständig, wenn es schlingenfrei ist.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	-1				+1					
P2	+1	-1								
P3		+1	-1							
P4			+1	-1						
P5				+1	-1					
P6						+1	-1			
P7							+1	-1		
P8								+1	-1	
P9									+1	-1
P10						-1				+1
P11	-1	+1		-1	+1	-1	+1		-1	+1
P12				-1			+1			
P13					+1	-1				

Abbildung 3.3: Inzidenzmatrix des Petrinetzmodells aus Abbildung 3.2

3.1.2 Markierungsgleichung

Jede Nachfolgemarkierung \mathbf{m}' einer Markierung \mathbf{m} , in der eine Transition t Konzeption besitzt, berechnet sich mittels $\mathbf{m}' = \mathbf{m} + \mathbf{t}$ wobei \mathbf{t} ein Vektor der Form $\mathbf{t} : P \rightarrow \mathbb{N}$ mit

$$\mathbf{t}(p) = \begin{cases} -K(p, t) & \text{falls } p \in \bullet t \setminus t \bullet \\ +K(t, p) & \text{falls } p \in t \bullet \setminus \bullet t \\ 0 & \text{sonst} \end{cases}$$

ist. Man beachte, daß diese Definition nur Petrinetze geeignet beschreibt, die frei von Schlingen sind. Über die Problematik mit Schlingen beinhaltenden Petrinetzen wird in Abschnitt 3.1.3 diskutiert.

Während \mathbf{t} ein Schalten einer Transition t beschreibt, läßt sich auch ein Vektor angeben, der das sukzessive Schalten mehrerer Transitionen angibt. Dieses Schalten mehrerer Transitionen wird als *Schaltsequenz* bezeichnet. Zu beachten ist, daß dieser (Zeilen-)Vektor, der sogenannte *Parikh-Vektor* ψ , nur die Anzahl des Schaltens jeder Transition einer Schaltsequenz ($\psi : T \rightarrow \mathbb{N}$), nicht jedoch deren Reihenfolge repräsentiert. Für eine gegebene Schaltsequenz ψ läßt sich die erreichbare Markierung \mathbf{m} wie folgt berechnen:

$$\mathbf{m}_0 + \mathbf{N} \cdot \psi = \mathbf{m}.$$

Für das Beispiel aus Abbildung 3.2 und das jeweils einmalige Schalten der Transitionen T1, T2, T3 und T4 aus der initialen Markierung heraus ergibt sich demnach folgende Markierung: Die Stellen P5 und P10 sind mit je einer Marke, die Stelle P12 mit zwei und die Stelle P13 mit fünf Marken belegt.

Nach Umstellung der genannten Gleichung erhalten wir eine Gleichung, die für die zukünftige Arbeit mit Petrinetzen auf Basis linearer Algebra von großer Bedeutung ist, die sogenannte *Markierungsgleichung*. Sei $N = (P, T, F, K, m_0)$ ein Petrinetz mit der Inzidenzmatrix N und m eine von m_0 aus erreichbare Markierung. Dann existiert eine natürlichzahlige Lösung von ψ für

$$N \cdot \psi = m - m_0.$$

3.1.3 Notwendig aber nicht hinreichend

Man beachte, daß diese Gleichung zwar als Testbedingung für die Erreichbarkeit einer Markierung verwendet werden kann, es sich aber nur um eine notwendige, nicht jedoch hinreichende Bedingung handelt. Die Gründe dafür liegen sowohl in der fehlenden Repräsentation von Schlingen in der Inzidenzmatrix als auch in der fehlenden Repräsentation der Schaltreihenfolge der Transitionen.

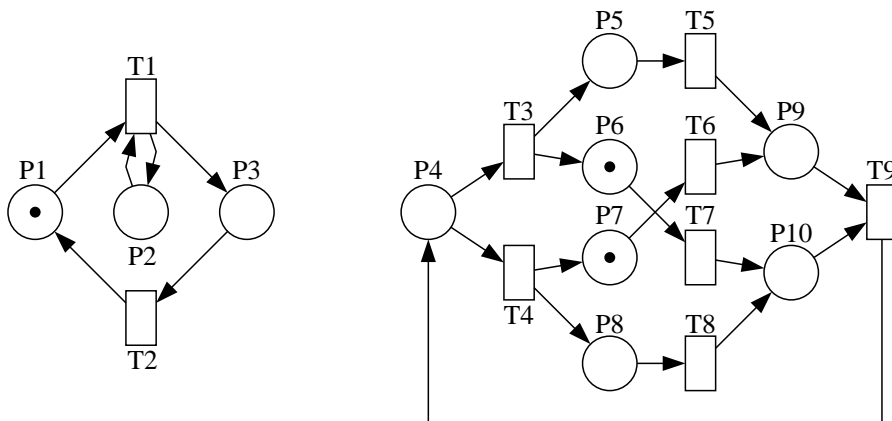


Abbildung 3.4: Markierungsgleichung testet nur notwendige Bedingung

Abbildung 3.4 zeigt zwei Beispiele, in denen mittels Markierungsgleichung nicht erreichbare Markierungen als erreichbar ermittelt werden. Das linke Beispiel verdeutlicht die mangelnde Repräsentation von Schlingen in der Inzidenzmatrix. Mittels Markierungsgleichung wird der Zustand, in dem nur die Stelle P_3 markiert ist, als erreichbar ermittelt. Die Nichtmarkierung der Stelle P_2 verhindert jedoch tatsächlich das Schalten der Transition T_1 . Die rechte Seite in Abbildung 3.4 zeigt ein Modell (aus [50]), in dem ein Zustand mit gleichzeitig markierten Stellen P_5 und P_8 nicht erreicht werden kann. Trotzdem wird mittels Markierungsgleichung diese Markierung als erreichbar berechnet, wenn die Transitionen T_3 und T_4 jeweils ein Mal und die Transitionen T_6 , T_7 und T_9 jeweils zwei Mal schalten. Bei Berücksichtigung der Schaltreihenfolge muß jedoch festgestellt werden, daß eine derartige Schaltsequenz nicht möglich ist.

3.1.4 Basialgorithmus zur Berechnung von Stellen-Invarianten

Lineare invariante Eigenschaften von Petrinetzen wurden erstmals in [90] untersucht. Diese Untersuchungen wurden in weiteren Arbeiten [96, 87] generalisiert. In [40] wurden Algorithmen zur Berechnung invarianter Eigenschaften von Petri-Netzmodellen verglichen und effizientere Verfahren vorgeschlagen. Auf der Basis linearalgebraischer Methoden wird in diesem Abschnitt die Grundidee der Berechnung minimaler Stellen-Invarianten am Beispiel aus Abbildung 3.2 skizziert.

Zur Berechnung aller minimalen Stellen-Invarianten wird die *transponierte* Inzidenzmatrix \mathbf{N}^T verwendet. Der (Spalten-)Vektor \mathbf{i} ist genau dann eine Stellen-Invariante, wenn $\mathbf{N}^T \cdot \mathbf{i} = \mathbf{0}$ gilt (vgl. Definition 3.3). Zur Berechnung aller minimalen Stellen-Invarianten wird die *transponierte* Inzidenzmatrix wie folgt partitioniert:

$$\mathbf{N}^T = \begin{array}{c} \begin{array}{cc} \xleftrightarrow{m-r} & \xleftrightarrow{r} \\ \mathbf{N}_{11}^T & \mathbf{N}_{12}^T \\ \mathbf{N}_{21}^T & \mathbf{N}_{22}^T \end{array} \quad \begin{array}{c} \updownarrow \\ r \\ \updownarrow \\ n-r \end{array} \end{array} \quad r \text{ ist der Rang von } \mathbf{N}^T$$

Dabei ist zu beachten, daß die Teilmatrix \mathbf{N}_{12}^T nicht-singulär, deren Determinante also ungleich Null sein muß. Das muß jedoch nicht für alle Anordnungen der Zeilen und Spalten von \mathbf{N}^T der Fall sein! Transponiert man die in Abbildung 3.3 angegebene Inzidenzmatrix des Beispielmotells, ist die Matrix \mathbf{N}_{12}^T singulär. Vertauscht man jedoch die Spalten geeignet, z.B. wie in Abbildung 3.5 dargestellt, erhält man eine nicht-singuläre Teilmatrix \mathbf{N}_{12}^T .

	P5	P2	P9	P4	P1	P6	P7	P8	P3	P10	P11	P12	P13
T1		+1			-1						-1		
T2		-1							+1		+1		
T10			-1							+1	+1		
T4	+1			-1							-1	-1	
T5	-1				+1						+1		+1
T6						+1				-1	-1		-1
T7						-1	+1				+1	+1	
T8							-1	+1					
T9			+1					-1			-1		
T3				+1					-1				

Abbildung 3.5: Transponierte Inzidenzmatrix des Petrinetzmodells aus Abb. 3.2

Der Rang der Inzidenzmatrix des Beispiels beträgt 9, die entsprechende Partitionierung in vier Untermatrizen ist in Abbildung 3.5 gezeigt. Unter Verwendung der (rücktransponierten) Untermatrizen \mathbf{N}_{11} und \mathbf{N}_{12} wird eine Matrix \mathbf{I}_N konstruiert mit

$$\mathbf{I}_N = [\mathbf{E}_{m-r} \mid -\mathbf{N}_{11}(\mathbf{N}_{12})^{-1}].$$

E_{m-r} steht dabei für die $(m-r) \times (m-r)$ Einheitsmatrix. Die konstruierte Matrix I_N besitzt die Dimension $(m-r) \times m$ und jede ihrer Zeilen ergibt einen Lösungsvektor für das Gleichungssystem $N^T \cdot \mathbf{i} = 0$. Dabei ist jedoch zu beachten, daß diese Zeilen auch negative Einträge enthalten können, die durch Linearkombination mit den anderen Zeilen eliminiert werden müssen, um der Definition für Stellen-Invarianten (siehe Definition 3.3) zu genügen. Für das gezeigte Beispiel ergibt sich folgende Matrix (man beachte dieselbe Stellenzuordnung der Spalten wie in Abbildung 3.5):

$$I_N = \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & +1 & +1 \\ 0 & +1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & +1 & -1 & -1 \\ 0 & 0 & +1 & 0 & 0 & +1 & +1 & +1 & 0 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & +1 & 0 & +1 & +1 & +1 & +1 & -1 & 0 & 0 \end{bmatrix}$$

Abbildung 3.6: Lösungsmatrix I_N des Petrinetzmodells aus Abbildung 3.2

Werden zur Zeile 4 die Zeilen 1 und 2 und zur Zeile 2 die Zeilen 1 und 3 addiert, ergibt sich folgende Matrix ohne negative Einträge:

$$I_N = \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & +1 & +1 \\ +1 & +1 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & +1 & +1 & +1 & 0 & +1 & 0 & 0 & 0 \\ +1 & +1 & 0 & +1 & +1 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Abbildung 3.7: Lösungsmatrix I_N ohne negative Einträge

Das Netz aus Abbildung 3.2 enthält demnach vier minimale Stellen-Invarianten:

$$\begin{aligned} P5 + P6 + P12 + P13 \\ P5 + P2 + P9 + P6 + P11 \\ P9 + P6 + P7 + P8 + P10 \\ P5 + P2 + P4 + P1 + P3. \end{aligned}$$

Nach Abbildung dieser Stellen-Invarianten auf ihre initiale Markierung ergeben sich die folgenden für jede erreichbare Markierung invarianten Markierungsgleichungen:

$$\begin{aligned} P5 + P6 + P12 + P13 &= 8 \\ P5 + P2 + P9 + P6 + P11 &= 1 \\ P9 + P6 + P7 + P8 + P10 &= 1 \\ P5 + P2 + P4 + P1 + P3 &= 1. \end{aligned}$$

Die in der zweiten Zeile beschriebene Stellen-Invariante ist der formale Ausdruck der zu Beginn des Abschnittes 3.0.5 erläuterten Interpretation der modellierten Eigenschaften des Roboters. Die erste Invariante interpretiert für das Modell aus Abbildung 3.1, daß 8 Lagerplätze entweder frei sind, frei werden, belegt sind oder belegt werden. In dieser Art läßt sich für jedes sinnvolle Modell eine geeignete Interpretation finden, die zusätzliche Aussagen über die korrekte Abbildung der Realität ermöglicht.

Der beschriebene Basisalgorithmus ist aus den genannten Gründen für eine effiziente Berechnung von Stellen-Invarianten kaum geeignet. Im folgenden wird ein Algorithmus vorgestellt, der Stellen-Invarianten aber auch Fallen und Siphons effizient berechnet.

3.2 Effizienter Algorithmus zur Berechnung von Stellen-Invarianten

Der neue, in TimeNET integrierte Algorithmus zur Berechnung von Stellen-Invarianten \mathbf{i} besteht aus zwei Schritten. Zunächst wird eine sogenannte \mathbb{Z} -Basis \mathbf{Z} bestimmt, die ganzzahlige Lösungen für \mathbf{i} liefert. Diese \mathbb{Z} -Basis \mathbf{Z} wird durch Überführen der transponierten Inzidenzmatrix in die *hermitesche Normalform* bestimmt. Dabei ist eine $m \times n$ Matrix \mathbf{M} mit dem Rang r in (Spalten-)hermitescher Normalform, wenn die ersten r Spalten von \mathbf{M} verschieden von Null und in unterer Dreiecksgestalt sind, wobei die Diagonalelemente h_{ii} für $1 \leq i \leq r$ sowohl größer Null als auch das Maximum jeder Zeile sind.

Zur Überführung der transponierten Inzidenzmatrix in die hermitesche Normalform wird ein in [68] vorgeschlagener Algorithmus verwendet. Dieser hat gegenüber den in [83, 39] vorgeschlagenen Methoden den Vorteil, daß die Einträge in der Transformationsmatrix, die mit der in die hermitesche Normalform zu überführende Matrix multipliziert wird, kleiner sind, als bei den bis dahin bekannten Methoden, was die weitere Bearbeitung begünstigt.

Der zweite Schritt besteht darin, die ganzzahligen Stellen-Invarianten \mathbf{i} in nicht-negative zu überführen. Um eine nicht-negative Lösung bei der Berechnung der Stellen-Invarianten \mathbf{i} zu ermöglichen, werden nicht Lösungen für Systeme nur von Gleichungen, sondern zusätzlich auch von Ungleichungen $\mathbf{i} \geq 0$ gelöst. Zur Lösung dieser sogenannten *Einschränkungssysteme* gibt es zwei Arten von Methoden. Zum einen gibt es pivotierende [8, 43, 95] Verfahren, wie das Simplexverfahren, zum anderen nicht-pivotierende Verfahren, zu denen die in Abschnitt 3.2.2 vorgestellte *doppelte Beschreibungsmethode* [103] zählt.

3.2.1 Anwendung am Beispiel

Anhand des Beispiels aus Abbildung 3.2 wird der nun in TimeNET verwendete Algorithmus beschrieben.

Zuerst wird die hermitesche Normalform \mathbf{H}_N der transponierten Inzidenzmatrix \mathbf{N}^T und die assoziierte Transformationsmatrix \mathbf{V} mit $\mathbf{H}_N = \mathbf{N}^T \cdot \mathbf{V}$ unter Verwendung des in [68] beschriebenen Algorithmus berechnet. Dem Algorithmus liegen initial die transponierte $m \times n$ -Inzidenzmatrix $\mathbf{H}_N := \mathbf{N}^T$ (siehe Abbildung 3.8) und die Einheitsmatrix $\mathbf{V} := \mathbf{E}_n$ als Eingangsmatrizen zugrunde. Durch geeignete Umstellungen wie das Addieren eines ganzzahligen Vielfachen einer Spalte auf eine andere, das Multiplizieren einer Spalte mit (-1) und das Vertauschen zweier Spalten wird die Matrix \mathbf{H}_N in hermitesche Normalform und \mathbf{V} als korrespondierende Transformationsmatrix berechnet und ausgegeben (siehe Abbildung 3.9).

$$\mathbf{N}^T = \begin{pmatrix} -1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ +1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & +1 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & +1 & +1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & +1 & 0 & 0 \end{pmatrix}$$

Abbildung 3.8: Transponierte Inzidenzmatrix \mathbf{N}^T des Petrinetzmodells aus Abb. 3.2

Die im folgenden Schritt gebildete \mathbb{Z} -Basis wird aus der Matrix \mathbf{V} ausgelesen. Der Rang r der Matrix \mathbf{N}^T betragt, wie an der Matrix \mathbf{H}_N leicht ablesbar, 9. Aus \mathbf{V} werden die $n - r$ letzten Spalten $\mathbf{v}_{r+1} \dots \mathbf{v}_n$, im Beispiel also $\mathbf{v}_{10} \dots \mathbf{v}_{13}$ ausgelesen und bilden die \mathbb{Z} -Basis \mathbf{Z}_N von \mathbf{N}^T (siehe Abbildung 3.10).

Mittels doppelter Beschreibungsmethode werden im dritten Schritt die Losungen \mathbf{U} des Ungleichungssystems $\mathbf{Z} \cdot \mathbf{u} \geq 0$ berechnet. Die Matrixmultiplikation der \mathbb{Z} -Basis \mathbf{Z}_N mit der Losungsmatrix \mathbf{U} ergibt die die Stellen-Invarianten beschreibende Matrix \mathbf{I}_N^T (siehe Abbildung 3.11). Dieser letzte Schritt wird im folgenden Abschnitt 3.2.2 beschrieben.

$$\mathbf{H}_N = \mathbf{N}^T \cdot \mathbf{V}$$

$$= \mathbf{N}^T \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 & -1 & -1 \\ +1 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & -1 \\ +1 & +1 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 & -1 & -1 \\ +1 & +1 & +1 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & 0 & 0 & 0 & +1 & 0 & 0 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & 0 & 0 & +1 & 0 & 0 \\ +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & 0 & +1 & +1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 \\ -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \end{pmatrix}$$

$$= \begin{pmatrix} +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung 3.9: Überführung der transponierten Inzidenzmatrix \mathbf{N}^T in die hermitesche Normalform \mathbf{H}_N mittels Transformationsmatrix \mathbf{V}

$$\mathbf{Z}_N = \begin{pmatrix} +1 & 0 & -1 & -1 \\ +1 & 0 & 0 & -1 \\ +1 & 0 & -1 & -1 \\ +1 & 0 & -1 & -1 \\ 0 & +1 & 0 & 0 \\ 0 & +1 & +1 & +1 \\ 0 & +1 & 0 & 0 \\ 0 & +1 & 0 & 0 \\ 0 & +1 & +1 & 0 \\ 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & +1 \end{pmatrix}$$

Abbildung 3.10: \mathbb{Z} -Basis \mathbf{Z}_N von \mathbf{N}^\top

$$\mathbf{I}_N^\top = \mathbf{Z}_N \cdot \mathbf{U} = \mathbf{Z}_N \cdot \begin{pmatrix} +1 & +1 & 0 & +1 \\ 0 & 0 & +1 & +1 \\ 0 & +1 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & +1 \\ 0 & +1 & 0 & +1 \\ 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & +1 \\ +1 & +1 & 0 & +1 \\ +1 & +1 & +1 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & +1 & +1 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & +1 & 0 & 0 \\ +1 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung 3.11: Berechnung der die Stellen-Invarianten beschreibenden Matrix \mathbf{I}_N aus der \mathbb{Z} -Basis \mathbf{Z}_N und der Lösungsmatrix \mathbf{U} des Ungleichungssystems $\mathbf{Z}_N \cdot \mathbf{u} \geq 0$

3.2.2 Die doppelte Beschreibungsmethode

Bei der doppelten Beschreibungsmethode werden Variablen des Einschränkungssystems durch Einsetzen aller paarweisen Linearkombinationen von Ungleichungen eliminiert. Dieses Verfahren wird anschaulicher durch das Wechseln in Beschreibungsmethoden der konvexen Geometrie: ein Polyeder kann auf zwei Arten beschrieben werden, als Lösungsraum eines Einschränkungssystems oder als konvexe Hülle seiner Ecken. Die erstere Beschreibungsmöglichkeit wird \mathcal{L} (ösungsraum)-Repräsentation, letztere \mathcal{H} (üllen)-Repräsentation genannt.

Abbildung 3.12 zeigt ein Beispiel und seine Definition in der jeweiligen Repräsentation. Auf die konvex-geometrische Definition der Begriffe *Gerade*, *Strahl* und *Ecke* wird aufgrund der Übereinstimmung mit dem intuitiven Verständnis an dieser Stelle verzichtet und stattdessen auf [130] verwiesen.

	\mathcal{L} -Repräsentation	\mathcal{H} -Repräsentation
Bsp.	$2x - y = 0$ $z \geq 1$	Gerade(1, 2, 0) Strahl(0, 0, 1) Ecke(0, 0, 1)
Def.	$\mathcal{P} = \{\mathbf{x} \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{C}\mathbf{x} \geq \mathbf{d}\}$	$\mathcal{P} = \{\mathbf{x} \mathbf{G}\boldsymbol{\lambda} + \mathbf{S}\boldsymbol{\mu} + \mathbf{E}\boldsymbol{\nu}, \boldsymbol{\mu}, \boldsymbol{\nu} \geq \mathbf{0}, \sum \nu_i = 1\}$

Abbildung 3.12: Doppelte Beschreibung eines Polyeders \mathcal{P}

Das Beispiel repräsentiert eine schief in den Raum projizierte Fläche, deren Ausbreitung in die „untere“ Richtung bei $z = 1$ begrenzt ist. Abbildung 3.13 zeigt die geometrische Darstellung des Beispiels. Die allgemeine Beschreibung der \mathcal{L} -Repräsentation enthält die Koeffizientenmatrizen \mathbf{A} und \mathbf{C} sowie die Einschränkungsvektoren \mathbf{b} und \mathbf{d} . Ein Polyeder in \mathbb{R}^n kann demnach auch als Schnittmenge endlich vieler Halbräume in \mathbb{R}^n aufgefaßt werden. Für die allgemeine Beschreibung eines Polyeders in der \mathcal{H} -Repräsentation sind die Spalten der Matrix \mathbf{G} die Geraden, die Spalten der Matrix \mathbf{S} repräsentieren die Strahlen und die der Matrix \mathbf{E} die Ecken.

Die doppelte Beschreibungsmethode gehört zu den sogenannten *einfügenden* Methoden. Im folgenden wird eine vereinfachende Veranschaulichung der auf der doppelten Beschreibungsmethode basierenden Verfahren zur Lösung von Einschränkungssystemen vorgestellt [94]. Als Ausgangspunkt der Veranschaulichung sei ein in Abbildung 3.14 dargestellter Polyeder \mathcal{P} angenommen, dessen Ecken E_1, E_2 durch die Kenntnis über die \mathcal{P} beschreibenden Strahlen bekannt sind. Wird nun eine neue Ungleichung repräsentierende Einschränkung H (für die Repräsentation der Einschränkung als sogenannte *Hyperebene*) eingefügt, entsteht ein neuer Polyeder \mathcal{P}' , der durch H begrenzt in H^+ liegt.

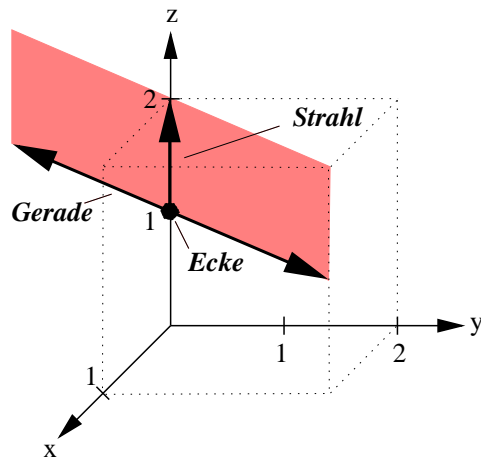


Abbildung 3.13: Geometrische Darstellung des Beispiels aus Abbildung 3.12

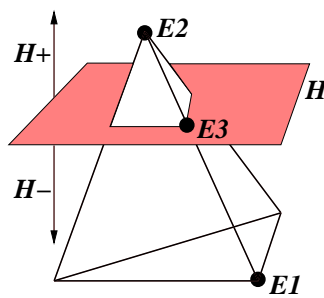


Abbildung 3.14: Geometrische Darstellung des Einfügens einer Einschränkung

Das Problem besteht nun darin, die neu entstandene Ecke E_3 , die eine Lösung des Einschränkungssystem repräsentiert, zu bestimmen. Dazu müssen alle das neue Polyeder P' beschreibenden Strahlen bestimmt werden. Bezeichnen wir alle die Strahlen, die den Polyeder P in H^- beschreiben mit S^- und alle Strahlen, die den Polyeder P in H^+ beschreiben mit S^+ müßte unter Hinzunahme der Informationen über H ein Kreuzprodukt aller Strahlen $S^+ \times S^-$ berechnet werden, um den neuen Polyeder P' geeignet bestimmen zu können. Dabei entstehen Kombinationen von Strahlen, die redundante Informationen über das Polyeder P' enthalten. Eine effizientere Methode als die Kombination aller S^+ und S^- bietet der Vorschlag [97], nur *adjazente* Strahlen miteinander konvex zu kombinieren. Dadurch erfolgt die Beschreibung von P' mit minimalen Informationen. Diese Beschreibung mit minimalen Informationen bietet einen großen Vorteil bei der Berechnung von Stellen-Invarianten. Repräsentiert der erzeugte Polyeder den Lösungsraum zur Berechnung von Stellen-Invarianten, so führt die Verwendung des Adjazenztests genau zur Berechnung der *minimalen* Stellen-Invarianten.

Ein neuer Adjazenztest-Algorithmus

In diesem Abschnitt wird der Begriff *adjazent* definiert und eine effiziente Implementierung des Adjazenztests vorgestellt.

Seien zwei Strahlen als *gesättigt* bezeichnet, wenn ihre Kombination eine Einschränkung repräsentierende Hyperebene H beschreibt, wie sie in Abbildung 3.14 vorgestellt wurde. Dann wird die Adjazenz zweier Strahlen wie folgt definiert (aus [75]):

3.4 Definition (adjazent)

Zwei Strahlen $\mathbf{r}_1, \mathbf{r}_2$ sind adjazent gdw. die gemeinsame Menge von Einschränkungen c , die von beiden Strahlen \mathbf{r}_1 und \mathbf{r}_2 gesättigt wird, nicht auch von einem anderen Strahl \mathbf{r} gesättigt wird: Sei $c = \{i | \mathbf{a}_i^T \mathbf{r}_1 = 0\} \cap \{i | \mathbf{a}_i^T \mathbf{r}_2 = 0\}$. Dann gibt es keinen Strahl \mathbf{r} mit $c_r = \{i | \mathbf{a}_i^T \mathbf{r} = 0\}$, so daß $c \subset c_r$.

Die Effizienz der doppelten Beschreibungsmethode hängt sehr stark vom Adjazenztest ab. In [62] wird ein effizienter Adjazenztest vorgeschlagen. Dieser Vorschlag wurde in die Stellen-Invarianten-Berechnung in `TimeNET` integriert [75]. Darin werden nicht mehr alle möglichen Paare von Strahlen geprüft, die durch eine neu eingefügte Hyperebene getrennt wurden, sondern Paare von adjazenten Strahlen gesucht, die *in* der neu eingefügten Hyperebene liegen *und* die von einer *zukünftig* eingefügten Hyperebene getrennt werden. Dadurch wird die Anzahl der zu testenden Paare von Strahlen deutlich reduziert und darüber hinaus erfolgt der Adjazenztest nun bereits in einem früheren Iterationsschritt, in dem der Polyeder noch durch weniger Strahlen repräsentiert wird.

3.2.3 Doppelte Beschreibungsmethode für die Berechnung von Fallen und Siphons

Mit Hilfe der doppelten Beschreibungsmethode lassen sich auch Fallen und Siphons [41] berechnen (siehe auch Abschnitt 3.0.5). Dazu wird die Inzidenzmatrix des zu untersuchenden Petrinetzes modifiziert [56, 55]. Für Fallen wird die **POST**-Matrix eines Petrinetzes N mit der Inzidenzmatrix $\mathbf{N} = \mathbf{POST} - \mathbf{PRE}$ derart geändert, daß die Gewichtung jeder Ausgangskante einer Transition durch die Summe der Gewichtung aller Eingangskanten der Transition ersetzt wird:

$$\mathbf{N}_F = \mathbf{POST}_F - \mathbf{PRE}_F \quad \text{mit} \quad \mathbf{PRE}_F = \mathbf{PRE}$$
$$\text{und} \quad \mathbf{POST}_F(p, t) = \begin{cases} \sum_{p' \in \bullet t} \text{pre}[p', t] & \text{wenn } p \in t^\bullet \\ 0 & \text{sonst.} \end{cases}$$

Zur Berechnung von Siphons wird die **PRE**-Matrix des zu untersuchenden Petrinetzes N mit der Inzidenzmatrix $\mathbf{N} = \mathbf{POST} - \mathbf{PRE}$ derart geändert, daß die

Modell aus	Anzahl Stellen-Invar.	TimeNET	PolyLib	cdd	GreatSPN	INA
[75] Abb.3.3	65536	78,0	260,6	248,0	787,4	10920,0
[70]	38941	218,1	1456,0	1727,6	-	-
[77] Abb.10 (4 Manager)	1023	0,9	0,6	3,0	2,1	8,5
[77] Abb.12 (3 Teilnehmer)	947	1,0	1,6	0,5	32,8	251,3
[24] Abb.1a	450	0,4	0,3	0,6	1,1	2,9

Abbildung 3.15: Vergleich der Dauern zur Berechnung von Stellen-Invarianten

Gewichtung jeder Eingangskante einer Transition durch die Summe der Gewichtung aller Ausgangskanten der Transition ersetzt wird:

$$N_s = \text{POST}_s - \text{PRE}_s \quad \text{mit} \quad \text{POST}_s = \text{POST}$$

$$\text{und} \quad \text{PRE}_s(p, t) = \begin{cases} \sum_{p' \in t \bullet} \text{post}[p', t] & \text{wenn } p \in \bullet t \\ 0 & \text{sonst.} \end{cases}$$

Unter Verwendung dieser modifizierten Inzidenzmatrizen wird das System von Ungleichungen der Form $\mathbf{y} \cdot \mathbf{N}_F \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$ ($\mathbf{y} \cdot \mathbf{N}_S \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$) gelöst, wobei $\|\mathbf{y}\|$ die Stellenmenge einer Falle (eines Siphons) des Netzes N beschreibt.

Der hier vorgestellte Algorithmus zur Berechnung von Fallen und Siphons ist im Modellierungs- und Analysewerkzeug für stochastische Petrinetze TimeNET [134] implementiert worden [75]. Vergleicht man die Dauern der Berechnung von Stellen-Invarianten durch TimeNET mit anderen Werkzeugen, so zählt TimeNET zu den effizientesten Werkzeugen im Bereich der strukturellen Analyse von Petrinetzen. Die Tabelle aus Abbildung 3.15 ist [76, 75] entnommen und zeigt einen entsprechenden Vergleich. Alle Zeitangaben sind in Sekunden. Für das in der zweiten Zeile der Tabelle angegebene Modell wurde die Berechnung der Stellen-Invarianten durch GreatSPN [67, 28] und INA [122] nach zwei Tagen ergebnislos abgebrochen. Die Werkzeuge PolyLib [130] und cdd [61] sind Implementierungen der doppelten Beschreibungsmethode. Diesen beiden Werkzeugen wurden im Test die bereits vorberechneten \mathbb{Z} -Basen der Modelle eingegeben. Die Messungen erfolgten auf einer SunU10-300 Workstation mit 128MB Hauptspeicher.

3.2.4 Begriffsklärungen

Ein abschließender Abschnitt ist der Begriffsklärung gewidmet, da sich in der Literatur unterschiedliche Bezeichnungen für dieselbe Eigenschaft finden.

Sowohl im deutschen als auch im englischen Sprachgebrauch wird einzig der Begriff Falle (engl. *trap*) einheitlich verwendet. Siphons wurden im englischen Sprachgebrauch ursprünglich als *deadlocks* bezeichnet. Durch die Verwechselbarkeit mit der gleichbezeichneten Eigenschaft von Petrinetzen hat sich mittlerweile jedoch der Begriff Siphon in Assoziation an einen Wasserspender durchgesetzt. Im Deutschen wird mit Siphon jedoch eher das Wasserauffangknie assoziiert, das sich aber gerade wie eine Falle für Wasser verhält. Es gibt daher Vorschläge, Siphons im Deutschen als *Co-Falle* zu bezeichnen. Da die Kenntnis über die Semantik des Begriffes Siphon in Beziehung zu Petrinetzen jedoch verbreiteter scheint als für den Begriff Co-Falle, wird trotz der Widersprüchlichkeit in der Assoziation dieser Begriff in dieser Arbeit verwendet.

Für den Begriff Stellen-Invariante gibt es die sinnvolle Entsprechung *place invariant* im Englischen. Der die Stellenmenge einer Stellen-Invariante beschreibende Vektor \mathbf{i} wird jedoch häufig auch als *p-flow* oder für $\mathbf{i} \geq \mathbf{0}$ *p-semiflow* bezeichnet. Diese Begriffe wurden eingeführt, um die Unterscheidung zu anderen Konstrukten zu verdeutlichen, wie z.B. Gleichungen, die eine invariante Eigenschaft beschreiben. Demnach wird die Gleichung $\mathbf{i} \cdot \mathbf{m} = \mathbf{i} \cdot \mathbf{m}_0 = k$ als *token conservation law* oder *marking invariant* bezeichnet.

Kapitel 4

Matrixreduzierende Verfahren

Zur Berechnung der stationären Zustandswahrscheinlichkeiten muß, wie in Abschnitt 2.5.1 beschrieben, der gesamte Erreichbarkeitsgraph des Netzes erstellt werden. Für jeden der n andauernden Zustände wird die Zustandswahrscheinlichkeit durch Lösung eines Gleichungssystems berechnet, wobei eine Generatormatrix der Größe $n \times n$ dieses Gleichungssystem beschreibt. Die Größe dieser Matrix ist neben der Größe des Erreichbarkeitsgraphen die stärkste Limitierung in der Anwendbarkeit der numerischen Analyse von Petrinetzen.

Unter Verwendung von Methoden der *Kronecker-Algebra* (oft auch als *Tensor-Algebra* bezeichnet) ist es möglich, nicht die gesamte Generatormatrix abzuspeichern, sondern nur einzelne Submatrizen, aus denen sich die Einträge der Generatormatrix berechnen lassen. Im Rahmen dieser Arbeit wurden Kronecker-Algebra basierte Methoden in das Modellierungs- und Analysewerkzeug für stochastische Petrinetze TimeNET integriert. Im folgenden wird nach einer Einleitung die Grundidee der Verwendung von Kronecker-Algebra zur Analyse stochastischer Petrinetze anhand eines Beispiels erläutert.

4.1 Kronecker-Algebra

Der erste Vorschlag zur Verwendung von Kronecker-algebraischen Methoden zur numerischen Analyse von stochastischen Petrinetzen findet sich in [114] für Netzwerke stochastischer Automaten. Dieser Vorschlag wurde auf allgemeine stochastische Petrinetze in [53, 84, 15, 34] angewendet. Neuere Arbeiten untersuchen bei der Anwendung von Kronecker-Operatoren die Komplexität verschiedener Methoden der Vektor-Matrix-Multiplikation [19], hierarchische Verwaltung von Zuständen bei der Erstellung des Erreichbarkeitsgraphen [20] und

die Analysemöglichkeiten von Netzen mit zeitlosen Transitionen als Synchronisationstransitionen [54].

4.1.1 Ein einführendes Beispiel

Die Anwendung der Kronecker-Operatoren wird anhand eines in Abbildung 4.1 gezeigten Beispiels verdeutlicht. Die Raten der Transitionen T1 und T3 seien λ_1 und λ_3 und μ die der Transition T2.

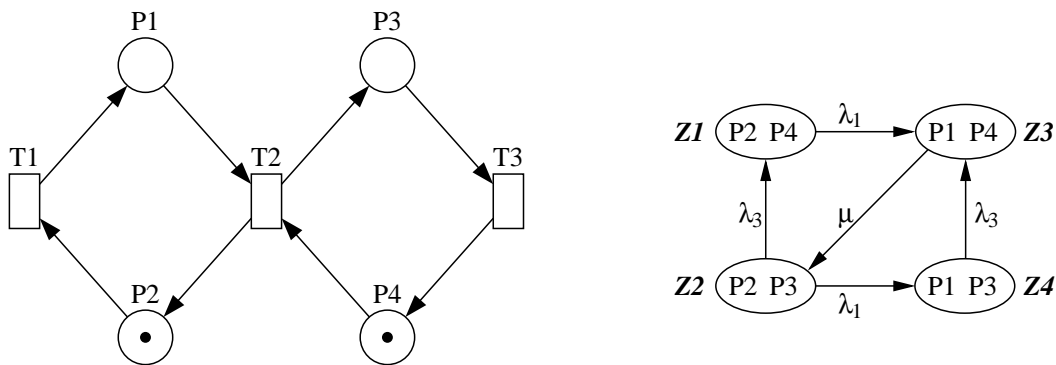


Abbildung 4.1: Beispiel – Gesamtmodell und Erreichbarkeitsgraph

Für die Analyse mit Hilfe von Kronecker-Operatoren wird das Gesamtmodell in mehrere sich synchronisierende Teilmodelle zerlegt. Dazu müssen zuerst die Synchronisationstransitionen des Gesamtmodells identifiziert werden. Danach wird die Möglichkeit überprüft, das Modell an den Synchronisationstransitionen so zu „zerschneiden“, daß voneinander im Weiteren unabhängige Teilmodelle entstehen. Für das in Abbildung 4.1 gezeigte Beispiel läßt sich das Modell in der beschriebenen Weise an der Synchronisationstransition T2 teilen. Die Zerlegung ist in Abbildung 4.2 dargestellt.

Die Transition T2 repräsentiert, wie bei der Zerlegung beschrieben, ein Synchronisationsereignis zweier Teilmodelle. Die durch die Transitionen T1 und T3 repräsentierten Ereignisse sind jeweils lokal. Diese Unterscheidung zwischen *lokalen* und *synchronen* Ereignissen ist Basis der Kronecker-algebraischen Analyse. Zur Vereinfachung wird zunächst die Erzeugung der Übergangsratenmatrix des Gesamtmodells mittels Kronecker-Operatoren gezeigt. Die Übergangsratenmatrizen \mathbf{R}_A und \mathbf{R}_B der Teilmodelle A und B sind in Abbildung 4.3 dargestellt. Außerdem werden Matrizen $\mathbf{R}_A^{\text{lokal}}$ und $\mathbf{R}_B^{\text{lokal}}$ eingeführt, die nur die Raten der lokalen Transitionen zeigen (siehe Abbildung 4.4). Das Synchronisationsereignis wird durch die Übergangswahrscheinlichkeitsmatrizen $\mathbf{W}_A^{\text{sync}}$ und $\mathbf{W}_B^{\text{sync}}$ repräsentiert. Im gezeigten Beispiel sind die Übergangswahrscheinlichkeiten von

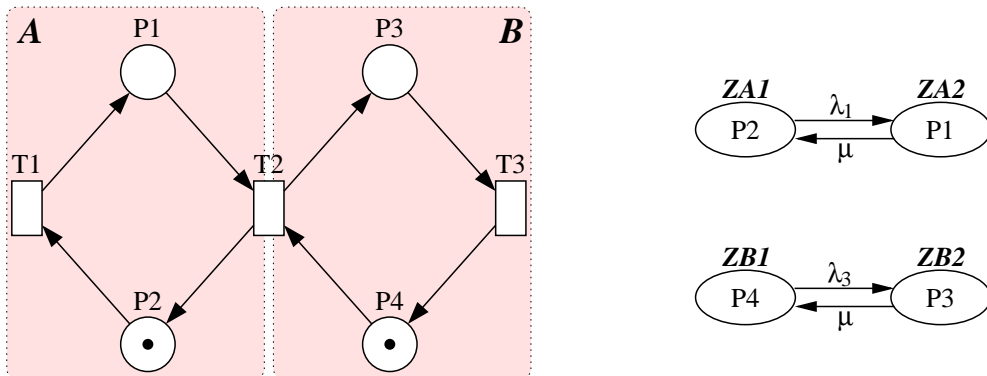


Abbildung 4.2: Beispiel – Teilmodelle und zugehörige Erreichbarkeitsgraphen

$$\mathbf{R}_A = \begin{pmatrix} 0 & \lambda_1 \\ \mu & 0 \end{pmatrix} \quad \mathbf{R}_B = \begin{pmatrix} 0 & \mu \\ \lambda_3 & 0 \end{pmatrix}$$

Abbildung 4.3: Übergangsratenmatrizen der Teilmodelle A und B

dem Zustand **ZA2** in den Zustand **ZA1** und von dem Zustand **ZB2** in den Zustand **ZB1** zu gelangen jeweils gleich 1 (siehe Abbildung 4.5).

Wie bereits erwähnt, beruht auf der Unterscheidung zwischen lokalen und synchronisierenden Ereignissen die Grundidee der Kronecker-Algebra basierten Analyse. Entsprechend dieser Unterscheidung werden zwei Kronecker-Operatoren eingeführt: die Kronecker-Summe \oplus und das Kronecker-Produkt \otimes . Für zwei Matrizen

$$\mathbf{A} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \quad \text{und} \quad \mathbf{B} = \begin{pmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{pmatrix}$$

sind das Kronecker-Produkt und die Kronecker-Summe definiert als

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{00} \cdot \mathbf{B} & a_{01} \cdot \mathbf{B} \\ a_{10} \cdot \mathbf{B} & a_{11} \cdot \mathbf{B} \end{pmatrix}$$

$$= \begin{pmatrix} a_{00} \cdot b_{00} & a_{00} \cdot b_{01} & a_{00} \cdot b_{02} & a_{01} \cdot b_{00} & a_{01} \cdot b_{01} & a_{01} \cdot b_{02} \\ a_{00} \cdot b_{10} & a_{00} \cdot b_{11} & a_{00} \cdot b_{12} & a_{01} \cdot b_{10} & a_{01} \cdot b_{11} & a_{01} \cdot b_{12} \\ a_{00} \cdot b_{20} & a_{00} \cdot b_{21} & a_{00} \cdot b_{22} & a_{01} \cdot b_{20} & a_{01} \cdot b_{21} & a_{01} \cdot b_{22} \\ a_{10} \cdot b_{00} & a_{10} \cdot b_{01} & a_{10} \cdot b_{02} & a_{11} \cdot b_{00} & a_{11} \cdot b_{01} & a_{11} \cdot b_{02} \\ a_{10} \cdot b_{10} & a_{10} \cdot b_{11} & a_{10} \cdot b_{12} & a_{11} \cdot b_{10} & a_{11} \cdot b_{11} & a_{11} \cdot b_{12} \\ a_{10} \cdot b_{20} & a_{10} \cdot b_{21} & a_{10} \cdot b_{22} & a_{11} \cdot b_{20} & a_{11} \cdot b_{21} & a_{11} \cdot b_{22} \end{pmatrix}$$

$$\mathbf{R}_A^{\text{lokal}} = \begin{pmatrix} 0 & \lambda_1 \\ 0 & 0 \end{pmatrix} \quad \mathbf{R}_B^{\text{lokal}} = \begin{pmatrix} 0 & 0 \\ \lambda_3 & 0 \end{pmatrix}$$

Abbildung 4.4: Übergangsratenmatrizen für die lokalen Ereignisse der Teilmodelle **A** und **B**

$$\mathbf{W}_A^{\text{sync}} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \mathbf{W}_B^{\text{sync}} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Abbildung 4.5: Übergangswahrscheinlichkeitsmatrizen für das Synchronisationsereignis der Teilmodelle **A** und **B**

$$\begin{aligned} \mathbf{A} \oplus \mathbf{B} &= \mathbf{A} \otimes \mathbf{E}_B + \mathbf{E}_A \otimes \mathbf{B} \\ &= \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{pmatrix} \\ &= \begin{pmatrix} a_{00} & 0 & 0 & a_{01} & 0 & 0 \\ 0 & a_{00} & 0 & 0 & a_{01} & 0 \\ 0 & 0 & a_{00} & 0 & 0 & a_{01} \\ a_{10} & 0 & 0 & a_{11} & 0 & 0 \\ 0 & a_{10} & 0 & 0 & a_{11} & 0 \\ 0 & 0 & a_{10} & 0 & 0 & a_{11} \end{pmatrix} + \begin{pmatrix} b_{00} & b_{01} & b_{02} & 0 & 0 & 0 \\ b_{10} & b_{11} & b_{12} & 0 & 0 & 0 \\ b_{20} & b_{21} & b_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{00} & b_{01} & b_{02} \\ 0 & 0 & 0 & b_{10} & b_{11} & b_{12} \\ 0 & 0 & 0 & b_{20} & b_{21} & b_{22} \end{pmatrix} \\ &= \begin{pmatrix} a_{00} + b_{00} & b_{01} & b_{02} & a_{01} & 0 & 0 \\ b_{10} & a_{00} + b_{11} & b_{12} & 0 & a_{01} & 0 \\ b_{20} & b_{21} & a_{00} + b_{22} & 0 & 0 & a_{01} \\ a_{10} & 0 & 0 & a_{11} + b_{00} & b_{01} & b_{02} \\ 0 & a_{10} & 0 & b_{10} & a_{11} + b_{11} & b_{12} \\ 0 & 0 & a_{10} & b_{20} & b_{21} & a_{11} + b_{22} \end{pmatrix} \end{aligned}$$

Mit Hilfe dieser Operatoren läßt sich die Übergangsratenmatrix des Gesamtmodells für das Beispiel aus Abbildung 4.1 wie folgt berechnen:

$$\begin{aligned} \mathbf{R} &= \underbrace{\mu \cdot (\mathbf{W}_A^{\text{sync}} \otimes \mathbf{W}_B^{\text{sync}})}_{\text{Synchronisationsereignis}} + \underbrace{(\mathbf{R}_A^{\text{lokal}} \oplus \mathbf{R}_B^{\text{lokal}})}_{\text{lokale Ereignisse}} \\ &= \mu \cdot \left(\left(\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \right) + \begin{pmatrix} 0 & \lambda_1 \\ 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 \\ \lambda_3 & 0 \end{pmatrix} \right) \end{aligned}$$

$$\begin{aligned}
&= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \lambda_1 & 0 \\ \lambda_3 & 0 & 0 & \lambda_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & \lambda_1 & 0 \\ \lambda_3 & 0 & 0 & \lambda_1 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \end{pmatrix}
\end{aligned}$$

Zur Berechnung der Generatormatrix des Gesamtmodells wird zusätzlich eine *Diagonalmatrix* \mathbf{D} von der Übergangsratenmatrix des Gesamtmodells subtrahiert. Die Einträge in der Diagonalen der Diagonalmatrix entsprechen der Summe der jeweiligen Zeileneinträge der Übergangsratenmatrix.

4.1.2 Allgemeine Anwendung der Kronecker-Operatoren

Im vorhergehenden Abschnitt wurden die Kronecker-Operatoren anhand eines sehr einfachen Beispiels eingeführt. In diesem Beispiel gab es nur zwei Teilmodelle, die sich über genau eine Transition synchronisierten. In diesem Abschnitt wird die Anwendung der Kronecker-Operatoren auf allgemeinere Modelle beschrieben. Mehrere Teilmodelle synchronisieren sich darin über mehrere Transitionen. Für die Anwendung des Kronecker-Produktes (der Kronecker-Summe) auf n Matrizen $\mathbf{M}_1 \dots \mathbf{M}_n$ schreiben wir $\bigotimes_{i=1}^n \mathbf{M}_i$ ($\bigoplus_{i=1}^n \mathbf{M}_i$). Das Kronecker-Produkt läßt sich leicht durch sukzessive Anwendung auf n Matrizen erweitern, da das Assoziativgesetz gilt [44]. Die allgemeine Anwendung der Kronecker-Summe ist wie folgt definiert:

$$\bigoplus_{i=1}^n \mathbf{M}_i = \sum_{i=1}^n (\mathbf{E}_1 \otimes \dots \otimes \mathbf{E}_{i-1} \otimes \mathbf{M}_i \otimes \mathbf{E}_{i+1} \otimes \dots \otimes \mathbf{E}_n)$$

\mathbf{E}_i steht für die Einheitsmatrix von demselben Typ wie die quadratische Matrix \mathbf{M}_i für alle $i = \{1, \dots, n\}$.

Sei $T^{\text{sync}} \subset T$ die Menge der Synchronisationstransitionen, durch die ein Petri-netz $\mathbf{N} = (P, T, F, K, m_0)$ in n Teilmodelle geteilt wird. Dann berechnet sich die Übergangsratenmatrix \mathbf{R}_N und die Generatormatrix \mathbf{Q}_N des Gesamtmodells wie folgt:

$$\mathbf{R}_N = \bigoplus_{i=1}^n \mathbf{R}_i^{\text{local}} + \sum_{t \in T^{\text{sync}}} (\text{rate}(t) \cdot \bigotimes_{i=1}^n \mathbf{W}_i^{\text{sync}})$$

$$\mathbf{Q}_N = \left(\bigoplus_{i=1}^n \mathbf{R}_i^{\text{local}} + \left(\sum_{t \in T^{\text{sync}}} (\text{rate}(t) \cdot \bigotimes_{i=1}^n \mathbf{W}_i^{\text{sync}}) \right) \right) - \mathbf{D}_N \quad \text{oder}$$

$$= \bigoplus_{i=1}^n \mathbf{Q}_i^{\text{local}} + \left(\sum_{t \in T^{\text{sync}}} (\text{rate}(t) \cdot \bigotimes_{i=1}^n \mathbf{W}_i^{\text{sync}}) - \mathbf{D}_N^{\text{sync}} \right)$$

Dabei gibt $\text{rate}(t)$ die Rate der Transition t an und \mathbf{D}_N ($\mathbf{D}_N^{\text{sync}}$) ist die \mathbf{R}_N ($\mathbf{Q}_N^{\text{sync}}$) zugeordnete Diagonalmatrix (siehe Abschnitt 4.1.1).

4.1.3 Zugriff auf einzelne Elemente der Generatormatrix des Gesamtsystems

Neben der Beschreibung einer geeigneten Zerlegung des Gesamtmodells ist der Zugriff auf Elemente oder Spalten der Untermatrizen das in der Literatur am wenigsten dokumentierte Detail der Kronecker-Algebra basierten Analyse. Wie in der Einführung beschrieben, liegt der Vorteil der Kronecker-Algebra basierten Analyse darin, die Generatormatrix des Gesamtsystems nicht flach abspeichern zu müssen. Vielmehr wird das Gleichungssystem zur Berechnung der Zustandswahrscheinlichkeiten erstellt, indem die Elemente der Generatormatrix des Gesamtsystems aus den Teilmatrizen berechnet werden. Dieser Abschnitt beschreibt die Berechnung der einzelnen Elemente der Generatormatrix des Gesamtsystems. Aus diesen Berechnungsvorschriften lassen sich leicht die Verfahren zur Berechnung von eine Gleichung des Gleichungssystems beschreibenden Spalten der gesamten Generatormatrix selektieren.

Für eine quadratische Matrix $\mathbf{M} = \bigotimes_{i=1}^n \mathbf{M}_i$ von der Dimension d beschreibe $\mathbf{M}(z, s)$ mit $z, s = \{0, \dots, d-1\}$ das Element aus Zeile z und Spalte s . Den Typ der quadratischen Matrizen \mathbf{M}_i notieren wir mit d_i . Seien außerdem für zwei natürliche Zahlen x, y die Operatoren $\text{ganz}(x, y)$ und $\text{rest}(x, y)$ mit der Semantik $\text{ganz}(x, y) \cdot y + \text{rest}(x, y) = x$ mit $\text{rest}(x, y) = x \bmod y$ definiert.

$$\text{Für } \mathbf{M} = \bigotimes_{i=1}^n \mathbf{M}_i \text{ ist } \mathbf{M}(z, s) = \prod_{i=1}^n \mathbf{M}_i(x_i, y_i)$$

$$\text{mit } \mathbf{M}_i(x_i, y_i) = \begin{cases} \mathbf{M}_n(\text{rest}(z, d_n), \text{rest}(s, d_n)) & , i = n \\ \mathbf{M}_i(\text{rest}(\text{temp}x_{i+1}, d_i), \text{rest}(\text{temp}y_{i+1}, d_i)) & , \text{sonst,} \end{cases}$$

$$\text{wobei } \text{temp}x_i = \begin{cases} \text{ganz}(z, d_n) & , i = n \\ \text{ganz}(\text{temp}x_{i+1}, d_i) & , \text{sonst.} \end{cases}$$

$$\text{und } \text{temp}y_i = \begin{cases} \text{ganz}(s, d_n) & , i = n \\ \text{ganz}(\text{temp}y_{i+1}, d_i) & , \text{sonst.} \end{cases}$$

Betrachten wir als Beispiel die Kronecker-Multiplikation der Matrizen \mathbf{A} und \mathbf{B} aus dem Einführungsabschnitt auf Seite 47. Das Element¹ $\mathbf{A} \otimes \mathbf{B}(3, 1) = a_{10} \cdot b_{01}$ soll aus den Teilmatrizen \mathbf{A} und \mathbf{B} berechnet werden. Wegen $z = 3, s = 1$ und da \mathbf{B} vom Typ $d_{\mathbf{B}} = 3$ ist, berechnet sich der Eintrag für \mathbf{B} aus $\mathbf{B}(\text{rest}(3, 3), \text{rest}(1, 3)) = \mathbf{B}(0, 1)$. Zur Berechnung des zugehörigen Eintrages der quadratischen 2-Matrix \mathbf{A} müssen wegen $\mathbf{A}(\text{rest}(\text{temp}x_{\mathbf{B}}, d_{\mathbf{A}}), \text{rest}(\text{temp}y_{\mathbf{B}}, d_{\mathbf{A}}))$ die Werte der Variablen $\text{temp}x_{\mathbf{B}} = \text{ganz}(3, 3) = 1$ und $\text{temp}y_{\mathbf{B}} = \text{ganz}(1, 3) = 0$ berechnet werden. Daraus ergibt sich $\mathbf{A}(\text{rest}(1, 2), \text{rest}(0, 2)) = \mathbf{A}(1, 0)$. Demnach ist der Eintrag der Gesamtmatrix $\mathbf{A} \otimes \mathbf{B}$ in der Zeile 3 und Spalte 1 gleich $\mathbf{A}(1, 0) \cdot \mathbf{B}(0, 1)$.

Für eine aus der Kronecker-Addition mehrerer Matrizen entstandene Matrix ist die Berechnung der einzelnen Elemente komplexer als für den Fall der Kronecker-Multiplikation, da wegen der Möglichkeit des Null-Eintrages drei statt zwei Fälle zu unterscheiden sind.

$$\text{Für } \mathbf{M} = \bigoplus_{i=1}^n \mathbf{M}_i \text{ ist } \mathbf{M}(z, s) = \sum_{i=n}^1 \mathbf{M}_i(x_i, y_i)$$

$$\text{mit } \mathbf{M}_i(x_i, y_i) = \begin{cases} \mathbf{M}_n(\text{rest}(z, d_n), \text{rest}(s, d_n)) & , \textcircled{1} \\ \mathbf{M}_i(\text{rest}(\text{temp}x_{i+1}, d_n), \text{rest}(\text{temp}y_{i+1}, d_n)) & , \textcircled{2} \\ 0 & , \textcircled{3} \end{cases}$$

- ① $i = n$ und $\text{temp}x_n = \text{temp}y_n$
- ② $i \neq n$ und $\text{temp}x_{i+1} = \text{temp}y_{i+1}$ und es gibt kein $k > i$ mit $\mathbf{M}_k(x_k, y_k) : x_k \neq y_k$
- ③ sonst

$$\text{mit } \text{temp}x_i = \begin{cases} \text{ganz}(z, d_n) & , i = n \\ \text{ganz}(\text{temp}x_{i+1}, d_i) & , \text{sonst.} \end{cases}$$

$$\text{und } \text{temp}y_i = \begin{cases} \text{ganz}(s, d_n) & , i = n \\ \text{ganz}(\text{temp}y_{i+1}, d_i) & , \text{sonst.} \end{cases}$$

Betrachten wir als Beispiel die Kronecker-Addition der Matrizen \mathbf{A} und \mathbf{B} aus dem Einführungsabschnitt auf Seite 48. Das Element $\mathbf{A} \oplus \mathbf{B}(2, 1) = b_{21}$ soll aus den Teilmatrizen \mathbf{A} und \mathbf{B} berechnet werden. Wegen $z = 2, s = 1$ und da \mathbf{B} vom Typ $d_{\mathbf{B}} = 3$ ist, berechnet sich der Eintrag für \mathbf{B} aus $\mathbf{B}(\text{rest}(2, 3), \text{rest}(1, 3)) = \mathbf{B}(2, 1)$. Für jede nachfolgende Berechnung gilt nun jedoch immer die Negation der Teilbedingung aus ②, denn es gibt im Weiteren stets ein $k > i$ mit $\mathbf{M}_k(x_k, y_k) : x_k \neq y_k$. Somit kann die Berechnung des Eintrages für die Gesamtmatrix an dieser Stelle abgebrochen werden, da es keinen Eintrag aus einer anderen Teilmatrix

¹Man beachte die Zählung der Zeilen und Spalten beginnend mit Null.

gibt, der zu dem bisher berechneten addiert werden muß. Der Grund für diese Bedingung liegt darin, daß es außer auf der Hauptdiagonalen keine Summe von Einträgen verschiedener Teilmatrizen gibt. Jeder Eintrag außerhalb der Hauptdiagonalen der Gesamtmatrix ist entweder Null oder aber genau ein Eintrag aus einer Teilmatrix, welcher in dieser Teilmatrix ebenfalls außerhalb der Hauptdiagonalen liegt. Wir betrachten noch ein weiteres Beispiel aus der Kronecker-Addition der Matrizen \mathbf{A} und \mathbf{B} aus dem Einführungsabschnitt auf Seite 48. Das Element $\mathbf{A} \oplus \mathbf{B}(0, 3) = a_{01}$ wird demnach wie folgt berechnet: Für den Eintrag der \mathbf{B} -Matrix ergibt sich wegen ① Null (und nicht $\mathbf{B}(\text{rest}(0, 3), \text{rest}(3, 3)) = \mathbf{B}(0, 0)!$), da $\text{temp}x_n = \text{ganz}(0, 3) \neq \text{temp}y_n = \text{ganz}(3, 3)$. Mit dieser Bedingung der Gleichheit der temp-Variablen wird getestet, ob der Eintrag überhaupt innerhalb eines Blockes von Einträgen der entsprechenden Teilmatrix liegt. Betrachten wir dazu nochmals das Beispiel der Anwendung der Kronecker-Addition auf die Matrizen \mathbf{A} und \mathbf{B} auf Seite 48. Die Einträge der Matrix \mathbf{B} befinden sich in $d_A = 2$ Blöcken entlang der Hauptdiagonalen. Allgemeiner befinden sich alle Einträge der Matrix \mathbf{M}_i in $\prod_{j=1}^{i-1} d_j$ Blöcken entlang der Hauptdiagonalen der Gesamtmatrix. Nicht jeder Eintrag in diesen Blöcken ist jedoch ein Eintrag der Teilmatrix \mathbf{M}_i . Es gibt ebenso Null-Einträge entsprechend der Kronecker-Multiplikation mit den Einheitsmatrizen $\mathbf{E}_{i+1} \otimes \dots \otimes \mathbf{E}_n$. Während mit der Gleichheit der temp-Variablen die Eintragung innerhalb eines Blockes überprüft wird, identifiziert die Berechnung mittels der rest-Variablen die Einträge der Teilmatrizen in diesen Blöcken. Der Eintrag der \mathbf{A} -Matrix wird aus $\mathbf{A}(\text{rest}(\text{ganz}(0, 3), 2), \text{rest}(\text{ganz}(3, 3), 2)) = \mathbf{A}(0, 1)$ berechnet.

4.1.4 Falsche Zustände und weitere Nachteile

Die Kronecker-Algebra basierte Analyse zählt zu den matrixreduzierenden Verfahren, nicht jedoch zu den zustandsraumreduzierenden! In dem in Abschnitt 4.1.1 zur Einführung der Kronecker-Operatoren verwendeten Beispiel hätte es tatsächlich genügt, die Erreichbarkeitsgraphen der Teilmodelle zu erstellen, um daraus die einzelnen Generator- und Übergangswahrscheinlichkeitsmatrizen zu entwickeln. Im Allgemeinen ist jedoch die Erstellung des Erreichbarkeitsgraphen für das Gesamtmodell erforderlich. Der Grund dafür liegt in der Existenz sogenannter *falscher Zustände*, die bei der Erstellung der einzelnen Erreichbarkeitsgraphen für die Teilmodelle erzeugt werden können. Die Erkennung falscher Zustände ist nur durch den Vergleich mit dem Erreichbarkeitsgraphen des Gesamtmodells möglich. Durch die eventuelle Erzeugung falscher Zustände sind die Möglichkeiten der Zerlegung des Gesamtmodells sehr beschränkt. Wird das Gesamtmodell ungünstig geteilt, entstehen unter Umständen so viele falsche Zustände, daß die Teilmatrizen größer sind als die Generatormatrix des Gesamtmodells [18]. In [86] wird eine strukturbasierte Erreichbarkeitsanalyse für *superposed*

GSPNs [84, 53, 114] vorgeschlagen. Im Rahmen dieser Arbeit entstand eine auf der doppelten Beschreibungsmethode basierende Erreichbarkeitsuntersuchung [75]. Dabei wird für einen gegebenen Zustand untersucht, ob er *nicht* erreichbar ist. Eine eventuelle Erreichbarkeit ist dabei nicht sicher berechenbar (siehe Abschnitt 3.1.3). Für den Einsatz des Verfahrens bei der Kronecker-Algebra basierten Analyse ist dies kein Nachteil, da gerade die Frage nach der Nichterreichbarkeit von Bedeutung ist. Das entwickelte Verfahren ist jedoch aus Zeitgründen kaum praktikabel, da jeder Zustand untersucht werden muß.

Es wäre wünschenswert, strukturelle Eigenschaften des Gesamtmodells zu finden, die einer hinreichenden Bedingung zur Vermeidung der Erzeugung falscher Zustände entsprechen. Für von Stellen-Invarianten überdeckte Modelle ist eine im folgenden an einem Beispiel begründete Vermutung für eine notwendige Bedingung zur Vermeidung falscher Zustände, daß bei der Zerlegung des Gesamtmodells minimale Stellen-Invarianten nicht „zerschnitten“ werden dürfen. Diese Forderung läßt sich im Allgemeinen nicht erfüllen. Das Beispiel aus Abbildung 3.2 läßt sich beispielsweise wie in Abbildung 4.6 dargestellt zerlegen.

In dieser Zerlegung wurden zwei minimale Stellen-Invarianten des Gesamtmodells zerschnitten: $P_1+P_2+P_3+P_4+P_5$ und $P_6+P_7+P_8+P_9+P_{10}$. Die Stelle P_5 ist wegen der mit der initialen Markierung gewichteten Stellen-Invariante $P_1+P_2+P_3+P_4+P_5=1$ im Gesamtmodell 1-beschränkt. Erstellt man jedoch den Erreichbarkeitsgraphen für das Teilmodell D , so kann die Stelle P_5 bis zu acht Marken enthalten. Dasselbe gilt für die Stelle P_6 (siehe alle Stellen-Invarianten des Beispiels in Abschnitt 3.1.4). Die Beschränkung dieser Stellen ist im Teilmodell D nicht sichtbar. Dadurch wird für das Teilmodell D ein Erreichbarkeitsgraph mit 165 Zuständen erstellt. Durch die Beschränkung der Stellen P_5 und P_6 ist der „richtige“ Erreichbarkeitsgraph jedoch nur 32 Zustände groß. Bereits für dieses kleine Beispiel werden demnach 133 falsche Zustände berechnet. Der dem Gesamtmodell zugrunde liegende Erreichbarkeitsgraph besteht aus 183 Zuständen!

Darüber hinaus ist weder eine automatische Zerlegung des Gesamtmodells möglich, noch läßt sich eine nutzerfreundliche Interaktion implementieren, wie es beispielsweise für die Zerlegung unstrukturierter Netze bei der im folgenden Kapitel 5 vorgestellten MIMO-Graph basierten Aggregation möglich ist. Stattdessen muß jedes Teilmodell vollständig und wegen möglicher Überschneidungen sukzessive durch den Nutzer angegeben werden.

4.1.5 Integration der Kronecker-Algebra basierten Analyse in TimeNET

Die Kronecker-Algebra basierte Analyse für farblose Petrinetze wurde innerhalb dieser Arbeit in TimeNET integriert. Zur Lösung des Gleichungssystems [124] wurde der bei der Anwendung der Kronecker-Operatoren am effizientesten [19,

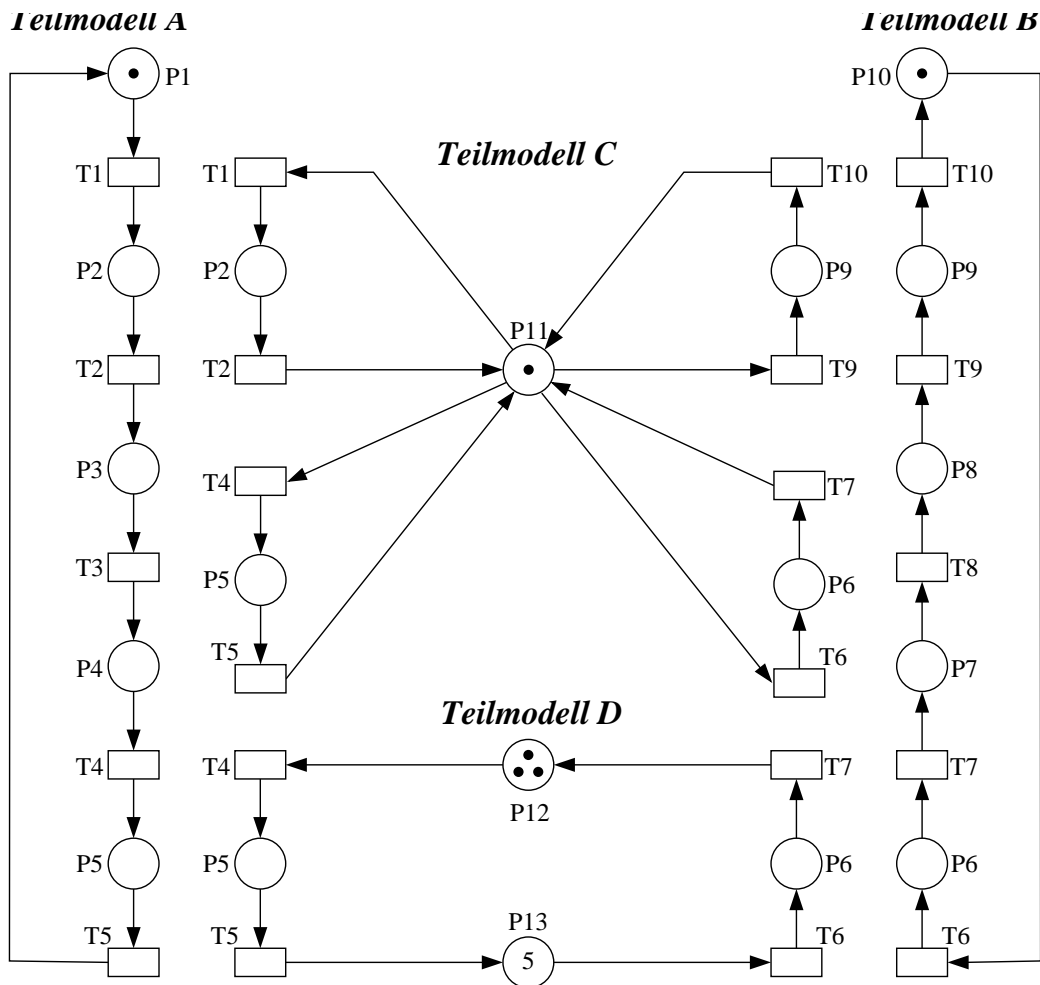


Abbildung 4.6: Mögliche Zerlegung für die Kronecker-Algebra basierte Analyse

29, 46] arbeitende Gauß-Seidel Algorithmus verwendet. Der Vektor der Zustandswahrscheinlichkeiten π wird darin doppelt iterativ berechnet. Eine innere Iteration führt zur Berechnung von π_i in K Schritten und eine äußere Iteration berechnet alle π_i mit $1 \leq i \leq N$ für die $N \times N$ -Matrix \mathbf{Q} wie folgt:

$$\pi_i^{(k+1)} \cdot \mathbf{Q}_{ii} = - \left(\sum_{j=1}^{i-1} (\pi_j^{(k+1)} \cdot \mathbf{Q}_{ji}) + \sum_{j=i+1}^N (\pi_j^k \cdot \mathbf{Q}_{ji}) \right)$$

Für die Kronecker-Algebra basierte Analyse farbiger Petrinetze wird in TimeNET derzeit noch auf die innerhalb dieser Arbeit implementierte Entfaltung farbiger in farblose Petrinetze zurückgegriffen. Wegen der in Abschnitt 4.1.4 erläuterten Restriktionen der Kronecker-Algebra basierten Analyse lassen sich nicht beliebig große, sondern vielmehr um ca. eine Größenordnung (für wenige spezielle Modelle auch größer [45]) größere Modelle als mit Standardanalysemetho-

den auswerten. In den folgenden Abschnitten wird eine Methode vorgestellt, die als Ergänzung zur Kronecker-Algebra basierten Analyse eine Auswertung noch größerer Modelle ermöglicht.

4.2 Überblick über weitere matrixreduzierende Verfahren

In diesem Abschnitt werden weitere matrixreduzierende Verfahren vorgestellt und diskutiert.

In [46] wird ein Verfahren vorgestellt, in dem ebenfalls nicht die gesamte Generatormatrix gespeichert wird. Stattdessen wird nur der im jeweiligen Iterationsschritt des Gauß-Seidel Lösungsverfahrens verwendete Zeilenvektor der transponierten Generatormatrix berechnet und gespeichert. Der Zeilenvektor wird durch Schalten von Transitionen im untersuchten Petrinetz ermittelt. Die positiven Einträge der i -ten Zeile der transponierten Generatormatrix repräsentieren die Raten der Übergänge, die zu dem Zustand z_i führen. Um diese Raten zu ermitteln, ist es deshalb bei gegebenem Zustand z_i nötig, das zu untersuchende Petrinetz zuerst rückwärts zu schalten, um alle Vorzustände des Zustandes z_i zu ermitteln. Erst nach Kenntnis aller Vorzustände von z_i läßt sich die entsprechende Zeile der Generatormatrix aufstellen. Der Nachteil dieser Methode liegt neben dem erhöhten Zeitaufwand darin, daß auch das jeweilige Diagonalelement der Zeile berechnet werden muß. Dieser Eintrag entspricht der negativen Summe der *ausgehenden* Raten des Zustandes z_i . Verwendet man einen zusätzlichen Vektor zur Speicherung der Diagonalelemente, wird das Zurückschalten des Petrinetzes vermieden. Jedoch ist die Größe des Vektors durch die Anzahl der andauernden Zustände bestimmt und benötigt entsprechend Speicherplatz.

Eine Methode, die gesamte Generatormatrix nur teilweise im Hauptspeicher und größtenteils auf einer Festplatte zu speichern, findet sich in [45]. Wiederum unter Verwendung des Gauß-Seidel Lösungsverfahrens wird die Generatormatrix in verschiedene Blöcke aufgeteilt, so daß eine teilweise parallele Berechnung der einzelnen Iterationsschritte ermöglicht wird.

4.2.1 Multi-Terminal- und Matrixdiagramme

Eine sehr junge Entwicklung auf dem Gebiet matrixreduzierender Verfahren zur Analyse von Petrinetzen ist die Verwendung von Multi-Terminal- und Matrixdiagrammen zur effizienten Speicherung der Übergangsratenmatrix. Diese gehen auf binäre Entscheidungsdiagramme (*BDDs*) [13] zurück. *BDDs* spielen im Bereich des *model checking* [37, 36] eine wichtige Rolle, da sich mit ihnen unter Ausnutzung von Symmetrien aussagen- und temporallogische Eigenschaf-

ten von Systemen mit über 10^{20} Zuständen [21, 22, 98, 113] beweisen lassen. Ein binäres Entscheidungsdiagramm ist ein gerichteter azyklischer Graph, der eine boolesche Funktion $f : \{0, 1\}^n \rightarrow \{0, 1\}$ repräsentiert. Die Knoten des BDDs, die keine Ausgangskanten besitzen, heißen Terminalknoten, alle anderen sind Nichtterminalknoten. In der Nutzung am effizientesten sind BDDs mit einer totalen Ordnung der Variablen und reduziert derart, daß es keine redundanten Knoten und Kanten gibt [14].

Beispielanwendung von Multi-Terminal BDDs

Eine Erweiterung der BDDs sind *Multi-Terminal* [6, 60] BDDs, die boolesche Funktionen $f : \{0, 1\}^n \rightarrow \mathbb{R}$ repräsentieren. Diese werden in [74] verwendet, um die Übergangsratenmatrix einer Markowkette effizient im Sinne sowohl des Zugriffes auf ein Element als auch der Speicherung der Gesamtmatrix zu repräsentieren. Während der oberste Nichtterminalknoten die Zeilen der betrachteten Matrix halbiert, teilen die direkt folgenden Nichtterminalknoten die Spalten der Matrix usw. im Wechsel, bis jeder Eintrag der Matrix genau einem Pfad durch das BDD zugeordnet ist. Die Effizienz dieser Methode hängt von der Anzahl unterschiedlicher Einträge in der Übergangsratenmatrix ab. Im schlechtesten Fall sind alle positiven Einträge der Matrix gleich und die Effizienz entspricht der eines Algorithmus zur Speicherung dünnbesetzter Matrizen. Abbildung 4.7 zeigt schematisch die Anwendung von Multi-Terminal BDDs zur Abspeicherung der auf Seite 49 dargestellten Übergangsratenmatrix des Beispiels aus Abbildung 4.1 auf Seite 46.

In Abbildung 4.7 sind die Nichtterminalknoten zum besseren Verständnis als die sie repräsentierenden Teilmatrizen dargestellt. Die Terminalknoten sind die einzelnen Einträge der Matrix. In dem dargestellten Multi-Terminal BDD gibt es noch redundante Knoten und Kanten. Sowohl redundante Terminal- als auch Nichtterminalknoten sind enthalten und werden durch eine Reduzierung des BDDs entfernt.

Beispielanwendung von Matrixdiagrammen

Die jüngste Anwendung von Matrixdiagrammen [99, 33] kann als ein entscheidender Fortschritt in der effizienten Speicherung und Verwaltung von Übergangsraten- oder Generatormatrizen bezeichnet werden. Matrixdiagramme sind ebenfalls gerichtete azyklische Graphen mit Nichtterminal- und Terminalknoten. In Matrixdiagrammen sind die Nichtterminalknoten jedoch Matrizen von Paaren bestehend aus einer reellen Zahl und einem Verweis auf andere Nichtterminal- oder Terminalknoten. Matrixdiagramme werden sowohl zur Speicherung von Erreichbarkeitsgraphen (siehe dazu Abschnitt 5.6) als auch zur Speicherung von Übergangsraten- und Generatormatrizen verwendet. In letzterem Fall hängt die

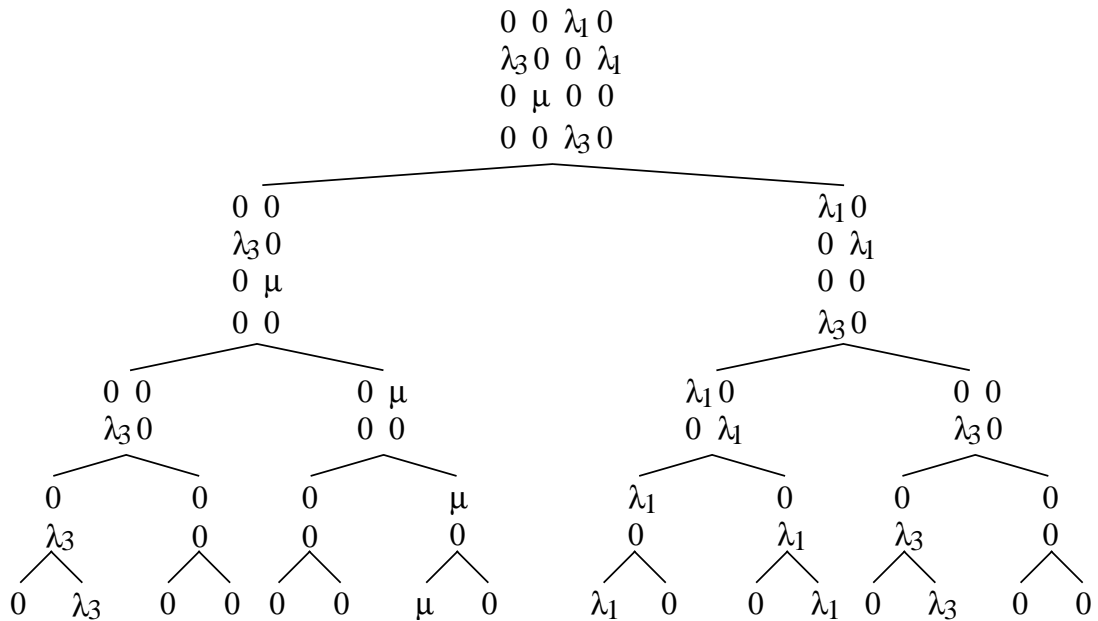


Abbildung 4.7: Beispiel Anwendung von Multi-Terminal BDDs

Effizienz der Methode stark von der Anzahl verschiedener Blöcke oder Einträge in den zu untersuchenden Matrizen ab. Diese Abhängigkeit soll durch ein kleines Beispiel verdeutlicht werden. Das Beispielmodell und der zugehörige Erreichbarkeitsgraph sind in Abbildung 4.8 dargestellt. Den Transitionen T1 und T3 sind entsprechend die Schaltraten λ_1 und λ_3 und der Transition T2 die Schaltrate μ zugeordnet.

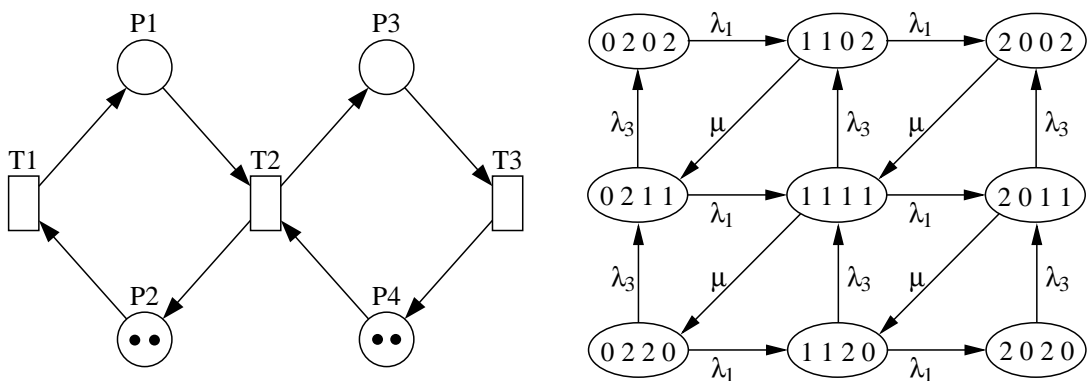


Abbildung 4.8: Beispiel – Modell und Erreichbarkeitsgraph

Die Inschriften der Zustände im Erreichbarkeitsgraph stehen für die Anzahl der Marken in den Stellen P1, P2, P3 und P4 in genannter Reihenfolge. Die Übergangsratenmatrix dieses Beispiels ist in Abbildung 4.9 dargestellt. Die Zeilen-

$$\begin{array}{c|ccc|ccc|ccc}
& \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
& \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} \\
& \mathbf{2} & \mathbf{1} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0} & \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\mathbf{2020} & 0 & \lambda_3 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{2011} & 0 & 0 & \lambda_3 & 0 & \lambda_1 & 0 & 0 & 0 & 0 \\
\mathbf{2002} & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\
\hline
\mathbf{1120} & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_1 & 0 & 0 \\
\mathbf{1111} & \mu & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \lambda_1 & 0 \\
\mathbf{1102} & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 \\
\hline
\mathbf{0220} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 \\
\mathbf{0211} & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & \lambda_3 \\
\mathbf{0202} & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & 0
\end{array} = \begin{array}{c|ccc}
& \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\mathbf{20} & \textcircled{1} & \textcircled{2} & \textcircled{3} \\
\mathbf{11} & \textcircled{4} & \textcircled{5} & \textcircled{6} \\
\mathbf{02} & \textcircled{7} & \textcircled{8} & \textcircled{9}
\end{array}$$

$$\begin{array}{c|ccc}
& \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\textcircled{1}, \textcircled{5}, \textcircled{9} = \mathbf{20} & 0 & \lambda_3 & 0 \\
& \mathbf{11} & 0 & 0 & \lambda_3 \\
& \mathbf{02} & 0 & 0 & 0
\end{array} \quad \begin{array}{c|ccc}
& \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\textcircled{2}, \textcircled{6} = \mathbf{20} & \lambda_1 & 0 & 0 \\
& \mathbf{11} & 0 & \lambda_1 & 0 \\
& \mathbf{02} & 0 & 0 & \lambda_1
\end{array}$$

$$\begin{array}{c|ccc}
& \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\textcircled{3}, \textcircled{7} = \mathbf{20} & 0 & 0 & 0 \\
& \mathbf{11} & 0 & 0 & 0 \\
& \mathbf{02} & 0 & 0 & 0
\end{array} \quad \begin{array}{c|ccc}
& \mathbf{2} & \mathbf{1} & \mathbf{0} \\
& \mathbf{0} & \mathbf{1} & \mathbf{2} \\
\hline
\textcircled{4}, \textcircled{8} = \mathbf{20} & 0 & 0 & 0 \\
& \mathbf{11} & \mu & 0 & 0 \\
& \mathbf{02} & 0 & \mu & 0
\end{array}$$

Abbildung 4.9: Verwaltung der Übergangsratenmatrix des Beispiels aus Abb. 4.8

und Spaltenbeschriftungen entsprechen den Zustandsinschriften des Erreichbarkeitsgraphen aus Abbildung 4.8.

Die jeweiligen Anzahlen der Marken auf den Stellen P1 und P2 sind fettgedruckt. Die Übergangsratenmatrix des Gesamtmodells lässt sich in kompakterer Form darstellen, wenn die Zeilen und Spalten nur noch mit den Anzahlen der Marken auf den Stellen P1 und P2 beschriftet sind. Diese Matrix enthält nur noch Verweise auf Untermatrizen. Diese Verweise sind in Abbildung 4.9 als umkreiste Zahlen dargestellt. Die Gleichheit von Untermatrizen ermöglicht das Zeigen mehrerer Verweise auf dieselbe Untermatrix (bspw. ③ und ⑦ zeigen gemeinsam auf die Nullmatrix). Dadurch bedarf es geringeren Speicherplatzes zur Verwaltung der Übergangsratenmatrix des Gesamtmodells.

Während die in [33] vorgestellte Methode die Kronecker-Repräsentation eines Petrinetzes ausnutzt, ist das Verfahren in [99] für beliebige Repräsentationen verall-

gemeinert. Daraus folgt, daß sich das zu untersuchende Petrinetz beliebig partitionieren läßt. Die Anwendung von Matrixdiagrammen erlaubt numerische Analysen von Petrinetzen mit mehr als 133 Millionen Zuständen [99]. Es ist zu erwarten, daß sich die Anwendung von Matrixdiagrammen zu einem Standard in der numerischen Analyse farbloser stochastischer Petrinetze entwickelt.

Kapitel 5

Zustandsraumreduzierende Verfahren

Die im vorangegangenen Kapitel 4 beschriebenen matrixreduzierenden Verfahren finden ihre Anwendung bei der numerischen Analyse großer stochastischer Petrinetze. Von noch größerer Bedeutung als matrixreduzierende Verfahren sind jedoch zustandsraumreduzierende. Das liegt einerseits daran, daß die Dimension der bei der numerischen Analyse erstellten Matrix der Anzahl der erreichbaren andauernden Zustände des analysierten stochastischen Petrinetzes entspricht. Die Erstellung der Matrix baut auf der Erstellung des Zustandsraumes auf. Andererseits spielt die Größe des Zustandsraumes nicht nur bei der numerischen Analyse eine restriktive Rolle. Viele der in Kapitel 3 beschriebenen strukturellen Eigenschaften, wie Lebendigkeit, Erreichbarkeit oder Beschränktheit basieren auf der Untersuchung des gesamten Zustandsraumes eines Petrinetzes. Daher finden zustandsraumreduzierende Verfahren ein breiteres Anwendungsgebiet als matrixreduzierende Verfahren.

Die Suche nach zustandsraumreduzierenden Verfahren ist so alt wie die zustandsraumbasierten Verfahren selbst. Eine der grundlegenden Ideen bei der Suche nach zustandsraumreduzierenden Verfahren ist es, das wegen seiner Größe nicht analysierbare Petrinetz in mehrere analysierbare Teilmodelle zu zerlegen, diese isoliert voneinander zu analysieren und von den Eigenschaften der Teilmodelle auf die des Gesamtmodells zu schließen. Jedoch ist dieses Verfahren für keine der genannten zu untersuchenden Eigenschaften praktikabel. Weder kann man aus der Lebendigkeit aller Teilmodelle auf die Lebendigkeit des Gesamtmodells schließen, noch bedeutet die Erreichbarkeit einer Menge von lokalen Zuständen die Erreichbarkeit des aus der Vereinigung dieser lokalen Zustände gebildeten globalen Zustandes des Gesamtmodells. Im Abschnitt 4.1.4 wurde bereits über die Problematik bei der Erstellung der Erreichbarkeitsgraphen von

Teilmodellen und des Schließens auf die Beschränktheitseigenschaft des Gesamtmodells diskutiert. Für die numerische Analyse ist das beschriebene Verfahren nur sehr eingeschränkt anwendbar. Je größer die Interaktion zwischen den Teilmodellen im Gesamtmodell ist, desto weniger praktikabel ist die isolierte Analyse der Teilmodelle. Ein zusätzlicher Aufwand ist nötig, um die Interaktion der Teilmodelle in die numerische Berechnung einfließen zu lassen.

Eine weniger als die Zerlegung des Gesamtmodells in Teilmodelle betrachtete Methode zur Reduzierung des Zustandsraumes ist die *Aggregation* des Gesamtmodells. Unter Aggregation wird dabei der Prozeß verstanden, ein großes, komplexes Modell in ein kleineres, weniger komplexes Modell zu transformieren. Das Problem dieser Transformation liegt darin, daß sie die zu untersuchenden Eigenschaften bewahren muß, da aus den Eigenschaften des aggregierten Modells auf die Eigenschaften des Originalmodells geschlossen werden soll. Die in diesem Kapitel vorgestellte *MIMO-Graphen basierte Aggregation* erfüllt diese Voraussetzungen für wichtige strukturelle Eigenschaften.

Ein bisher kaum betrachtetes Problem ist die geeignete Aggregation stochastischer Petrinetze. Bisher sind keine allgemeinen Aggregierungsmethoden für stochastische Petrinetze bekannt, die bspw. gewährleisten, daß der Durchsatz des aggregierten Modells dem des Originalmodells entspricht. Erstmals werden in diesem Kapitel für die MIMO-Graphen basierte Aggregation Formeln präsentiert, die eine Aggregation komplexer Strukturen derart ermöglichen, daß die aggregierten Modelle näherungsweise denselben Durchsatz aufweisen wie das Originalmodell. Für einige Petrinetzmodelle ist die Aggregation bezüglich des Durchsatzes sogar exakt. Die untersuchten stochastischen Petrinetze besitzen einschränkend nur zeitlose und zeitbehaftete Transitionen mit negativ exponentiell verteilter Schaltzeit.

5.1 MIMO-Graphen basierte Aggregation

In diesem Abschnitt wird eine neue Aggregierungsmethode vorgestellt. Dazu werden sogenannte *MIMO-Graphen* (MIMO für *multiple input/multiple output*) eingeführt. MIMO-Graphen sind gerichtete Graphen mit zwei voneinander disjunkten Knotenmengen, der Menge der *MIMO-P*- und der Menge der *MIMO-T-Elemente*.

Zunächst werden MIMO-Graphen zur Aggregation farbloser Petrinetze verwendet. In Abschnitt 5.3 wird die Methode zur Anwendung auf spezielle farbige Petrinetze erweitert.

Die generelle Idee der MIMO-Graphen basierten Aggregation besteht darin, ein Petrinetz in einen MIMO-Graphen zu transformieren. Dieser MIMO-Graph wird in einem zweiten Schritt aggregiert und in einem dritten Schritt wieder zurück in

ein jetzt aggregiertes Petrinetz transformiert.

5.1.1 Ein einführendes Beispiel

In diesem Abschnitt wird zunächst anhand des in Abbildung 3.2 auf Seite 31 dargestellten Beispiels die MIMO-Graphen basierte Aggregierung erläutert.

Bei der Transformation in einen MIMO-Graphen werden alle die Elemente übernommen, die mehrere Eingangs- und/oder Ausgangskanten besitzen und alle Stellen, die initial markiert sind.

Die Stellen P_1 , P_{10} , P_{11} , P_{12} und P_{13} sind initial markiert. Sie werden transformiert in gleich beschriftete MIMO-P-Elemente und in eckige Klammern wird die Anzahl der Marken auf diesen Stellen notiert. Die Transitionen T_1 , T_4 , T_6 und T_9 besitzen mehrere Eingangs-, während die Transitionen T_2 , T_5 , T_7 und T_{10} mehrere Ausgangskanten besitzen. Sie werden zu gleichbeschrifteten MIMO-T-Elementen transformiert. Alle anderen Elemente des Petrinetzmodells besitzen nur genau eine Eingangs- und eine Ausgangskante. Diese Elemente werden bei der Transformation in einen MIMO-Graphen nicht berücksichtigt. Die (transitiven) Kantenbeziehungen zwischen den Elementen bleiben im MIMO-Graphen erhalten. Abbildung 5.1 zeigt den aus dem Petrinetzmodell erzeugten MIMO-Graphen.

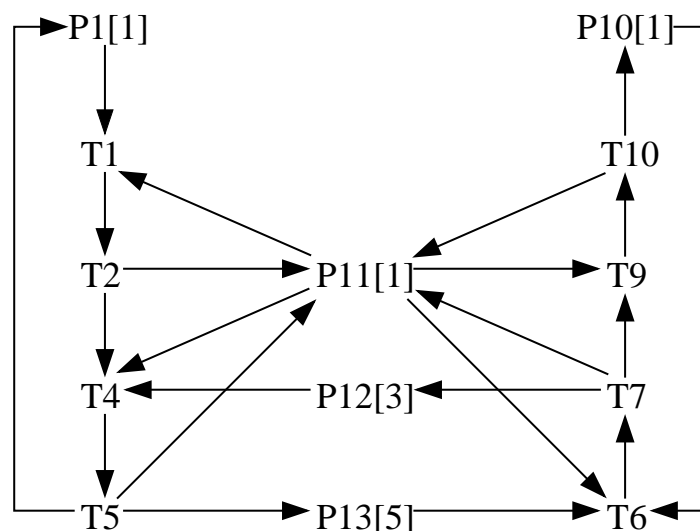


Abbildung 5.1: Ergebnis erster Schritt: MIMO-Graph

Die Transformation aus einem Petrinetz in einen MIMO-Graphen ist bereits der erste Schritt einer Reduktion des Modells, da alle Sequenzen innerhalb des Petrinetzes entfallen. In einem zweiten Schritt wird der MIMO-Graph aggregiert. Dazu werden im folgenden zwei generelle Aggregierungsregeln vorgestellt.

Die Idee der Aggregation des MIMO-Graphen besteht darin, benachbarte Elemente gleichen Typs unter bestimmten Bedingungen zu einem Element zu verschmelzen. Benachbart sind zwei Elemente, wenn sie durch eine Kante miteinander verbunden sind. Elemente des MIMO-Graphen sind entweder vom Typ MIMO-P oder vom Typ MIMO-T. Zwei benachbarte Elemente A und B gleichen Typs können mit Ausnahme folgender Einschränkung miteinander verschmolzen werden. Wenn die A und B verbindende Kante von A nach B verläuft, so dürfen A und B nur dann *nicht* miteinander verschmolzen werden, wenn A weitere Ausgangs- und B weitere Eingangskanten besitzen.

Im angegebenen Beispiel sind die MIMO-T-Elemente T1 und T2 verschmelzbar. Sie sind vom gleichen Typ und weder besitzt T1 weitere Ausgangs- noch T2 weitere Eingangskanten als die zum jeweils anderen Element. T2 und T4 sind nicht verschmelzbar, da sowohl T2 weitere Ausgangskanten (zu P11) als auch T4 weitere Eingangskanten (von P11 und P12) besitzen. Neben T1 und T2 können auch T4 und T5, T6 und T7 und auch T9 und T10 miteinander verschmolzen werden. Abbildung 5.2 zeigt den aggregierten MIMO-Graphen. Die aus Verschmelzung entstandenen MIMO-T-Elemente sind darin T_{m1}, \dots, T_{m4} benannt.

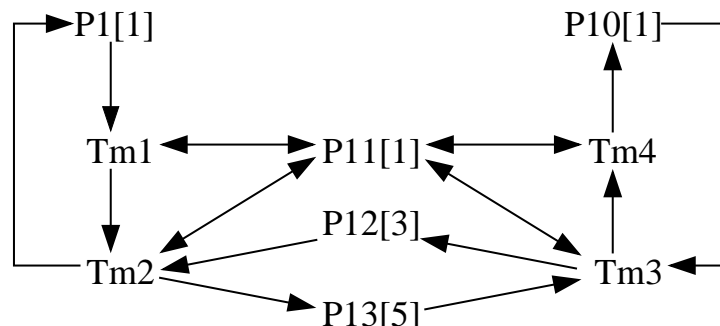


Abbildung 5.2: Ergebnis zweiter Schritt: aggregierter MIMO-Graph

Der dritte und letzte Schritt der MIMO-Graphen basierten Aggregation ist die Rücktransformation des aggregierten MIMO-Graphen in ein Petrinetz. Dabei werden MIMO-T-Elemente zu Transitionen und MIMO-P-Elemente zu Stellen rücktransformiert. Die Kantenbeziehungen bleiben erhalten. Zu beachten ist einzig, daß im aggregierten MIMO-Graphen noch zum Teil Elemente gleichen Typs benachbart sind. Zwischen diese Elemente muß ein Element des anderen Typs eingefügt werden. Im angegebenen Beispiel werden bei der Rücktransformation zwischen die Transitionen T_{m1} und T_{m2} eine Stelle P_{a1} und zwischen T_{m3} und T_{m4} eine Stelle P_{a2} eingefügt. Abbildung 5.3 zeigt das aggregierte Petrinetzmodell des Beispiels. Es ist ein Petrinetzmodell entstanden, dessen Zustandsraum deutlich kleiner als der des Originalmodells ist.

Nach dieser informellen Einführung in die MIMO-Graph basierte Aggregation

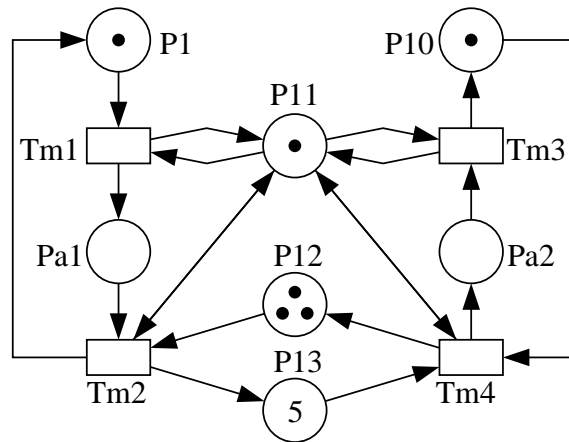


Abbildung 5.3: Ergebnis dritter Schritt: aggregiertes Petrinetzmodell

folgt nun die Definition des MIMO-Graphen und die Formalisierung der angegebenen Schritte.

5.1.2 Formelle Beschreibung farbloser MIMO-Graphen

Jedem Petrinetz läßt sich eindeutig ein MIMO-Graph zuordnen.

5.1 Definition (MIMO-Graph (für farblose Netze))

Ein MIMO-Graph wird durch ein 5-Tupel

$$G_{\text{MIMO}} = (T_{\text{MIMO}}, P_{\text{MIMO}}, F_{\text{MIMO}}, K_{\text{MIMO}}, m_{0\text{MIMO}})$$

beschrieben, wobei

- T_{MIMO} und P_{MIMO} endliche und disjunkte Mengen,
- $F_{\text{MIMO}} \subseteq (P_{\text{MIMO}} \cup T_{\text{MIMO}}) \times (T_{\text{MIMO}} \cup P_{\text{MIMO}})$,
- K_{MIMO} eine Abbildung $F_{\text{MIMO}} \rightarrow \mathbb{N} \setminus 0$ und
- $m_{0\text{MIMO}}$ eine Abbildung $P_{\text{MIMO}} \rightarrow \mathbb{N} \cup 0$ sind.

Die Elemente aus T_{MIMO} werden MIMO-T-Elemente und die aus P_{MIMO} MIMO-P-Elemente genannt.

Bei der Transformation eines Petrinetzes in einen MIMO-Graphen werden besondere Elemente des Petrinetzes ausgezeichnet. Die sogenannten *MIMO-Elemente* sind solche, die mehrere oder gar keine Eingangs- oder Ausgangskanten besitzen, deren Eingangskantenkardinalität sich von deren Ausgangskantenkardinalität unterscheidet oder Stellen, denen eine initiale Markierung zugeordnet ist.

5.2 Definition (MIMO-Elemente (für farblose Netze))

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. Ein Element $x \in (P \cup T)$ ist ein MIMO-Element genau dann, wenn mindestens eine der folgenden Bedingungen erfüllt ist

1. $|\bullet x| > 1$ oder
2. $|x\bullet| > 1$ oder
3. $\exists x_1, x_2 \in (P \cup T)$ mit $K(x_1, x) \neq K(x, x_2)$ oder
4. $m_0(x) > 0$ für $x \in P$.

Stellen und Transitionen eines Petrinetzes, die keine MIMO-Elemente sind, heißen *SISO-Elemente* (SISO für *single input/single output*). Im folgenden wird der *Typ* von MIMO-Elementen entsprechend ihres Ursprungs in Petrinetzen unterschieden. Stellen des Petrinetzes werden durch MIMO-P-Elemente im MIMO-Graphen, Transitionen durch MIMO-T-Elemente repräsentiert.

Über die MIMO-Elemente wird im folgenden eine Nachbarschaftsrelation definiert. Die Nachbarschaftsrelation in Petrinetzen wird üblicherweise so definiert, daß Stellen und Transitionen genau dann benachbart sind, wenn zwischen ihnen eine Kante existiert. Demnach sind in Petrinetzen nie zwei Stellen oder zwei Transitionen benachbart. In MIMO-Graphen können dagegen auch MIMO-Elemente gleichen Typs benachbart sein.

5.3 Definition (MIMO-Nachbarschaftsrelation)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz und F^* die transitive Hülle der Kantenrelation F . Seien x, y beliebige MIMO-Elemente. x und y sind MIMO-Nachbarn genau dann, wenn $(x, y) \in F^*$ existiert und entweder $(x, y) \in F$ oder mindestens ein *Pfad* $(x, z_1), (z_1, z_2), (z_2, z_3), \dots, (z_n, y)$ existiert mit $(x, z_1), (z_i, z_{i+1}), (z_n, y) \in F$ für alle $i = 1, \dots, n-1$, und alle z_i und z_n sind keine MIMO-Elemente. Für solch einen Pfad von MIMO-Element x zu MIMO-Element y mit der Eigenschaft, daß kein weiteres MIMO-Element Teil dieses Pfades ist, schreiben wir $\text{MIMO-Pfad}(x, y)$. Für alle MIMO-Nachbarn x, y aus N ist (x, y) Element der MIMO-Nachbarschaftsrelation.

Für einen MIMO-Pfad von x nach y mit $(x, z_1), (z_1, z_2), (z_2, z_3), \dots, (z_n, y)$ und kein z_i ist MIMO-Element oder $z_1 = y$ gilt, daß $K(x, z_1) = K(z_1, z_2) = K(z_2, z_3) = \dots = K(z_n, y)$, da ein z_i sonst MIMO-Element wäre. Dieses K bezeichnen wir als *Kanten kardinalität* des MIMO-Pfades und schreiben $K(\text{MIMO-Pfad}(x, y))$. Existieren mehrere MIMO-Pfade zwischen x und y , läßt sich eine Summe der Kanten kardinalitäten der MIMO-Pfade eindeutig bestimmen. Wir schreiben dafür vereinfachend $\sum K(\text{MIMO-Pfad}(x, y))$.

Die MIMO-Graph basierte Aggregation eines Petrinetzes erfolgt in drei Schritten. Zuerst wird das Petrinetz in einen MIMO-Graphen transformiert. Dann wird

dieser MIMO-Graph aggregiert und zuletzt erfolgt eine Rücktransformation des aggregierten MIMO-Graphen in ein Petrinetz. Der folgende Abschnitt beschreibt formal die Transformation eines Petrinetzes in einen MIMO-Graphen.

5.1.3 Transformation eines farblosen Petrinetzes in einen MIMO-Graphen

Bei der Transformation eines Petrinetzes in einen MIMO-Graphen werden nur MIMO-Elemente übernommen. Die MIMO-Nachbarschaftsrelation wird durch Kanten im MIMO-Graphen beschrieben.

5.4 Definition (Transformation Petrinetz in MIMO-Graph)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz. N wird transformiert in den MIMO-Graphen $G_{\text{MIMO}}^N = (T_{\text{MIMO}}^N, P_{\text{MIMO}}^N, F_{\text{MIMO}}^N, K_{\text{MIMO}}^N, m_{0\text{MIMO}}^N)$ wobei

1. jedem $x_{\text{MIMO}} \in (P_{\text{MIMO}}^N \cup T_{\text{MIMO}}^N)$ eindeutig ein MIMO-Element $x \in (P \cup T)$ zugeordnet ist.
2. Sei $x_{\text{MIMO}}, y_{\text{MIMO}} \in (P_{\text{MIMO}}^N \cup T_{\text{MIMO}}^N)$. Dann ist $(x_{\text{MIMO}}, y_{\text{MIMO}}) \in F_{\text{MIMO}}^N$ genau dann, wenn die $x_{\text{MIMO}}, y_{\text{MIMO}}$ zugeordneten $x, y \in (P \cup T)$ MIMO-Nachbarn sind.
3. Seien $x, y \in (P \cup T)$ MIMO-Nachbarn in N und $x_{\text{MIMO}}, y_{\text{MIMO}}$ die x, y zugeordneten MIMO-Elemente aus G_{MIMO}^N .
Dann ist $K_{\text{MIMO}}(f_{\text{MIMO}}(x_{\text{MIMO}}, y_{\text{MIMO}})) = \sum K(\text{MIMO} - \text{Pfad}(x, y))$ für alle transitiven Kantenbeziehungen zwischen x und y , in denen kein weiteres MIMO-Element auftritt.

Die Definition der den Kanten im MIMO-Graphen zugeordneten Kardinalitäten soll an folgendem Beispiel verdeutlicht werden. Abbildung 5.4 zeigt die Anwendung der Definition.

Die linke Seite der Abbildung zeigt ein Petrinetz. Die Transitionen $T1$ und $T2$ sind nach Definition 5.2 MIMO-Elemente, denn sie besitzen mehrere Ausgangs- und keine Eingangs- ($T1$) bzw. mehrere Eingangs- und keine Ausgangskanten ($T2$). Sie sind darüber hinaus MIMO-Nachbarn, denn sie sind transitiv über Kanten miteinander verbunden und es gibt einen Pfad $(T1, P1), (P1, T2)$ mit $P1$ ist kein MIMO-Element. Es gibt noch einen über das Nicht-MIMO-Element $P2$ verlaufenden Pfad. Für die Zugehörigkeit von $(T1, T2)$ zur MIMO-Nachbarschaftsrelation genügt aber die Existenz eines Pfades. $P3$ ist ein MIMO-Element, da $K(T1, P3) \neq K(P3, T2)$ ist. Da für die MIMO-Nachbarschaft die Existenz der bedingten transitiven Kantenbeziehung genügt, gibt es im MIMO-Graphen keine

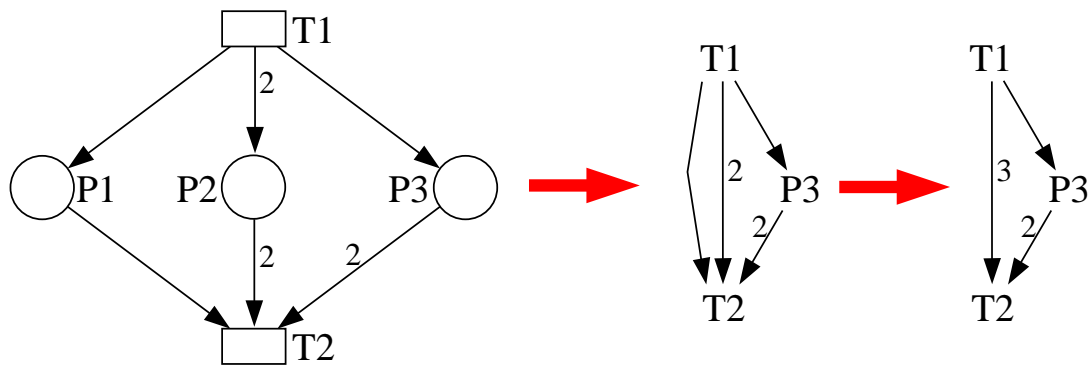


Abbildung 5.4: Kantenkardinalität im MIMO-Graph

mehrfachen Kanten zwischen Nachbarn. Die Kantenkardinalitäten aller bedingten transitiven Kantenbeziehungen werden addiert. Unbeschriftete Kanten besitzen die Kantenkardinalität 1. Diese Addition und das „Verschmelzen“ von Kanten zeigt Abbildung 5.4. Da eine Kante im MIMO-Graphen eine transitive Kantenbeziehung zweier Elemente im Petrinetz repräsentiert, kann es vorkommen, daß eine Kante im MIMO-Graphen für eine „Kette“ von Kanten im Petrinetz steht. Problematisch wäre es, wenn in dieser „Kette“ von Kanten verschiedene Kardinalitäten aufträten. Wie an dem Beispiel und der Stelle P_3 ersichtlich, kann dieses Problem durch die Definition von MIMO-Elementen (siehe Definition 5.2) nicht auftreten, denn ein Wechsel der Kantenkardinalität erfolgt stets durch ein MIMO-Element.

Zusammenfassend läßt sich die Idee der Transformation eines Petrinetzes in einen MIMO-Graphen beschreiben als das Weglassen rein sequentieller Elemente. Sequentielle Elemente werden als solche bezeichnet, wenn sie einen rein sequentiellen Markenstrom unterstützen. Das sind z.B. Stellen, in die Marken aus genau einer und aus denen Marken in genau eine Richtung fließen, ohne die Menge des Stromes (Kardinalität) zu verändern, sowie Transitionen, die Marken aus genau einer Stelle abziehen und diese Anzahl von Marken in eine Stelle transportieren.

Der folgende Abschnitt behandelt die Aggregation des MIMO-Graphen, des zweiten Schritts der MIMO-Graph basierten Aggregation.

5.1.4 Aggregation des farblosen MIMO-Graphen

Nach der Transformation des Petrinetzes in einen MIMO-Graphen wird der MIMO-Graph aggregiert. Dabei werden benachbarte, also durch eine Kante verbundene Elemente gleichen Typs zu einem Element verschmolzen, wenn sie bestimmten Bedingungen genügen. Diese Bedingungen sind in Abbildung 5.5 gra-

fisch veranschaulicht.

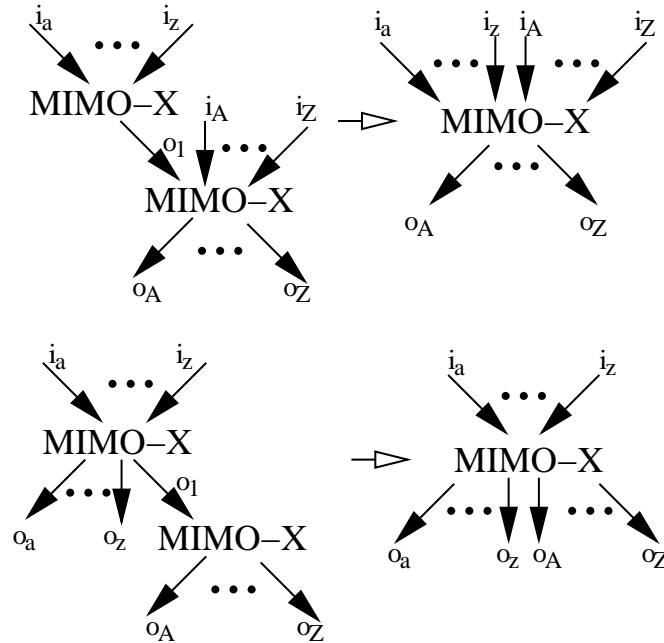


Abbildung 5.5: Aggregierungsregeln

Die obere Regel besagt, daß zwei Elemente gleichen Typs (x steht für entweder T oder P) dann verschmolzen werden können, wenn das Element, von dem eine Kante aus zu dem anderen verläuft, keine weiteren Ausgangskanten besitzt. Die alternative Regel ist unten in Abbildung 5.5 dargestellt. Wenn das Element, zu dem die Kante zwischen den beiden zu verschmelzenden Elementen hin verläuft, keine weiteren Eingangskanten besitzt, können die beiden Elemente miteinander verschmolzen werden. Beim Verschmelzen wird die Kante zwischen den beiden verschmelzenden Elementen gelöscht und alle anderen Ein- und Ausgangskanten der beiden Elemente sind Ein- und Ausgangskanten des neuen verschmolzenen Elementes.

5.5 Definition (Aggregierung von MIMO-Elementen)

Sei $G_{\text{MIMO}}^N = (T_{\text{MIMO}}^N, P_{\text{MIMO}}^N, F_{\text{MIMO}}^N, K_{\text{MIMO}}^N, m_{0\text{MIMO}}^N)$ ein MIMO-Graph. Seien $x, y \in (T_{\text{MIMO}}^N \cup P_{\text{MIMO}}^N)$ gleichen Typs mit $(x, y) \in F_{\text{MIMO}}^N$ und gelten beide folgenden Aggregierungsbedingungen.

1. Wenn ein $z \neq y$ mit $(x, z) \in F_{\text{MIMO}}^N$ existiert, dann existiert kein $z' \neq x$ mit $(z', y) \in F_{\text{MIMO}}^N$ und
2. wenn ein $z \neq x$ mit $(z, y) \in F_{\text{MIMO}}^N$ existiert, dann existiert kein $z' \neq y$ mit $(x, z') \in F_{\text{MIMO}}^N$.

Dann werden x, y zu einem neuen MIMO-Element w derart aggregiert, daß ein aggregierter MIMO-Graph $G_{\text{MIMO}'^N} = (T_{\text{MIMO}'^N}, P_{\text{MIMO}'^N}, F_{\text{MIMO}'^N}, K_{\text{MIMO}'^N}, m_{0\text{MIMO}'^N})$ entsteht mit

- $P_{\text{MIMO}'^N} = ((P_{\text{MIMO}^N} \setminus (x \cup y)) \cup w)$, falls $x, y \in P_{\text{MIMO}^N}$
- $T_{\text{MIMO}'^N} = ((T_{\text{MIMO}^N} \setminus (x \cup y)) \cup w)$, falls $x, y \in T_{\text{MIMO}^N}$
-

$$F_{\text{MIMO}'^N} = \begin{cases} ((F_{\text{MIMO}^N} \setminus ((z, x) \cup (z, y))) \cup (z, w)) & , \textcircled{1} \text{ und} \\ ((F_{\text{MIMO}^N} \setminus ((x, z) \cup (y, z))) \cup (w, z)) & , \textcircled{2} \text{ und} \\ F_{\text{MIMO}^N} \setminus (x, y) & \end{cases}$$

- ① für alle $z \neq x, y$ mit $(z, x) \in F_{\text{MIMO}^N}$ oder/und $(z, y) \in F_{\text{MIMO}^N}$
- ② für alle $z \neq x, y$ mit $(x, z) \in F_{\text{MIMO}^N}$ oder/und $(y, z) \in F_{\text{MIMO}^N}$

- Für alle $(z, w) \in F_{\text{MIMO}'^N}$ ist

$$K_{\text{MIMO}'^N}(z, w) \begin{cases} K_{\text{MIMO}^N}(z, x) & , \textcircled{1} \\ K_{\text{MIMO}^N}(z, y) & , \textcircled{2} \\ K_{\text{MIMO}^N}(z, x) + K_{\text{MIMO}^N}(z, y) & , \textcircled{3} \end{cases}$$

- ① falls $(z, x) \in F_{\text{MIMO}^N}$ und $(z, y) \notin F_{\text{MIMO}^N}$
- ② falls $(z, y) \in F_{\text{MIMO}^N}$ und $(z, x) \notin F_{\text{MIMO}^N}$
- ③ falls $(z, x), (z, y) \in F_{\text{MIMO}^N}$

und für alle $(w, z) \in F_{\text{MIMO}'^N}$ ist

$$K_{\text{MIMO}'^N}(w, z) \begin{cases} K_{\text{MIMO}^N}(x, z) & , \textcircled{1} \\ K_{\text{MIMO}^N}(y, z) & , \textcircled{2} \\ K_{\text{MIMO}^N}(x, z) + K_{\text{MIMO}^N}(y, z) & , \textcircled{3} \end{cases}$$

- ① falls $(x, z) \in F_{\text{MIMO}^N}$ und $(y, z) \notin F_{\text{MIMO}^N}$
- ② falls $(y, z) \in F_{\text{MIMO}^N}$ und $(x, z) \notin F_{\text{MIMO}^N}$
- ③ falls $(x, z), (y, z) \in F_{\text{MIMO}^N}$

- Falls $x, y \in P_{\text{MIMO}^N}$, dann ist $m_{0\text{MIMO}'^N}(w) = m_{0\text{MIMO}^N}(x) + m_{0\text{MIMO}^N}(y)$

5.1.5 Löschen von SISO-Elementen in MIMO-Graphen

Während der Aggregation können aus Elementen mit MIMO-Eigenschaft solche mit nur einer Eingangs- und Ausgangskante mit identischer Kantenkardinalität werden. Diese Elemente werden SISO-Elemente (siehe Abschnitt 5.1.2) genannt. Abbildung 5.6 zeigt solch ein Beispiel.

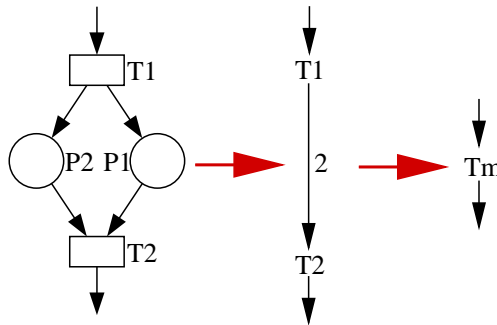


Abbildung 5.6: SISO-Elemente in MIMO-Graphen

Die linke Seite der Abbildung zeigt den Ausschnitt eines Petrinetzes. Die Transitionen T_1 und T_2 sind MIMO-Transitionen, während die beiden Stellen keine MIMO-Stellen sind. Bei der Transformation in einen MIMO-Graphen werden die beiden MIMO-T-Elemente durch eine Kante mit der Kantenkardinalität 2 verbunden. Die MIMO-T-Elemente können miteinander zu einem MIMO-T-Element T_m verschmolzen werden. Dieses Element besitzt jedoch keine MIMO-Eigenschaften mehr. Elemente mit SISO-Eigenschaft werden aus dem MIMO-Graphen gelöscht. Das MIMO-Element aus dem Vorbereich und das aus dem Nachbereich des gelöschten Elementes werden durch eine Kante mit der entsprechenden Kardinalität verbunden.

Die Aggregation des MIMO-Graphen ist beendet, wenn es keine benachbarten aggregierbaren Elemente gleichen Typs und keine SISO-Elemente mehr gibt.

5.1.6 Rücktransformation des aggregierten MIMO-Graphen in ein Petrinetz

Im dritten Schritt der MIMO-Graphen basierten Aggregation wird der MIMO-Graph zurücktransformiert in ein Petrinetz. Nach der Aggregation des MIMO-Graphen sind fast ausschließlich Elemente verschiedenen Typs benachbart. Bei der Rücktransformation werden MIMO-T-Elemente als Transitionen und MIMO-P-Elemente als Stellen repräsentiert. Die Kantenbeziehungen und deren Kardinalitäten im Petrinetz entsprechen denen des MIMO-Graphen. Sind im aggregierten

MIMO-Graph Elemente gleichen Typs benachbart, so wird bei der Rücktransformation ein Element des anderen Typs eingefügt. Mit der Rücktransformation in ein (aggregiertes) Petrinetz ist der Aggregierungsalgorithmus abgeschlossen.

5.2 Konsistenz der MIMO-Graphen basierten Aggregation

In diesem Abschnitt wird gezeigt, daß die strukturellen Eigenschaften von Petrinetzmodellen Beschränktheit und Lebendigkeit bei der MIMO-Graphen basierten Aggregation bewahrt bleiben. Die Bewahrung dieser wichtigen Eigenschaften fassen wir in dem Begriff *Konsistenz* zusammen.

5.6 Definition (Konsistenz)

Die Aggregation eines Petrinetzes N in ein (aggregiertes) Petrinetz N' heißt konsistent, wenn für N und N' folgende Zusammenhänge gelten:

1. Wenn alle Transitionen in N lebendig sind, dann sind auch alle Transitionen in N' lebendig.
2. Wenn N beschränkt ist, dann ist auch N' beschränkt.

Für die in obiger Definition verwendeten Begriffe wird auf die Definitionen 3.1 und 3.2 in Abschnitt 3.0.4 auf Seite 27 verwiesen.

Es muß hier angemerkt werden, daß es einen Konsistenzbegriff für eine strukturelle Eigenschaft von Petrinetzen gibt, der eine vollständige Überdeckung eines Petrinetzes mit Transitions-Invarianten beschreibt. Auf diesen Begriff wird in dieser Arbeit nicht eingegangen und stattdessen auf [119] verwiesen. Der hier vorgestellte Begriff folgt dem in [123] verwendeten. In [123] wurde die Konsistenz der dort vorgestellten Reduktionsregeln (auf der Basis der Regeln aus [9, 10]) gezeigt. Im Anschluß an den folgenden Beweis wird die MIMO-Graphen basierte Aggregation mit diesen Reduktionsregeln verglichen.

Für den folgenden Beweis der Konsistenz der MIMO-Graphen basierten Aggregation wird angenommen, daß das Originalmodell lebendig und beschränkt ist. Zu zeigen ist, daß das entstehende aggregierte Modell ebenfalls lebendig und beschränkt ist.

5.2.1 Konsistenz beim Löschen von SISO-Elementen

In diesem Abschnitt wird gezeigt, daß der Übergang vom Petrinetz in einen MIMO-Graphen konsistent ist. Dazu wird angenommen, daß wir ein Petrinetz in

einen MIMO-Graphen überführen und ohne Verschmelzungen wieder zurück in ein Petrinetz. Bei diesem Vorgang werden ausschließlich SISO-Elemente gelöscht (siehe Abschnitte 5.1.3 und 5.1.6). Das bedeutet, daß in Abbildung 5.7 dargestellte Strukturen aggregiert werden.

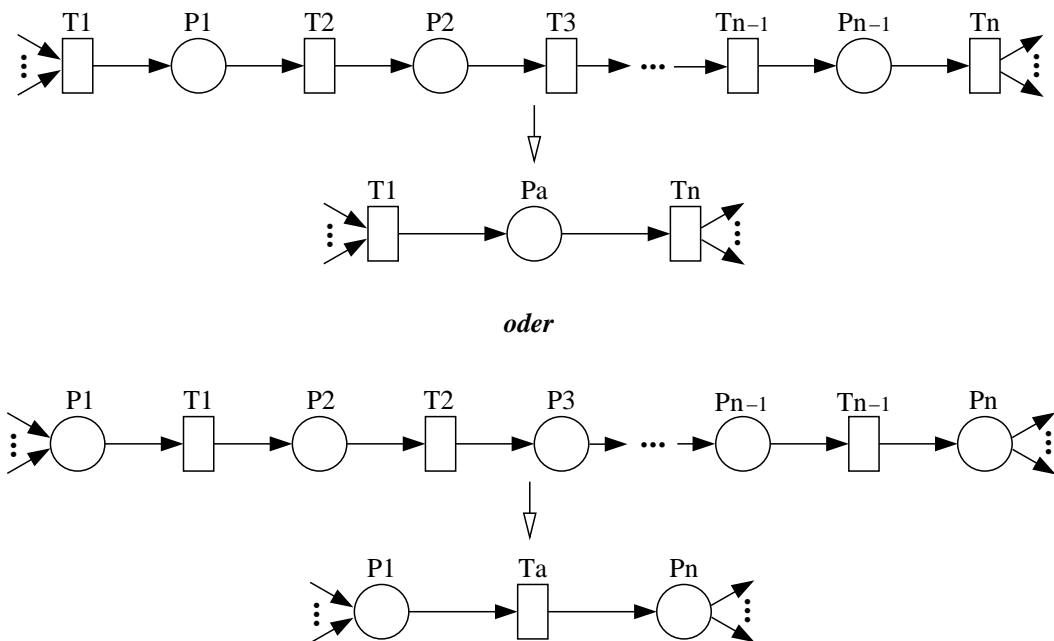


Abbildung 5.7: Löschen von SISO-Elementen

Es gibt vier verschiedene Möglichkeiten dieser Sequenzen von Transitionen und Stellen. Zwei von diesen Möglichkeiten sind in Abbildung 5.7 dargestellt. Diese Sequenzen beginnen und enden mit demselben Typ von Netzelementen. Diese Transitionen T_1 , T_n bzw. diese Stellen P_1 , P_n sind MIMO-Elemente. Mehrere Eingangs- bzw. Ausgangskanten bei diesen Elementen sollen auf die MIMO-Eigenschaft aufmerksam machen. Es sei aber angemerkt, daß z.B. T_1 statt mehrerer Eingangs- auch weitere Ausgangskanten haben könnte oder eine Eingangskante mit einer Kardinalität größer eins oder aber keine Eingangskante. Die Definition eines MIMO-Elementes ist in Definition 5.2 auf Seite 66 gegeben. Für die beiden dargestellten Möglichkeiten von Sequenzen wird bei der Transformation in einen MIMO-Graphen eine Struktur erzeugt, die die beiden MIMO-Elemente gleichen Typs (T_1 und T_n bzw. P_1 und P_n) mit einer Kante verbindet, während alle anderen Elemente wegen ihrer SISO-Eigenschaft gelöscht werden. Bei der Rücktransformation werden eine zusätzliche Stelle P_a bzw. eine zusätzliche Transition T_a eingefügt, so daß die in Abbildung 5.7 dargestellten Strukturen entstehen.

Die beiden anderen Möglichkeiten von Sequenzen von Stellen und Transitionen

beginnen und enden mit MIMO-Elementen verschiedenen Typs. Bei der Transformation in einen MIMO-Graphen werden die SISO-Elemente aus diesen Sequenzen gelöscht und im MIMO-Graphen verbleiben benachbarte MIMO-Elemente verschiedenen Typs. Diese Struktur kann ohne Einfügen zusätzlicher Elemente in ein Petrinetz rücktransformiert werden.

Zu zeigen ist nun, daß das Löschen von SISO-Elementen konsistent ist. Anhand der oberen in Abbildung 5.7 dargestellten Aggregation soll dieser Beweis geführt werden. Die Nichtkonsistenz der Aggregation wäre bspw. gegeben, wenn die Transition T_n lebendig im Originalmodell aber nicht lebendig oder tot wäre im aggregierten Modell. Dazu müsste sich jedoch durch die Aggregation das Schaltverhalten der Transition T_1 ändern. Dieses Schaltverhalten bleibt jedoch unbeeinflusst. Die Beschränktheit betreffend gilt, wenn die Stelle P_1 beschränkt ist, dann auch die Stellen P_2, \dots, P_{n-1} , da die Transitionen T_2, \dots, T_{n-1} wegen der SISO-Eigenschaft weder Marken erzeugen noch vernichten, denn es gilt $K(T_x, P_x) = K(P_x, T_{x+1})$ und $K(P_x, T_{x+1}) = K(T_{x+1}, P_{x+1})$ für alle $x = \{1, \dots, n-1\}$. Wenn aber P_1 beschränkt ist, so auch P_a , da sich das Schaltverhalten der Transition T_1 durch die Aggregation nicht ändert und sowohl P_1 als auch P_a nur durch diese Transition beeinflusst werden.

Das Zeigen der Konsistenz für den in Abbildung 5.7 unten dargestellten Fall erfolgt analog. Wegen der genannten Gründe gilt dabei, daß wenn T_1 lebendig, dann auch T_2, \dots, T_{n-1} . Wenn aber T_1 lebendig, dann auch T_a . Die Beschränktheit der Stellen P_1 und P_n bleiben unbeeinflusst.

Die Idee der Definition 5.2 von MIMO-Elementen besteht demnach gerade darin, Elemente zu erkennen, die die Lebendigkeit und Beschränktheit von Petrinetzen beeinflussen können. Elemente mit der SISO-Eigenschaft beeinflussen diese Eigenschaften nicht, so daß ihre Entfernung aus dem Petrinetz Konsistenz bewahrt. Es bleibt zu zeigen, daß die Verschmelzungsregeln von MIMO-Elementen desselben Typs konsistent sind. Wir betrachten dazu zunächst den Fall, daß die zu verschmelzenden MIMO-Elemente MIMO-T-Elemente sind.

5.2.2 Konsistenz der Verschmelzung von Transitionen

MIMO-T-Elemente werden wie in Abbildung 5.5 auf Seite 69 dargestellt verschmolzen. Die beiden in Abbildung 5.8 dargestellten Fälle zeigen die Anwendung der in Abschnitt 5.1.4 eingeführten Aggregierungsregeln für Transitionen.

Sind zwei MIMO-Elemente im MIMO-Graphen benachbart, so befanden sich im originalen Petrinetz zwischen diesen Elementen nur Elemente mit der SISO-Eigenschaft. Die Konsistenz des Löschens von SISO-Elementen wurde bereits im vorhergehenden Abschnitt 5.2.1 gezeigt. Es genügt deshalb, die Konsistenz der Verschmelzung von Transitionen mit nur einer Stelle anstelle einer Sequenz aus

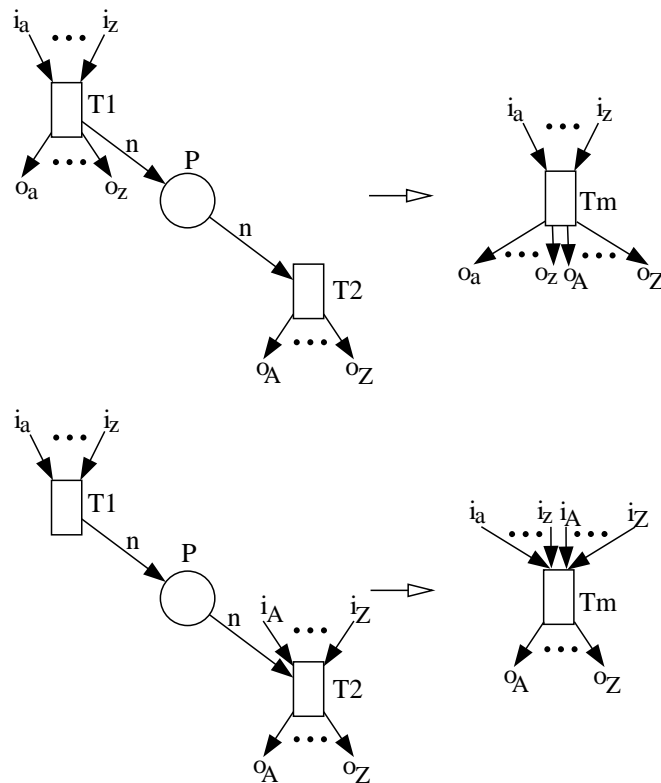


Abbildung 5.8: Verschmelzung von Transitionen

Stellen und Transitionen mit jeweiliger SISO-Eigenschaft zwischen diesen Transitionen zu zeigen.

Da entweder T_1 keine weiteren Ausgangskanten oder T_2 keine weiteren Eingangskanten besitzen darf, gibt es neben dem in Abbildung 5.8 dargestellten SISO-Element P kein MIMO-Element P_1 mit $P_1 \in (T_1 \bullet \cap \bullet T_2)$. Existieren zwischen T_1 und T_2 weitere SISO-P-Elemente werden die Kantenkardinalitäten entsprechend Abschnitt 5.1.3 zusammengefaßt. Es ist jedoch zu beachten, daß die MIMO-T-Elemente darüber hinaus transitiv über die Kanten o_x und i_y mit $X = \{A, \dots, Z\}$, $y = \{a, \dots, z\}$ im MIMO-Graphen verbunden sein können.

Lebendigkeit

Wir zeigen zunächst die Bewahrung der Lebendigkeit bei der Verschmelzung von Transitionen.

Für den in Abbildung 5.8 oben dargestellten Fall gilt, genau wenn T_1 lebendig ist, dann auch T_2 , denn wenn T_1 lebendig, so gibt es eine nichtleere Menge M_{T_1} von erreichbaren Markierungen, unter denen Transition T_1 Konzession besitzt.

Für alle Markierungen $m_{T1} \in M_{T1}$ gilt $m_{T1} \xrightarrow{T1} m$ mit $T2$ besitzt Konzession in m . Außerdem gibt es keine Transition t mit $t \neq T1$ und keine Markierungen $m1, m2$ mit $m1 \xrightarrow{t} m2$, so daß $T2$ keine Konzession in $m1$ aber in $m2$ besitzt. Somit ist $T2$ genau dann lebendig, wenn $T1$ lebendig ist. Wegen $\bullet T1 = \bullet Tm$ und da für alle $p \in \bullet T1, \bullet Tm$ gilt $K(p, T1) = K(p, Tm)$, gilt aber auch, daß Tm im aggregierten Modell genau dann lebendig ist, wenn es $T1$ im Originalmodell ist.

Für den in Abbildung 5.8 unten dargestellten Fall gilt aus der Annahme der Lebendigkeit aller Transitionen, daß sowohl $T1$ als auch $T2$ lebendig sind. Zu zeigen ist, daß dann auch Tm im aggregierten Modell lebendig ist. Die Transition Tm wäre nur dann tot (nicht lebendig), wenn durch die Verschmelzung eine Schlinge zu einer unmarkierten (markierten) Stelle entstünde, die außer dieser Hin- und Rückkante zur Transition Tm keine weitere Eingangskante hätte.

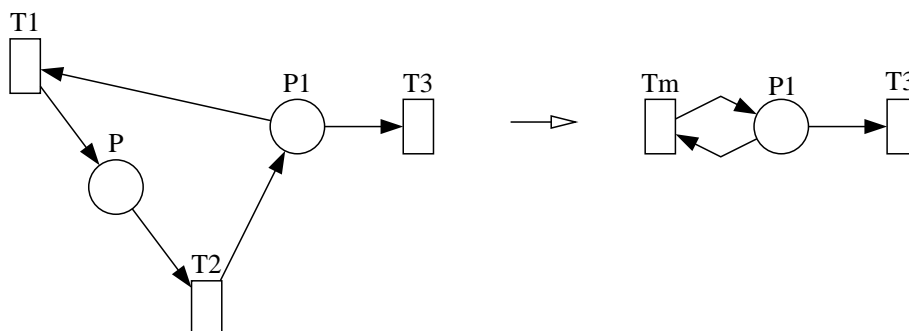


Abbildung 5.9: Entstehen einer Schlinge bei der Verschmelzung

Abbildung 5.9 zeigt den betreffenden Ausschnitt eines Petrinetzmodells für den die Verschmelzung der Transitionen $T1$ und $T2$ in die Transition Tm zu einer Schlinge führt. Eine Schlinge zu einer Stelle $P1$ mit den genannten Eigenschaften kann nur dann entstehen, wenn im originalen Netz ein Pfad der transitiven Kantenbeziehungen $F^*(T2, P1)$ und $F^*(P1, T1)$ existiert, da entweder $T1$ keine weiteren Ausgangskanten oder $T2$ keine weiteren Eingangskanten außer denen zu und von P haben darf. Wäre nun aber $P1$ unmarkiert (markiert) und hätte keine weiteren Eingangskanten, so wäre auch $T1$ tot (nicht lebendig) und somit auch $T2$, denn P wäre nie markiert (irgendwann nie mehr). Das aber ist ein Widerspruch zur der Annahme, daß sowohl $T1$ als auch $T2$ lebendig sind.

Beschränktheit

In diesem Abschnitt wird gezeigt, daß die Beschränktheit eines Petrinetzmodells durch die in Abschnitt 5.1.4 beschriebene und in Abbildung 5.8 dargestellte Verschmelzung von Transitionen bewahrt bleibt.

Wir betrachten wieder Abbildung 5.8. Aus der Annahme der Beschränktheit des Originalmodells folgt [119, 51] die Existenz einer das Originalnetz überdeckenden Stellen-Invariante i . Aus Definition 3.3 folgt

$$\begin{aligned}\sum_{p \in \bullet T_1} i(p) \cdot K(p, T_1) &= \sum_{p' \in T_1 \bullet} i(p') \cdot K(T_1, p') \\ \sum_{p \in \bullet T_2} i(p) \cdot K(p, T_2) &= \sum_{p' \in T_2 \bullet} i(p') \cdot K(T_2, p')\end{aligned}$$

Aus der SISO-Eigenschaft von P folgt

$$\begin{aligned}&\sum_{p \in \bullet T_1} i(p) \cdot K(p, T_1) + \sum_{p \in \bullet T_2 \setminus P} i(p) \cdot K(p, T_2) \\ &= \sum_{p' \in T_1 \bullet \setminus P} i(p') \cdot K(T_1, p') + \sum_{p' \in T_2 \bullet} i(p') \cdot K(T_2, p')\end{aligned}$$

Daraus ergibt sich

$$\begin{aligned}&\sum_{\substack{p \in \bullet T_1 \\ p \notin \bullet T_2 \setminus P}} i(p) \cdot K(p, T_1) + \sum_{\substack{p \in \bullet T_2 \setminus P \\ p \notin \bullet T_1}} i(p) \cdot K(p, T_2) + \sum_{p \in (\bullet T_1 \cap (\bullet T_2 \setminus P))} i(p) \cdot (K(p, T_1) + K(p, T_2)) \\ &= \sum_{\substack{p' \in T_1 \bullet \setminus P \\ p' \notin T_2 \bullet}} i(p') \cdot K(T_1, p') + \sum_{\substack{p' \in T_2 \bullet \\ p' \notin T_1 \bullet \setminus P}} i(p') \cdot K(T_2, p') + \sum_{p' \in ((T_1 \bullet \setminus P) \cap (T_2 \bullet))} i(p') \cdot (K(T_1, p') + K(T_2, p')).\end{aligned}$$

Das entspricht (siehe auch Definition 5.5 auf Seite 69) aber gerade

$$\sum_{p \in \bullet T_m} i(p) \cdot K(p, T_m) = \sum_{p' \in T_m \bullet} i(p') \cdot K(T_m, p').$$

Die Stellen-Invarianten des Originalmodells werden bei der Verschmelzung von Transitionen demnach nur darin geändert, daß die SISO-Stellen zwischen diesen Transitionen nicht mehr in den Stellen-Invarianten des aggregierten Modells abgebildet werden. Ansonsten werden die das Originalmodell überdeckenden Stellen-Invarianten nicht geändert. Das aggregierte Modell ist demnach ebenso wie das Originalmodell beschränkt.

Die Verschmelzung von Transitionen ist demnach konsistent. Es bleibt zu zeigen, daß auch die Verschmelzung von Stellen eine konsistente Aggregation ist.

5.2.3 Konsistenz der Verschmelzung von Stellen

Abbildung 5.10 zeigt die beiden zu betrachtenden Fälle der Verschmelzung von Stellen.

Durch das Verschmelzen der Stellen P_1 und P_2 zur Stelle P_m kann nur das Schaltverhalten der mit diesen Stellen verbundenen Transitionen beeinflußt werden.

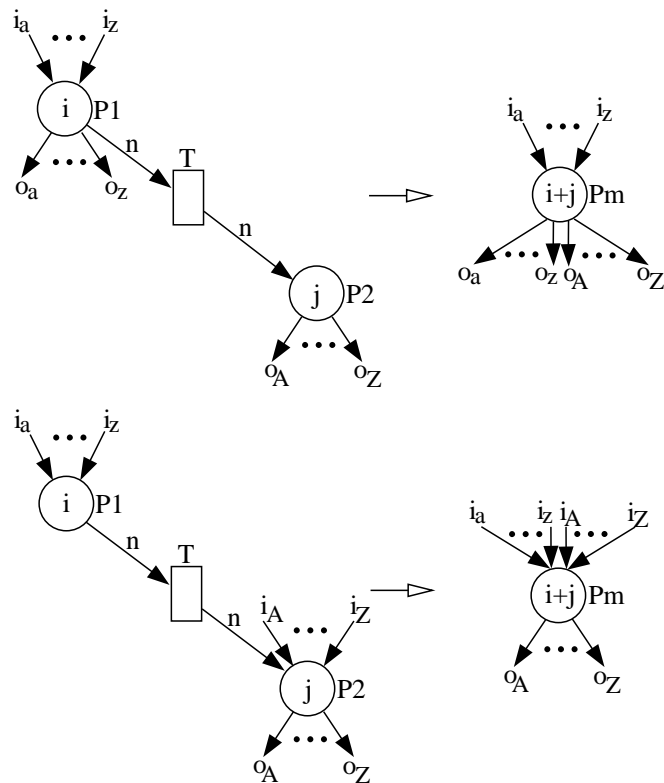


Abbildung 5.10: Verschmelzung von Stellen

Dabei wird sogar das Schaltverhalten der Transitionen bewahrt, die nur entweder mit $P1$ oder mit $P2$ eine gemeinsame Kante besitzen. Abbildung 5.11 zeigt dagegen die Fälle, in denen eine Transition $T1$ mit beiden Stellen verbunden ist.

Unter den gezeigten Strukturen ist die jeweils nach der Verschmelzung der Stellen $P1$ und $P2$ entstandene aggregierte Struktur dargestellt. Die Transition $T1$ kann sowohl die MIMO- als auch die SISO-Eigenschaft besitzen. In Abbildung 5.11 sei aber angenommen, daß es sich bei $T1$ um eine Transition mit der MIMO-Eigenschaft handelt, so daß sie beim Übergang des Petrinetzes in einen MIMO-Graphen nicht gelöscht wird. Die Konsistenz des Löschs von SISO-Elementen wurde bereits in Abschnitt 5.2.1 gezeigt.

Lebendigkeit

Zunächst wird die Lebendigkeit der Transition $T1$ für die in Abbildung 5.11 dargestellten Fälle diskutiert. Da $T1$ in dem in der Mitte der Abbildung dargestellten Ausschnitt eines Petrinetzes keine Eingangskante besitzt, die mit den Stellen $P1$ oder $P2$ verbunden ist, ist ihr Schaltverhalten durch die Verschmelzung der beiden Stellen nicht beeinflusst. Im auf der linken Seite dargestellten Ausschnitt

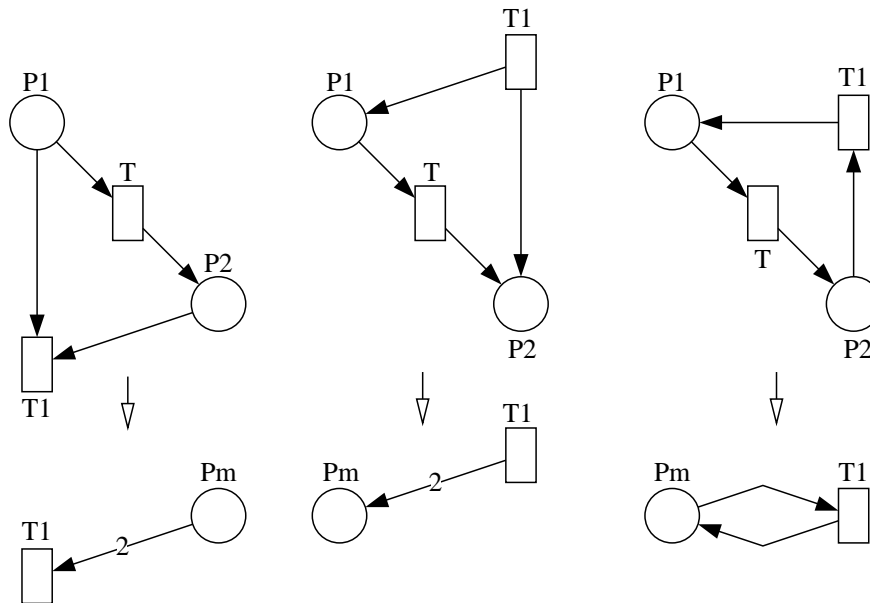


Abbildung 5.11: Lebendigkeit von Transitionen

besitzt T_1 Konzession genau dann, wenn sowohl P_1 als auch P_2 markiert sind. Im Originalmodell schaltet T_1 demnach unter Verbrauch zweier P_1 und P_2 markierender Marken. Nach der Verschmelzung dieser beiden Stellen zieht T_1 im aggregierten Modell ebenfalls zwei Marken aus der Stelle P_m ab. Die Lebendigkeit dieser Transition kann demnach nicht beeinflusst sein. In dem auf der rechten Seite in Abbildung 5.11 dargestellten Fall besitzt T_1 im Originalmodell nur dann Konzession, wenn P_2 markiert ist. Das Schalten der Transition T_1 führt zur Markierung von P_1 . Nach der Verschmelzung von P_1 und P_2 zur Stelle P_m ist T_1 im aggregierten Modell über eine Schlinge mit P_m verbunden. T_1 besitzt demnach im aggregierten Modell nur dann Konzession, wenn P_m markiert ist. Das Schalten von T_1 im aggregierten Modell führt zur Markierung von P_m . Demnach ist auch für diesen Fall die Lebendigkeit der Transition T_1 unbeeinflusst.

Beschränktheit

Die Verschmelzung von Stellen ist nunmehr konsistent, wenn sich die Beschränktheitseigenschaft des originalen Petrinetzes nicht ändert. Betrachten wir wieder Abbildung 5.10. Aus der Annahme der Beschränktheit von N folgt, daß N mit Stellen-Invarianten überdeckt ist. Wegen der SISO-Eigenschaft von T und aus Definition 3.3 folgt sogar, daß für alle Stellen-Invarianten i gilt, daß $i(P_1) = i(P_2)$. Durch die Verschmelzung der Stellen ändert sich die Inzidenzmatrix N des Originalmodells N in die Inzidenzmatrix N' des aggregierten Modells. Dabei wird die Zeile für P_2 zu der Zeile P_1 addiert und es entsteht die Zeile für P_m . In der

Spalte für T gibt es in N nur die Einträge $-K(P1, T)$ und $K(T, P2)$. Wegen der SISO-Eigenschaft von T gilt $K(P1, T) = K(T, P2)$. Nach Addition der Zeilen von $P1$ und $P2$ ist der Spaltenvektor für T ein Nullvektor, der durch die Aggregation gelöscht wird. Interpretiert man in allen Stellen-Invarianten i mit $N^T \cdot i = 0$ die Einträge für $P1$ als Einträge für Pm und löscht die Einträge für $P2$, so erhält man alle Stellen-Invarianten i' mit $|i| = |i'| + 1$ und $N'^T \cdot i' = 0$. Das durch die Inzidenzmatrix N' beschriebene aggregierte Modell N' ist somit beschränkt unter der Annahme, daß das Originalmodell N beschränkt ist.

Die MIMO-Graphen basierte Aggregation ist demnach konsistent. Im folgenden Abschnitt wird die MIMO-Graphen basierte Aggregation mit aus der Literatur bekannten strukturellen Aggregationen verglichen.

5.2.4 Vergleich mit bekannten Reduktionsregeln

Die MIMO-Graphen basierte Aggregation ist aufgrund ihrer Anwendung auf die Struktur von Petrinetzen mit sogenannten *Reduktionsregeln* aus der Literatur vergleichbar. In [88, 11, 80, 105, 126] finden sich erste Vorschläge für die Reduktion sehr einfacher Strukturen von Petrinetzen, die z.T. nur für eingeschränkte Klassen von Petrinetzen, wie z.B. markierte Graphen anwendbar sind. Einen einführenden Überblick bieten auch [104, 118, 1].

Die in [123] vorgestellten Reduktionen basieren auf den Arbeiten von Berthelot [10, 9] und sind in ihrem Umfang und ihrer Anwendbarkeit der hier präsentierten MIMO-Graphen basierten Aggregation vergleichbar. In diesem Abschnitt wird dieser Vorschlag diskutiert und es werden Unterschiede und Gemeinsamkeiten mit der MIMO-Graphen basierten Aggregation verdeutlicht.

In [123] werden neun Reduktionsregeln vorgestellt, die hier sukzessive diskutiert werden sollen. Die erste Reduktionsregel besagt, daß alle Transitionen ohne eine Stelle im Vorbereich zusammen mit den Stellen im Nachbereich dieser Transition gelöscht werden. Abbildung 5.12 zeigt die Anwendung der ersten Reduktionsregel aus [123] an einem Beispiel.

Die linke Seite der Abbildung zeigt das Originalmodell. Die Transition $T1$ ist eine Transition ohne eine Stelle im Vorbereich. Sie soll nach der ersten Reduktionsregel zusammen mit der Stelle $P1$ gelöscht werden. Die Stelle $P1$ ist unbeschränkt und somit das Originalmodell. Das aus der Reduktion entstandene Petrinetz, das auf der rechten Seite der Abbildung dargestellt ist, ist jedoch beschränkt, da alle Stellen des Netzes 1-beschränkt sind. Für die in dieser Arbeit mittels MIMO-Graphen aggregierte Modelle nehmen wir die Beschränktheit der Originalmodelle explizit an. Für das in Abbildung 5.12 dargestellte Beispiel würde jedoch mittels MIMO-Graphen dasselbe auf der rechten Seite in Abbildung 5.12 gezeigte aggregierte Modell entstehen.

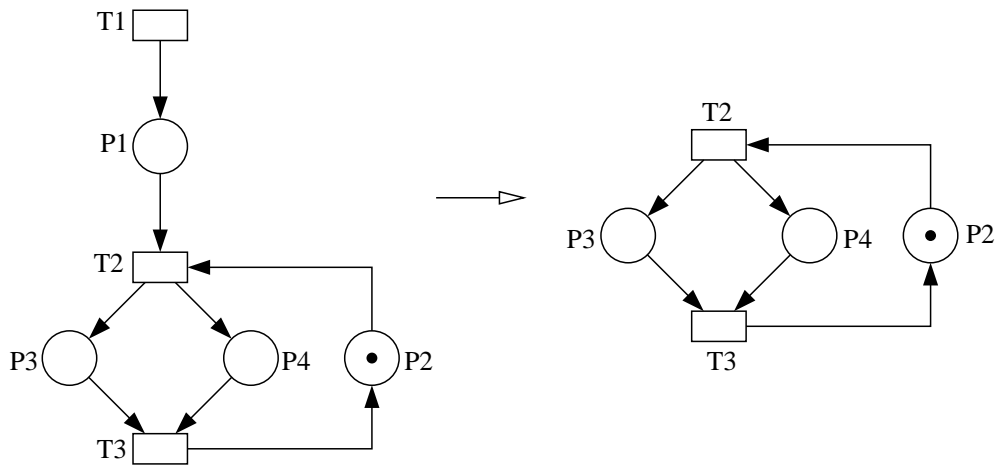


Abbildung 5.12: Erste Reduktionsregel aus [123]

Die zweite Reduktionsregel aus [123] besagt, daß alle *toten* Stellen und alle folgenden toten Transitionen gelöscht werden können. Dabei gelten als tote Stellen solche, die keine Transition im Vorbereich besitzen und deren Markierung weniger Marken beinhaltet als die Kantenkardinalität zu den Transitionen im Nachbereich erfordert. Diese Transitionen sind demnach tot (siehe Definition 3.1 auf Seite 28). Diese Reduktion führt nicht zur Verringerung des Zustandsraumes des originalen Petrinetzes.

Die dritte Reduktionsregel aus [123] betrifft eine ähnlich triviale Struktur, die in realistischen Modellen selten auftritt. Dabei werden sogenannte *parallele* Elemente gelöscht. Zwei Elemente $x, y \in (P \cup T)$ heißen *parallel*, wenn sie denselben Vorbereich und denselben Nachbereich mit jeweils derselben Kantenkardinalität besitzen, d.h. es muß $\bullet x = \bullet y$ und $x \bullet = y \bullet$ sowie für alle $z \in \bullet x$ und alle $z' \in x \bullet$ gelten, daß $K(z, x) = K(z, y)$ und $K(x, z') = K(y, z')$. Sind zwei Elemente parallel, so kann eines von beiden gelöscht werden. Sind die beiden Elemente Stellen, so wird die Stelle gelöscht, die initial weniger Marken enthält. Diese Reduktion reduziert wiederum nicht den Zustandsraum des Netzes. Bei der MIMO-Graphen basierten Aggregation werden parallele Elemente nur dann gelöscht, wenn sie die SISO-Eigenschaft besitzen.

Die vierte Reduktionsregel aus [123] betrifft sogenannte *äquivalente* Stellen. Äquivalente Stellen besitzen jeweils genau eine Transition in ihrem Nachbereich. Diese Transitionen besitzen einerseits denselben Vorbereich mit Ausnahme dieser äquivalenten Stellen und andererseits denselben Nachbereich.

Abbildung 5.13 zeigt ein Beispiel für die vierte Reduktionsregel aus [123]. Die Stellen P2 und P3 sind äquivalent. Sie haben jeweils nur eine Transition im Nachbereich (T3 und T4), die wiederum mit Ausnahme der äquivalenten Stellen denselben Vor- und Nachbereich besitzen. Die Reduktionsregel besagt nun, daß ei-

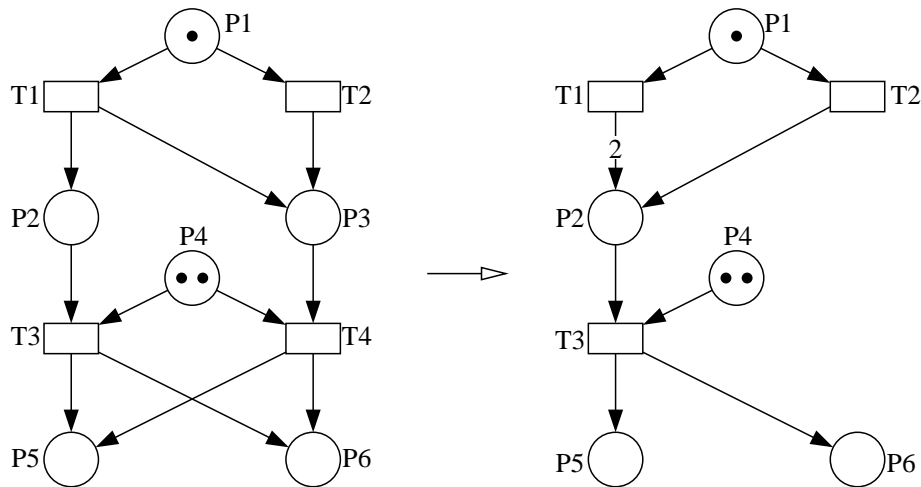


Abbildung 5.13: Vierte Reduktionsregel aus [123]

ne der äquivalenten Stellen und ihre Nachtransition gelöscht werden können, wobei alle Kanten, die auf die gelöschte Stelle zeigten, auf die äquivalente Stelle gelenkt werden. Wenn eine entsprechende Kante bereits existiert (wie im Beispiel die Kante von T1 zu P2), so wird die Kardinalität dieser Kante entsprechend erhöht. Bei der MIMO-Graphen basierten Aggregation werden äquivalente Stellen nicht explizit erkannt. Abbildung 5.13 zeigt ein Beispiel, das durch die generelle Aggregationsregeln der MIMO-Graphen basierten Aggregation nicht aggregiert würde. Das in [123] präsentierte Beispiel für äquivalente Stellen würde dagegen durch die MIMO-Graphen basierte Aggregation aggregiert. Um äquivalente Stellen zu erkennen, müsste jede Stelle des Netzes auf die Existenz einer äquivalenten Stelle hin untersucht werden. Auf eine solche sukzessive Untersuchung des Netzes auf detaillierte strukturelle Eigenschaften wird bei der MIMO-Graphen basierten Aggregation verzichtet.

Für die fünfte und sechste Reduktionsregel aus [123] werden Stellen P identifiziert, deren Transitionen aus dem Nachbereich entweder keine weiteren Stellen im Vorbereich besitzen als P oder deren Transitionen im Vorbereich zu keiner weiteren Transition im Konflikt stehen. Diese Stellen können gelöscht werden und jede Transition aus dem Vorbereich wird mit jeder Transition aus dem Nachbereich von P verschmolzen.

Abbildung 5.14 zeigt die Anwendung der fünften und sechsten Reduktionsregel aus [123]. Die in der Abbildung oben dargestellte Reduktion erfolgt nach Regel fünf. Die Stelle $P2$ erfüllt die Voraussetzungen zur Anwendung dieser Regel, da alle Transitionen aus ihrem Nachbereich ($T2$ und $T3$) keine weiteren Stellen in ihrem Vorbereich besitzen außer $P2$. Die Stelle $P2$ kann demnach gelöscht werden und die Transitionen $T2$ und $T3$ werden jeweils mit der Transition $T1$ verschmolzen. Bei der Anwendung der MIMO-Graphen basierten Aggregation würden

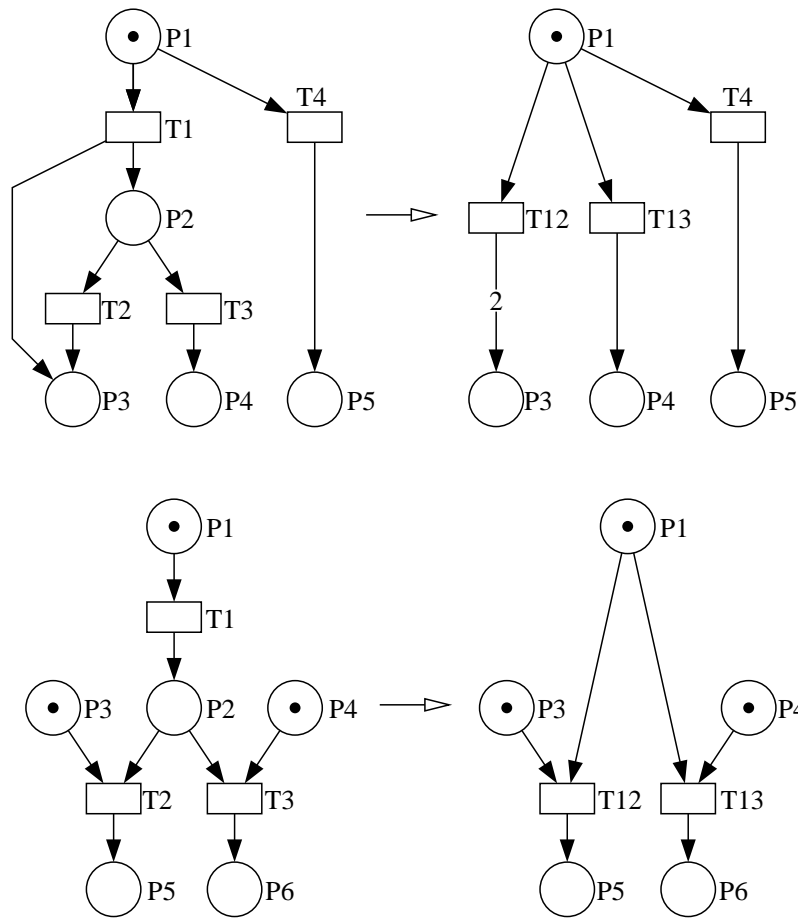


Abbildung 5.14: Fünfte und sechste Reduktionsregel aus [123]

für dieses Modell die Stellen P3 und P4 jeweils mit der Stelle P2 verschmolzen werden, so daß dasselbe aggregierte Modell entstünde wie bei der Anwendung der fünften Reduktionsregel aus [123]. Die in Abbildung 5.14 auf der unteren Hälfte dargestellte Reduktion zeigt die Anwendung der sechsten Reduktionsregel. Die Stelle P2 erfüllt die Anforderungen zur Anwendung dieser Regel, da die Transition T1 aus deren Vorbereich zu keiner anderen Transition in Konflikt steht. Die Stelle P2 kann demnach gelöscht und die Transitionen T2 und T3 jeweils mit der Transition T1 verschmolzen werden. Bei Anwendung der MIMO-Graphen basierten Aggregierung für dieses Modell würden die Stellen P1 und P2 miteinander verschmolzen werden, so daß dasselbe aggregierte Modell entstünde wie bei Anwendung der sechsten Reduktionsregel. Um die Voraussetzungen für die Anwendung der fünften und sechsten Reduktionsregel zu überprüfen, müssen wiederum alle Stellen und deren Transitionen im Vor- und Nachbereich durchsucht werden. Es sei betont, daß es Modelle geben kann, die Stellen mit Eigenschaften zur Anwendbarkeit der fünften oder sechsten Reduktionsregel enthal-

ten, für die die MIMO-Graphen basierte Aggregation jedoch zu keiner Reduktion des Modells führen würde. Es gilt dabei wiederum das bereits für die vierte Aggregierungsregel Gesagte, so daß auf die Diskussion am Ende dieses Abschnittes verwiesen wird.

Die siebte und achte Reduktionsregel aus [123] betreffen Schlingen. Eine Stelle, die nur über Schlingen mit je gleicher Kantenkardinalität mit Transitionen verbunden ist, kann gelöscht werden, wenn sie für die Konzession dieser Transitionen genügend markiert ist. Ebenso kann eine Transition gelöscht werden, wenn sie nur über Schlingen mit je gleicher Kantenkardinalität mit Stellen verbunden ist und diese Stellen weitere Ausgangstransitionen besitzt.

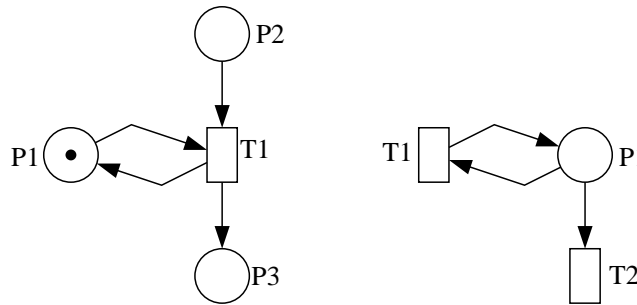


Abbildung 5.15: Siebte und achte Reduktionsregel aus [123]

Abbildung 5.15 zeigt auf der linken Seite ein Modell bei dem die siebte Reduktionsregel durch Löschen von P1 angewendet werden kann. Die rechte Seite in Abbildung 5.15 zeigt ein Modell mit den Voraussetzungen zur Anwendung der achten Reduktionsregel. Die Transition T1 kann gelöscht werden, denn es steht eine Transition T2 in Konflikt zu T1 und es gilt $K(P1, T1) \leq K(P1, T2)$. Weder die Anwendung der siebten als auch die der achten Reduktionsregel führen zur Reduzierung des Zustandsraumes des Modells.

Die neunte und letzte Reduktionsregel aus [123] betrifft Transitionen mit nur einer Stelle im Vorbereich. Diese Transition kann gelöscht und die Stelle aus dem Vorbereich jeweils mit allen Stellen aus dem Nachbereich unter Addition ihrer initialen Markierungen verschmolzen werden.

Abbildung 5.16 zeigt ein aus der Literatur (z.B. [119]) bekanntes Beispiel.

An diesem Beispiel soll die algorithmische Problematik der genannten Reduktionsregeln verdeutlicht werden. Zur Anwendbarkeit der Reduktionsregeln müssen zunächst alle Stellen und alle Transitionen dahingehend überprüft werden, ob sie Bedingungen einzelner Reduktionsregeln erfüllen. Bei Überprüfung des Modells aus Abbildung 5.16 wird daraufhin festgestellt, daß die Stellen P1 und P4 die Voraussetzungen zur Anwendung der fünften Reduktionsregel erfüllen und die Transitionen T3 und T4 jeweils die Voraussetzungen zur Anwendung

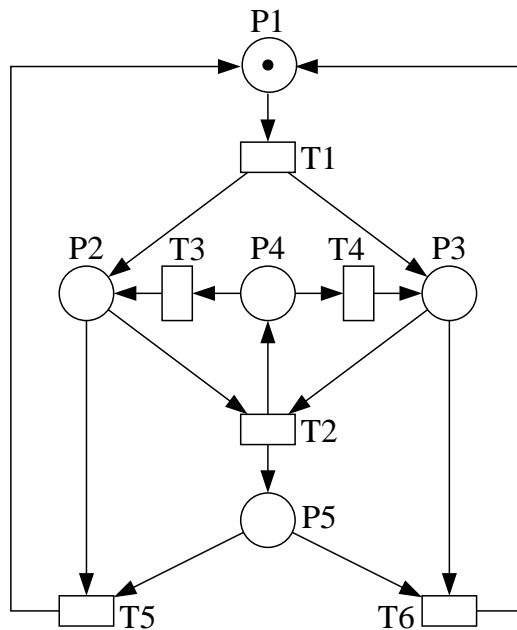


Abbildung 5.16: Beispiel

der neunten Reduktionsregel. Auf die Transitionen T3 und T4 läßt sich jedoch die neunte Reduktionsregel nicht *sukzessive* anwenden, da P4 die einzige Stelle im Vorbereich sowohl von T3 als auch von T4 ist und diese bei der Reduktion gelöscht wird. Tatsächlich müßten zunächst alle neu hinzukommenden Transitionen und ihre Kantenbeziehungen erstellt werden, bevor die betreffenden Stellen und Transitionen gelöscht werden können. Dabei entsteht jedoch das nächste Problem, daß bei der Erstellung der Transitionen geprüft werden muß, ob eine Transition mit derselben Schaltsemantik nicht bereits existiert. Beispielsweise wird bei der Anwendung der neunten Reduktionsregel für Transition T3 und bei der Anwendung der fünften Reduktionsregel für die Stelle P4 jeweils eine neue, der Verschmelzung der Transitionen T3 und T2 entsprechende Transition erzeugt und dann T2, P4 und T3 gelöscht. Wendet man die Reduktionsregel nun derart an, daß zunächst alle neuen Transitionen erzeugt werden, ehe die betreffenden Elemente gelöscht werden, so entsteht die der Verschmelzung der Transitionen T3 und T2 entsprechende Transition T32 doppelt (ebenso T42). Der Vergleich zu den in diesem Abschnitt diskutierten Reduktionsregeln verdeutlicht den Vorteil der algorithmischen Idee der MIMO-Graphen basierten Aggregation. Die Reihenfolge der Anwendungen der generellen Aggregationsregeln der MIMO-Graphen basierten Aggregation ist irrelevant.

Abbildung 5.17 zeigt das nach (paralleler !) Anwendung der genannten Reduktionsregeln und des Entfernens von Transitionen mit derselben Schaltsemantik aus dem Beispiel aus Abbildung 5.16 entstandene Modell.

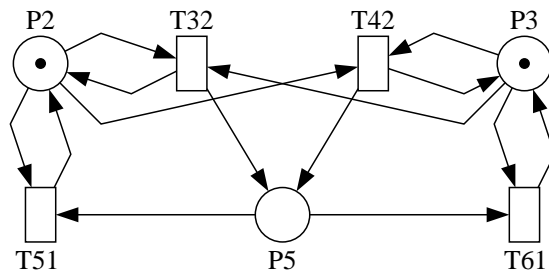


Abbildung 5.17: Beispiel

Dieses reduzierte Modell verdeutlicht ein weiteres Problem bei der Anwendung der angegebenen Reduktionsregeln. Während das Originalmodell ausschließlich lebendige Transitionen enthielt, besitzen die Transitionen T32 und T42 im reduzierten Modell nur in der initialen Markierung Konzession. Nachdem beispielsweise T32 geschaltet hat, besitzt nur die Transition T51 Konzession und kann ein Mal schalten. Danach ist das Netz tot. Schaltet unter der initialen Markierung die Transition T42, kann dann nur die Transition T61 schalten. Nach dem Schalten von T61 ist das Netz ebenfalls tot. Das bedeutet, daß es im reduzierten Modell weder einen Ablauf gibt, in dem alle Transitionen mindestens einmal geschaltet haben, noch ist das Modell reversibel (siehe Seite 28). Wäre das gezeigte Modell ein stochastisches Petrinetz, so ließe sich zwar eine stationäre numerische Analyse des Originalmodells, nicht jedoch des reduzierten Modells durchführen. Gerade eine solche Änderung der Lebendigkeit wird aber bei der MIMO-Graphen basierten Aggregation vermieden.

Implizite Stellen

In [110, 112, 111, 82] wird zur Aggregation des Petrinetzes eine Methode vorgeschlagen, die auf *impliziten Stellen* [39] basiert. Implizite Stellen sind Stellen eines Petrinetzes, die gelöscht werden können, ohne die Abläufe des Netzes zu beeinflussen. Abbildung 5.18 zeigt ein Beispiel für ein Petrinetz mit einer impliziten Stelle.

Die Stelle P2 ist eine implizite Stelle. Die Abläufe des Netzes wären dieselben ohne diese Stelle. P2 ist darüber hinaus sogar eine *markierungsstrukturelle* implizite Stelle (im folgenden mit *MSIP* abgekürzt). MSIPs werden für die Aggregation basierend auf impliziten Stellen verwendet. Während implizite Stellen abhängig von einer bestimmten initialen Markierung eines Petrinetzmodells definiert sind, sind strukturelle implizite Stellen für jede beliebige initiale Markierung eines Petrinetzmodells implizite Stellen. MSIPs sind darüber hinaus strukturelle implizite Stellen, deren Markierung sich aus der Markierung des Originalmodells berechnen läßt. Die Stelle P2 aus Abbildung 5.18 ist eine implizite Stelle für jede beliebige

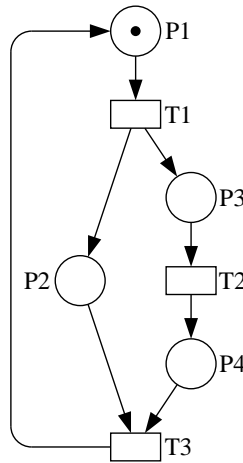


Abbildung 5.18: P2 ist eine implizite Stelle

ge initiale Markierung des dargestellten Petrinetzmodells und somit strukturell. Außerdem läßt sich die Markierung von P2 als die Summe der Markierungen der Stellen P3 und P4 berechnen. P2 ist somit eine markierungsstrukturelle implizite Stelle.

5.7 Definition (MSIP)

Sei $N = (P, T, F, K, m_0)$ ein Petrinetz mit der Inzidenzmatrix N . Eine Stelle $p \neq P$ ist eine markierungsstrukturelle implizite Stelle mit dem Inzidenzvektor n_p genau dann, wenn ein $y \geq 0$ existiert, so daß gilt $n_p = y \cdot N$. Die Menge der Stellen in $\|y\|$ werden die von p implizierten Stellen genannt, wobei $\|y\|$ die Menge der Trägerkomponenten (der von Null verschiedenen Komponenten) von y ist.

Für das in Abbildung 5.18 dargestellte Beispiel gibt es ein y mit $y = (0, 1, 1)$, so daß für die Inzidenzmatrix

$$\begin{array}{rcc}
 & & \begin{array}{ccc} T1 & T2 & T3 \end{array} \\
 \mathbf{N} = & \begin{array}{l} P1 \\ P3 \\ P4 \end{array} & \begin{array}{ccc} -1 & 0 & +1 \\ +1 & -1 & 0 \\ 0 & +1 & -1 \end{array}
 \end{array}$$

gilt $y \cdot N = (+1, 0, -1) = n_{P2}$.

Der Algorithmus zur Berechnung von MSIPs basiert auf der Berechnung minimaler Stellen-Invarianten des Netzes (siehe Abschnitt 3.2) und eines linearen Optimierungsproblems für die Berechnung der initialen Markierung der markierungsstrukturellen impliziten Stellen. In dem im folgenden dargestellten y ausgerechnet. Die Menge H beschreibt die Menge der markierungsstrukturellen impliziten Stellen p_y . Für alle diese Stellen wird mittels eines Optimierungsalgorithmus die initiale Markierung errechnet. Der Operator $\lceil \cdot \rceil$ angewendet auf eine

reelle Zahl v steht für die v nächstgrößere natürliche Zahl. $\mathbf{Pre}[\cdot, t]$ steht für die Spalte der Matrix \mathbf{Pre} , die den Transitionsvektor für t beschreibt.

Eingabe $N = (P, T, F, K, m_0)$ ein Petrinetz mit der Inzidenzmatrix \mathbf{N} und der Pre-matrix \mathbf{Pre}

Ausgabe Menge von MSIPs H mit ihrer initialen Markierung

Beginn

$H = \{p_y \mid p_y \text{ ist eine Stelle mit Inzidenzvektor } \mathbf{n}_{p_y} = \mathbf{y} \cdot \mathbf{N}\}$

für alle $p_y \in H$

$\mathbf{m}_0[p_y] = \lceil \text{Minimum } \mathbf{y} \cdot \mathbf{m}_0 + \mu \rceil$

für $\mathbf{y} \cdot \mathbf{N} = \mathbf{n}_{p_y}$

$\mathbf{y} \cdot \mathbf{Pre}[\cdot, t] + \mu \geq \mathbf{Pre}[p_y, t], \forall t \in p_y^\bullet$

$\mathbf{y} \geq \mathbf{0}, \mathbf{y}[p_y] = 0$

ende für alle

ende

Ein Problem bei der Aggregation unter Verwendung impliziter Stellen besteht darin, daß sogenannte *falsche* Zustände durch ein Hinzufügen impliziter Stellen entstehen können (siehe dazu bspw. [110]). Falsche Zustände sind solche, die zwar nicht im Original, aber im aggregierten Modell auftreten können. Abbildung 5.19 zeigt oben das Originalmodell mit den impliziten Stellen p_{i1} und p_{i2} . In diesem Modell ist kein Zustand erreichbar, indem sowohl p_1 als auch p_4 markiert sind.

In dem in Abbildung 5.19 unten dargestellten aggregierten Modell ist ein solcher Zustand jedoch erreichbar. Ein Algorithmus zum Entfernen falscher Zustände ist ebenfalls in [110] vorgestellt.

Die in [123] vorgestellten Reduktionsregeln sind für die Aggregation stochastischer Modelle nicht anwendbar, da lebendige Modelle in nichtlebendige Modelle aggregiert werden können. Die Aggregation mittels impliziter Stellen ist in dieser Hinsicht per Definition konsistent. Bei der Aggregation mittels impliziter Stellen ist es jedoch nicht möglich, den Einfluß der entfernten zeitbehafteten Transitionen auf das Zeitverhalten des aggregierten Modells zu berücksichtigen. Wird eine Struktur aus Stellen und Transitionen durch eine implizite Stelle ersetzt, wird diese ersetzte Struktur aus dem Petrinetz gelöscht. Die MIMO-Graphen basierte Aggregation ist nicht nur eine konsistente Aggregation, sie ermöglicht auch, das Zeitverhalten der aggregierten zeitbehafteten Transitionen zu berücksichtigen. Ein Vorschlag dieser Berücksichtigung wird im Abschnitt 5.5 vorgestellt.

Die Aggregation mittels impliziter Stellen ist in TimeNET für die Anwendung auf spezielle farbige Petrinetze integriert worden. Im Kapitel 6 wird eine Anwendungsmöglichkeit der MIMO-Graphen basierten Aggregation vorgestellt. Darin wird für eine iterative Analyseverfahren, bei der bisher Teilmodelle mittels

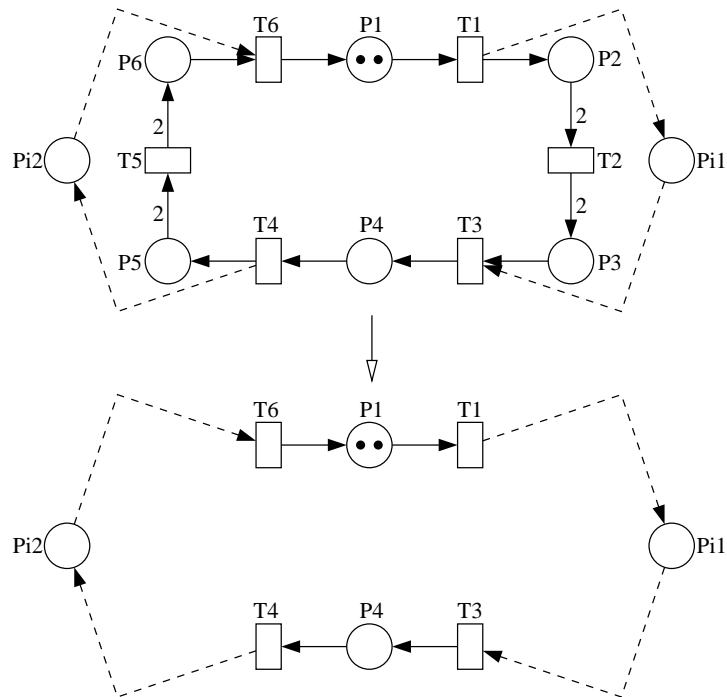


Abbildung 5.19: Falsche Zustände

impliziter Stellen aggregiert wurden, als Aggregierungsmethode die MIMO-Graphen basierte Aggregierung verwendet.

5.3 MIMO-Graphen basierte Aggregierung spezieller farbiger Petrinetze

Die MIMO-Graphen basierte Aggregierung spezieller farbiger Petrinetze (siehe Abschnitt 2.4.1 auf Seite 10) entspricht strukturell der MIMO-Graphen basierten Aggregierung für farblose Petrinetze. Ein entscheidender Unterschied liegt darin, daß auch Transitionen mit nur einer Ein- und Ausgangskante die MIMO-Eigenschaft besitzen können, wenn sie beim Schalten die Farbe einer Marke ändern.

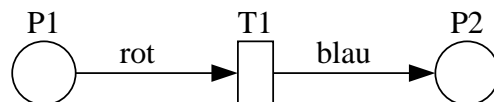


Abbildung 5.20: T1 besitzt MIMO-Eigenschaft

Abbildung 5.20 zeigt eine solche Transition, die von P1 eine rote Marke abzieht

und auf P2 eine blaue Marke legt. Diese Transition besitzt die MIMO-Eigenschaft. Sind einer Transition mehrere Übergänge zugeordnet, so besitzt die Transition die MIMO-Eigenschaft, wenn mindestens einer dieser Übergänge die Farbe einer Marke ändert.

Die grundlegende Erweiterung bei der MIMO-Graphen basierten Aggregation spezieller farbiger Petrinetze zu der von farblosen Petrinetzen liegt in der Verschmelzung von Transitionen.

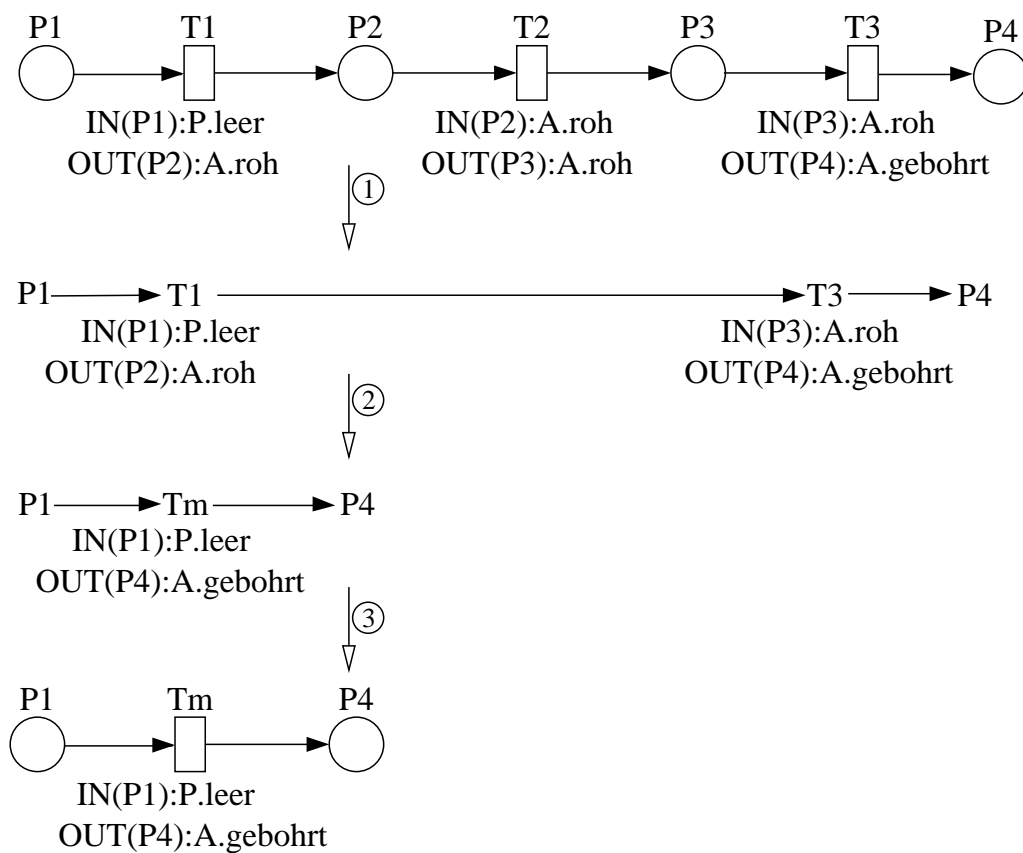


Abbildung 5.21: Beispiel Verschmelzung farbiger Transitionen

Abbildung 5.21 zeigt ein einfaches Beispiel der Verschmelzung. In der Abbildung ist oben das Originalmodell dargestellt. Die Transitionen T1 und T3 besitzen die MIMO-Eigenschaft, denn sie ändern die Farbe von Marken. Unter den Transitionen sind die Übergänge der Transitionen in TimeNET-ähnlicher Notation dargestellt. Die Transition T1 zieht von der Stelle P1 die Stelle P.leer ab und legt auf die Stelle P2 die Marke A.roh ab. Die Transition T2 besitzt nur jeweils eine Ein- und Ausgangskante und ändert weder die Markenanzahl noch die Farbe der Marken. Sie besitzt demnach die SISO-Eigenschaft und wird beim Übergang ① vom Petri-Netz in den MIMO-Graphen ebenso wie die Stellen P2 und P3 gelöscht.

Der MIMO-Graph besteht aus den MIMO-P-Elementen P_1 und P_4 und den MIMO-T-Elementen T_1 und T_3 . Die letztgenannten sind benachbart und können verschmolzen werden, da T_1 keine weitere Ausgangs- und sogar T_3 keine weitere Eingangskante besitzt. Bei der Verschmelzung von MIMO-T-Elementen werden deren Übergänge transitiv verschmolzen. In dem Beispiel aus Abbildung 5.21 wird bei der Aggregation des MIMO-Graphen ② der Übergang von T_1 mit dem von T_3 derart verschmolzen, daß der für T_m dargestellte Übergang entsteht. Zum leichteren Verständnis lassen sich Übergänge in den Eingangs- (IN) und in den Ausgangsteil (OUT) zerlegen und jeweils als Eingangs- oder Ausgangskante interpretieren. Die Verschmelzung erfolgt dann nach der in Abbildung 5.5 dargestellten Regel. Sind Transitionen mehrere Übergänge zugeordnet, so erfolgt die Identifikation der jeweils verschmelzbaren Übergänge über die Farben der Marken.

Die MIMO-Graphen basierte Aggregation spezieller farbiger Petrinetze ist in TimeNET integriert. Werden zwei im MIMO-Graphen benachbarte MIMO-P-Elemente nicht verschmolzen, so wird bei der Rücktransformation des MIMO-Graphen in ein Petrinetz zwischen diesen Stellen eine Transition eingefügt. Um für diese Transition mit SISO-Eigenschaft die korrekte Übergangstabelle erstellen zu können, wurde der Übergang von einem Petrinetz in einen MIMO-Graphen so implementiert, daß die Übergänge von Transitionen den Kanten des MIMO-Graphen zugeordnet sind. Somit wurde eine Lösung des Problems zusätzlich eingefügter Transitionen möglich.

5.4 Teilmodelle: Zerlegung und Aggregation

Eine Anwendungsmöglichkeit der MIMO-Graphen basierten Aggregation wird in Abschnitt 6.2 gezeigt. Die MIMO-Graphen basierte Aggregation findet darin Anwendung bei einem aus der Literatur bekannten iterativen Analyseverfahren [81, 82, 23, 24, 110, 112, 111]. Dieses Verfahren basiert auf der Dekomposition [42] des zu analysierenden Petrinetzmodells. In Dekompositionsverfahren wird das große, zu analysierende Petrinetzmodell in mehrere kleinere Teilmodelle zerlegt und diese unter Einbeziehung ihrer gegenseitigen Abhängigkeiten analysiert. In Abschnitt 5.4.1 wird die Zerlegung großer Modelle in Teilmodelle erläutert.

Ein wichtiger Vorteil der MIMO-Graphen basierten Aggregation besteht darin, daß diese Aggregierungsmethode keine Anforderungen an die Zerlegungsmethode großer Petrinetze stellt. Das bedeutet, daß die Zerlegungsmethode unabhängig von der Aggregierungsmethode ist. Für die meisten aus der Literatur bekannten Verfahren [114, 25, 66, 73, 91, 5, 131, 35] ist das nicht der Fall. Ist die Zerlegung jedoch nicht unabhängig von der späteren Aggregierungsmethode, bedeutet das eine Einschränkung bezüglich der Möglichkeit der Reduzierung des

Zustandsraumes. Denn nur dadurch, daß ein Modell in beliebig viele und somit kleine Teilmodelle zerlegbar ist, kann eine Analyse beliebig großer Gesamtmodelle ermöglicht werden. Im folgenden Abschnitt wird die in dieser Arbeit verwendete Zerlegung beschrieben.

5.4.1 Zerlegung großer Modelle in Teilmodelle

Wie eingangs erwähnt, stellt die MIMO-Graphen basierte Aggregation keine Anforderungen an die Zerlegung der Petrinetzmodelle. Es ist sogar unbedeutend, ob die Modelle an Transitionen oder an Stellen zerlegt werden. Da in dieser Arbeit jedoch nur eine iterative Analysemethode (siehe Abschnitt 6.2) untersucht wird, die eine Zerlegung der Modelle an Stellen bedingt, ist nur diese Zerlegungsmethode Inhalt dieses Abschnittes.

Die Stellen, an denen das Modell zerlegt werden soll, werden *Puffer* genannt. Für die Zerlegung farbloser Modelle genügt es, die Puffer interaktiv zu markieren. Die Zerlegung erfolgt anhand der markierten Puffer automatisch. Zunächst werden alle Kanten, die Puffer mit Transitionen verbinden gelöscht aber gespeichert. Alle verbleibenden zusammenhängende Teile sind *Teilmodelle*. Zu jedem Teilmodell werden die entfernten Kanten und die entsprechenden Puffer wieder hinzugefügt. Danach ist die Zerlegung farbloser Petrinetze abgeschlossen. Es sei an dieser Stelle angemerkt, daß nicht jede beliebige Markierung von Puffern eine sinnvolle Zerlegung des Gesamtmodells zur Folge hat.

Für spezielle farbige Petrinetze erfolgt die Zerlegung sogar vollständig automatisch. Sie ist damit nach Wissen des Autors die einzige vollständig automatische Zerlegungsmethode großer Petrinetzmodelle. Bei speziellen farbigen Petrinetzmodellen kann die Struktur des Modells bei der Zerlegung ausgenutzt werden. Einzelne Komponenten des Modells werden bereits bei der Modellierung als Teilmodelle deklariert, indem sie als Detaillierung einer Vertretertransition modelliert werden. Die Zerlegungsmethode für spezielle farbige Petrinetze nutzt diese Modellierungstechnik aus. Somit ist jedes unter einer Vertretertransition verfeinerte Modell ein Teilmodell. Die Stellen, die mit einer Vertretertransition über Kanten verbunden sind, sind die Puffer. Bei speziellen farbigen Petrinetzen wird das aus dem Strukturmodell und den diesem Modell assoziierten Arbeitsplänen erzeugte Petrinetzmodell zerlegt. Dieses erzeugte Petrinetzmodell hat dieselbe Struktur wie das Strukturmodell, wobei den Transitionen die in den Arbeitsplänen beschriebenen Übergänge in einer Übergangstabelle zugeordnet sind (siehe Abschnitt 2.4.1 oder [132, 136]). Abbildung 5.22 zeigt das bereits aus Abschnitt 2.4.1 bekannte Petrinetzmodell einer Fertigungsanlage.

Wird auf das in Abbildung 5.22 gezeigte Modell die automatische Zerlegung angewendet, so sind alle in dieser Abbildung dargestellten Stellen Puffer. Alle Vertretertransitionen repräsentieren ein Teilmodell. Neben der automatischen Zer-

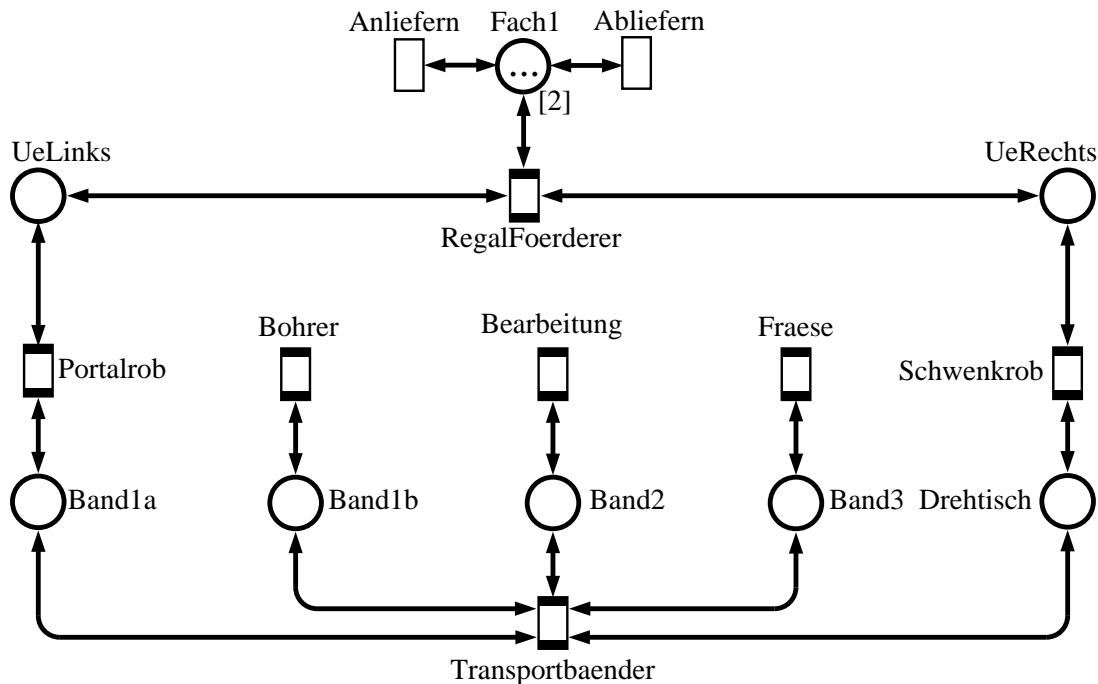


Abbildung 5.22: Petrinetzmodell einer Fertigungsanlage

legung ist darüber hinaus ein Markieren der Puffer durch den Nutzer möglich, um mehrere Teilmodelle zu einem zusammenfassen zu können. Da die Teilmodelle für den Bohrer (siehe Abbildung 6.21 auf Seite 144), die Bearbeitung, die Fräse und die Transportbänder des Modells der Fertigungsanlage aus Abbildung 5.22 bspw. nicht sehr komplex sind, ist eine Zusammenfassung dieser Teilmodelle zu einem Teilmodell geeignet.

Abbildung 5.23 zeigt nochmals das bereits aus Abschnitt 2.4.1 bekannte Teilmodell des Schwenkarmroboters. Die MIMO-Graphen basierte Aggregation eines solchen Teilmodells wird im folgenden Abschnitt erläutert.

5.4.2 Aggregation von Teilmodellen

In Teilnetzen sind einzelne Elemente ausgezeichnet. Diese Elemente stellen die Verbindungen zu anderen Teilnetzen dar. In dem im vorhergehenden Abschnitt 5.4.1 vorgestellten Zerlegungsalgorithmus erfolgt die Interaktion mit anderen Teilnetzen über Stellen, den sogenannten Puffern. Die MIMO-Graphen basierte Aggregation ist nicht auf die Auszeichnung von Stellen beschränkt. Ebenso könnten zu aggregierende Teilnetze auch über Transitionen miteinander interagieren. In diesem Abschnitt wird das Verfahren jedoch auf die hier beschriebene Zerlegung asynchron interagierender Teilmodelle angewendet.

Die Aggregation erfolgt generell wie in Abschnitt 5.1 beschrieben. Für den er-

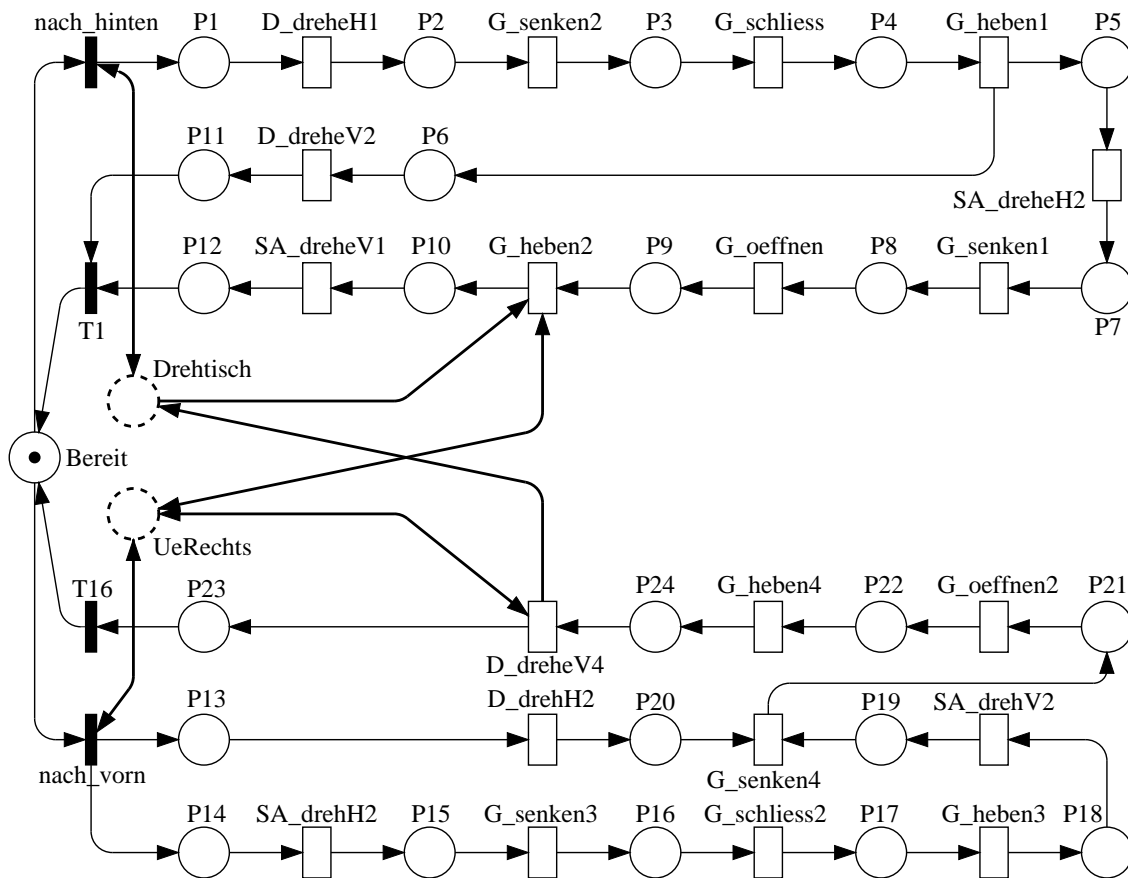


Abbildung 5.23: Verfeinertes Modell des Schwenkarmroboters

sten Schritt der Aggregation, den Übergang von einem Petrinetz in einen MIMO-Graphen, gibt es jedoch folgende Ausnahme: sowohl alle Puffer als auch alle mit Puffern verbundene Transitionen sind MIMO-Elemente! Diese Elemente werden demzufolge bei der Transformation von einem Petrinetz in einen MIMO-Graphen bewahrt. Weitere Unterschiede zum allgemeinen Fall existieren für den ersten Transformationsschritt nicht.

Für den zweiten Schritt, der Aggregation des MIMO-Graphen betrifft die einzige Einschränkung im Gegensatz zur generellen Aggregation ebenfalls mit Puffern verbundene MIMO-T-Elemente. Sie dürfen auch dann nicht gelöscht werden, wenn sie die SISO-Eigenschaft aufweisen. Diese Einschränkung ist darin begründet, daß vermieden werden soll, ein Teilnetz soweit zu aggregieren, daß selbst die Puffer miteinander verschmolzen werden. Eine weitere Einschränkung liegt darin, zeitlose Transitionen repräsentierende MIMO-T-Elemente nicht zu verschmelzen, wenn sie eine Eingangskante von einem Puffer besitzen. Das bedeutet, daß zeitlose Transitionen, die Marken von einem Puffer abziehen, auch nach der Aggregation erhalten bleiben müssen. Diese Einschränkung ist darin begründet, daß eventuell weitere zeitlose Transitionen aus anderen Teilmodellen

Marken von diesem Puffer abziehen. Diese Transitionen hätten immer Priorität, wenn man derart mit Puffern verbundene zeitlose Transitionen in zeitbehaftete verschmilzt. Ein strukturell sehr simples Teilnetz wäre dann einzig durch eine Stelle repräsentiert. Durch die erläuterte Einschränkung bleibt zumindest eine die Puffer verbindende Transition bestehen, die in stochastischen Petrinetzen das Zeitverhalten des Teilnetzes beschreibt. Alle weiteren Aggregierungsregeln gelten, wie in Abschnitt 5.1.4 beschrieben.

Die Rücktransformation eines MIMO-Graphen in ein aggregiertes Teilmodell erfolgt wie für den Standardalgorithmus beschrieben.

5.5 Berechnung der Raten verschmolzener Transitionen

Während die Aggregation der Struktur eines Petrinetzmodells die Voraussetzungen schafft, um sehr große Modelle analysieren zu können, besteht eine weitere, komplexe Aufgabe darin, das Zeitverhalten des aggregierten Modells dem des Originalmodells anzugleichen. In dieser Arbeit erfolgt die Angleichung mit dem Ziel, daß der Durchsatz des aggregierten Modells dem des Originalmodells angenähert entspricht. Vor dieser Arbeit gab es in diesem Bereich nahezu keine Forschungsergebnisse, die eine adäquate Aggregation zeitbehafteter Transitionen bei einem zeitlich äquivalentem Verhalten des aggregierten und des Originalnetzes unterstützte. Eine Berechnung der Schaltrate von T_m in Abhängigkeit der Schalraten der verschmelzenden Transitionen T_1 und T_2 für eine so einfache Aggregation wie die in Abbildung 5.24 dargestellt, war nicht bekannt.

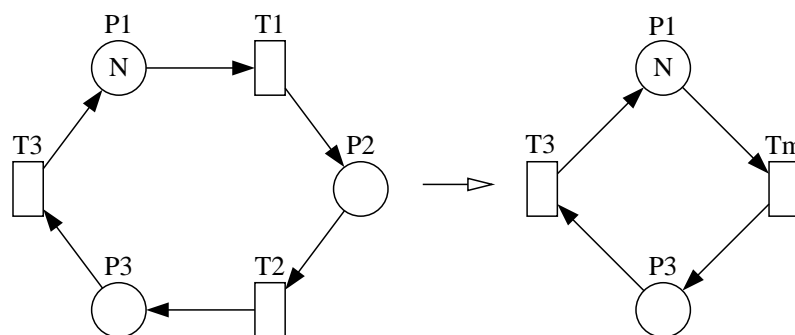


Abbildung 5.24: Beispiel zeitliche Äquivalenz

In Abbildung 5.24 ist auf der linken Seite ein einfaches Petrinetzmodell dargestellt. Die rechte Seite zeigt das aggregierte Modell, in dem die Transitionen T_1 und T_2 des Originalmodells zu der Transition T_m verschmolzen sind. Nehmen

wir an, die Transitionen T_1 , T_2 und T_3 sind zeitbehaftete Transitionen, denen eine negativ exponentielle Schaltzeitverteilung mit den Raten λ_1 , λ_2 , λ_3 zugeordnet ist. Für das aggregierte Petrinetzmodell ist folgende Frage zu beantworten.

Welche Schaltrate muß der Transition T_m zugeordnet werden, so daß bei Erhalt der Schaltrate von T_3 der Erwartungswert der Anzahl der Marken auf P_3 des Originalmodells gleich dem des aggregierten Modells ist?

Eine Antwort auf diese Frage gibt Abschnitt 5.5.2. Zunächst wird der für unsere Untersuchungen gewählte Ansatz beschrieben. Da es keine Forschungsergebnisse im Bereich der strukturbasierten Zuordnung von Schaltraten zu aggregierten zeitbehafteten Transitionen gab und um detaillierte Untersuchungen durchführen zu können, wurde bei der Integration der MIMO-Graphen basierten Aggregation stochastischer Petrinetze in TimeNET folgende Entwicklungsreihenfolge gewählt. Zunächst wurden die verschiedenen Einflüsse auf das Zeitverhalten des Petrinetzes bei der Aggregation identifiziert. Dabei sind drei verschiedene Einflüsse zu unterscheiden.

1. Verschmelzung von MIMO-T-Elementen im MIMO-Graphen.
2. Verschmelzen zweier von einem MIMO-T-Element ausgehenden Kanten.
3. Verschmelzen zweier von einem MIMO-P-Element ausgehenden Kanten.

Aus dieser Unterteilung heraus werden drei Operatoren eingeführt, mit denen der Einfluß auf das Zeitverhalten bei der Aggregation beschrieben wird. Dem in obiger Liste unter Punkt eins aufgeführten Einfluß ist der *Sequenzoperator* \diamond zugeordnet, während dem unter Punkt zwei aufgeführten Einfluß der *Paralleloperator* \parallel zugeordnet ist. Dem unter Punkt drei obiger Liste aufgeführten Einfluß auf das Zeitverhalten des Petrinetzes bei der Aggregation ist der *Alternativoperator* \triangle zugeordnet. Mit diesen drei Operatoren wird in TimeNET während der Aggregation zunächst eine syntaktische Beschreibung der neuen Raten der zeitbehafteten Transitionen erstellt. Unter Verwendung dieser syntaktischen Beschreibungen der Schaltraten wird in einem letzten Entwicklungsschritt eine Semantik der Operatoren entwickelt.

5.5.1 Syntaktische Beschreibung der Schaltraten

In diesem Abschnitt wird gezeigt, wie unter Verwendung des Sequenz-, Parallel- und Alternativoperators die Schaltraten der Transitionen des aggregierten Petrinetzmodells beschrieben werden. Beim Löschen von SISO-Elementen in der ersten Phase der MIMO-Graphen basierten Aggregation, dem Übergang vom

Petrinetz in einen MIMO-Graphen, wird je nach Einbettung des SISO-Elementes in eine bestimmte Struktur einer der drei Operatoren verwendet. Da die Aggregation des MIMO-Graphen ein iterativer Prozeß ist, bei dem durch Verschmelzung von Elementen und Kanten MIMO-Elemente zu SISO-Elementen werden können, wird der selektive Einsatz der Operatoren unter Umständen auch während der Aggregation des MIMO-Graphen nötig. Bei der Verschmelzung von MIMO-T-Elementen findet der Sequenzoperator Anwendung. Beim Übergang vom aggregierten MIMO-Graphen zum aggregierten Petrinetz werden eventuell hinzugefügten Transitionen die entsprechenden Schaltraten zugeordnet. Auf die einzelnen Anwendungsfälle wird in den folgenden Abschnitten detailliert eingegangen.

Der Sequenzoperator

Wie in Abschnitt 5.1 beschrieben, werden beim Übergang vom Petrinetzmodell in einen MIMO-Graphen die Elemente mit SISO-Eigenschaft weggelassen. Ein Entfernen einer zeitbehafteten Transition bewirkt jedoch einen zeitlichen Einfluß in der Art, daß die Marken von dieser Transition nicht mehr zeitlich verzögert werden. Läßt man diese Verzögerung unberücksichtigt, so weist ein aggregiertes Modell einen anderen Durchsatz als das Originalmodell auf. Die Raten der entfernten zeitbehafteten Transitionen müssen demnach bei dem Übergang vom Petrinetz in einen MIMO-Graphen berücksichtigt bleiben. Die Idee bei der Bewahrung der Schaltraten entfernter Transitionen besteht darin, sie mit den Schaltraten der strukturell *folgenden* Transitionen zu verknüpfen. Zur Ermittlung der folgenden Transitionen genügt es jedoch nicht, die Richtungen der Kanten zu verfolgen und die Schaltrate der entfernten Transition mit Hilfe des Sequenzoperators mit der Schaltrate der im Pfad *nächsten* Transition zu verknüpfen. Der Grund dafür liegt darin, daß bspw. die Transition im Originalnetz zwischen zwei Stellen mit MIMO-Eigenschaft liegt und diese MIMO-P-Elemente des MIMO-Graphen die Voraussetzung zur Verschmelzung nicht erfüllen. Zwischen diesen MIMO-P-Elementen muß bei der Rücktransformation des MIMO-Graphen in ein Petrinetz eine Transition eingefügt werden, die die Schaltrate der entfernten Transition zugewiesen bekommt. In diesem Fall fände der Sequenzoperator erst gar keine Anwendung. Abbildung 5.25 zeigt diesen Fall.

Bei der Implementierung der MIMO-Graphen basierten Aggregation werden deshalb die Schaltraten zunächst an Kanten gekoppelt.

Werden jedoch MIMO-P-Elemente des MIMO-Graphen miteinander verschmolzen und ist der die MIMO-P-Elemente verbindenden Kante eine Schaltrate zugeordnet, so wird diese Schaltrate allen Ausgangskanten des verschmolzenen MIMO-P-Elementes zugeordnet. Ist einer Ausgangskante bereits eine Schaltrate zugeordnet, so wird diese mittels Sequenzoperators mit der neuen Schaltrate

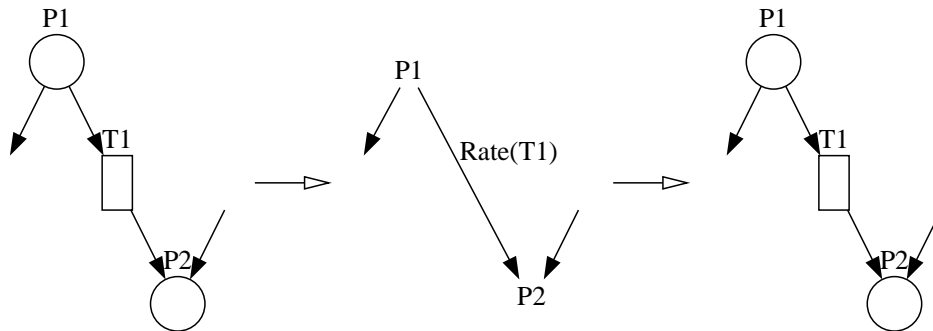


Abbildung 5.25: Zuordnung der Schaltraten zu Kanten

verknüpft. Abbildung 5.26 zeigt die Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-P-Elementen.

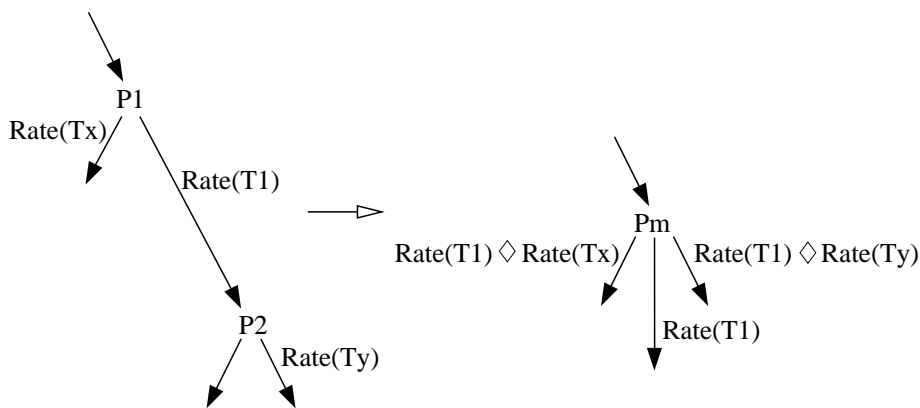


Abbildung 5.26: Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-P-Elementen

Die Zuweisung der Schaltraten an die Kanten führt dazu, daß jede Ausgangskante eines MIMO-T-Elementes des MIMO-Graphen mit einer Schaltrate attribuiert ist. Wird bei der Erstellung des MIMO-Graphen eine Transition mit SISO-Eigenschaft entfernt und führt die Kante, der die Rate der entfernten Transition zugewiesen ist, auf ein MIMO-T-Element, so werden die Raten auf allen Ausgangskanten des MIMO-T-Elementes mittels Sequenzoperator mit der Rate der entfernten Transition verknüpft. Abbildung 5.27 zeigt ein entsprechendes Beispiel.

Nach der Erstellung des MIMO-Graphen können benachbarte MIMO-Elemente desselben Typs verschmolzen werden, wenn sie gewissen Anforderungen genügen (siehe Abschnitt 5.1). Bei der Verschmelzung von MIMO-T-Elementen, deren Ausgangskanten mit einer Schaltrate belegt sind, ist eine Zuordnung einer neuen

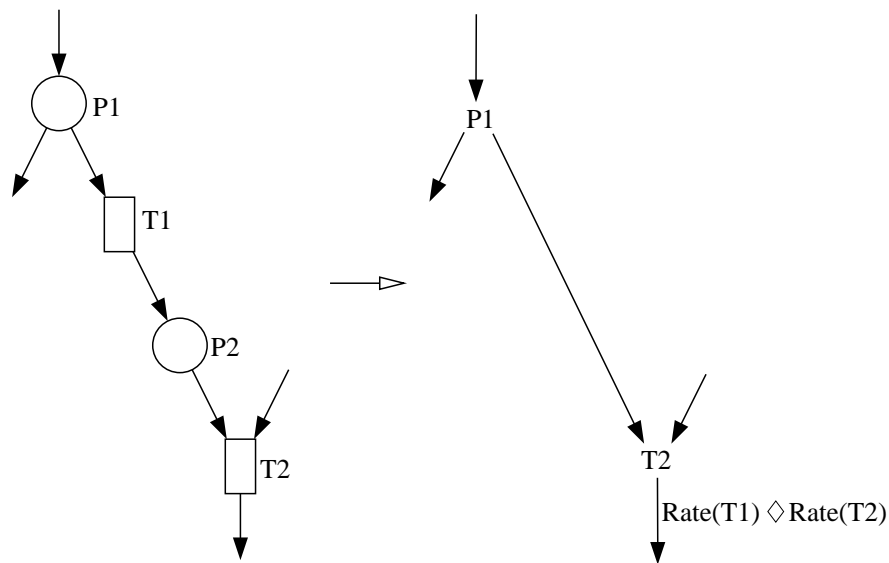


Abbildung 5.27: Anwendung des Sequenzoperators bei folgendem MIMO-T-Element

Schaltrate zum verschmolzenen Element auf der Basis der Schaltraten der verschmolzenen Elemente notwendig. Diese Zuordnung erfolgt ebenfalls mit Hilfe des Sequenzoperators \diamond .

Abbildung 5.28 zeigt eine weitere Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-T-Elementen. Zu beachten ist, daß die $\text{Rate}(T1)$ der Ausgangskante von T1, die nicht zu T2 führt, auch nicht mittels Sequenzoperator mit der $\text{Rate}(T2)$ verknüpft wird. Diese Regel der Anwendung des Sequenzoperators führt zu der wichtigen Eigenschaft, daß die Reihenfolge der Verschmelzungen beliebig ist und stets zu demselben Resultat führen. Abbildung 5.29 zeigt ein Beispiel der mehrfachen Anwendung des Sequenzoperators.

Bei der Rücktransformation des MIMO-Graphen in ein Petrinetz entsteht jedoch ein Fehler, wenn ein MIMO-T-Element mehrere Ausgangskanten besitzt, die mit unterschiedlichen Raten verknüpft sind. In stochastischen Petrinetzen sind die Schaltraten nicht für die Kanten sondern für die Transitionen definiert. Zum jetzigen Zeitpunkt wurde für dieses Problem noch keine Lösung gefunden. Die derzeitige Implementierung ermittelt den Durchschnitt der mit den Ausgangskanten verknüpften Raten. Dieser Durchschnitt wird der Transition zugeordnet. Es ließe sich an dieser Stelle jedoch ein weiterer Operator definieren und unterschiedliche Semantiken dieses Operators untersuchen, um den Approximationsfehler zu minimieren. Eine weitere Quelle des Approximationsfehlers wird im folgenden diskutiert.

Zur Erläuterung einer weiteren Ursache des Approximationsfehlers bei der MIMO-Graphen basierten Aggregation betrachten wir das Beispiel aus Abbildung

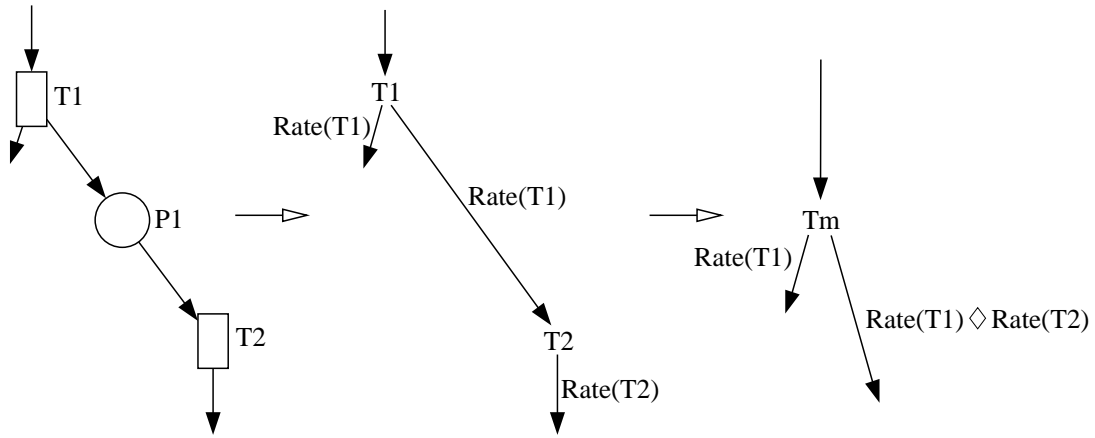


Abbildung 5.28: Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-T-Elementen

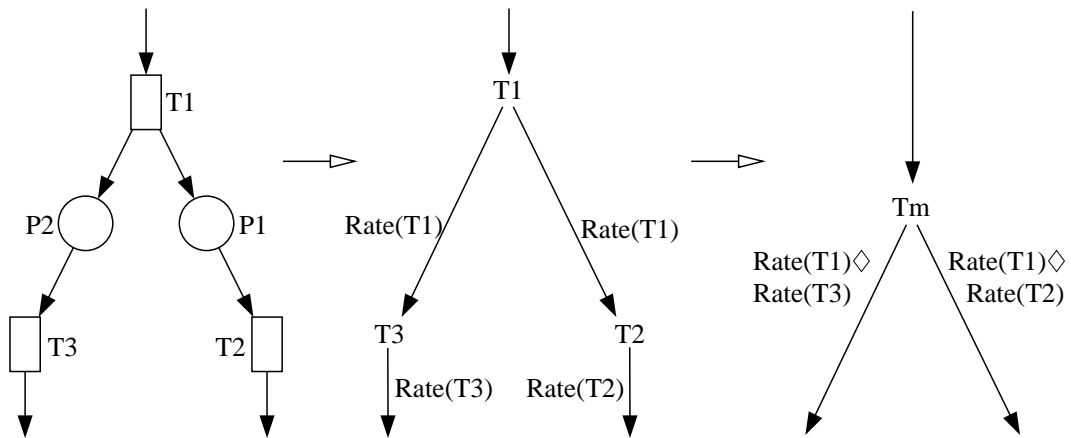


Abbildung 5.29: Anwendung des Sequenzoperators bei mehrfachem Verschmelzen von MIMO-T-Elementen

5.30.

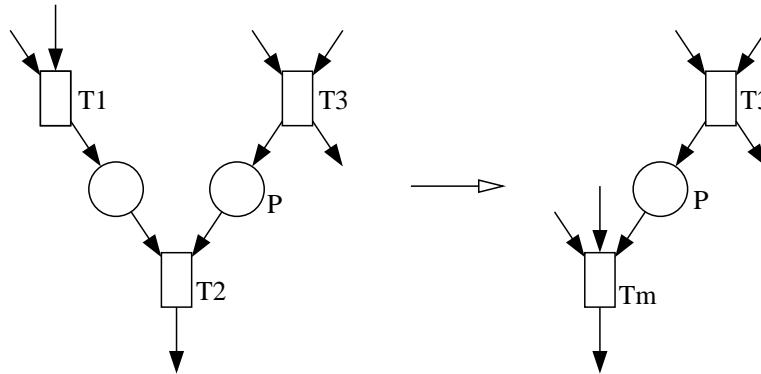


Abbildung 5.30: Ein weiterer Grund für den Approximationsfehler

Abbildung 5.30 zeigt auf der linken Seite den Ausschnitt eines Originalmodells, dessen Transitionen $T1$ und $T2$ zur Transition T_m im auf der rechten Seite dargestellten aggregierten Modell verschmolzen werden. Bei der Berechnung der Schaltzeit der Transition T_m werden die Schaltzeiten der Transitionen $T1$ und $T2$ unter Verwendung des Sequenzoperators verknüpft. Angenommen, das Schalten der Transition $T3$ ist ein sehr seltenes Ereignis im Verhältnis zum Schalten der Transition $T1$ im Originalmodell. Dann besitzt die Transition $T2$ im Originalmodell nur selten Konzession, da P nur selten markiert ist. Ist „endlich“ P markiert, so vergeht eine $T2$ zugeordnete Schaltzeit t_2 . Ist aber P im aggregierten Modell „endlich“ markiert, so vergeht eine T_m zugeordnete Schaltzeit, die im Durchschnitt größer ist als t_2 . Die Anwendung des Sequenzoperators ist per Definition zu allgemein, um für alle Anwendungsfälle eine exakte Berechnung der Schaltzeit verschmolzener Transitionen zu gewährleisten.

Der Alternativoperator

Der Alternativoperator Δ findet seine Anwendung bei der Verschmelzung von Schaltraten zugeordneten Kanten, die von einem MIMO-P-Element ausgehen. Abbildung 5.31 zeigt ein Beispiel der Anwendung des Alternativoperators.

In dem in Abbildung 5.31 gezeigten Beispiel werden beim Übergang vom Petrietz in einen MIMO-Graphen die Transitionen $T1$ und $T2$ aufgrund ihrer jeweiligen SISO-Eigenschaft entfernt und die beiden Kanten zwischen $P1$ und $P2$ miteinander verschmolzen. Die Schaltraten der beiden entfernten Transitionen werden mittels Alternativoperators verknüpft und der verschmolzenen Kante zugeordnet. Die verknüpften Schaltraten werden auch dann mit der verschmolzenen Kante verknüpft, wenn diese Kante zu einem MIMO-T-Element führt. Beim

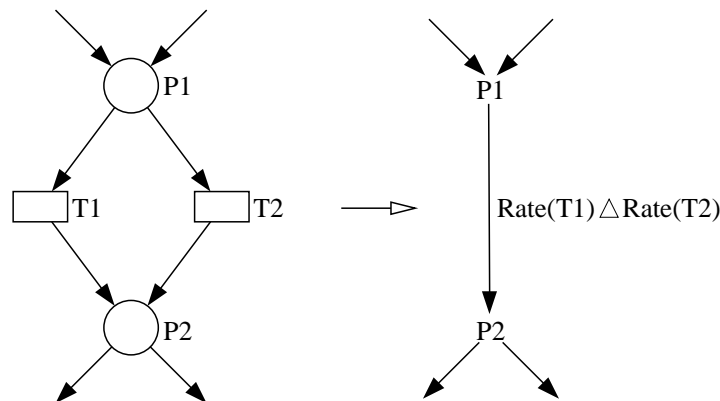


Abbildung 5.31: Anwendung des Alternativoperators

Übergang vom MIMO-Graphen in ein Petrinetz wird die mittels Alternativoperator verknüpfte Kante unter Verwendung des Sequenzoperators mit der Schaltrate dieses MIMO-Elementes verknüpft und der dem MIMO-Element entsprechenden Transition zugeordnet.

Der Paralleloperator

Werden Kanten miteinander verschmolzen, die von einem MIMO-T-Element ausgehen, wird der Paralleloperator verwendet, um den zeitlichen Einfluß der entfernten SISO-T-Elemente zu repräsentieren. Die Verknüpfung mittels Paralleloperators erfolgt stets nach dem MIMO-Element, zu dem zu verschmelzende Kanten hinführen. Abbildung 5.32 zeigt die Anwendung des Paralleloperators für den Fall, daß die Kanten zu einem MIMO-T-Element führen.

Für den in Abbildung 5.32 dargestellten Fall werden die Raten von T2 und T3 mittels Paralleloperator und dieser Term mittels Sequenzoperator mit der Rate von T4 vernüpft. Werden mehr als zwei Kanten miteinander verschmolzen, so werden zunächst alle mit diesen Kanten verbundenen Schaltraten unter Verwendung des Paralleloperators verknüpft und anschließend der Sequenzoperator verwendet.

Alle drei hier beschriebenen Operatoren können beim Löschen von SISO-Elementen Anwendung finden. Die jeweilige Anwendbarkeit hängt von der Struktur ab, in der die zu löschenden SISO-Elemente eingebettet sind. Ein großer Vorteil der MIMO-Graphen basierten Aggregation liegt hierbei darin, daß eine umfangreiche Analyse der Struktur unnötig ist. Wird ein SISO-T-Element gelöscht, so attribuiert man die verbleibende Kante mit der Rate der gelöschten Transition. Wird die Kante mit keiner weiteren Kante verschmolzen, so wird diese Rate mittels Sequenzoperator mit einer eventuell bereits an dieser Kante attribuierten Rate

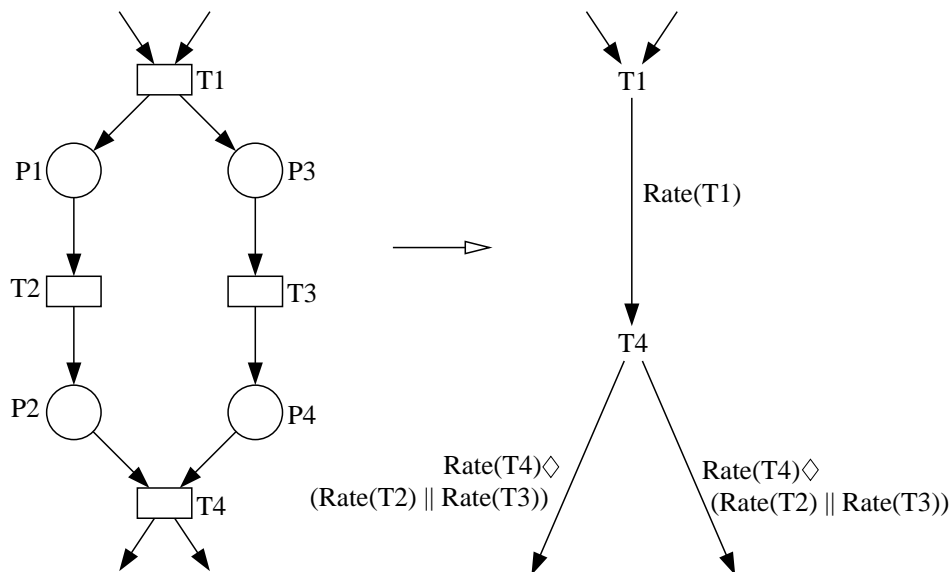


Abbildung 5.32: Spezielle Anwendung des Paralleloperators

verknüpft. Wird die Kante jedoch mit einer weiteren Kante verknüpft, so muß nur der Typ des MIMO-Elementes bestimmt werden, von dem diese Kante ausgeht. Ist dieses Element ein MIMO-P-Element, so werden die Raten mittels des Alternativoperators verknüpft, im Falle eines MIMO-T-Elementes mittels des im nächsten Abschnitt beschriebenen Paralleloperators.

5.5.2 Semantik der Operatoren zur Berechnung der Schaltraten

In diesem Abschnitt wird den in Abschnitt 5.5.1 beschriebenen Operatoren zur Berechnung der Schaltraten verschmolzener Operatoren eine Semantik zugeordnet. Die Bestimmung einer geeigneten Semantik der einzelnen Operatoren, so daß der Durchsatz des aggregierten Petrinetzes dem des Originalnetzes entspricht, ist eine sehr komplexe Aufgabe, die derzeit noch nicht vollständig gelöst ist. Die Komplexität dieser Aufgabe ist schon dadurch ersichtlich, daß vor dieser Arbeit selbst für ein so einfaches Modell wie das in Abbildung 5.24 auf Seite 95 dargestellte keine geschlossene Formel zur Berechnung der Schaltrate von T_m auf Basis der Schaltraten von T_1 und T_2 existiert. In diesem Abschnitt wird nicht nur eine Formel zur Berechnung der Schaltrate von T_m für ein beliebiges N des in Abbildung 5.24 dargestellten Modells vorgestellt, sondern darüber hinaus für die Verschmelzung von n in einer Sequenz angeordneten Transitionen.

Allen im folgenden betrachteten zeitbehafteten Transitionen ist eine negativ exponentielle Schaltzeit zugeordnet. Eine geeignete Semantik der Operatoren für die Verschmelzung zeitbehafteter Transitionen mit allgemeiner Schaltzeitvertei-

lung existiert derzeit nicht.

Der Sequenzoperator

In diesem Abschnitt wird die Semantik des Sequenzoperators vorgestellt. Darüber hinaus wird die von uns gewählte Vorgehensweise zur Ermittlung der Semantik erläutert. Zur Ermittlung der Semantik des Sequenzoperators sind wir von einem einfachen Modell ausgegangen, in dem zunächst zwei in Sequenz angeordnete Transitionen $T1$ und $T2$ zu einer verschmolzen werden. Abbildung 5.33 zeigt das bereits in Abbildung 5.24 vorgestellte Modell und die Anwendung des Sequenzoperators.

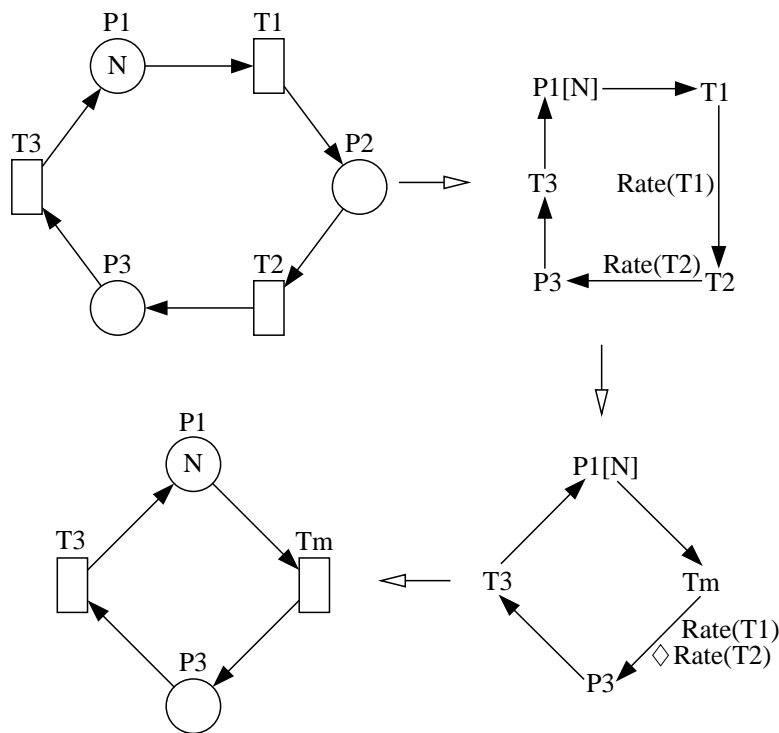


Abbildung 5.33: Referenzmodell zur Ermittlung des Sequenzoperators

Der Transition T_m ist demnach die Rate $Rate(T1) \diamond Rate(T2)$ zugeordnet. Es soll die Frage beantwortet werden, welcher Wert dieser Rate zugeordnet werden muß, damit im Original- und im aggregierten Modell die Erwartungswerte für die Anzahl der Marken auf den Stellen $P3$ identisch sind unter der Annahme, daß die Schaltrate der Transition $T3$ in beiden Modellen gleich ist.

Wir untersuchen diese Fragestellung zunächst für $N = 1$.

Abbildung 5.34 zeigt den Erreichbarkeitsgraphen des Originalmodells aus Abbildung 5.33 auf der linken und den des aggregierten Modells auf der rechten

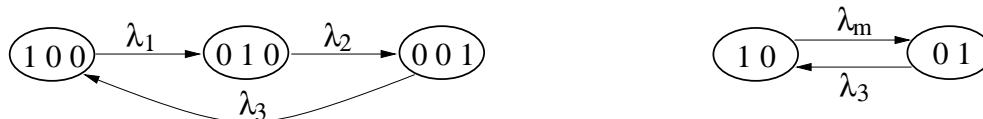


Abbildung 5.34: Erreichbarkeitsgraphen des Original- und des aggregierten Modells

Seite für jeweils $N = 1$. Die die Zustandsübergänge beschreibenden Kanten sind darin bereits mit den Schaltraten der entsprechenden Transitionen beschriftet. Die Zustände werden mittels der Anzahl der Marken auf den einzelnen Stellen beschrieben. Der Zustand 100 steht bspw. für die Markierung des Originalmodells, in der die Stelle $P1$ markiert und die Stellen $P2$ und $P3$ unmarkiert sind. Die Schaltrate für die Transition T_m ist durch λ_m beschrieben und die restlichen Schaltraten sind dem Index entsprechend den Transitionen zugeordnet. Aus dem Erreichbarkeitsgraphen lassen sich folgende Gleichungssysteme zur Berechnung der stationären Zustandswahrscheinlichkeiten aufstellen:

$$\begin{aligned}\lambda_1 \pi_{100} &= \lambda_3 \pi_{001} \\ \lambda_1 \pi_{100} &= \lambda_2 \pi_{010} \\ \pi_{100} + \pi_{010} + \pi_{001} &= 1\end{aligned}$$

für das Originalmodell, wobei bspw. π_{100} die stationäre Zustandswahrscheinlichkeit des Zustandes 100 beschreibt. Für das aggregierte Modell lautet das entsprechende Gleichungssystem

$$\begin{aligned}\lambda_m \pi_{10} &= \lambda_3 \pi_{01} \\ \pi_{10} + \pi_{01} &= 1\end{aligned}$$

wobei ebenfalls π zur Beschreibung der stationären Zustandswahrscheinlichkeit verwendet wird, da sich die einzelnen Zustandswahrscheinlichkeiten durch die Stelligkeit des Index mit denen des Originalmodells unterscheiden lassen. Der Erwartungswert der Anzahl der Stellen auf $P3$ im Originalmodell ist gleich π_{001} , während der des aggregierten Modells gleich π_{01} ist. Wegen

$$\pi_{001} = \frac{1}{\frac{\lambda_3}{\lambda_1} + \frac{\lambda_3}{\lambda_2} + 1}$$

und

$$\pi_{01} = \frac{\lambda_m}{\lambda_3 + \lambda_m}$$

folgt aus $\pi_{001} = \pi_{01}$

$$\lambda_m = \frac{1}{\frac{1}{\lambda_1} + \frac{1}{\lambda_2}}$$

Das heißt, daß

$$\frac{1}{\lambda_m} = \frac{1}{\lambda_1} + \frac{1}{\lambda_2}.$$

Das Reziproke der Schaltraten wird auch als *Schaltzeit* bezeichnet. Dieses Ergebnis für $N = 1$ war bereits vor dieser Arbeit bekannt. Im folgenden wird diskutiert, wie sich λ_m für beliebige $N > 0$ beschreiben läßt.

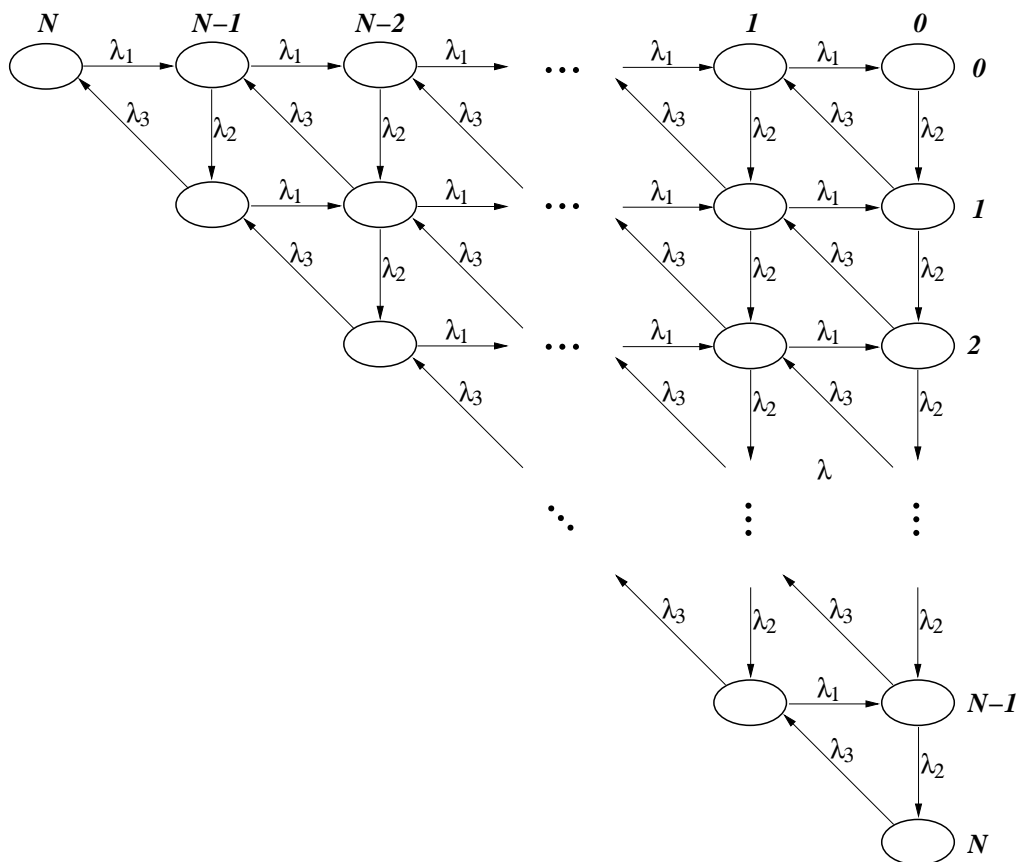


Abbildung 5.35: Erreichbarkeitsgraphen des Originalmodells für beliebiges N

Abbildung 5.35 zeigt den Erreichbarkeitsgraphen für beliebiges N . Auf die Zustandsinschriften wurde dabei aus Platzgründen verzichtet. Stattdessen geben die fettgedruckten Werte in der obersten Reihe die Anzahl der Marken in der Stelle P_1 an, die für die Markierung aller Zustände in der jeweils darunter liegenden Spalte gilt. Die fettgedruckten Werte am rechten Rand geben die Anzahl der Marken in der Stelle P_3 an, die für die Markierung aller Zustände in der jeweils nebenliegenden Zeile gilt. Wegen der mit der initialen Markierung gewichteten Stellen-Invariante $P_1 + P_2 + P_3 = N$ läßt sich daraus die Anzahl der Marken auf P_2 für jeden Zustand berechnen. Der Zustand in der linken oberen Ecke wird

demnach mittels $N00$ beschrieben, während der Zustand in der rechten oberen Ecke mit $0N0$ und der in der linken unteren Ecke mit $00N$ beschrieben wird. Um keine Mehrdeutigkeiten zu erlauben, werden im folgenden die Markenanzahlen pro Stelle durch Komma getrennt, so daß wir bspw. $N, 0, 0$ schreiben. Aus dem Erreichbarkeitsgraphen läßt sich das Gleichungssystem zur Berechnung der stationären Zustandswahrscheinlichkeiten π erstellen:

$$\begin{aligned}\lambda_1 \pi_{N,0,0} &= \lambda_3 \pi_{N-1,0,1} \\ \lambda_1 \pi_{1,N-1,0} &= \lambda_2 \pi_{0,N,0} \\ \lambda_2 \pi_{0,1,N-1} &= \lambda_3 \pi_{0,0,N}\end{aligned}$$

und für alle $x = 0, \dots, N-2$:

$$\begin{aligned}\lambda_1 \pi_{N-x,x,0} + \lambda_3 \pi_{N-2-x,x+1,1} &= (\lambda_1 + \lambda_2) \pi_{N-1-x,x+1,0} \\ \lambda_2 \pi_{N-1-x,1,x} + \lambda_3 \pi_{N-2-x,0,x+2} &= (\lambda_1 + \lambda_3) \pi_{N-1-x,0,x+1} \\ \lambda_2 \pi_{0,N-x,x} + \lambda_1 \pi_{1,N-2-x,x+1} &= (\lambda_2 + \lambda_3) \pi_{0,N-1-x,x+1}\end{aligned}$$

und für alle $x = 2, \dots, N-1$ und alle $y = 0, \dots, x-2$:

$$\begin{aligned}\lambda_2 \pi_{N-x,x-y,y} + \lambda_1 \pi_{N-(x-1),x-y-2,y+1} + \lambda_3 \pi_{N-(x+1),x-y-1,y+2} \\ = (\lambda_1 + \lambda_2 + \lambda_3) \pi_{N-x,x-y-1,y+1}\end{aligned}$$

Für das aggregierte Modell läßt sich für ein beliebiges $N > 0$ das folgende Gleichungssystem zur Berechnung der stationären Zustandswahrscheinlichkeit π erstellen:

$$\begin{aligned}\lambda_m \pi_{N,0} &= \lambda_3 \pi_{N-1,1} \\ \lambda_m \pi_{1,N-1} &= \lambda_3 \pi_{0,N}\end{aligned}$$

für alle $x = 0, \dots, N-2$:

$$\lambda_m \pi_{N-x,x} + \lambda_3 \pi_{N-2-x,x+2} = (\lambda_m + \lambda_3) \pi_{N-1-x,x+1}$$

Die Summe der einzelnen Zustandswahrscheinlichkeiten für jedes Modell ist darüber hinaus gleich eins.

Der Erwartungswert der Anzahl der Marken auf der Stelle P3 berechnet sich für das Originalmodell aus $\sum_{x=0}^N \sum_{y=0}^x y \cdot \pi_{N-x,x-y,y}$ und für das aggregierte Modell aus

$$\sum_{x=0}^N x \cdot \pi_{N-x,x}.$$

Nicht nur, daß das Gleichungssystem für das Originalmodell ab $N = 4$ von zu hoher Ordnung ist, um es symbolisch zu lösen, es gilt bereits ab $N = 2$ eine wichtige Eigenschaft nicht mehr, die noch für $N = 1$ galt: Die Schaltrate von T_m läßt sich nicht mehr nur aus den Schaltraten von T_1 und T_2 ausdrücken. Sie ist vielmehr auch von der Schaltrate von T_3 (und der Anzahl der Marken N im Petri-netz) abhängig. Diese Abhängigkeitseigenschaft von der „Umgebung“ des zu aggregierenden Teils des Petri-netzes würde einen unvermeidbaren Aufwand bei der

Aggregation bedeuten und würde zur Unbrauchbarkeit der Aggregierungsmethode führen, denn es wäre eine qualitative Analyse des Netzes notwendig, die eine Erstellung des vollständigen Erreichbarkeitsgraphen des Originalmodells erfordern würde. Zum Beispiel müßte ein k für jede Stelle des Originalmodells ermittelt werden, das die k -Beschränktheit der Stelle beschreibt, um den oben genannten Einfluß des N in die Berechnung der Schaltrate für T_m berücksichtigen zu können. Eine geschlossene Form für eine feste Schaltrate von T_m , die nur die Schaltraten der Transitionen T_1 und T_2 berücksichtigt, gibt es demnach nicht!

Die Lösung des Problems wird jedoch durch die Verwendung markierungsabhängiger Schaltraten für T_m ermöglicht. Abbildung 5.36 zeigt den Erreichbarkeitsgraphen des aggregierten Modells für ein beliebiges $N > 0$, in dem die Zustandsübergänge bereits mit den Schaltraten der jeweiligen Übergänge beschriftet sind.

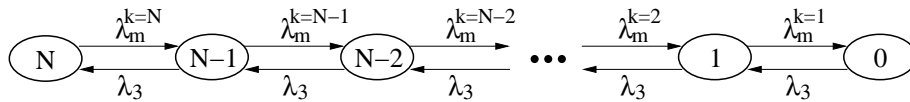


Abbildung 5.36: Erreichbarkeitsgraphen des aggregierten Modells für beliebiges N und markierungsabhängige Schaltraten

Die Zustände in Abbildung 5.36 sind jeweils nur mit der Anzahl der Marken in P_1 beschriftet. Wegen der mit der Anzahl der Marken der initialen Markierung gewichteten Stellen-Invariante $P_1 + P_3 = N$ für das aggregierte Modell läßt sich die Anzahl der Marken auf P_3 für jeden Zustand berechnen. Die Schaltraten für die Transition T_m sind abhängig von der Anzahl der Marken in P_1 . Sind bspw. drei Marken in P_1 , so schaltet T_m mit der Schaltrate $\lambda_m^{k=3}$.

Verwendet man markierungsabhängige Schaltraten für T_m , so ergibt sich für

$$\lambda_m^{k=x} = \begin{cases} \frac{\frac{1}{\lambda_1^x} - \frac{1}{\lambda_2^x}}{\frac{1}{\lambda_1^{x+1}} - \frac{1}{\lambda_2^{x+1}}} = \lambda_1 \lambda_2 \frac{\lambda_2^x - \lambda_1^x}{\lambda_2^{x+1} - \lambda_1^{x+1}} & \text{für } \lambda_1 \neq \lambda_2 \\ \frac{x \lambda_1}{1+x} & \text{für } \lambda_1 = \lambda_2. \end{cases}$$

Setzt man in obige Formel $k = 1$, so erhält man für T_m die für $N = 1$ berechnete Schaltrate.

Die Vorgehensweise zur Bestimmung dieser und auch der folgenden Formeln wird am Beispiel des Referenzmodells zur Bestimmung des Sequenzoperators aus Abbildung 5.33 erläutert. Zunächst wurde der Erwartungswert der Anzahl der Marken auf P_3 für konkrete Schaltratenbelegungen der Transitionen des Originalmodells mit Hilfe von *TimeNET* berechnet. Für das aggregierte Modell wurden dann markierungsabhängige Werte für die Schaltrate von T_m derart gesucht,

daß der Erwartungswert der Anzahl der Marken auf P3 im aggregierten Modell gleich dem des Originalmodells ist. Aus den ermittelten Werten der markierungsabhängigen Schaltzeiten von Tm wurden Gesetzmäßigkeiten entwickelt, die in obigen Formeln resultierten.

Im folgenden werden anstelle der Formeln für die Schaltzeiten λ die der Schaltzeiten $d = \frac{1}{\lambda}$ angegeben, da in TimeNET zur Attribuierung zeitbehafteter Transitionen ebenfalls die Schaltzeit verwendet wird. Oben genannte Formel läßt sich danach wie folgt umstellen:

$$d_m^{k=x} = \begin{cases} \frac{d_1^{x+1} - d_2^{x+1}}{d_1^x - d_2^x} & \text{für } d_1 \neq d_2 \\ \frac{x+1}{x} d_1 & \text{für } d_1 = d_2. \end{cases} = d_1 \frac{1}{\sum_{i=0}^{x-1} \left(\frac{d_2}{d_1}\right)^i} + d_2$$

Der hintere Term in obiger Formel gibt eine Möglichkeit zur Berechnung der Schaltzeit ohne Fallunterscheidung an. Diese Formel wird im folgenden bei der Verallgemeinerung der Anwendung des Sequenzoperators für $n(> 1)$ aggregierte Transitionen verwendet. Abbildung 5.37 zeigt die Verallgemeinerung der Aggregation sequentieller Transitionen. Die Transitionen T1 bis Tn werden zu Tm aggregiert.

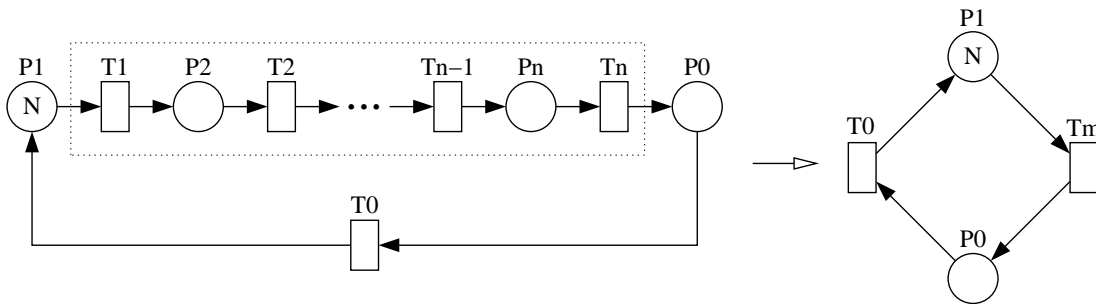


Abbildung 5.37: Aggregation von n sequentiellen Transitionen

Gesucht werden die markierungsabhängigen Schaltzeiten für Tm, so daß der Erwartungswert für die Anzahl der Marken auf P0 im aggregierten Modell gleich dem des Originalmodells ist. Die Entwicklung der Formel wird im folgenden für 2, ..., n aggregierte Transitionen vorgestellt:

$$d_m^{k=x}(d_1, d_2) = d_1 \cdot \frac{1}{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}} + d_2$$

$$d_m^{k=x}(d_1, d_2, d_3) = \left(d_1 \cdot \frac{1}{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}} + d_2 \right) \cdot \frac{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}}{\sum_{i_1=0}^{x-1} \left(\left(\frac{d_2}{d_1}\right)^{i_1} \sum_{i_2=0}^{x-i_1-1} \left(\frac{d_3}{d_1}\right)^{i_2} \right)} + d_3$$

$$d_m^{k=x}(d_1, d_2, d_3, d_4) = \left(\left(d_1 \cdot \frac{1}{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}} + d_2 \right) \cdot \frac{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}}{\sum_{i_2=0}^{x-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right)} + d_3 \right) \cdot \frac{\sum_{i_2=0}^{x-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right)}{\sum_{i_3=0}^{x-1} \left(\left(\frac{d_4}{d_1}\right)^{i_3} \sum_{i_2=0}^{x-i_3-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_3-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right) \right)} + d_4$$

⋮

$$d_m^{k=x}(d_1, \dots, d_n) = \left(\left(\left(\left(d_1 \cdot z_1 + d_2 \right) \cdot z_2 + d_3 \right) \cdot z_3 + d_4 \right) \dots \right) \cdot z_{n-1} + d_n$$

$$= \sum_{j=1}^{n-1} \left(d_j \cdot \prod_{l=j}^{n-1} z_l \right) + d_n$$

Für zwei aufeinanderfolgende Faktoren z_i und z_{i+1} kürzt sich der Nenner von z_i und der Zähler von z_{i+1} . Wir zeigen dieses Kürzen in $\prod_{l=j}^{n-1} z_l$ zunächst für $j = 1$:

$$\prod_{l=1}^{n-1} z_l = \frac{1}{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}} \cdot \frac{\sum_{i_1=0}^{x-1} \left(\frac{d_2}{d_1}\right)^{i_1}}{\sum_{i_2=0}^{x-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right)} \cdot \frac{\sum_{i_2=0}^{x-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right)}{\sum_{i_3=0}^{x-1} \left(\left(\frac{d_4}{d_1}\right)^{i_3} \sum_{i_2=0}^{x-i_3-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-i_3-i_2-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right) \right)}$$

$$\dots \cdot \frac{\sum_{i_{n-2}=0}^{x-1} \left(\left(\frac{d_{n-1}}{d_1}\right)^{i_{n-2}} \sum_{i_{n-3}=0}^{x-i_{n-2}-1} \left(\left(\frac{d_{n-2}}{d_1}\right)^{i_{n-3}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^{n-2} i_y-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^{n-2} i_y-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right) \right) \dots \right)}{\sum_{i_{n-1}=0}^{x-1} \left(\left(\frac{d_n}{d_1}\right)^{i_{n-1}} \sum_{i_{n-2}=0}^{x-i_{n-1}-1} \left(\left(\frac{d_{n-1}}{d_1}\right)^{i_{n-2}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^{n-1} i_y-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^{n-1} i_y-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right) \right) \dots \right)}$$

$$= \frac{1}{\sum_{i_{n-1}=0}^{x-1} \left(\left(\frac{d_n}{d_1}\right)^{i_{n-1}} \sum_{i_{n-2}=0}^{x-i_{n-1}-1} \left(\left(\frac{d_{n-1}}{d_1}\right)^{i_{n-2}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^{n-1} i_y-1} \left(\left(\frac{d_3}{d_1}\right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^{n-1} i_y-1} \left(\frac{d_2}{d_1}\right)^{i_1} \right) \right) \dots \right)}$$

Allgemeiner gilt für ein beliebiges $j = \{1, \dots, n-1\}$:

$$\prod_{l=j}^{n-1} z_l = \frac{\sum_{i_j=0}^{x-1} \left(\left(\frac{d_{j+1}}{d_1} \right)^{i_j} \sum_{i_{j-1}=0}^{x-i_j-1} \left(\left(\frac{d_j}{d_1} \right)^{i_{j-1}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^j i_y-1} \left(\left(\frac{d_3}{d_1} \right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^j i_y-1} \left(\left(\frac{d_2}{d_1} \right)^{i_1} \right) \right) \dots \right) \right)}{\sum_{i_{n-1}=0}^{x-1} \left(\left(\frac{d_n}{d_1} \right)^{i_{n-1}} \sum_{i_{n-2}=0}^{x-i_{n-1}-1} \left(\left(\frac{d_{n-1}}{d_1} \right)^{i_{n-2}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^{n-1} i_y-1} \left(\left(\frac{d_3}{d_1} \right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^{n-1} i_y-1} \left(\left(\frac{d_2}{d_1} \right)^{i_1} \right) \right) \dots \right) \right)}$$

Da der Nenner in dem Term für $\prod_{l=j}^{n-1} z_l$ unabhängig von dem Summenzähler j aus

$d_m^{k=x}(d_1, \dots, d_n) = \sum_{j=1}^{n-1} \left(d_j \cdot \prod_{l=j}^{n-1} z_l \right) + d_n$ ist, läßt er sich aus der Summe ausklammern. Daraus folgt:

$$d_m^{k=x}(d_1, \dots, d_n) =$$

$$\frac{1}{\sum_{i_{n-1}=0}^{x-1} \left(\left(\frac{d_n}{d_1} \right)^{i_{n-1}} \sum_{i_{n-2}=0}^{x-i_{n-1}-1} \left(\left(\frac{d_{n-1}}{d_1} \right)^{i_{n-2}} \dots \sum_{i_2=0}^{x-\sum_{y=3}^{n-1} i_y-1} \left(\left(\frac{d_3}{d_1} \right)^{i_2} \sum_{i_1=0}^{x-\sum_{y=2}^{n-1} i_y-1} \left(\left(\frac{d_2}{d_1} \right)^{i_1} \right) \right) \dots \right) \right)}$$

$$\cdot \sum_{j=1}^{n-1} \left(d_j \cdot \sum_{i_j=0}^{x-1} \left(\left(\frac{d_{j+1}}{d_1} \right)^{i_j} \sum_{i_{j-1}=0}^{x-i_j-1} \left(\left(\frac{d_j}{d_1} \right)^{i_{j-1}} \dots \sum_{i_1=0}^{x-\sum_{y=2}^j i_y-1} \left(\left(\frac{d_2}{d_1} \right)^{i_1} \right) \dots \right) \right) \right) + d_n$$

Auch wenn diese Formel komplex erscheint, enthält sie nur sehr schnell berechenbare Additionen und Multiplikationen und beschreibt die Aggregation eines Petrinetzes mit $\binom{N + \#P - 1}{N}$ Zuständen in ein Petrinetz mit $N + 1$ Zuständen, wobei N die initiale Markenanzahl und $\#P$ die Anzahl der Stellen im Originalnetz beschreibt. Der Durchsatz der beiden Transitionen T_0 im Original- und im aggregierten Modell aus Abbildung 5.37 ist dabei gleich.

Der Alternativoperator

In diesem Abschnitt wird die Semantik des Alternativoperators vorgestellt. Diese Semantik wurde anhand eines Referenzmodells ermittelt, das in Abbildung 5.38 dargestellt ist.

Im Gegensatz zum Sequenzoperator sind beim Alternativoperator erstmals zeitlose Transitionen zu berücksichtigen. Zeitlose Transitionen werden, wie in Abschnitt 2.5.1 ab Seite 17 beschrieben, insbesondere für die Modellierung von Alternativen verwendet. Durch eine Zuordnung von Wahrscheinlichkeiten zu in Konflikt stehenden zeitlosen Transitionen sind bestimmte Alternativen priorisierbar. Für das in Abbildung 5.38 auf der linken Seite dargestellte Originalmodell

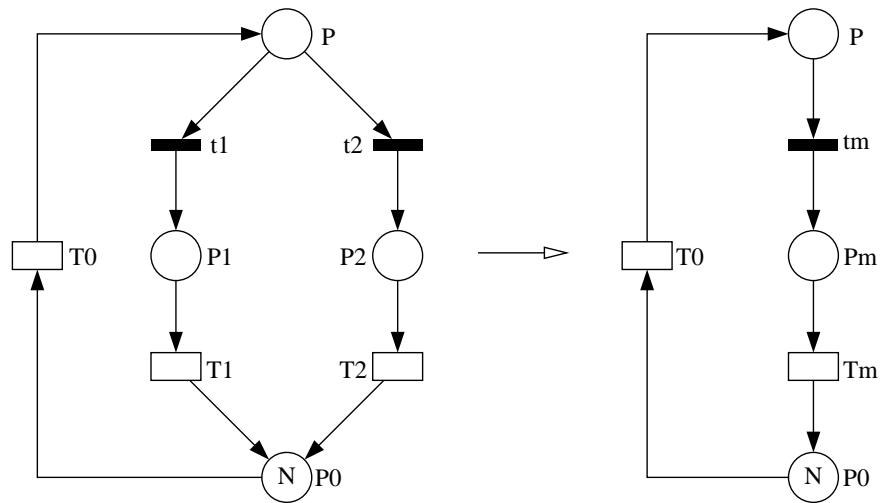


Abbildung 5.38: Referenzmodell zur Ermittlung der Semantik des Alternativoperators

mit Alternative ordnen wir den zeitlosen Transition t_1 und t_2 die Schaltwahrscheinlichkeiten w_1 und w_2 entsprechend zu. Ist P markiert, so schaltet demnach t_1 mit Wahrscheinlichkeit $\frac{w_1}{w_1+w_2}$ bzw. t_2 mit der Wahrscheinlichkeit $\frac{w_2}{w_1+w_2}$.

Für die in Abbildung 5.38 dargestellte Aggregation soll im folgenden die Schaltzeit der zeitbehafteten Transition T_m des aggregierten Modells unter Verwendung der Schaltzeiten der Transitionen T_1 und T_2 des Originalmodells derart bestimmt werden, daß die Erwartungswerte für die Anzahl der Marken auf den Stellen P_0 des Original- und des aggregierten Netzes übereinstimmen. Die Schaltzeit der Transition T_0 ist im Original- und im aggregierten Modell identisch. Die zeitlose Transition t_m im aggregierten Modell ist einflußlos und könnte ebenso weggelassen werden, da die dem Modell zugrunde liegende Markowkette identisch mit der dem Modell ohne zeitlose Transitionen zugrunde liegenden Markowkette ist. Sie ist trotzdem aufgeführt, da diese Aggregation auch dann Anwendung findet, wenn sich in Konflikt zu den Transitionen t_1 und t_2 des Originalmodells und somit zur Transition t_m des aggregierten Modells eine zeitlose Transition mit MIMO-Eigenschaft befindet. Würde die im Originalmodell in Abbildung 5.38 dargestellte Struktur einzig zu einer zeitbehafteten Transition aggregiert, wäre sie bei einer in Konflikt stehenden zeitlosen Transition t_{ot} , da zeitlose Transitionen stets eine höhere Priorität als zeitbehaftete Transitionen besitzen.

Wie schon für den Sequenzoperator erläutert, führt auch bei der Beschreibung der Semantik des Alternativoperators die Lösung zu einer markierungsabhängigen Schaltzeit von T_m . Wir schreiben für diese markierungsabhängige Schaltzeit von T_m wiederum $d_m^{k=x}$, wobei x für die Anzahl der Marken auf der Stelle P_m im

aggregierten Modell steht. Diese Schaltzeit berechnet sich wie folgt:

$$d_m^{k=x} = \frac{\sum_{i=0}^x \left(\frac{w_2 \cdot d_2}{w_1 \cdot d_1}\right)^i}{\sum_{j=0}^{x-1} \left(\frac{w_2 \cdot d_2}{w_1 \cdot d_1}\right)^j} \cdot \frac{w_1 \cdot d_1}{w_1 + w_2}$$

Der zeitlosen Transition t_m wird die Summe der Schaltwahrscheinlichkeiten der Transitionen t_1 und t_2 als Schaltwahrscheinlichkeit w_m zugeordnet, die jedoch, wie beschrieben, nur für Modelle mit zu t_m in Konflikt stehenden zeitlosen Transitionen relevant ist.

Das Referenzmodell für die Verallgemeinerung des Alternativoperators auf $n (> 1)$ Alternativen ist in Abbildung 5.39 dargestellt.

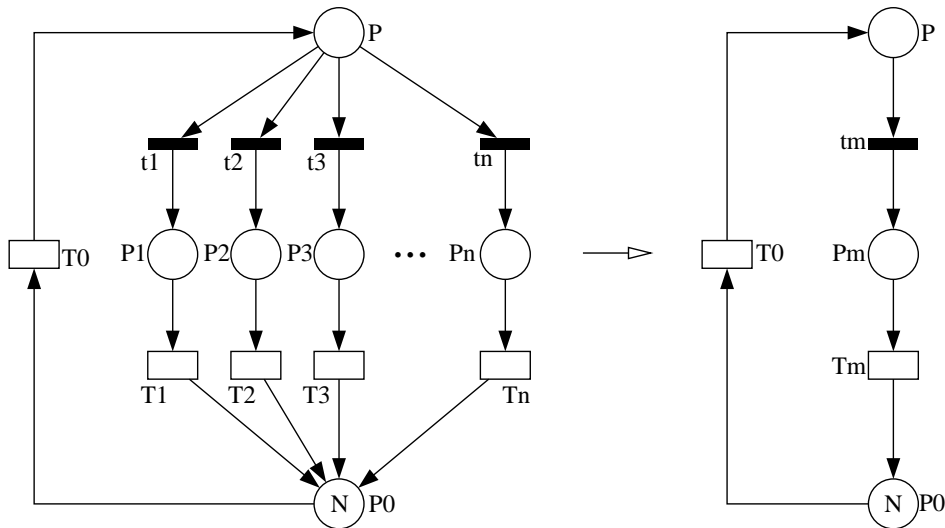


Abbildung 5.39: Referenzmodell für n Alternativen

Wiederum ist die markierungsabhängige Schaltzeit der Transition T_m derart gesucht, daß die Erwartungswerte für die Anzahl der Marken auf den Stellen P_0 des Original- und des aggregierten Modells identisch sind unter der Voraussetzung, daß die Schaltzeit der Transition T_0 in beiden Modellen gleich ist. Die Formel für diese allgemeine Aggregation von Alternativen lautet:

$$d_m^{k=x} = \frac{\sum_{i_{n-1}=0}^x \left(v_n^{i_{n-1}} \cdot \sum_{i_{n-2}=0}^{x-i_{n-1}} \left(v_{n-1}^{i_{n-2}} \cdot \sum_{i_{n-3}=0}^{x-i_{n-1}-i_{n-2}} \dots \sum_{i_2=0}^{x-\sum_{j=3}^{n-1} i_j} \left(v_3^{i_2} \cdot \sum_{i_1=0}^{x-\sum_{j=2}^{n-1} i_j} v_2^{i_1} \right) \right) \right)}{\sum_{i_{n-1}=0}^{x-1} \left(v_n^{i_{n-1}} \cdot \sum_{i_{n-2}=0}^{x-1-i_{n-1}} \left(v_{n-1}^{i_{n-2}} \cdot \sum_{i_{n-3}=0}^{x-1-i_{n-1}-i_{n-2}} \dots \sum_{i_2=0}^{x-1-\sum_{j=3}^{n-1} i_j} \left(v_3^{i_2} \cdot \sum_{i_1=0}^{x-1-\sum_{j=2}^{n-1} i_j} v_2^{i_1} \right) \right) \right)} \cdot \frac{w_1 \cdot d_1}{\sum_{i=1}^n w_i}$$

wobei $v_i = \frac{w_i \cdot d_i}{w_1 \cdot d_1}$ für alle $i = \{2, \dots, n\}$ ist. Die Anzahl der Alternativen ist, wie in Abbildung 5.39 dargestellt, gleich n und $d_m^{k=x}$ steht für die Schaltzeit der Transition T_m , wenn die Stelle P_m mit x Marken markiert ist. Der Transition t_m wird die Summe der Schaltwahrscheinlichkeiten der Transitionen t_1, \dots, t_n als Schaltwahrscheinlichkeit w_m zugeordnet.

Für das in Abbildung 5.39 dargestellte Referenzmodell ist die Aggregation unter Verwendung obiger Formel hinsichtlich der Berechnung des Durchsatzes der Transition T_0 beider Modelle exakt. Das auf der linken Seite der Abbildung 5.39 dargestellte Originalmodell hat $(n + 1)^N$ andauernde Zustände, während das aggregierte Modell nur $N + 1$ Zustände beschreibt.

Der Paralleloperator

In diesem Abschnitt wird die Semantik des Paralleloperators vorgestellt. Das zur Ermittlung der Semantik verwendete Referenzmodell ist in Abbildung 5.40 dargestellt.

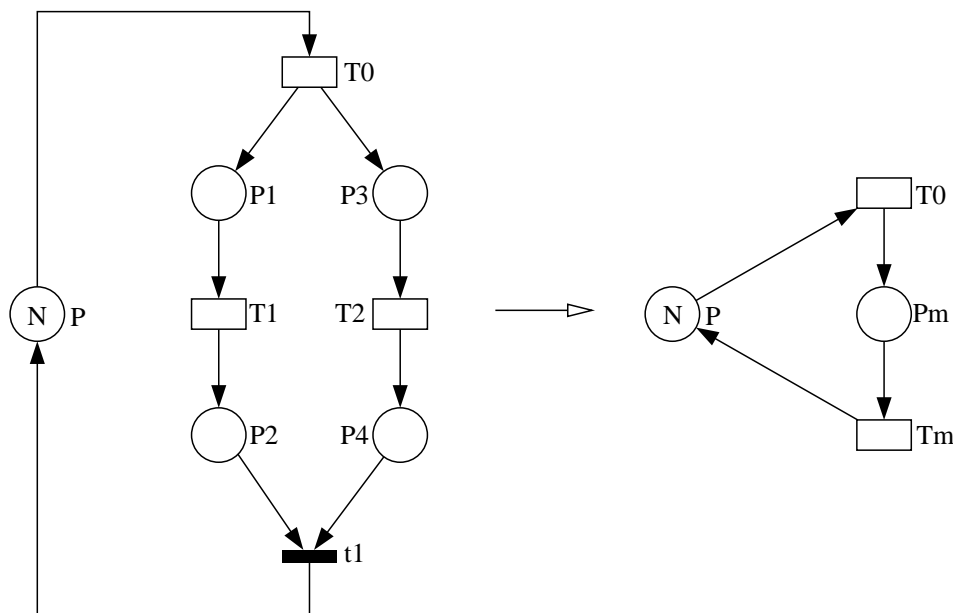


Abbildung 5.40: Referenzmodell zur Ermittlung der Semantik des Paralleloperators

Die linke Seite in Abbildung 5.40 zeigt das zu aggregierende Modell mit den parallelen Transitionen T_1 und T_2 . Diese Transitionen wurden in dem auf der rechten Seite der Abbildung dargestellten aggregierten Modell zur Transition T_m verschmolzen. Gesucht ist eine (markierungsabhängige) Schaltzeit von T_m , so daß

die Erwartungswerte der Anzahl der Marken auf P im originalen und aggregierten Modell identisch sind.

Für den Paralleloperator konnte im Gegensatz zum Sequenz- und dem Alternativoperator keine Semantik ermittelt werden, die eine bezüglich des Durchsatzes *exakte* Aggregation paralleler Transitionen mit negativ exponentieller Schaltzeit ermöglicht. Der Grund dafür liegt darin, daß die markierungsabhängigen Schaltzeiten für die Transition T_m von der Schaltzeit der Transition T_0 abhängen. Die markierungsabhängigen Schaltzeiten der Transition T_m werden, wie auch bereits bei dem Sequenz- und dem Alternativoperator, mit $d_m^{k=x}$ notiert. Dabei steht x für die Anzahl der Marken auf der Stelle P_m . Die markierungsabhängige Schaltzeit $d_m^{k=1}$ für den Zustand, in dem sich nur eine Marke auf P_m befindet, ist noch unabhängig von der Schaltzeit d_0 der Transition T_0 . Für alle Schaltzeiten von T_m unter der Markierung, daß sich mehr als eine Marke auf P_m befinden, gilt jedoch die Abhängigkeit von d_0 . Abbildung 5.41 zeigt anhand des Beispiels aus Abbildung 5.40 die Abhängigkeit der markierungsabhängigen Schaltzeit $d_m^{k=2}$ von der Schaltzeit d_0 .

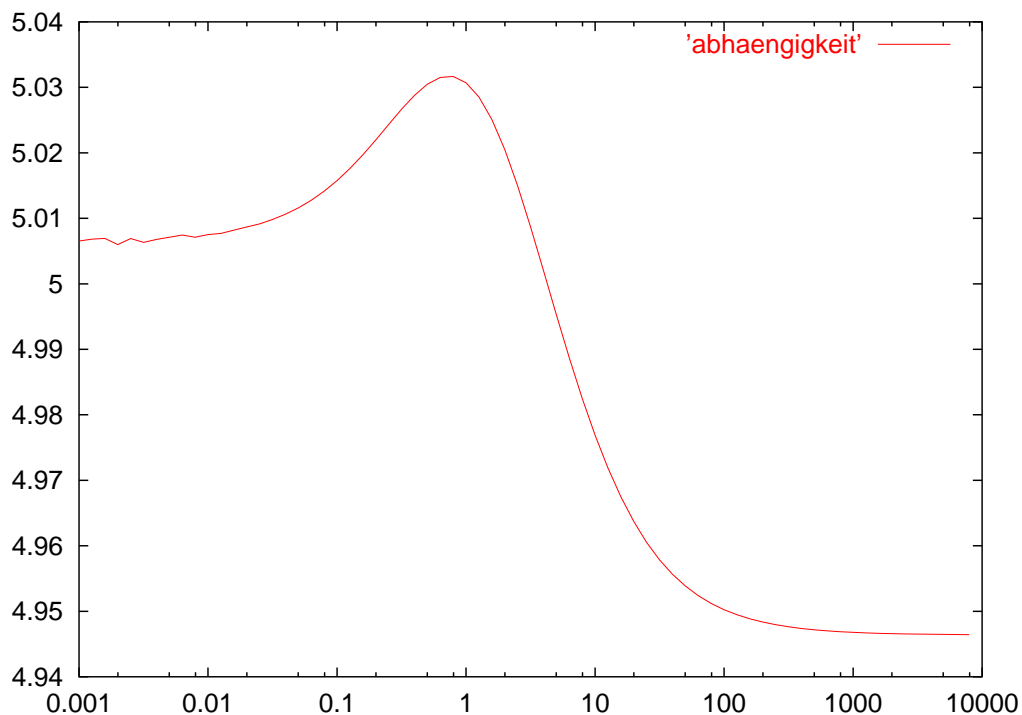


Abbildung 5.41: Abhängigkeit $d_m^{k=2}$ von d_0

Die x-Achse in Abbildung 5.41 listet in logarithmischer Anordnung Werte für die Schaltzeit d_0 auf, während auf der y-Achse die Werte für $d_m^{k=2}$ eingetragen sind. Die Kurve zeigt die *exakten* Werte für $d_m^{k=2}$ unter der Annahme, daß den

Transitionen T1 und T2 im Originalmodell aus Abbildung 5.40 die Schaltzeiten $d_1 = 1$ und $d_2 = 5$ zugeordnet sind. Wäre $d_m^{k=2}$ unabhängig von d_0 , so beschriebe die in Abbildung 5.41 dargestellte Kurve eine horizontale Linie.

Die Form der die Abhängigkeit beschreibenden Kurve ist sicherlich überraschend und nur mit Vermutungen interpretierbar. Betrachtet man erneut Abbildung 5.40, so wäre eine erste intuitive Vermutung für die Schaltzeit d_m sicherlich das Maximum aus d_1 und d_2 . Unter der für die Abbildung 5.41 getroffenen Annahme, daß $d_1 = 1$ und $d_2 = 5$ sind, entspricht der tatsächliche Wert auch angenähert dieser Vermutung (siehe Abbildung 5.41). Das Maximum für $d_m^{k=2}$ liegt bei $d_0 = 0,7313$ und beträgt $5,03172438$. Für $d_0 = 10.000$ beträgt der exakte Wert für $d_m^{k=2} = 4,946426$ und für $d_0 = 100.000$ ist $d_m^{k=2} = 4,94639$. Der Ausdruck $k = 2$ in $d_m^{k=2}$ bedeutet, daß diese Schaltzeit im aggregierten Modell unter der Markierung verwendet wird, daß sich auf P_m genau zwei Marken befinden (siehe Abbildung 5.40). Das bedeutet, daß das Schalten der Transition T_m mit der Schaltzeit $d_m^{k=2}$ ein Schalten der Transition T1 des originalen Modells unter der Markierung, daß P1 mit zwei Marken belegt ist oder das Schalten der Transition T2 des originalen Modells unter der Markierung, daß P3 mit zwei Marken belegt ist, abbildet. Im folgenden ist deshalb der Zustandsübergang interessant, der eine zweite Marke in die Stelle P1 und/oder P3 schaltet. Wenn d_0 sehr klein ist, so ist die Wahrscheinlichkeit sehr groß, daß auf *beide* Stellen eine zweite Marke geschaltet wird. Der seltene Fall, daß für die Transition T1 eine hohe Schaltzeit ausgewürfelt wird, führt vermutlich zu dem Ergebnis, daß $d_m^{k=2}$ größer als fünf ist. Ist jedoch die Schaltzeit von d_0 sehr groß, so führt der seltene Fall, daß für die Transition T1 eine hohe Schaltzeit ausgewürfelt wird, vermutlich zu dem Ergebnis, daß $d_m^{k=2}$ kleiner als fünf ist, denn der Zustand mit einer Belegung von zwei Marken auf P1 und einer Marke auf P3 wird „schnell“ durch das Schalten von T1 wieder verlassen. Doch selbst mit dieser Interpretation ist der „Buckel“ in der Kurve noch nicht begründet.

Da bis zum Zeitpunkt der Fertigstellung dieser Arbeit keine Formel für die Beschreibung der Schaltzeit einer aus der Verschmelzung zweier paralleler Transitionen entstandenen Transition gefunden wurde, die eine exakte Angleichung des Durchsatzes des aggregierten Modells an den Durchsatz des Originalmodells beschreibt, wird die markierungsabhängige Schaltzeit für $d_m^{k=x}$ approximativ wie folgt berechnet.

$$d_m^{k=x} = d_1 + d_2 \cdot \frac{1}{\sum_{i=0}^{2x-1} \left(\frac{d_1}{d_2}\right)^i}$$

Wie bereits erläutert, berechnet diese Formel nicht den exakten Wert für $d_m^{k=x}$. Die in Tabelle 5.42 dargestellten Vergleiche zeigen anhand von Beispielwerten für die Schaltzeiten der Transitionen aus Abbildung 5.40 die Abweichungen der durch die angegebene Formel berechneten Werte von den exakten.

x	d ₀ , d ₁ , d ₂ = 1		d ₀ , d ₂ = 1, d ₁ = 2	
	d _m ^{k=x} exakt	d _m ^{k=x} berechnet	d _m ^{k=x} exakt	d _m ^{k=x} berechnet
1	1.5	1.5	2.33333	2.33333
2	1.25	1.25	2.07442	2.06667
3	1.16546	1.16667	2.02539	2.01587
4	1.12137	1.125	2.01039	2.00392
5	1.0958	1.1	2.00507	2.00098
6	1.07923	1.08333	2.00256	2.00024
7	1.06758	1.07143	2.00124	2.00006
8	1.05895	1.0625	2.00056	2.00002
9	1.05228	1.05556	2.00024	2
10	1.04697	1.05	2.0001	2
x	d ₀ , d ₁ = 1, d ₂ = 5		d ₀ = 5, d ₁ , d ₂ = 1	
	d _m ^{k=x} exakt	d _m ^{k=x} berechnet	d _m ^{k=x} exakt	d _m ^{k=x} berechnet
1	5.16667	5.16667	1.5	1.5
2	5.0307	5.00641	1.22245	1.25
3	5.00924	5.00026	1.1113	1.16667
4	5.00153	5.00001	1.05561	1.125
5	5.0002	5	1.02819	1.1
6	5.00002	5	1.01402	1.08333
7	5	5	1.00648	1.07143
8	5	5	1.00238	1.0625
9	5	5	1.00097	1.05556
10	5	5	0.998521	1.05

Abbildung 5.42: Vergleiche der berechneten und exakten $d_m^{k=x}$

Abbildung 5.43 zeigt das Referenzmodell zur Ermittlung der Semantik des Paralleloperators für ein beliebiges $n(> 1)$ paralleler Transitionen.

Für den allgemeinen Fall von n parallelen Transitionen konnte keine geeignete Semantik gefunden werden. Für alle bis zum Zeitpunkt der Fertigstellung dieser Arbeit untersuchten Formeln war der Approximationsfehler der Aggregation inakzeptabel groß. In Kapitel 6 wird für mehr als zwei parallele Transitionen die Güte der Approximation untersucht, wenn aus den n parallelen Transitionen die beiden mit den größten Schaltzeiten ausgesucht werden und diese Schaltzeiten in die Formel für die Aggregation von zwei parallelen Transitionen eingesetzt wird.

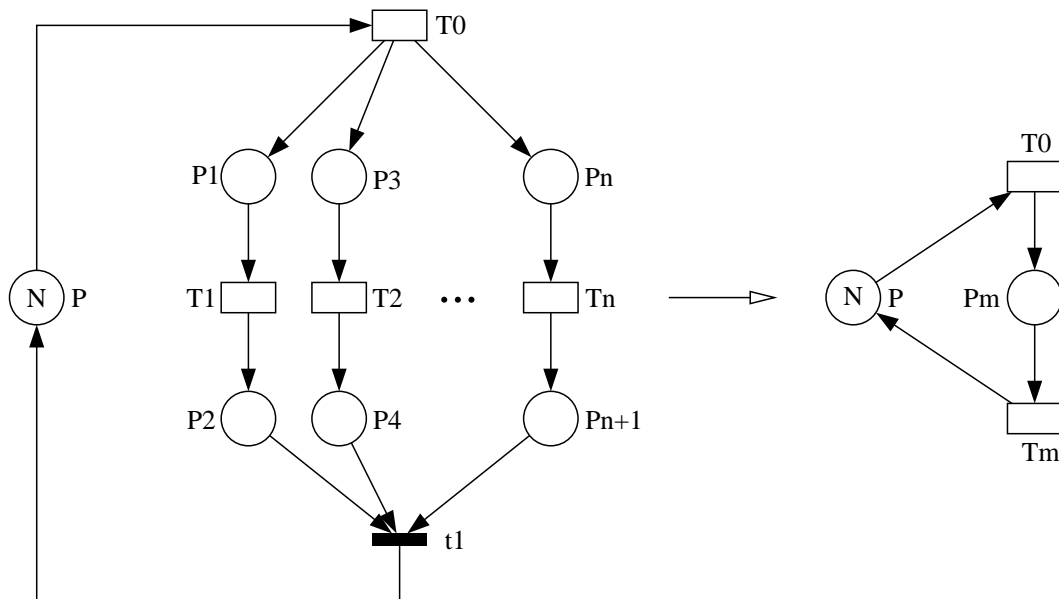


Abbildung 5.43: Referenzmodell für n parallele Transitionen

5.5.3 Ursachen des Approximationsfehlers

Nachdem in Abschnitt 5.5.1 bereits zwei Ursachen des Approximationsfehlers bei der Verwendung des Sequenzoperators erläutert wurden, soll in diesem Abschnitt über weitere Gründe des entstehenden Fehlers bei der Aggregation diskutiert werden. Neben der Fehlerquelle der im vorhergehenden Abschnitt erläuterten Anwendung des Paralleloperators ist die Kombination der Operatoren ein weiteres, bisher noch ungelöstes Problem bei der strukturbasierten Aggregation.

Am Beispiel des Sequenz- und des Alternativoperators soll der gegenwärtige Stand der Forschung bei der Möglichkeit von Kombinationen der vorgestellten Operatoren erläutert werden. Abbildung 5.44 zeigt auf der linken Seite ein Modell, das aggregiert werden soll.

Die obere Hälfte des Originalmodells weist die aus Abbildung 5.39 bekannte Struktur zur Anwendung des Alternativoperators auf. Die Transitionen T_{s1}, \dots, T_{sn} können unter Anwendung des Sequenzoperators miteinander verschmolzen werden. Das in Abbildung 5.44 rechts oben dargestellte Modell zeigt die Verschmelzung der Alternativen in die Transitionen t_m und T_m sowie die Verschmelzung der in einer Sequenzstruktur wie in Abbildung 5.37 angeordneten Transitionen T_{s1}, \dots, T_{sn} in die Transition T_{sm} . Diese Aggregation ist jedoch unter Anwendung der MIMO-Graphen basierten Aggregation noch nicht vollständig. Die Transitionen t_m, T_m und T_{sm} wären weiter verschmelzbar. Jedoch ist den Transitionen T_m und T_{sm} nach Anwendung der im Abschnitt 5.5.2

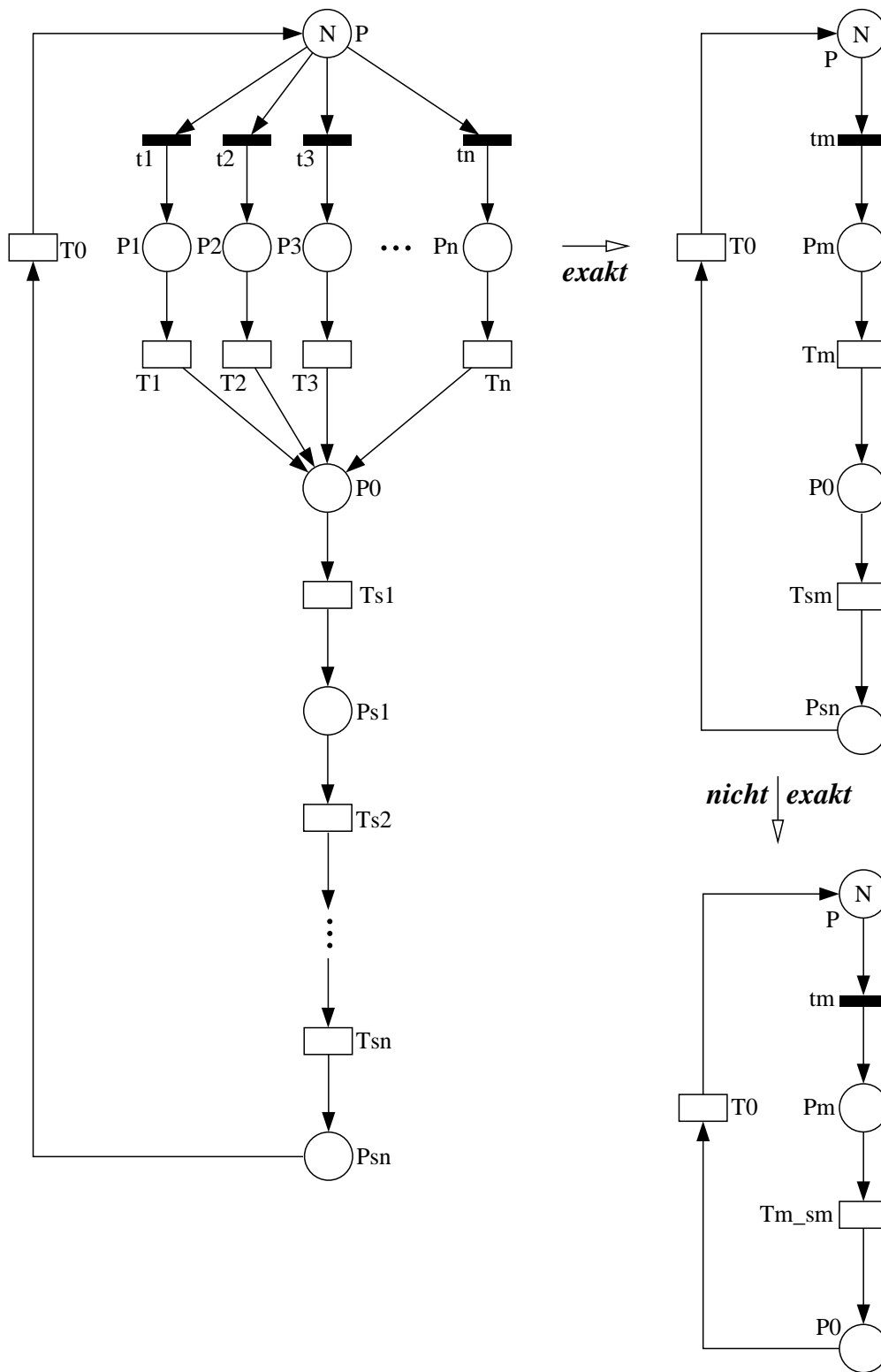


Abbildung 5.44: Kombinationsmöglichkeiten und -grenzen der Operatoren

vorgeschlagenen Formeln zur Berechnung der neuen Schaltzeiten verschmolzener Transitionen jeweils eine markierungsabhängige Schaltzeit zugeordnet. Dabei ist die Schaltzeit der Transition T_{sm} von der Anzahl der Marken auf der Stelle P_0 abhängig. Nach Verschmelzung der Transitionen T_m und T_{sm} zur Transition $T_{m.sm}$, dargestellt in Abbildung 5.44 rechts unten, existiert diese Stelle jedoch nicht mehr. Dieses Problem tritt bei beliebiger Kombination aller drei vorgeschlagenen Operatoren auf und ist zum Zeitpunkt der Fertigstellung dieser Arbeit noch ungelöst.

Es sei an dieser Stelle auch nochmals betont, daß nur zeitbehaftete Transitionen mit exponentiell verteilter Schaltzeit untersucht werden. Insbesondere für das in dieser Arbeit untersuchte Anwendungsgebiet der Fertigungssysteme genügt die ausschließliche Verwendung zeitbehafteter Transitionen mit exponentiell verteilter Schaltzeit nicht den Anforderungen der Realität. Transitionen mit deterministischer oder allgemeiner Schaltzeit sind für eine realistischere Abbildung eines Fertigungssystems auf ein Petrinetzmodell unerlässlich.

Erstmals werden in dieser Arbeit jedoch Formeln präsentiert, die eine struktur-basierte Behandlung aggregierter zeitbehafteter Transitionen ermöglichen, so daß das aggregierte Modell angenähert denselben Durchsatz wie das Originalmodell aufweist. Für einige wenige Modelle ist gar ein aggregiertes Modell erstellbar, daß exakt denselben Durchsatz wie das Originalmodell besitzt. Obwohl es sich bei diesen Modellen um strukturell sehr einfache Modelle handelt, sind sie doch deutlich komplexer als die bisher in der Literatur [72, 73, 71, 38, 121] untersuchten, deren Durchsatz ebenfalls mittels Formeln exakt berechenbar ist. Es sei jedoch betont, daß sich selbst bei Modellen, für die der Durchsatz bei der Aggregation exakt bewahrt bleibt, die Zustandswahrscheinlichkeiten auch der nichtaggregierten Zustände ändern kann. Die Aggregation referenziert nur den Erwartungswert von Marken auf Stellen, also einen Mittelwert. Der in Abbildung 5.44 verwendete Begriff *exakt* bezieht sich nur auf die Beschreibung dieses Mittelwertes. Die Modelle, die auch hinsichtlich der Zustandswahrscheinlichkeiten exakt aggregierbar sind, müssen sehr einschränkenden Bedingungen genügen [72, 73]. In Kapitel 6 werden neben den Erwartungswerten für einige Modelle auch Wahrscheinlichkeiten einer bestimmten Anzahl von Marken auf einer Stelle untersucht.

Ein offenes Problem bleibt die Abschätzung des Approximationsfehlers. In Kapitel 6 werden Modelle experimentell untersucht und Heuristiken diskutiert, in denen sich aus aggregierten Strukturen von Petrinetzmodellen Aussagen über die Größe des Approximationsfehlers ableiten lassen.

5.6 Überblick über weitere zustandsraumreduzierende Verfahren

Zustandsraumreduzierende Verfahren lassen sich in drei verschiedene Arten klassifizieren. Die meisten Methoden nutzen Symmetrien aus. Eine zweite Klasse mit zunehmender Bedeutung basiert auf der Verwendung binärer Entscheidungsdiagramme (*binary decision diagrams*–BDDs). Die dritte Klasse basiert auf der Beachtung von Nebenläufigkeit in Petrinetzen und geht auf Valmari [127, 128] zurück. Die Beachtung der Nebenläufigkeit oder vielmehr die Beachtung des Nichtbeeinflussens von Transitionen erlaubt eine qualitative Analyse von Petrinetzen unter Aufspannung eines reduzierten Zustandsraumes. Der Erreichbarkeitsgraph ist derart reduziert, daß nur Zustandsübergänge betrachtet werden, die durch das Schalten *sturer* Transitionen entstehen. Dabei wird ausgenutzt, daß der aus nebenläufigen Ereignissen resultierende Zustand unabhängig von der Reihenfolge dieser Ereignisse ist. Betrachten wir das Beispielmmodell aus Abbildung 4.1 auf Seite 46 unter der Annahme, daß die Stellen P_2 und P_3 markiert seien. Unter dieser Markierung haben die Transitionen T_1 und T_3 Konzession und können unabhängig voneinander schalten. Die Reihenfolge des Schaltens spielt dabei keine Rolle, es wird ein Zustand erreicht, in dem die Stellen P_1 und P_4 markiert sind. Mit Hilfe der Theorie über sture Transitionsmengen lassen sich wichtige qualitative Eigenschaften eines Petrinetzes untersuchen. Insbesondere die Lebendigkeit und die Identifizierung toter Zustände sind mit z.T. deutlich geringerem Speicherplatzbedarf analysierbar.

Für das Ausnutzen von Symmetrien gibt es eine fast unüberschaubare Anzahl an Vorschlägen, von denen nur einige hier kurz skizziert werden können. Für die qualitative Analyse farbiger Petrinetze kann das Gleichverhalten verschieden farbiger Marken durch sogenannte *äquivalente Markierungen* [77, 79] bzw. *parametrisierte Erreichbarkeitsgraphen* [93] ausgenutzt werden. Da sich bspw. die speisenden Philosophen [52] äquivalent verhalten, kann demnach bei der Erstellung des Erreichbarkeitsgraphen des als farbiges Petrinetz modellierten Algorithmus vom Index der Philosophen abstrahiert werden.

In derselben Weise läßt sich das äquivalente Verhalten verschiedenfarbiger Marken für die numerische Analyse stochastischer Petrinetze ausnutzen. In *well-formed* [26, 27] Netzen werden Objektklassen und Farbdomänen derart definiert, daß sich Marken mit Farben aus derselben Objektklasse symmetrisch verhalten. Durch die Zuordnung der Objektklassen zu Farbdomänen lassen sich elementare Funktionen wie z.B. eine Nachbarschaftsrelation über die speisenden Philosophen definieren, um einen reduzierten, symbolischen Erreichbarkeitsgraphen aufzustellen.

In [109, 107, 108] sind erstmals BDDs verwendet worden, um den Erreichbarkeitsgraphen von (1-beschränkten) Petrinetzen zu erstellen und effizient abzu-

speichern. Durch die Zerlegung des Petrinetzes mit n Stellen in n Teilnetze mit je einer Stelle war jedoch ein enormer Zeitaufwand nötig, um die Erreichbarkeit der Zustände zu ermitteln. Trotzdem gelang eine BDD-Repräsentation der erreichbaren Zustände eines Petrinetzes unter geringem Speicherplatzbedarf.

Jüngste Arbeiten [31, 32] benutzen eine Erweiterung binärer Entscheidungsdiagramme zur Erstellung des Erreichbarkeitsgraphen. Diese Erweiterung wird *multi-valued decision diagrams–MDDs* genannt, die Funktionen der Form

$$f : \{0, \dots, k_1\} \times \dots \times \{0, \dots, k_N\} \rightarrow \{0, \dots, M\}$$

mit $k_1, \dots, k_N, M \in \mathbb{N}$ beschreiben. Für den Spezialfall $k_1 = \dots = k_N = M = 1$ ist das durch f beschriebene MDD ein BDD. Für die Erstellung des Erreichbarkeitsgraphen eines Petrinetzes werden MDDs der Form

$$f : \{0, \dots, k_1\} \times \dots \times \{0, \dots, k_N\} \rightarrow \{0, 1\}$$

verwendet, wobei die Belegung der Zahlenvariablen globalen Zuständen des zu untersuchenden Petrinetzes entsprechen, die bei dem Funktionswert 1 erreichbar und bei dem Funktionswert 0 nicht erreichbar sind.

Bei der auf MDDs basierenden Erstellung des Erreichbarkeitsgraphen wird das zu analysierende Petrinetz zunächst in mehrere Teilnetze derart geteilt, daß die paarweise disjunkten Stellenmengen der Teilnetze der Stellenmenge des Gesamtnetzes entspricht. Betrachten wir dazu das in Abbildung 4.1 auf Seite 46 dargestellte Beispiel unter der in Abbildung 4.2 gezeigten Zerlegung. Für jedes Teilmodell wird zunächst der lokale Erreichbarkeitsgraph erstellt. Abbildung 5.45 zeigt das MDD für die initiale Markierung des Petrinetzmodells aus Abbildung 4.1.

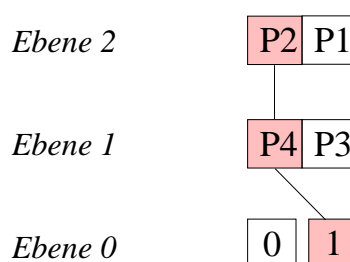


Abbildung 5.45: MDD für die initiale Markierung des Beispiels aus Abbildung 4.1

Das gezeigte MDD besteht aus drei Ebenen 0, 1, 2. Die beiden oberen Ebenen 2 und 1 zeigen sogenannte Nichtterminale des MDD und beschreiben die Zustände der jeweiligen Teilmodelle. In Ebene 2 sind die Zustände des in Abbildung 4.2 mit A bezeichneten Teilmodells dargestellt, wobei bspw. $P1$ für den lokalen Zustand steht, in dem die Stelle $P1$ markiert ist. Die unterste Ebene 0 stellt die Terminale 0 und 1 dar und gibt an, ob der durch das MDD beschriebene Pfad aus Nichtterminalen und Kanten einen erreichbaren Zustand des Gesamtmodells darstellt

(Pfad führt zum Terminalknoten 1) oder nicht (Pfad führt zum Terminalknoten 0). Die Richtung der Kanten ist im Bild nicht dargestellt, stattdessen vereinbaren wir stets, daß die Kanten von der übergeordneten zur untergeordneten Ebene verlaufen. Die Nichtterminale und die Terminale werden als Knoten des MDD bezeichnet. Der initiale Zustand $P2 \ P4$ ist in Abbildung 5.45 dunkel unterlegt und per Definition (siehe Def. 2.3) erreichbar.

Eine wichtige Unterscheidung findet wie auch schon bei der Kronecker-Algebra basierten Analyse zwischen lokalen und synchronen Ereignissen statt. Für das Teilmodell A ist das Schalten der Transition T1 wie auch für das Teilmodell b das Schalten der Transition T3 ein lokales Ereignis, die Transition T2 synchronisiert beide Teilmodelle. Beeinflußt eine Transition, wie im dargestellten Beispiel die Transition T2, ein oder mehrere Teilmodelle, so wird eine Markierung $ErsteEbene(T2)$ an die höchste Ebene angehängt, die durch die Transition T2 beeinflusst wird. Im gezeigten Beispiel ist es die höchste Ebene des MDD, die Ebene 2.

Abbildung 5.46 zeigt die schrittweise Erstellung des MDD basierten Erreichbarkeitsgraphen für das Beispiel aus Abbildung 4.1 auf Seite 46.

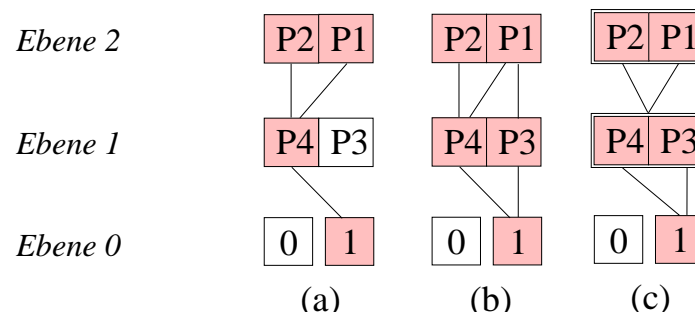


Abbildung 5.46: Erstellung des MDD basierten Erreichbarkeitsgraphen

Da in Ebene 1 im lokalen Zustand P4 keine lokale Transition schaltfähig ist, werden zunächst die Zustände auf Ebene 2 berechnet. In der Ebene 2 ist unter der Markierung P2 die Transition T1 schaltfähig. Der lokale Zustand P1 ist erreichbar und wird wegen der Kante von P2 zu P4 ebenfalls mit P4 verbunden¹. Dieser Schritt ist in Abbildung 5.46(a) dargestellt. Ebene 2 trägt darüber hinaus die Markierung $ErsteEbene(T2)$, so daß in dieser Ebene das durch das Schalten der Transition T2 beschriebene Ereignis ausgewertet wird. Das Schalten führt in Ebene 1 zu einem Zustand P3, die Kanten P1 zu P3 und von dort zum Terminalknoten 1 werden gezogen (Schritt (b)). Durch das Schalten der lokalen Transition T3 ist der Zustand P4 von P3 aus erreichbar. Die Ebene 1 ist somit vollständig beschrieben (in [32]: *gesättigt*), dargestellt durch ein doppeltes Rechteck in Schritt (c). Somit sind bereits sowohl $\{P2\} \times \{P3, P4\}$ als auch $\{P1\} \times \{P3, P4\}$ erreichbare Zustände.

¹Jedes Ausführen eines Ereignisses bewirkt die Erstellung eines neuen MDDs, welches mit dem bestehenden *vereinigt* werden muß. Für elementare Operationen auf MDDs siehe z.B. [32]

Das Schalten der Transition T_2 führt in Ebene 2 zum Zustand P_1 , so daß auch diese Ebene vollständig beschrieben ist. Nach Schritt (c) ist ein vollständig beschriebener Erreichbarkeitsgraph mittels eines MDDs dargestellt.

In diesem einfachen Beispiel ist das Kreuzprodukt der lokalen Zustände gleich der Menge der globalen Zustände, so daß es keinen Pfad durch das MDD gibt, der am Terminalknoten 0 endet. Ein neben dem geringen Speicherbedarf weiterer Vorteil ist aber die Berechnung des tatsächlichen Erreichbarkeitsgraphen. Es entstehen also nicht, wie beispielsweise bei der Kronecker-Algebra basierten Analyse, falsche Zustände. In Verbindung mit der in Abschnitt 4.2 beschriebenen Matrixdiagramm basierten Darstellung der Übergangsraten- oder Generatormatrix lassen sich bereits jetzt Petrinetze mit z.T. fast 10^{627} [99] Zuständen analysieren. Der große Nachteil dieser Methode liegt jedoch derzeit noch in der Restriktion, daß die Zerlegung der in [34, 31] beschriebenen *Produktform* genügen muß. Insbesondere ist eine Analyse von Petrinetzen mit zeitlosen Transitionen oder markierungsabhängigen Kanten kardinalitäten auf der Basis von MDDs derzeit noch nicht möglich.

Kapitel 6

Anwendungen

In diesem Abschnitt wird die MIMO-Graphen basierte Aggregierung an Beispielen angewendet. Es gibt zwei Arten der Anwendung der MIMO-Graphen basierten Aggregierung bei der numerischen Analyse stochastischer Petrinetze. Einerseits lässt sich ein zu untersuchendes Petrinetzmodell in seiner Gesamtheit aggregieren. Das entstehende aggregierte Petrinetz wird anschließend analysiert. Im Abschnitt 6.1 wird dieser Ansatz der Anwendung für einige Beispiele untersucht. Insbesondere werden in Abschnitt 6.1 die Möglichkeiten und Grenzen der Semantik der vorgeschlagenen Operatoren zur Verschmelzung zeitbehafteter Transitionen demonstriert. Anhand einfacher Modelle wird zum einen die Anwendung der in Abschnitt 5.5.2 vorgeschlagenen Formeln zur Berechnung markierungsabhängiger Schaltzeiten aggregierter Transitionen demonstriert. Zum anderen werden Modelle untersucht, die in Abschnitt 5.5.1 als problematisch für die MIMO-Graphen basierte Aggregierung diskutiert wurden.

Eine weitere Anwendungsmöglichkeit der MIMO-Graphen basierten Aggregierung zur Leistungsbewertung stochastischer Petrinetze besteht in der Integration der Aggregierung in bekannte iterative, approximative Verfahren, die auf einer Zerlegung des Gesamtmodells basieren. Ein solches Verfahren wird in Abschnitt 6.2 vorgestellt. In diesem Verfahren wird zunächst das Gesamtmodell in mehrere Teilmodelle zerlegt. Dabei findet die in Abschnitt 5.4.1 vorgestellte automatische Zerlegung spezieller farbiger Petrinetzmodelle Anwendung. Anschließend werden mittels MIMO-Graphen basierter Aggregierung die Teilmodelle aggregiert. In einem dritten Schritt werden diese Teilmodelle unter Beachtung der Interaktion untereinander iterativ analysiert. Durch die in dieser Arbeit vorgestellte strukturbasierte Berechnung der Schaltzeiten verschmolzener Transitionen ist es möglich, den Approximationsfehler bei der untersuchten iterativen Analyse akzeptabel klein zu halten.

6.1 Anwendungsbeispiele

In diesem Abschnitt werden die in Abschnitt 5.5.2 vorgeschlagenen Semantiken der Operatoren experimentell untersucht, die bei der Verschmelzung zeitbehalteter Transitionen Anwendung finden.

Mit den ersten in diesem Abschnitt vorgestellten Modellen soll die Anwendung des Sequenz- und Alternativoperators demonstriert werden. Insbesondere wird die Berechnung der markierungsabhängigen Schaltzeiten erläutert. Für alle vorgestellten Modelle sind die Werte der markierungsabhängigen Schaltzeiten angegeben. Im zweiten Teil dieses Abschnittes werden zwei in Abschnitt 5.5.1 diskutierte Probleme anhand von Modellen experimentell untersucht. Im dritten Teil dieses Abschnittes werden Modelle mit parallelen Transitionen untersucht.

Modelle mit Sequenzen und Alternativen

Zunächst wird an einem einfachen Petrinetz die Anwendung des Sequenzoperators verdeutlicht. Abbildung 6.1 zeigt oben das zu analysierende Originalmodell. Die Transitionen T1...T10 werden mittels Sequenzoperator zur Transition T_m verschmolzen. Das aggregierte Modell ist in Abbildung 6.1 unten dargestellt.

Die Schaltzeiten der Transitionen sind entsprechend ihres Index gewählt. Der Transition T3 ist demnach die Schaltzeit $d_3 = 3$ und der Transition T10 die Schaltzeit $d_{10} = 10$ zugeordnet. Die markierungsabhängige Schaltzeit $d_m^{k=x}$, wobei x für die Anzahl der Marken auf der Stelle P steht, wird wie folgt berechnet:

$$d_m^{k=x} = \frac{1}{\sum_{i_9=0}^{x-1} \left(\left(\frac{10}{1}\right)^{i_9} \sum_{i_8=0}^{x-i_9-1} \left(\left(\frac{9}{1}\right)^{i_8} \dots \sum_{i_2=0}^{x-i_9-\dots-i_3-1} \left(\left(\frac{3}{1}\right)^{i_2} \sum_{i_1=0}^{x-i_9-\dots-i_2-1} \left(\frac{2}{1}\right)^{i_1} \right) \dots \right) \right)} \cdot \sum_{j=1}^9 \left(d_j \cdot \sum_{i_j=0}^{x-1} \left(\left(\frac{d_{j+1}}{1}\right)^{i_j} \sum_{i_{j-1}=0}^{x-i_j-1} \left(\left(\frac{d_j}{1}\right)^{i_{j-1}} \dots \sum_{i_1=0}^{x-i_j-\dots-i_2-1} \left(\frac{2}{1}\right)^{i_1} \right) \dots \right) \right) + 10$$

Abbildung 6.2 zeigt die errechneten Werte für $d_m^{k=x}$.

Das in Abbildung 6.1 oben dargestellte Originalmodell hat 184756 Zustände, während das aggregierte Modell nur 11 Zustände besitzt. Der Erwartungswert der Anzahl der Marken auf der Stelle P0 beträgt in beiden Modellen 0,087124. Die Wahrscheinlichkeit, daß sich zwei Marken auf der Stelle P0 befinden, ist in beiden Modellen 0,005767. Die stationäre Analyse des aggregierten Modells benötigt 0,37 Sekunden. Die Werte für $d_m^{k=x}$ werden unabhängig von der Schaltzeit d_0 der Transition T0 ermittelt. Die Werte der angegebenen Leistungsmaße wurden für $d_0 = 1$ ermittelt. Die Berechnung des Originalmodells dauert 603,2 Sekunden auf einem Pentium-III PC mit 512MB Hauptspeicher.

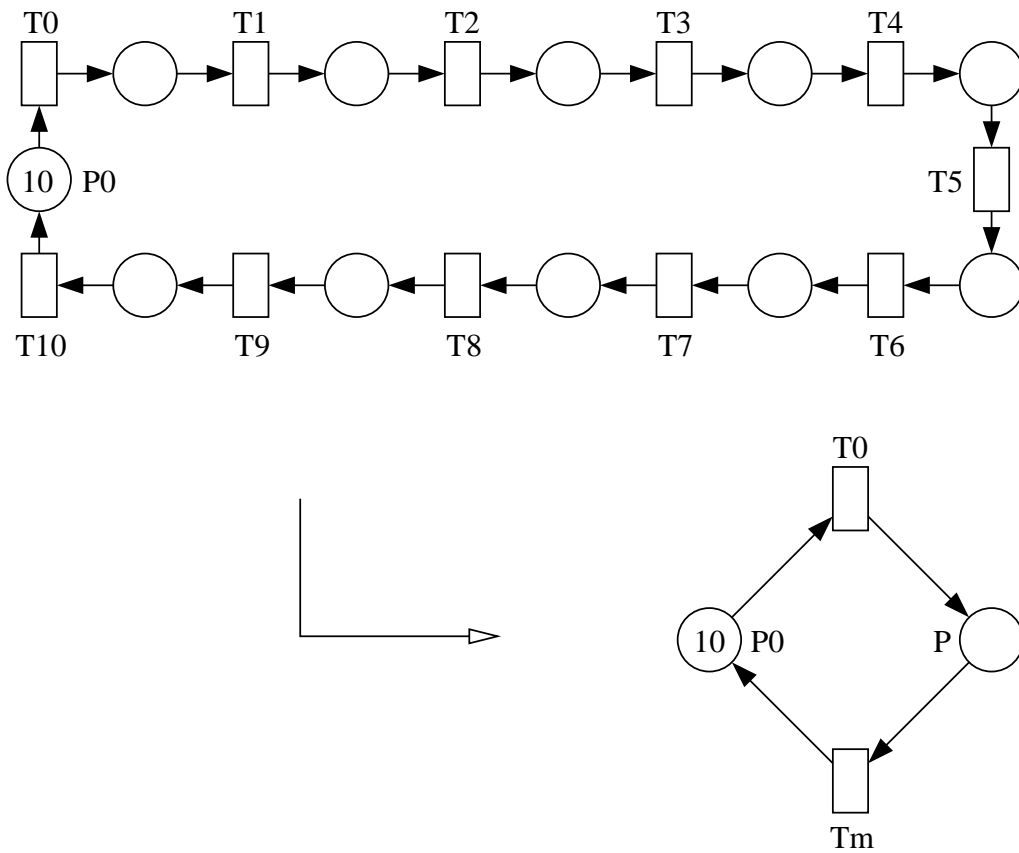


Abbildung 6.1: Anwendung des Sequenzoperators

x	$d_m^{k=x}$
1	55
2	31
3	23,0645161290
4	19,1418181818
5	16,8218085106
6	15,3012987013
7	14,2361967038
8	13,4545569118
9	12,8609325050
10	12,3981180392

Abbildung 6.2: Beispiel – Werte für $d_m^{k=x}$

Ein weiteres Beispiel, an dem die Anwendung des Sequenz- und des Alternativoperators gezeigt werden soll, ist in Abbildung 6.3 dargestellt. Abbildung 6.4 zeigt das daraus erzeugte aggregierte Modell.

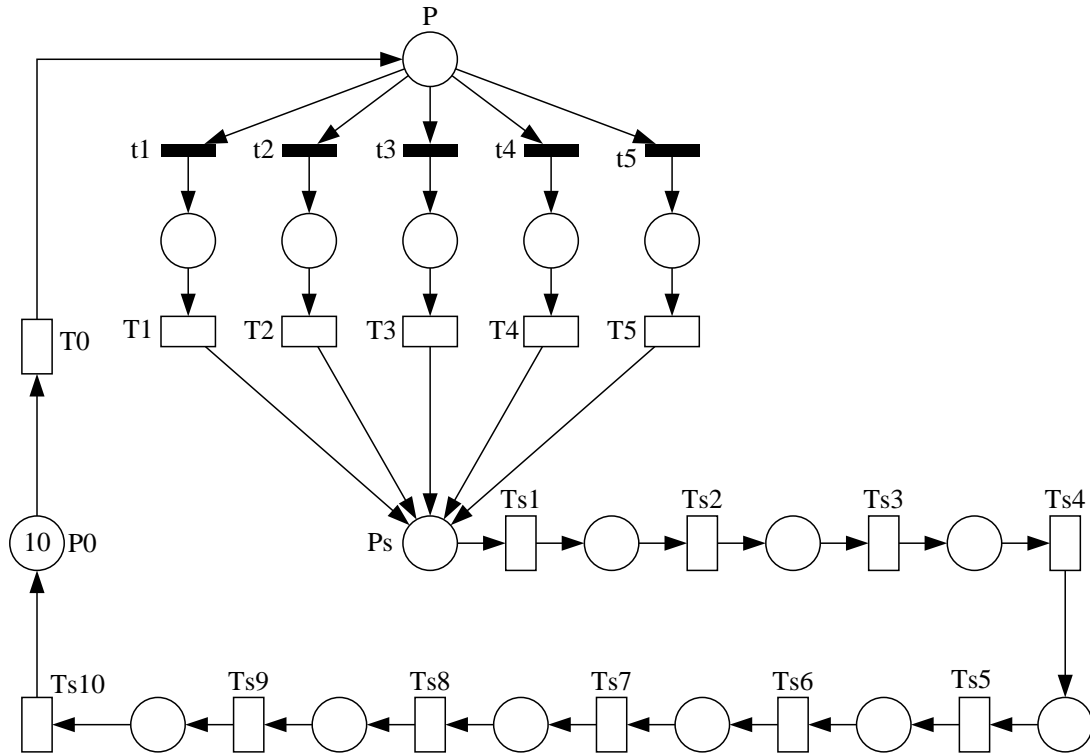


Abbildung 6.3: Anwendung des Sequenz- und Alternativoperators

Die Transitionen T_{am} und T_{sm} beschreiben die aus der Aggregation der Alternativen und der Sequenz entstandenen Transitionen. Eine weitere Aggregation dieser beiden Transitionen wäre zwar strukturell möglich, ist aber wegen des noch nicht gelösten Problems der Kombination von Operatoren derzeit nicht handhabbar (siehe Abschnitt 5.5.3). Den mittels Sequenzoperator verschmolzenen Transitionen $Ts_1 \dots Ts_{10}$ sind, wie bereits beim in Abbildung 6.1 dargestellten Modell, ihrem Index entsprechende Schaltzeiten zugeordnet. Die markierungsabhängige Schaltzeit $d_{sm}^{k=x}$ besitzt deshalb dieselben in Abbildung 6.2 dargestellten Werte in Abhängigkeit der Anzahl der Marken auf P_s .

Den Transitionen $Ta_1 \dots Ta_5$ sind ebenfalls Schaltzeiten entsprechend ihres Index zugeordnet. Die zeitlosen Transitionen $ta_1 \dots ta_5$ besitzen jeweils dieselbe Schaltwahrscheinlichkeit 0.2. Die markierungsabhängige Schaltzeit $d_{am}^{k=x}$ der Transition Ta_m in Abhängigkeit der Anzahl der Marken auf P berechnet sich wie folgt:

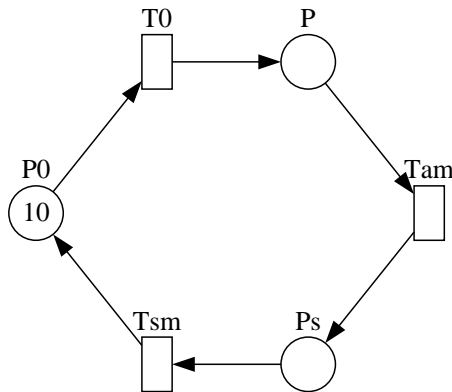


Abbildung 6.4: Anwendung des Sequenz- und Alternativoperators

$$d_{am}^{k=x} = \frac{\sum_{i_4=0}^x \left(\left(\frac{0.2 \cdot 5}{0.2 \cdot 1} \right)^{i_4} \sum_{i_3=0}^{x-i_4} \left(\left(\frac{0.2 \cdot 4}{0.2 \cdot 1} \right)^{i_3} \sum_{i_2=0}^{x-i_4-i_3} \left(\left(\frac{0.2 \cdot 3}{0.2 \cdot 1} \right)^{i_2} \sum_{i_1=0}^{x-i_4-i_3-i_2} \left(\frac{0.2 \cdot 2}{0.2 \cdot 1} \right)^{i_1} \right) \right) \right)}{\sum_{i_4=0}^{x-1} \left(\left(\frac{0.2 \cdot 5}{0.2 \cdot 1} \right)^{i_4} \sum_{i_3=0}^{x-i_4-1} \left(\left(\frac{0.2 \cdot 4}{0.2 \cdot 1} \right)^{i_3} \sum_{i_2=0}^{x-i_4-i_3-1} \left(\left(\frac{0.2 \cdot 3}{0.2 \cdot 1} \right)^{i_2} \sum_{i_1=0}^{x-i_4-i_3-i_2-1} \left(\frac{0.2 \cdot 2}{0.2 \cdot 1} \right)^{i_1} \right) \right) \right)} \cdot \frac{0.2 \cdot 1}{1}$$

Abbildung 6.5 zeigt die errechneten Werte für $d_{am}^{k=x}$. Das in Abbildung 6.3 darge-

x	$d_{am}^{k=x}$
1	3
2	1,8666666666
3	1,5
4	1,324
5	1,2235649546
6	1,1603997648
7	1,1181453410
8	1,0886618819
9	1,0674576723
10	1,0518614394

Abbildung 6.5: Beispiel – Werte für $d_{am}^{k=x}$

stellte Originalmodell besitzt 3 268 760 Zustände, während das aggregierte Modell aus Abbildung 6.4 66 Zustände besitzt. Das Originalmodell ist zu groß, um mit TimeNET analysiert werden zu können. Mit Hilfe der ereignisdiskreten Simulation wurde in einem Konfidenzintervall von 99% bei einem Fehler von 2% für den Erwartungswert der Anzahl der Marken auf der Stelle P0 der Wert 0,087602

und für die Wahrscheinlichkeit, daß sich genau drei Marken auf der Stelle P0 befinden, der Wert = 0,000399 berechnet. Unter Verwendung des aggregierten Modells wird für den Erwartungswert der Anzahl der Marken auf der Stelle P0 der Wert 0,086214 und für die Wahrscheinlichkeit, daß sich genau drei Marken auf der Stelle P0 befinden, der Wert 0,000415 berechnet. Die Belegungen von $d_{am}^{k=x}$ und $d_{sm}^{k=x}$ sind unabhängig von der Schaltzeit d_0 der Transition T0. Die Werte für die Wahrscheinlichkeit und den Erwartungswert wurden für $d_0 = 1$ ermittelt.

Erfahrungen mit kleineren Modellen belegen, daß die für die aggregierten Modelle berechneten Werte mit denen der entsprechenden Originalmodelle übereinstimmen. Es ist davon auszugehen, daß es sich bei den Abweichungen um Fehler der Simulation handelt.

Problematische Anwendungen des Sequenzoperators

Im folgenden werden einige Modelle vorgestellt, die innerhalb dieser Arbeit bereits als problematisch für die Anwendung der vorgestellten Operatoren diskutiert wurden. Zunächst werden zwei in Abschnitt 5.5.1 diskutierte Probleme betrachtet. Abbildung 6.6 zeigt auf der linken Seite ein Beispiel, in dem die Transitionen T1, T2 und T3 mittels Sequenzoperator verschmolzen werden. Das aggregierte Modell ist auf der rechten Seite in Abbildung 6.6 dargestellt.

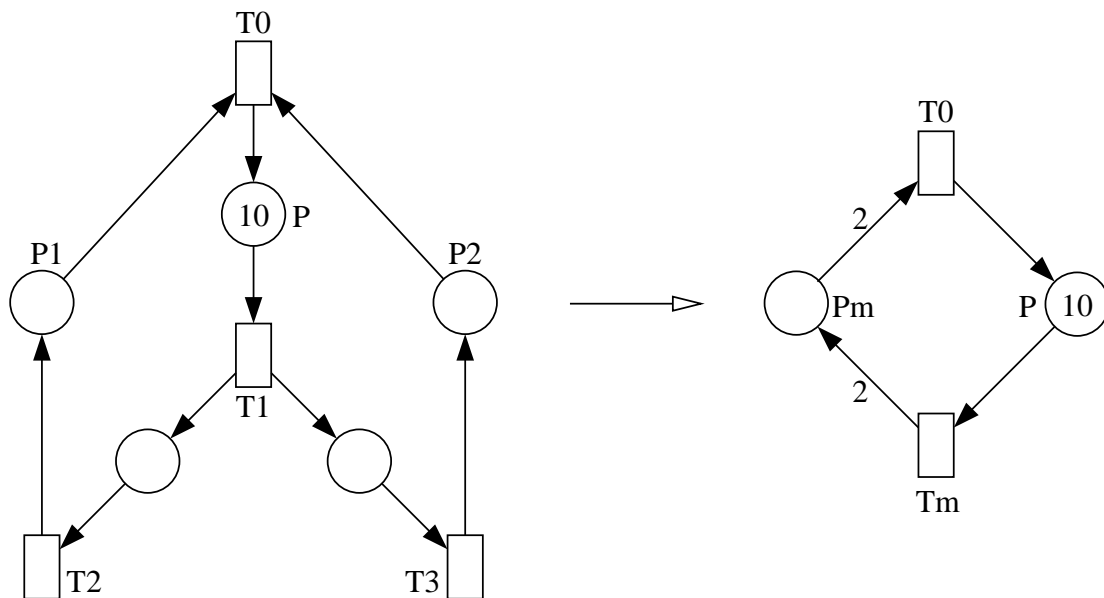


Abbildung 6.6: Problematische Anwendung des Sequenzoperators – 1

Das bei der Aggregation entstehende Problem wurde in Abschnitt 5.5.1 anhand der Abbildung 5.29 auf Seite 100 erläutert. Dieses Problem entsteht bei der Ag-

gregation, wenn den Ausgangskanten eines MIMO-T-Elementes verschiedene Schaltzeiten zugeordnet sind. Bei der Rücktransformation des MIMO-Graphen in ein Petrinetz wird der diesem MIMO-T-Element entsprechenden Transition der Mittelwert der Schaltzeiten der Ausgangskanten zugeordnet. Die markierungsabhängige Schaltzeit $d_m^{k=x}$ für die Transition T_m in Abbildung 6.6 wird unter Verwendung des Mittelwertes der Schaltzeiten d_2 und d_3 der Transitionen T_2 und T_3 berechnet. Im folgenden wird die Auswirkung dieser Festlegung für variierende Schaltzeiten d_3 der Transition T_3 untersucht. Wir betrachten Schaltzeiten der Transitionen T_1 und T_2 mit $d_1 = d_2 = 1$. Die markierungsabhängigen Schaltzeiten $d_m^{k=x}$ in Abhängigkeit der Schaltzeit d_3 der Transition T_3 werden wie folgt berechnet.

$$d_m^{k=x} = \frac{1}{\sum_{i=0}^{x-1} \frac{1+d_3}{2 \cdot 1}} + \frac{1+d_3}{2 \cdot 1}$$

Dabei ergibt sich der Term $\frac{1+d_3}{2 \cdot 1}$ aus $\frac{d_{23}}{d_1}$, wobei d_{23} den Mittelwert $\frac{d_2+d_3}{2}$ der Schaltzeiten der Transitionen T_2 und T_3 beschreibt. Die Werte der markierungsabhängigen Schaltzeiten $d_m^{k=x}$ in Abhängigkeit der Schaltzeiten d_3 von Transition T_3 sind in Abbildung 6.7 dargestellt.

In Abbildung 6.8 sind die Durchsätze der Transition T_0 vergleichend dargestellt. Der Durchsatz der Transition wird berechnet, indem die Wahrscheinlichkeit dafür berechnet wird, daß die Transition Konzession besitzt. Diese Wahrscheinlichkeit wird durch die Schaltzeit der Transition T_0 dividiert. Die Wahrscheinlichkeit, daß die Transition T_0 im auf der linken Seite in Abbildung 6.6 dargestellten Originalmodell Konzession besitzt, ist gleich der Wahrscheinlichkeit, daß die Stelle P_1 und die Stelle P_2 markiert sind. Da der Transition T_0 eine Schaltzeit $d_0 = 1$ im Beispiel zugeordnet ist, ist diese Wahrscheinlichkeit gleich dem zu ermittelnden Durchsatz der Transition. Die Wahrscheinlichkeit, daß die Transition T_0 im auf der rechten Seite in Abbildung 6.6 dargestellten aggregierten Modell Konzession besitzt, ist gleich der Wahrscheinlichkeit, daß sich auf der Stelle P_m mindestens zwei Marken befinden.

Die Vergleiche zeigen, daß der Mittelwert von Schaltzeiten nicht die geeignete Wahl sein kann, wenn das Problem bei der Aggregation auftritt, daß den Ausgangskanten eines MIMO-T-Elementes verschiedene Schaltzeiten zugeordnet sind. Wie bereits in Abschnitt 5.5.1 erwähnt, scheint die Einführung eines weiteren Operators notwendig, um dieses Problem zu lösen.

Ein weiteres in Abschnitt 5.5.1 diskutiertes Problem tritt bei der Aggregation des in Abbildung 6.9 dargestellten Petrinetzmodells auf.

In dem in Abbildung 6.9 oben dargestellten Originalmodell sind zwar die Transitionen T_1 und T_2 , nicht aber die Transitionen T_2 und T_3 verschmelzbar, da T_2 weitere Eingangskanten und T_3 weitere Ausgangskanten besitzen. Das bereits in

x	$d_m^{k=x}$				
	$d_3 = 1$	$d_3 = 2$	$d_3 = 3$	$d_3 = 4$	$d_3 = 5$
1	2	2,5	3	3,5	4
2	1,5	1,9	2,33333	2,78571	3,25
3	1,33333	1,71052	2,14286	2,60256	3,07692
4	1,25	1,62307	2,06666	2,53941	3,025
5	1,2	1,57583	2,03226	2,51552	3,00826
6	1,16666	1,54812	2,01587	2,50617	3,00275
7	1,14285	1,53108	2,00787	2,50246	3,00091
8	1,125	1,52030	2,00392	2,50098	3,0003
9	1,11111	1,51335	2,00196	2,50039	3,0001
10	1,1	1,50882	2,00098	2,50016	3,00003
x	$d_m^{k=x}$				
	$d_3 = 6$	$d_3 = 7$	$d_3 = 8$	$d_3 = 9$	$d_3 = 10$
1	4,5	5	5,5	6	6,5
2	3,72222	4,2	4,68182	5,16666	5,65385
3	3,55970	4,04762	4,53883	5,03226	5,52721
4	3,51677	4,01176	4,50856	5,00641	5,50492
5	3,50477	4,00293	4,50190	5,00128	5,50089
6	3,50136	4,00073	4,50042	5,00026	5,50016
7	3,50039	4,00018	4,50009	5,00005	5,50003
8	3,50011	4,00005	4,50002	5,00001	5,50001
9	3,50003	4,00001	4,5	5	5,5
10	3,50001	4	4,5	5	5,5

Abbildung 6.7: Werte für $d_m^{k=x}$ in Abhängigkeit von d_3

d_3	Durchsatz von T0 im		relativer
	Originalmodell	aggreg. Modell	Fehler
1	0,806714	0,833333	3,29 %
2	0,498635	0,651731	30,70 %
3	0,333305	0,498653	49,61 %
4	0,249998	0,399834	59,93 %
5	0,199999	0,333306	66,65 %
6	0,166666	0,285708	71,42 %
7	0,142857	0,249999	74,99 %
8	0,124999	0,222222	77,78 %
9	0,111111	0,200000	80,00 %
10	0,099999	0,181818	81,82 %

Abbildung 6.8: Vergleich der Durchsätze der Transition T0

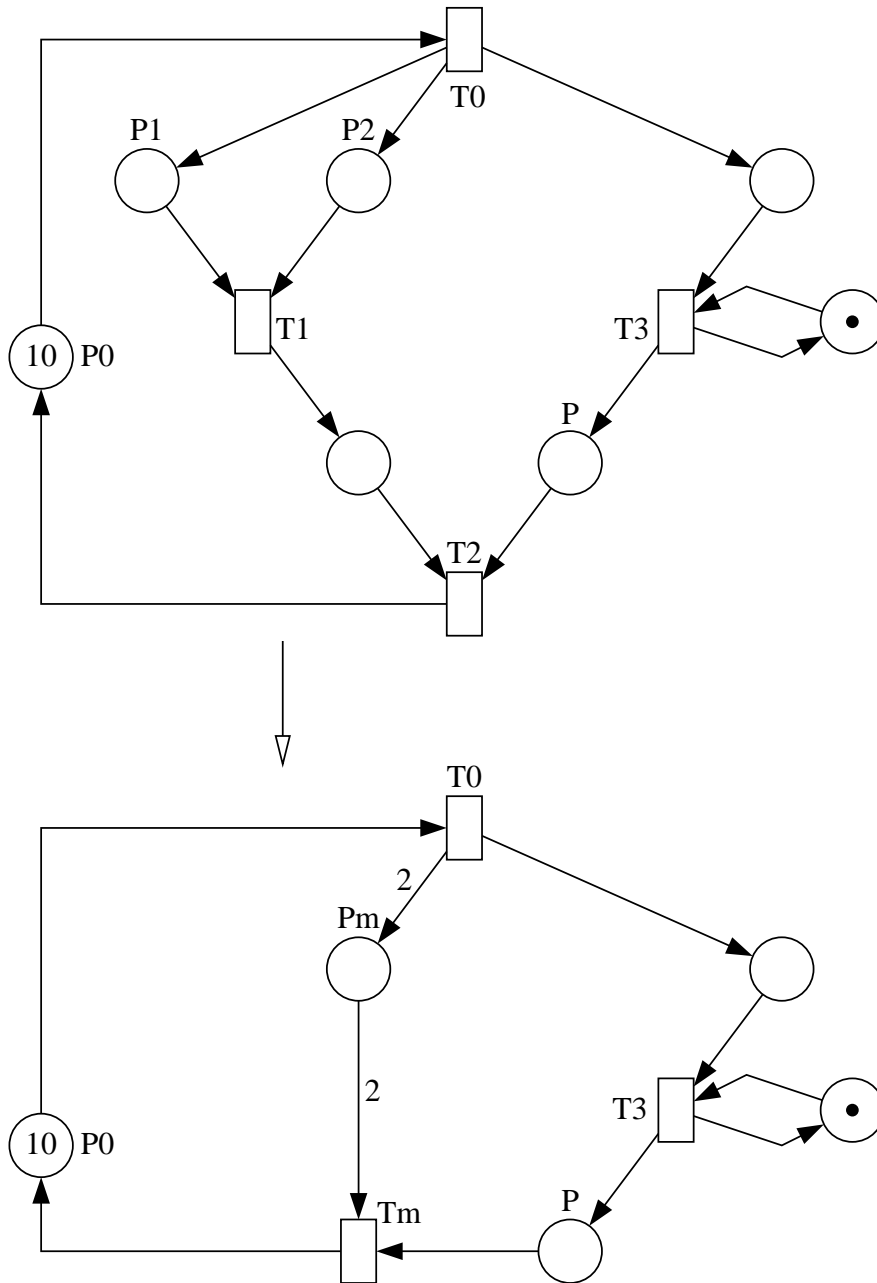


Abbildung 6.9: Problematische Anwendung des Sequenzoperators – 2

Abschnitt 5.5.1 diskutierte Problem besteht nun darin, daß nach der Verschmelzung der Transitionen T1 und T2 zur Transition Tm eine Marke auf der Stelle P im Originalmodell um die Schaltzeit d2 der Transition T2, im aggregierten Modell aber um die im Durchschnitt höhere Schaltzeit $d_m^{k=2x}$ der Transition Tm verzögert wird. Diese Änderung der Verzögerung soll im folgenden beispielhaft untersucht werden.

Für die Schaltzeiten der Transitionen T0, T1 und T2 soll eine Schaltzeit $d_0 = d_1 = d_2 = 1$ angenommen werden. Die markierungsabhängigen Schaltzeiten $d_m^{k=2x}$ der Transition Tm sind in der Tabelle in Abbildung 6.10 dargestellt¹.

x	$d_m^{k=2x}$
1	2
2	1.5
3	1.33333333
4	1.25
5	1.2
6	1.16666666
7	1.14285714
8	1.125
9	1.11111111
10	1.1

Abbildung 6.10: Beispiel – Werte für $d_m^{k=2x}$

Die in Abbildung 6.11 dargestellte Tabelle zeigt den Vergleich der Erwartungswerte für die Anzahl der Marken auf der Stelle P0 in Abhängigkeit der Schaltzeit d3 der Transition T3. Da das an dieser Stelle diskutierte Problem erst scheinbar relevant wird, wenn im Modell aus Abbildung 6.9 die Schaltzeit der Transition groß gegenüber der Schaltzeit der Transition T2 ist, variieren die Werte für d3 zwischen 10 und 20 und sind auch für d3 = 100 und d3 = 1000 aufgeführt.

Der in Abbildung 6.11 dargestellte Vergleich überrascht, da die Erwartungswerte im Original- und im aggregierten Modell fast dieselben sind. Tatsächlich sind die relativen Approximationsfehler bei kleineren Schaltzeiten d3 der Transition T3 sogar größer. Bei der Berechnung von Wahrscheinlichkeiten der Anzahl der Marken auf der Stelle P0 sind ähnliche Ergebnisse erzielt worden. Allerdings verkleinert die dargestellte Aggregation des Beispiels den Zustandsraum nur unwesentlich. Das Originalmodell beschreibt 506, das aggregierte Modell 66 Zustände.

¹Wir schreiben $k = 2x$, da die markierungsabhängigen Schaltzeiten für Tm in Abhängigkeit der Anzahl der Marken auf der Stelle Pm berechnet werden. Da die Kantenkardinalität der Kante von Pm zu Tm zwei beträgt, steht das x hier nicht für die Anzahl der Marken auf Pm, sondern das k. An diesem Beispiel wird der Grund der Schreibweise für das übliche $d_m^{k=x}$ deutlich.

d_3	$P\{\#P0\}$ im		relativer Fehler
	Originalmodell	aggreg. Modell	
10	0,11111135	0,11101614	0,08 %
11	0,10000034	0,09992663	0,08 %
12	0,09090928	0,09085080	0,06 %
13	0,08333347	0,08328588	0,06 %
14	0,07692321	0,07688370	0,05 %
15	0,07142877	0,07139539	0,05 %
16	0,06666681	0,06663833	0,04 %
17	0,06250013	0,06247549	0,04 %
18	0,05882367	0,05880211	0,04 %
19	0,05555565	0,05553669	0,03 %
20	0,05263165	0,05261483	0,03 %
100	0,01010101	0,01010101	0
1000	0,001001	0,001001	0

Abbildung 6.11: Vergleich der Erwartungswerte

Generell scheint aber dieses Problem einen weniger starken Einfluß auf den Approximationsfehler zu haben, als das vorher diskutierte.

Modelle mit parallelen Transitionen

Eine weitere Ursache des Approximationsfehlers ist in der Anwendung des Paralleloperators begründet. Für das Referenzmodell mit zwei parallelen Transitionen wurden bereits in Abschnitt 5.5.2 Untersuchungen durchgeführt. Darin wurde erläutert, daß für den Paralleloperator, im Gegensatz zum Sequenz- und Alternativoperator, keine Formel existiert, die die Verschmelzung von Transitionen bezüglich des Durchsatzes exakt beschreibt. Insbesondere ist eine Verschmelzung von mehr als zwei parallelen Transitionen ein noch ungelöstes Problem. In Abschnitt 5.5.2 wurde für diesen Fall vorgeschlagen, daß die beiden Transitionen mit den größten Schaltzeiten ausgewählt werden. Nach dieser Wahl wird eine Struktur von mehr als zwei parallelen Transitionen so verschmolzen, als bestünde sie nur aus zwei parallelen Transitionen. Der zeitliche Einfluß der anderen parallelen Transitionen bleibt unberücksichtigt. In diesem Abschnitt werden zunächst Modelle mit zwei parallelen Transitionen untersucht und später die Güte des Verfahrens mit mehr als zwei parallelen Transitionen getestet.

Abbildung 6.12 zeigt die bereits aus Abbildung 5.40 auf Seite 114 bekannte Aggregation eines Petrinetzmodells mit zwei parallelen Transitionen T_1 und T_2 zur Transition T_m . Es werden die Erwartungswerte der Anzahl der Marken auf der Stelle P_0 ($E\{\#P_0\}$) sowie die Wahrscheinlichkeiten, daß sich auf dieser Stelle

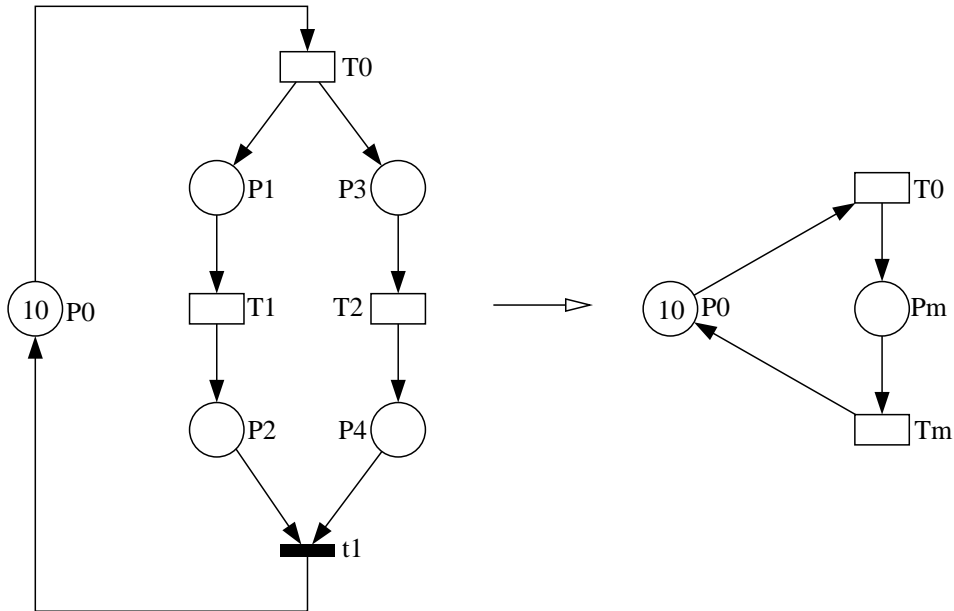


Abbildung 6.12: Anwendung des Paralleloperators

d_0, d_1, d_2	$E\{\#P0\}$ im		relativer
	Originalmodell	aggreg. Modell	Fehler in %
1,1,1	4,02555	4	0,63
1,2,1	0,99227	0,99326	0,1
1,5,1	0,25	0,25	0
5,1,1	9,63125	9,625	0,63
d_0, d_1, d_2	$P\{\#P0=3\}$ im		relativer
	Originalmodell	aggreg. Modell	Fehler in %
1,1,1	0,11234	0,11076	1,4
1,2,1	0,06259	0,06254	0,08
1,5,1	0,0064	0,0064	0
5,1,1	0,00002	0,00003	50

Abbildung 6.13: Vergleich der Erwartungswerte und Wahrscheinlichkeiten

drei Marken befinden ($P\{\#P_0=3\}$), für das Original- und das aggregierten Modell berechnet. Die Werte dieser Leistungsmaße werden zunächst für die in Abbildung 5.42 auf Seite 117 untersuchten Schaltzeiten verglichen, da darin bereits die berechneten markierungsabhängigen Schaltzeiten $d_m^{k=x}$ für die Transition T_m aufgeführt sind. Abbildung 6.13 zeigt diese Vergleiche. Im folgenden werden dieselben Vergleiche für eine Belegung der Schaltzeiten $d_0 = 1$, $d_1 = 9$ durchgeführt, wobei d_2 zwischen 10 und 19 variiert. Abbildung 6.14 zeigt die berechneten markierungsabhängigen Schaltzeiten $d_m^{k=x}$ der Transition T_m , die abhängig von der Anzahl der Marken auf der Stelle P_m sind. Abbildung 6.15 zeigt den Vergleich der

x	$d_m^{k=x}$				
	$d_2 = 10$	$d_2 = 11$	$d_2 = 12$	$d_2 = 13$	$d_2 = 14$
1	14,26315	15,04999	15,85714	16,68181	17,52173
2	11,90782	12,62401	13,38857	14,19290	15,02982
3	11,13420	11,85707	12,64953	13,49489	14,37970
4	10,75582	11,50255	12,33375	13,22284	14,15022
5	10,53533	11,31061	12,17902	13,10379	14,06100
6	10,39359	11,19778	12,09813	13,04908	14,02503
7	10,29662	11,12820	12,09813	13,02337	14,01031
8	10,22744	11,08404	12,05442	13,01117	14,00425
9	10,17660	11,05548	12,03037	13,00534	14,00175
10	10,13840	11,03680	12,01700	13,00256	14,00072
x	$d_m^{k=x}$				
	$d_2 = 15$	$d_2 = 16$	$d_2 = 17$	$d_2 = 18$	$d_2 = 19$
1	18,375	19,24	20,11538	21	21,89285
2	15,89338	16,77875	17,68201	18,6	19,53013
3	15,29363	16,22898	17,18010	18,14285	19,11425
4	15,10249	16,07086	17,04967	18,03529	19,02541
5	15,03650	16,02226	17,01386	18,00879	19,00569
6	15,01308	16,00703	17,00388	18,00219	19,00127
7	15,00470	16,00222	17,00108	18,00054	19,00028
8	15,00169	16,00070	17,00030	18,00013	19,00006
9	15,00061	16,00022	17,00008	18,00003	19,00001
10	15,00022	16,00007	17,00002	18,00001	19,00000

Abbildung 6.14: Werte für $d_m^{k=x}$ in Abhängigkeit von d_2

Erwartungswerte der Anzahl der Marken auf der Stelle P_0 und Abbildung 6.16 den Vergleich der Wahrscheinlichkeiten, daß sich genau drei Marken auf der Stelle P_0 befinden, für das Original- und das aggregierte Modell bei variierenden d_2 .

Die Abbildungen 6.15 und 6.16 belegen, daß die für die Aggregation zweier par-

d_2	$E\{\#P0\}$ im		relativer Fehler in %
	Originalmodell	aggreg. Modell	
10	0,10889438	0,109323	0,39
11	0,09934015	0,099593	0,25
12	0,09068271	0,090818	0,15
13	0,08324187	0,083312	0,08
14	0,07688131	0,076918	0,05
15	0,07140788	0,071427	0,03
16	0,06665584	0,066666	0,02
17	0,06249410	0,062500	0,01
18	0,05882021	0,058823	0
19	0,05555363	0,055555	0

Abbildung 6.15: Vergleich der Erwartungswerte

d_2	$P\{\#P0=3\}$ im		relativer Fehler in %
	Originalmodell	aggreg. Modell	
10	0,00079034	0,000854	8,05
11	0,00063558	0,000672	5,73
12	0,00050880	0,000528	3,77
13	0,00040981	0,000420	2,48
14	0,00033328	0,000338	1,41
15	0,00027393	0,000276	0,75
16	0,00022750	0,000229	0,66
17	0,00019082	0,000192	0,62
18	0,00016152	0,000162	0,30
19	0,00013788	0,000138	0,09

Abbildung 6.16: Vergleich der Wahrscheinlichkeiten

alleler Transitionen vorgeschlagene Formel einen geeigneten, wenn auch approximativen, Vorschlag darstellt. Im folgenden wird die Güte der Wahl untersucht, für die Aggregation von mehr als zwei parallelen Transitionen diejenigen zu wählen, die die größten Schaltzeiten besitzen und mit diesen die Aggregierungsformel für zwei parallele Transitionen anzuwenden. Als Referenzmodell dieser Untersuchung wird dabei das in Abbildung 6.17 dargestellte gewählt.

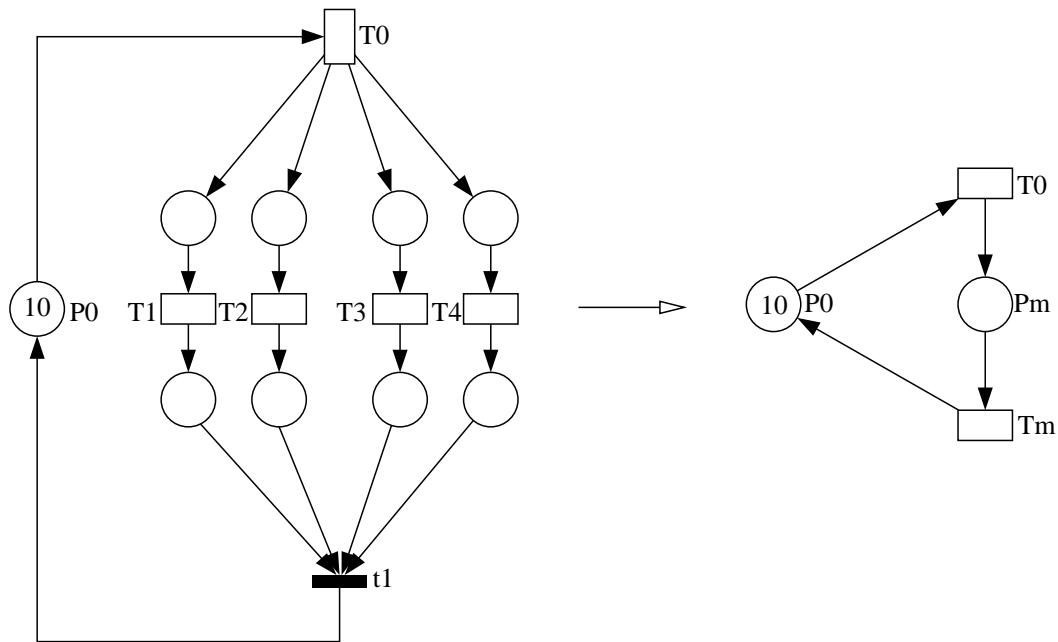


Abbildung 6.17: Anwendung des Paralleloperators für mehrere Transitionen

Der Transition T_1 ist die Schaltzeit $d_1 = 9$ und der Transition T_2 ist die Schaltzeit $d_2 = 10$ zugeordnet. Unter Verwendung dieser beiden Schaltzeiten wird die markierungsabhängige Schaltzeit $d_m^{k=x}$ der Transition T_m berechnet. In der linken oberen Spalte in Abbildung 6.14 sind diese Schaltzeiten in Abhängigkeit der Anzahl x der Marken auf der Stelle P_m dargestellt. Die erste Zeile in der in Abbildung 6.15 dargestellten Tabelle zeigt den Erwartungswert der Anzahl der Marken auf P_0 im aggregierten Modell. Er beträgt 0,109323. Mit diesem Wert werden die oben in Abbildung 6.18 aufgeführten Werte für das Originalmodell unter variierenden Schaltzeiten d_3 und d_4 der Transitionen T_3 und T_4 verglichen. Die in Abbildung 6.16 dargestellte Tabelle zeigt in der ersten Zeile den Wert der Wahrscheinlichkeit dafür, daß sich drei Marken auf der Stelle P_0 im aggregierten Modell befinden. Dieser Wert beträgt 0,000854. In der unteren Hälfte der in Abbildung 6.18 dargestellten Tabelle sind diese Wahrscheinlichkeiten für variierende Schaltzeiten der Transitionen T_3 und T_4 des Originalmodells angegeben. Die relative Abweichung der Werte im Originalmodell von denen des aggregierten Modells sind in der rechten Spalte aufgeführt.

d_3, d_4	$E\{\#P0\}$ im Original	rel. Fehler in %
1,1	0,108895	0,39
1,8	0,108539	0,72
9,10	0,103023	6,11
1,20	0,052629	107,72
11,20	0,052626	107,73
20,20	0,049860	119,26
d_3, d_4	$P\{\#P0=3\}$ im Original	rel. Fehler in %
1,1	0,000790	8,10
1,8	0,000763	11,92
9,10	0,000629	35,77
1,20	0,000118	623,73
11,20	0,000118	623,73
20,20	0,000091	838,46

Abbildung 6.18: Vergleich der Erwartungswerte und Wahrscheinlichkeiten

Abbildung 6.18 verdeutlicht, daß die Entscheidung, die beiden Transitionen mit der größten Schaltzeit auszuwählen, durch die ermittelten Werte als die mit der besten Annäherung an das Zeitverhalten des Originalmodells bestätigt wird. Das Originalmodell besitzt 14 641 Zustände, während das aggregierte Modell 11 Zustände beschreibt.

Mit Ausnahme des Problems, daß den Ausgangskanten eines MIMO-T-Elementes im MIMO-Graphen verschiedene Schaltraten zugeordnet sind, weisen alle in dieser Arbeit vorgeschlagenen Lösungen zur Aggregation verschiedener Strukturen in eine Transition mit markierungsabhängiger Schaltzeit einen sehr geringen Approximationsfehler bei der Analyse des Erwartungswertes oder des Durchsatzes auf. Selbst die eine Approximation beschreibende Formel für die Semantik des Paralleloperators ist von akzeptabler Güte. Die Ähnlichkeit der untersuchten Wahrscheinlichkeiten von Original- und aggregiertem Modell war unter der gewählten Methode der Angleichung der Erwartungswerte nicht zu erwarten und somit überraschend.

Die entscheidende Einschränkung für die Untersuchung komplexerer Modelle liegt darin, die Operatoren nicht kombinieren zu können. Diese Möglichkeit der Kombination muß einer nächster Schritt bei der Entwicklung dieser Aggregierungsmethode sein. Um Operatoren miteinander kombinieren zu können, muß eine Berechnungsvorschrift entwickelt werden, die eine Verschmelzung von Transitionen mit markierungsabhängigen Schaltzeiten beschreibt.

Eine weitere Möglichkeit der Anwendung der MIMO-Graphen basierten Aggregierung ist Inhalt des nächsten Kapitels.

6.2 Eine iterative numerische Analysemethode

In diesem Abschnitt wird die in Abschnitt 5.4 vorgeschlagene Zerlegungsmethode und anschließende MIMO-Graphen basierte Aggregation von Teilnetzen an einer aus der Literatur [81, 82, 23, 24, 110, 112, 111] bekannten iterativen Analysemethode angewendet. Die Methode wurde in [57] für die Anwendung auf spezielle stochastische Petrinetze erweitert. In [58, 59] wurde erstmals die Verwendung der MIMO-Graphen basierten Aggregation für diese iterative Analysemethode gezeigt. Die präsentierte Analysemethode berechnet den Durchsatz eines Petrinetzmodells approximativ. Dazu werden sogenannte *teilaggregierte* Gesamtmodelle und ein *vollständig aggregiertes* Gesamtmodell des zu analysierenden Petrinetzes erzeugt. In einem teilaggregierten Gesamtmodell ist *ein* Teilmodell im Original bewahrt, während alle anderen Teilmodelle aggregiert sind. In dem vollständig aggregierten Gesamtmodell sind alle Teilmodelle aggregiert. Jedem Teilmodell wird ein teilaggregiertes Gesamtmodell zugeordnet, so daß der Anzahl der Teilmodelle entsprechend viele teilaggregierte Gesamtmodelle erzeugt werden.

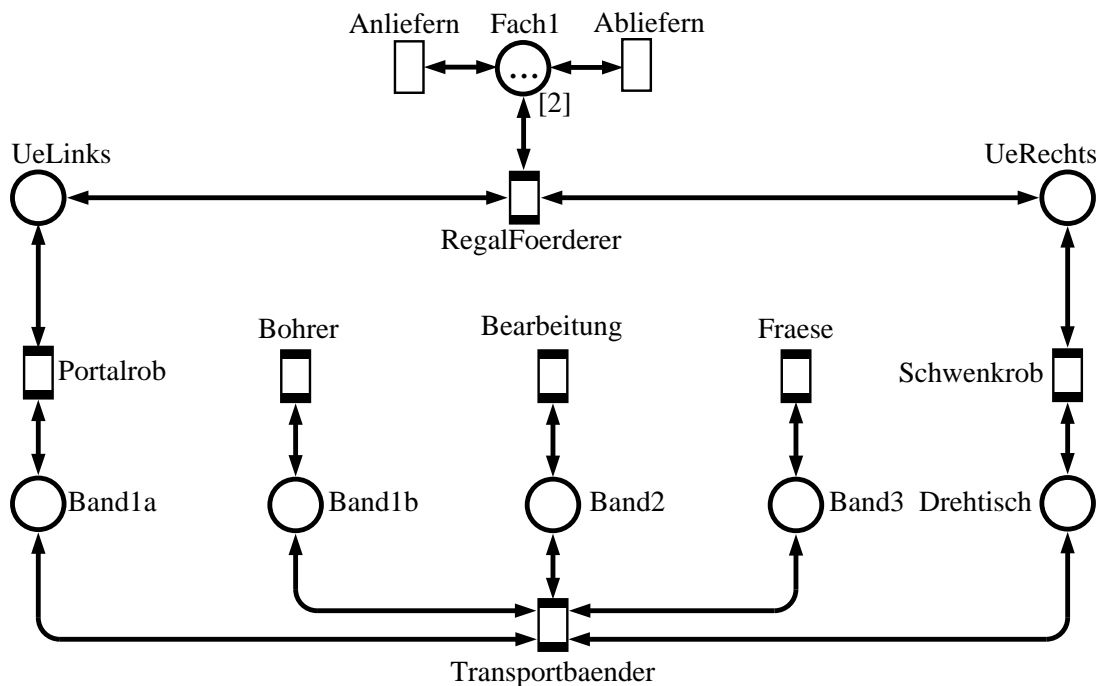


Abbildung 6.19: Fertigungssystemmodell

Betrachten wir zur Erläuterung teilaggregierter Gesamtmodelle das bereits bekannte Modell eines Fertigungssystems. Das teilaggregierte Gesamtmodell des Regalförder-Teilmodells beläßt dieses Teilmodell im detaillierten Original und verwendet die aggregierten Teilmodelle des Portalroboters, des aus den Teil-

modellen des Bohrers, der Bearbeitung, der Fräse und den Transportbändern zusammengefaßte Teilmodell und des Schwenkarmroboters.

6.2.1 Zerlegung und Aggregierung des Beispielmodells

In diesem Abschnitt wird für das Petrinetzmodell des Fertigungssystembeispiels aus Abbildung 6.19 unter der in Abschnitt 5.4.1 erläuterten Zerlegung die teil-aggregierten Gesamtmodelle und das vollständig aggregierte Gesamtmodell erzeugt.

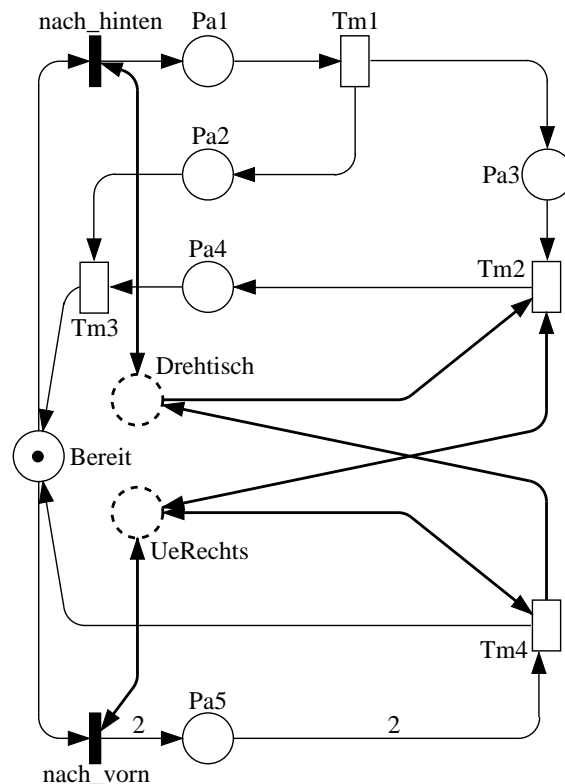


Abbildung 6.20: Aggregiertes Teilmodell des Schwenkarmroboters

Abbildung 6.20 zeigt das aggregierte Teilmodell des in Abbildung 5.23 auf Seite 94 im Original dargestellten Schwenkarmroboters. Die Transition Tm1 ist durch Verschmelzung der Transitionen D_dreheH1, G_senken2, G_schliess und G_heben1 entstanden. Die Schaltzeit von Tm1 wird durch die Anwendung des Sequenzoperators über die Schaltzeiten der genannten Transitionen bestimmt. Die Transition Tm2 ist aus der Verschmelzung der Transitionen SA_dreheH2, G_senken1, G_oeffnen und G_heben2 entstanden, deren Schaltzeiten ebenfalls mittels Sequenzoperator verknüpft werden. Die Transition Tm3 ist aus der Verschmelzung der Transitionen D_dreheV2, SA_dreheV1 und T1 entstanden.

Die Schaltzeiten von `D_dreheV2` und `SA_dreheV1` werden mittels Paralleloperator verknüpft und der Transition `Tm3` als neue Schaltzeit zugeordnet. Die Transition `Tm4` ist aus der Verschmelzung aller in Abbildung 5.23 auf der unteren Hälfte gezeigten Transitionen mit Ausnahme der zeitlosen Transition `nach_vorn` entstanden. Ihre Schaltzeit berechnet sich zunächst aus den mittels Sequenzoperator verknüpften Schaltzeiten der Transitionen `SA_drehH2`, `G_senken3`, `G_schliess2`, `G_heben3` und `SA_drehV2`. Diese neue Schaltzeit wird mittels Paralleloperator mit der Schaltzeit der Transition `D_drehH2` und dieser Term wiederum mittels Sequenzoperator mit den Schaltzeiten der Transition `G_senken4`, `G_oeffnen2`, `G_heben4` und `D_dreheV4` verknüpft. Die so berechnete Schaltzeit wird der Transition `Tm4` zugeordnet. Wie in Abschnitt 5.5.3 auf Seite 118 erläutert, ist diese Kombination der Operatoren für derart komplexe Strukturen noch nicht beliebig möglich. Im oben gezeigten Beispiel sind jedoch keine markierungsabhängigen Schaltzeiten nötig, da die Anzahl der Marken jeder Stelle im Vorbereich einer Transition mit Konzession entweder genau zwei auf der Stelle `Pa5` oder genau eins auf allen anderen Stellen beträgt.

Die Transitionen, die nur dünngezeichnete Ein- und Ausgangskanten besitzen, schalten nur Elementarmarken (siehe Abschnitt 2.4.1 ab Seite 10). Ihnen ist jeweils nur ein Übergang zugeordnet, der das Abziehen der Elementarmarken von den Stellen im Vorbereich und das Markieren mit Elementarmarken der Stellen im Nachbereich beschreibt. Die Transition `nach_hinten` im in Abbildung 5.23 dargestellten Originalmodell schaltet darüber hinaus genau dann, wenn sich auf dem Puffer `Drehtisch` die farbigen Marken `Palette.leer` und `A.gefraest` befinden. Die dickgezeichnete Hin- und Rückkante zur Stelle `Drehtisch` repräsentiert diese Bedingung. Das Schalten der Transition `G_heben2` des Originalmodells repräsentiert das Legen des gefrästen Werkstückes `A` auf die Palette und das Heben auf die Pufferstelle `UeRechts`. Der Übergang dieser Transition beschreibt demnach ein gemeinsames Entfernen der Marken `Palette.leer` und `A.gefraest` von der Stelle `Drehtisch` und das Markieren der Stelle `UeRechts` mit `A.fertig` (siehe auch Arbeitsplan des Originalmodells ab Seite 13). Da die Transition `nach_hinten` bei der Aggregation nicht verschmolzen wird, da sie eine zeitlose Transition mit einer Eingangskante von einem Puffer besitzt, bleibt nur die Verschmelzung der Transition `G_heben2` für den oberen Teil des Modells bei der Verschmelzung der Übergänge zu berücksichtigen. Da die Transition `G_heben2` jedoch mit keiner Transition verschmolzen wird, die ebenfalls farbige Marken schaltet, reduziert sich die Verschmelzung der Übergänge auf die Elementarmarken betreffenden Teile der Übergänge. Bei der Transformation des Originalteilmodells aus Abbildung 5.23 in einen MIMO-Graphen werden aus dem oberen Teil des Modells nur die Transitionen `nach_hinten`, `G_heben1`, `G_heben2` und `T1` als MIMO-Elemente übernommen, alle Stellen aus dem oberen Teil und alle weiteren Transitionen besitzen die SISO-Eigenschaft. Da die genannten MIMO-T-Elemente nicht weiter verschmolzen werden können, werden bei der Rücktransformation in ein Petrinetz die Stellen `Pa1`, ..., `Pa5` eingefügt. Somit

ändern sich für alle Transitionen aus dem oberen Teil des in Abbildung 6.20 dargestellten aggregierten Modells die Übergänge die Elementarmarken betreffend (siehe auch zur Zuordnung der Übergänge zu den Kanten des MIMO-Graphen in Abschnitt 5.3). Das bedeutet, daß bei der neuen Erstellung der Übergänge der Transitionen nach_hinten, Tm1, Tm2 und Tm3 die entfernten Stellen aus dem jeweiligen Vor- und Nachbereich durch die neuen Stellen ersetzt werden müssen. Die die farbigen Marken betreffenden Teile der Übergänge bleiben, wie bereits erwähnt, unverändert.

Abbildung 6.21 zeigt das Original- und aggregierte Teilmodell des Bohrers. Die Teilmodelle der Bearbeitung und der Fräse sind strukturell dem des Bohrers identisch.

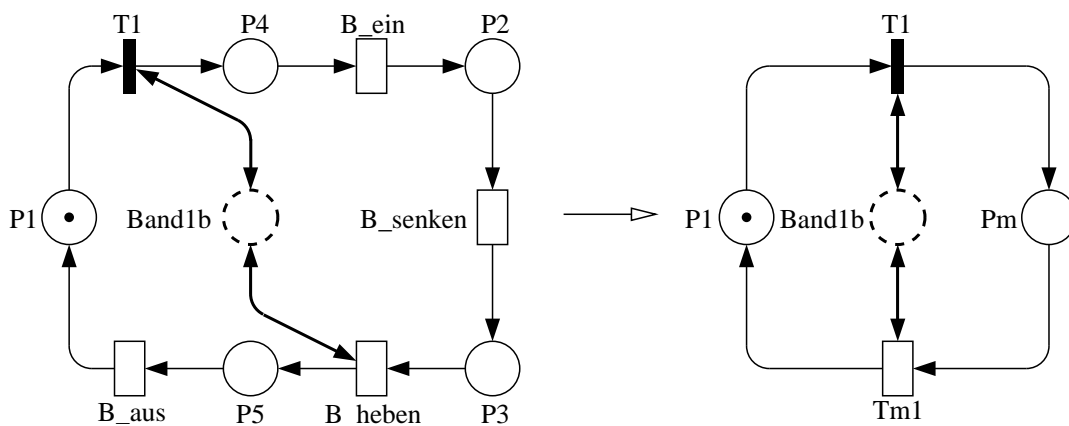


Abbildung 6.21: Original- und aggregiertes Teilmodell des Bohrers

Das Originalmodell des Portalroboters ist in Abbildung 6.22 dargestellt, während Abbildung 6.23 das aggregierte Teilmodell des Portalroboters zeigt.

Wie in Abschnitt 5.4.1 erläutert, werden nachfolgend die bei der automatischen Zerlegung identifizierten Teilmodelle des Bohrers, der Bearbeitung, der Fräse und der Transportbänder zu einem Teilmodell zusammengefaßt. Dieses Teilmodell ist aggregiert in Abbildung 6.24 auf Seite 146 dargestellt.

6.2.2 Iterative Analyse des Beispielmodells

Unter Verwendung des ersten teilaggregierten Gesamtmodells wird die iterative Analyse gestartet. Berechnet werden die Durchsätze der Ausgangstransitionen des im Originalen bewahrten Teilmodells des teilaggregierten Gesamtmodells. Als Ausgangstransitionen eines Teilmodells werden dabei alle die Transitionen bezeichnet, die eine Ausgangskante zu einem Puffer besitzen. Für das teilaggregierte Gesamtmodell des Regalförderlagers werden demnach die Durchsätze aller Ausgangstransitionen des Regalförder-Teilmodells berechnet. Der Durch-

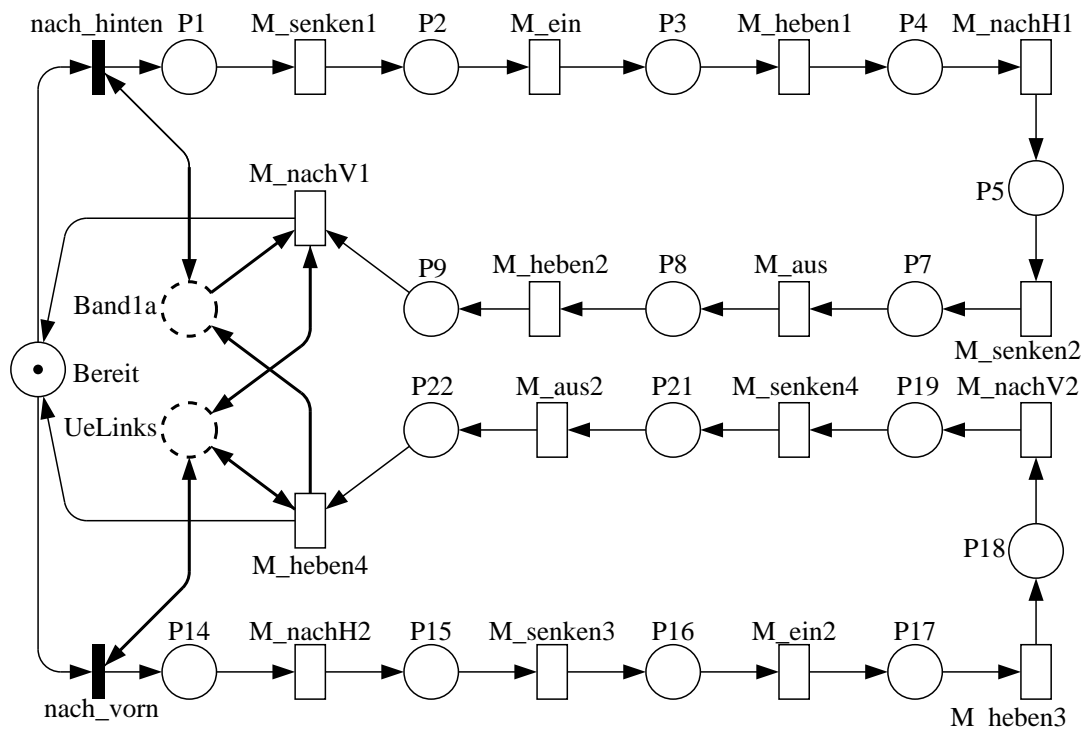


Abbildung 6.22: Teilmodell des Portalroboters

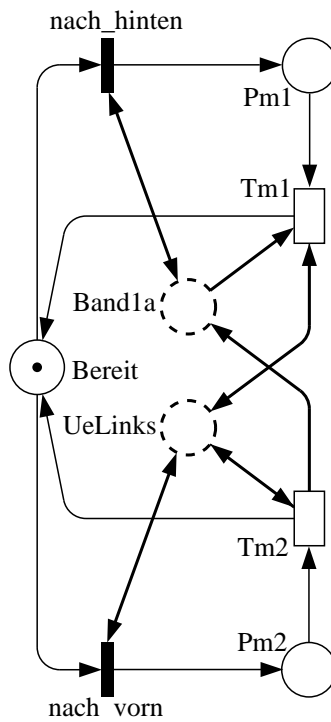


Abbildung 6.23: Aggregiertes Teilmodell des Portalroboters

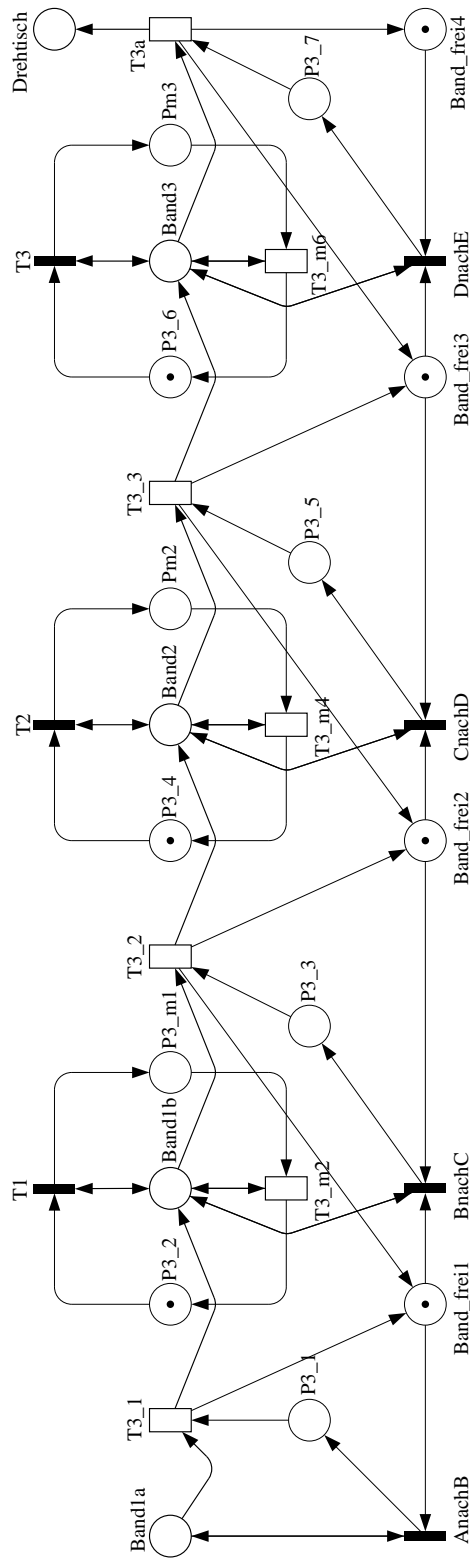


Abbildung 6.24: Aggregiertes zusammengefasstes Teilmodell des Bohrers, der Bearbeitung, der Fräse und der Transportbänder

satz einer Transition berechnet sich aus der Wahrscheinlichkeit, daß die Transition Konzession besitzt multipliziert mit der Schaltrate der Transition.

Im zweiten Schritt wird das vollständig aggregierte Gesamtmodell verwendet. Unter Verwendung dieses Modells werden iterativ die Schaltzeiten der Ausgangstransitionen des aggregierten Regalförder-Teilmodells derart verändert, daß die Durchsätze der Ausgangstransitionen des aggregierten Regalförder-Teilmodells im vollständig aggregierten Gesamtmodell denen der im ersten Schritt berechneten gleichen. In diesem Schritt werden somit neue Schaltraten der Ausgangstransitionen für das aggregierte Regalförder-Teilmodell ermittelt. In allen verbleibenden teilaggregierten Gesamtmodellen werden diese neuen Schaltraten den Ausgangstransitionen des darin aggregierten Regalförder-Teilmodells gesetzt.

In dem teilaggregierten Gesamtmodell des Portalroboters sind demnach die Ausgangstransitionen des darin aggregierten Regalförder-Teilmodells im folgenden Schritt bereits neu gesetzt. In diesem dritten Schritt werden die Durchsätze der Ausgangstransitionen des Portalroboter-Teilmodells berechnet.

Im vierten Schritt wird wiederum das vollständig aggregierte Gesamtmodell verwendet. In diesem Modell sind die Schaltraten des aggregierten Regalförder-Teilmodells bereits neu gesetzt. Iterativ werden wiederum durch verändern der Schaltraten der Ausgangstransitionen des Portalroboter-Teilmodells die Durchsätze dieser Transitionen denen der im dritten Schritt berechneten angeglichen. Die so ermittelten Schaltraten der Ausgangstransitionen des aggregierten Portalroboter-Teilmodells werden in allen anderen teilaggregierten Gesamtmodellen und im vollständig aggregierten Gesamtmodell gesetzt.

Mit diesem Verfahren werden sukzessive alle Durchsätze der Ausgangstransitionen der im Original bewahrten Teilmodelle der teilaggregierten Gesamtmodelle berechnet und mittels vollständig aggregiertem Gesamtmodell innerhalb der aggregierten Teilmodelle angeglichen.

Dieses iterative Analyseverfahren besteht somit aus zwei iterativen Prozessen. Die „große“ Iteration besteht darin, unter Verwendung der teilaggregierten Gesamtmodelle die Durchsätze *aller* aggregierten Teilmodelle denen der Originalteilmodelle anzugleichen. Die „kleine“ Iteration paßt den Durchsatz *eines* aggregierten Teilmodells dem des Originals an.

Tabelle 6.25 zeigt die Durchsätze der Ausgangstransitionen der Teilmodelle in den einzelnen Iterationsschritten.

Bereits im zweiten Iterationsschritt ändert sich nur noch die Schaltzeit der Ausgangstransition im aus den Teilmodellen des Bohrers, der Bearbeitung, der Fräse und den Transportbändern zusammengesetzten Teilmodell. Die Iteration konvergiert im dritten Schritt. Diese wenigen Iterationsschritte bis zur Konvergenz entsprechen unseren Erfahrungen bei der Anwendung der hier vorgestellten iterativen Analyse. Bis jetzt untersuchten wir kein Modell, für das das Verfahren nicht

	Teilaggr. Modell	Durchsatz der Ausgangstrans.	Neue Schaltzeit der Ausgangstrans. im vollaggr. Modell	Durchsatz der Ausgangstrans. im vollaggregierten Modell
1	RegalF	0,0154205	5	0,0153869
	PortalR	0,0154653	7,8	0,0154171
	Baender	0,000951519	995	0,000953346
	SchwenkR	0,000953372	4	0,000953408
2	RegalF	0,000953418	5	0,000953408
	PortalR	0,000953243	7,8	0,000953408
	Baender	0,000951588	998	0,000950689
	SchwenkR	0,000950653	4	0,000950689

Abbildung 6.25: Iterationsschritte für das Fertigungssystemmodell

konvergierte. Ein formaler Beweis dafür, daß dieses Verfahren stets konvergiert, bleibt jedoch ein offenes Problem.

Existieren mehrere Ausgangstransitionen in einem Teilmodell, so wird im vollaggregierten Modell für alle Ausgangstransitionen dieselbe Schaltzeit ermittelt. In dem Teilmodell des Portalroboters existieren bspw. mehrere Ausgangstransitionen (siehe Abbildungen 6.22 und 6.23). In dem teilaggregierten Modell des Portalroboters ist das Teilmodell des Portalroboters im Original bewahrt und alle anderen Teilmodelle sind aggregiert. Nachdem in einem ersten Iterationsschritt die neue Schaltzeit der Ausgangstransitionen des Förderregalteilmodells im vollaggregierten Modell ermittelt wurde, wird diese neue Schaltzeit im teilaggregierten Modell des Portalroboters gesetzt. Wie in Abbildung 6.22 ersichtlich, sind die Transitionen M_{nachV1} und M_{heben4} die Ausgangstransitionen des Portalroboterteilmodells. Für diese Transitionen werden die Durchsätze berechnet. Da beide Transitionen die Schaltzeit 1 besitzen, genügt es, die Wahrscheinlichkeiten zu berechnen, daß die Transitionen Konzession besitzen. Für die Transition M_{nachV1} wird demnach das Maß $P[\#Band1a(Palette.leer) > 0 \ \& \ \#P9=1]$ berechnet, für die Transition M_{heben4} das Maß $P[\#UeLinks(A.geliefert) > 0 \ \& \ \#P22=1]$. Wäre die Schaltzeit beider Transitionen ungleich eins, so würde die Schaltzeit durch obige Wahrscheinlichkeit geteilt. Für beide Wahrscheinlichkeiten ergibt sich der in Tabelle 6.25 dargestellte Wert 0,0154653. In der „kleinen“ Iteration werden anschließend unter Verwendung des vollaggregierten Modells die Schaltzeiten $d1$ und $d2$ der Transitionen $Tm1$ und $Tm2$ derart verändert, daß die Maße $P[\#Band1a(Palette.leer) > 0 \ \& \ \#Pm1=1]/d1$ und $P[\#UeLinks(A.geliefert) > 0 \ \& \ \#Pm2=1]/d2$ den Wert 0,0154653 aufweisen (siehe Abbildung 6.23). Wie eingangs erwähnt, werden dabei nicht zwei verschiedene Werte $d1$ und $d2$, sondern nur ein Wert $d1 = d2$ berechnet. Wie aus Tabelle 6.25 ersichtlich, ist für $d1 = d2 = 7,8$ der Durchsatz der Transitionen

0,0154171.

Die den teilaggregierten Modellen zugrunde liegenden Markowketten beschreiben 287, 213, 1932 und 287 andauernde Zustände. Dem vollaggregierten Modell liegt eine Markowkette mit 87 andauernden Zuständen zugrunde. Das Originalmodell hat 20964 andauernde Zustände. Der Durchsatz des Fertigungssystemmodells wird mittels $P[\#Fach1(A.fertig)=1]$ berechnet, da der Transition *Ab liefern* die Schaltzeit eins zugeordnet ist (siehe Abbildung 5.22). Für das Originalmodell beträgt der Wert dieses Maßes 0,000951402, während für das vollaggregierte Modell der Wert 0,000950689 berechnet wird. Das entspricht einem Approximationsfehler von nur 0,075%. Auch wenn das Verfahren nicht für alle von uns untersuchten Modelle einen so geringen Approximationsfehler aufweist, so verdeutlicht es doch die Anwendbarkeit der Methode. Für die meisten der von uns untersuchten Modelle lag der Approximationsfehler bei unter 3%.

Das in [82, 24, 111] vorgestellte Verfahren basiert auf der Aggregation mittels impliziter Stellen (siehe Abschnitt 5.2.4). Die Aggregation mittels impliziter Stellen ist wie die MIMO-Graphen basierte Aggregation konsistent. Für die Aggregation mittels impliziter Stellen existiert jedoch keine Methode, die Schaltzeiten der verbleibenden Transitionen neu zu berechnen. Während der Aggregation mittels impliziter Stellen wird demnach das Zeitverhalten des aggregierten Modells nicht dem des Originalmodells angepaßt. Diese Anpassung erfolgt erst durch die hier ebenfalls verwendete iterative Analyse. Da jedoch das Zeitverhalten bei den mittels impliziten Stellen aggregierten Teilmodellen z.T. sehr stark von dem der originalen Teilmodelle abweichen kann, werden in oben genannter Literatur die teilaggregierten Modelle anders als in dieser Arbeit vorgeschlagen. In dieser Arbeit besteht ein teilaggregiertes Modell aus *einem* originalen Teilmodell während alle anderen Teilmodelle aggregiert sind. In der in [81, 82, 23, 24, 110, 112, 111] vorgeschlagenen Methode besteht ein teilaggregiertes Modell aus einem aggregierten Teilmodell während alle anderen im Original bewahrt bleiben. Die Approximationsergebnisse sind mit denen der hier vorgeschlagenen Methode vergleichbar, jedoch unterscheiden sich die teilaggregierten Modelle hinsichtlich ihrer Größe. Ist bspw. in einem teilaggregierten Modell bei der wie in Abschnitt 5.4.1 erläuterten und in Abschnitt 5.4.2 verwendeten Zerlegung des Fertigungssystembeispiels aus Abbildung 5.22 nur das Teilmodell des Portalroboters aggregiert und alle anderen Teilmodelle im Original bewahrt, hat die diesem teilaggregierten Modell zugrunde liegende Markowkette 15252 Zustände. Ist nur der Portalroboter im Original und alle anderen Teilmodelle aggregiert ist die diesem teilaggregierten Modell zugrunde liegende Markowkette 213 andauernde Zustände groß.

Kapitel 7

Zusammenfassung

Petrinetze sind ein sehr geeignetes Konzept zur Modellierung komplexer verteilter Systeme. Trotzdem sind sie im außerakademischen Bereich nur von geringer Bedeutung. Die Vorteile bei der Anwendung von Petrinetzen sind vielfältig.

Petrinetze sind mit einer sehr einfachen, intuitiv verständlichen grafischen Veranschaulichung darstellbar. Jedem Petrinetzmodell liegt eine formale Semantik zugrunde. Selbst komplexere Strukturen, wie Alternativen und Synchronisationen sind eindeutig bestimmt. Außerdem sind Modelle, in denen der Zeitaspekt von Bedeutung ist und sogar Modelle mit stochastischen Parametern wohl definiert. Darüber hinaus bietet die Petrinetztheorie eine Vielzahl qualitativer und quantitativer Analysemethoden. Jede wichtige Fragestellung über technische oder organisatorische Systeme läßt sich mit Analysemethoden auf der Basis von Petrinetzen beschreiben und beantworten. Petrinetze grenzen sich mit diesen Eigenschaften von weiter verbreiteten Konzepten, wie z.B. *Ereignisprozeßketten* oder *message sequence charts* ab. Der wohl revolutionärste Vorteil von Petrinetzen liegt jedoch im Konzept der Darstellung der Kausalität *lokaler* Ereignisse und *lokaler* Bedingungen. Auf eine „erzwungene“ totale Ordnung unabhängiger Ereignisse und Bedingungen wird verzichtet. Bereits mit kleineren Petrinetzmodellen lassen sich leicht entsprechende Vorteile gegenüber *Mealy*- oder *Moore*-Automaten begünden. Die Anwendbarkeit von Petrinetzen ist nicht auf wenige Anwendungsgebiete beschränkt.

Daß Petrinetze nicht verbreiteter sind, liegt hauptsächlich an fehlenden Softwarewerkzeugen. Zum einen gibt es nur wenige Petrinetze verarbeitende Programme, zum Anderen sind diese oft auf einen kleinen Ausschnitt der Analysemöglichkeit von Petrinetzen beschränkt. Es gibt separate Werkzeuge im Bereich des *model checking*, für die numerische Analyse stochastischer Petrinetze und für die qualitative Analyse allgemeiner Petrinetze. Vorschläge zur Standardisierung von Dateiformaten oder aber generische Petrinetzwerkzeuge wie bspw. TimeNET oder

Möbius [100] sind sehr nützlich und finden doch kaum Unterstützung. Darüber hinaus bietet außer einem Ansatz in TimeNET keines der bekannten Petrinetzwerkzeuge eine Schnittstelle für den in Petrinetzen unerfahrenen Modellierer. Eine Modellierung mit Petrinetze unterlegten Symbolen ist weniger eine Frage mangelnder Konzepte denn mangelnder Ressourcen für die Implementierung. Daß sich die auch in dieser Arbeit vorgestellten speziellen farbigen Petrinetze zur Modellierung von Fertigungssystemen aber auch von *workflow management* Systemen eignen, ist in mehreren Arbeiten gezeigt worden [47, 48, 49]. Doch auch für diese Netzklasse fehlt die anwendungsfreundliche Schnittstelle zum Nutzer, die bspw. eine Modellierung der Arbeitspläne erleichtert, indem die Vorgaben des Strukturplans ausgenutzt und dem Modellierer angeboten werden.

In einem großen Vorteil der Petrinetze, deren Analysierbarkeit, sind auch deren Nachteile begründet. In sehr vielen Analysemethoden läßt sich das Prinzip der lokalen Kausalität von Bedingungen und Ereignissen nicht ausnutzen. Stattdessen basieren viele Analysemethoden auf einer totalen Ordnung über die Bedingungen und Ereignisse des gesamten Modells, die zu einer sogenannten „Explosion“ des Zustandsraumes führen. Für die numerische Analyse werden gar Matrizen mit der Dimension dieses Zustandsraumes benötigt. Dieses Problem und Vorschläge zur Vermeidung und Überwindung werden in dieser Arbeit diskutiert und vorgestellt. Approximative Analysemethoden werden angewandt, wenn Fragestellungen der Optimierung zu beantworten sind. Im Bereich der Fertigungssysteme sind Fragestellungen bspw. derart, ob es günstiger ist, mehr Lagerkapazitäten oder schnellere Maschinen zur Verfügung zu stellen. Zu einer genauen Berechnung eines Leistungsparameters wird im Falle der Nichtanwendbarkeit der exakten numerischen Analyse anstelle approximativer Verfahren auf die ereignisdiskrete Simulation des Petrinetzmodells zurückgegriffen. Für die Simulation ist im Gegensatz zu den meisten approximativen Verfahren eine Abschätzung des Fehlers möglich.

Das zweite ungelöste Problem bei der numerischen Analyse stochastischer Petrinetze ist neben der Zustandsraumexplosion das der zeitlichen Komplexität des Modells. Modelle mit generellen Schaltzeiten zugeordneten Transitionen sind nur unter starken Einschränkungen analysierbar. Dieses Problem blieb in dieser Arbeit unbetrachtet.

Die wohl einzige Überwindungsmöglichkeit des Problems der Zustandsraumexplosion liegt darin, den totalen Zustandsraum nicht aufspannen zu müssen. Anstatt Wahrscheinlichkeiten globaler Zustände, genügt es in vielen Fällen, Wahrscheinlichkeiten lokaler Zustände zu berechnen. Eine Abbildung des total geordneten Zustandsraumes auf eine Markowkette ließe sich durch eine Abbildung des halbgeordneten Zustandsraumes auf eine Markowkette ersetzen. Eine solche Methode ist jedoch nicht in Sicht. Die auch in dieser Arbeit beschriebenen Kronecker-Algebra basierten Methoden gehen jedoch im Ansatz diesen Weg. Darin werden lokale und globale Informationen getrennt und getrennt ausge-

nutzt. Da jedoch selbst bei Anwendung Kronecker-Algebra basierter Methoden die Wahrscheinlichkeiten globaler Zustände berechnet werden, ist eine Erstellung und Lösung eines Gleichungssystems nötig mit der den globalen Zuständen entsprechenden Anzahl von Unbekannten. Geschlossene Formeln zur Berechnung der Zustandswahrscheinlichkeiten von Petrinetzen könnten eine weitere Möglichkeit zur Überwindung des Problems der Zustandsraumexplosion sein, auch wenn diese neue, numerische Probleme aufwerfen.

In dieser Arbeit werden erstmals Formeln zur strukturbasierten Aggregierung zeitbehafteter Transitionen vorgestellt, die das Durchsatzverhalten des aggregierten Modells dem des Originalmodells annähern. Dadurch existiert erstmals eine Aggregierung, die umfangreichen Bedingungen zur Anwendung auf stochastische Modelle genügt. Die MIMO-Graphen basierte Aggregierung bewahrt sowohl strukturelle Eigenschaften des Modells, wie Lebendigkeit und Beschränktheit und sie bewahrt darüber hinaus zeitbehaftete Eigenschaften des Modells, wie den Durchsatz. Durch die sowohl automatische Zerlegung großer Modelle als auch deren automatische Aggregierung erweist sich die MIMO-Graphen basierte Aggregierung als ein geeignetes Verfahren zur approximativen Analyse großer Petrinetzmodelle.

Die hier vorgestellte, neue Aggregierungsmethode wurde ebenso wie die in dieser Arbeit beschriebene neue Methode zur Berechnung von Stellen-Invarianten, Fallen und Siphons in `TimeNET` integriert. Darüber hinaus ist die Implementierung der numerischen Analyse von stochastischen Petrinetzen auf der Basis der in dieser Arbeit beschriebenen Kronecker-Operatoren weit fortgeschritten und bereits für farblose Petrinetzmodelle anwendbar. Ebenso wurde die im Anwendungskapitel vorgestellte iterative Analysemethode sowohl unter Aggregierung mittels impliziter Stellen als auch mittels MIMO-Graphen basierter Aggregierung in das Analysewerkzeug integriert.

7.1 Ausblick

Weitere der in dieser Arbeit vorgestellten neueren Methoden auf der Basis bspw. von Matrixdiagrammen müssen in `TimeNET` integriert werden. Die Integration Kronecker-algebraischer Methoden ist ein Schritt in die Richtung, den Anschluß an effizientere Werkzeuge wie `SMART` [120] oder `Möbius` [100] wieder herzustellen.

Die in dieser Arbeit präsentierten Formeln zur Berechnung der neuen Schaltzeiten verschmolzener Transitionen können nur als erste Ergebnisse verstanden werden. Die Entwicklung von Formeln zur Verschmelzung von exponentiellen Transitionen mit markierungsabhängigen Schaltzeiten ist notwendig, um die hier präsentierten Formeln geeignet kombinieren zu können. Andererseits ist eine

Entsprechung der Formeln für die Anwendung auf Transitionen mit genereller Schaltzeit erforderlich, um realistischen Ansprüchen zu genügen.

Abbildungsverzeichnis

2.1	Ein Nachfrage-Angebot-Beispiel	2
2.2	Petrinetzmodell des Nachfrage-Angebot-Beispiels	3
2.3	Schalten einer Transition	3
2.4	Erreichbarkeitsgraph des Nachfrage-Angebot-Beispiels	5
2.5	Vergleich Komplementärstelle – hemmende Kante	6
2.6	Unbeschränktes Modell mit hemmender Kante und möglicher Interpretation	6
2.7	Abräumkanten mit Weiterverwendung und Vernichtung	7
2.8	Farbloses Petrinetzmodell – Ausschnitt einer Fertigungsanlage für Autos	8
2.9	Farbiges Petrinetzmodell – Ausschnitt einer Fertigungsanlage für Autos	9
2.10	Übergangstabelle der Transition Montage	9
2.11	Strukturmodell der höchsten Hierarchiestufe	11
2.12	Verfeinertes Modell des Schwenkarmroboters	12
2.13	Arbeitsplanmodell für ein Werkstück auf der obersten Hierarchiestufe	13
2.14	Detailliertes Arbeitsplanmodell für den Schwenkarmroboter	14
2.15	Verwendung zeitloser Transitionen zur Modellierung eines beschränkten und kritischen Bereichs	18
2.16	Verwendung zeitloser Transitionen zur Modellierung einer Alternative	19
2.17	Farbloses stochastisches Petrinetzmodell eines Fertigungssystems	20
2.18	Schaltzeiten und Schaltwahrscheinlichkeiten des Fertigungssystembeispiels	22
2.19	Anzahl der andauernden Zustände und Resultate für variierende N des Fertigungssystembeispiels	22

3.1	Petrinetzmodell einer Fertigungsanlage	26
3.2	Petrinetzmodell einer Fertigungsanlage mit geänderten Beschriftungen	31
3.3	Inzidenzmatrix des Petrinetzmodells aus Abbildung 3.2	32
3.4	Markierungsgleichung testet nur notwendige Bedingung	33
3.5	Transponierte Inzidenzmatrix des Petrinetzmodells aus Abb. 3.2	34
3.6	Lösungsmatrix \mathbf{I}_N des Petrinetzmodells aus Abbildung 3.2	35
3.7	Lösungsmatrix \mathbf{I}_N ohne negative Einträge	35
3.8	Transponierte Inzidenzmatrix \mathbf{N}^T des Petrinetzmodells aus Abb. 3.2	37
3.9	Überführung der transponierten Inzidenzmatrix \mathbf{N}^T in die hermitesche Normalform \mathbf{H}_N mittels Transformationsmatrix \mathbf{V}	38
3.10	\mathbb{Z} -Basis \mathbf{Z}_N von \mathbf{N}^T	39
3.11	Berechnung der die Stellen-Invarianten beschreibenden Matrix \mathbf{I}_N aus der \mathbb{Z} -Basis \mathbf{Z}_N und der Lösungsmatrix \mathbf{U} des Ungleichungssystems $\mathbf{Z}_N \cdot \mathbf{u} \geq 0$	39
3.12	Doppelte Beschreibung eines Polyeders \mathcal{P}	40
3.13	Geometrische Darstellung des Beispiels aus Abbildung 3.12	41
3.14	Geometrische Darstellung des Einfügens einer Einschränkung	41
3.15	Vergleich der Dauern zur Berechnung von Stellen-Invarianten	43
4.1	Beispiel – Gesamtmodell und Erreichbarkeitsgraph	46
4.2	Beispiel – Teilmodelle und zugehörige Erreichbarkeitsgraphen	47
4.3	Übergangsratenmatrizen der Teilmodelle \mathbf{A} und \mathbf{B}	47
4.4	Übergangsratenmatrizen für die lokalen Ereignisse der Teilmodelle \mathbf{A} und \mathbf{B}	48
4.5	Übergangswahrscheinlichkeitsmatrizen für das Synchronisationereignis der Teilmodelle \mathbf{A} und \mathbf{B}	48
4.6	Mögliche Zerlegung für die Kronecker-Algebra basierte Analyse	54
4.7	Beispiel Anwendung von Multi-Terminal BDDs	57
4.8	Beispiel – Modell und Erreichbarkeitsgraph	57
4.9	Verwaltung der Übergangsratenmatrix des Beispiels aus Abb. 4.8	58
5.1	Ergebnis erster Schritt: MIMO-Graph	63
5.2	Ergebnis zweiter Schritt: aggregierter MIMO-Graph	64
5.3	Ergebnis dritter Schritt: aggregiertes Petrinetzmodell	65

5.4	Kantenkardinalität im MIMO-Graph	68
5.5	Aggregierungsregeln	69
5.6	SISO-Elemente in MIMO-Graphen	71
5.7	Löschen von SISO-Elementen	73
5.8	Verschmelzung von Transitionen	75
5.9	Entstehen einer Schlinge bei der Verschmelzung	76
5.10	Verschmelzung von Stellen	78
5.11	Lebendigkeit von Transitionen	79
5.12	Erste Reduktionsregel aus [123]	81
5.13	Vierte Reduktionsregel aus [123]	82
5.14	Fünfte und sechste Reduktionsregel aus [123]	83
5.15	Siebte und achte Reduktionsregel aus [123]	84
5.16	Beispiel	85
5.17	Beispiel	86
5.18	P_2 ist eine implizite Stelle	87
5.19	Falsche Zustände	89
5.20	T_1 besitzt MIMO-Eigenschaft	89
5.21	Beispiel Verschmelzung farbiger Transitionen	90
5.22	Petrinetzmodell einer Fertigungsanlage	93
5.23	Verfeinertes Modell des Schwenkarmroboters	94
5.24	Beispiel zeitliche Äquivalenz	95
5.25	Zuordnung der Schaltraten zu Kanten	98
5.26	Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-P-Elementen	98
5.27	Anwendung des Sequenzoperators bei folgendem MIMO-T-Element	99
5.28	Anwendung des Sequenzoperators bei der Verschmelzung von MIMO-T-Elementen	100
5.29	Anwendung des Sequenzoperators bei mehrfachem Verschmelzen von MIMO-T-Elementen	100
5.30	Ein weiterer Grund für den Approximationsfehler	101
5.31	Anwendung des Alternativoperators	102
5.32	Spezielle Anwendung des Paralleloperators	103
5.33	Referenzmodell zur Ermittlung des Sequenzoperators	104

5.34	Erreichbarkeitsgraphen des Original- und des aggregierten Modells	105
5.35	Erreichbarkeitsgraphen des Originalmodells für beliebiges \mathbb{N}	106
5.36	Erreichbarkeitsgraphen des aggregierten Modells für beliebiges \mathbb{N} und markierungsabhängige Schaltraten	108
5.37	Aggregation von n sequentiellen Transitionen	109
5.38	Referenzmodell zur Ermittlung der Semantik des Alternativopera- tors	112
5.39	Referenzmodell für n Alternativen	113
5.40	Referenzmodell zur Ermittlung der Semantik des Paralleloperators	114
5.41	Abhängigkeit $d_m^{k=2}$ von d_0	115
5.42	Vergleiche der berechneten und exakten $d_m^{k=x}$	117
5.43	Referenzmodell für n parallele Transitionen	118
5.44	Kombinationsmöglichkeiten und -grenzen der Operatoren	119
5.45	MDD für die initiale Markierung des Beispiels aus Abbildung 4.1	122
5.46	Erstellung des MDD basierten Erreichbarkeitsgraphen	123
6.1	Anwendung des Sequenzoperators	127
6.2	Beispiel – Werte für $d_m^{k=x}$	127
6.3	Anwendung des Sequenz- und Alternativoperators	128
6.4	Anwendung des Sequenz- und Alternativoperators	129
6.5	Beispiel – Werte für $d_{am}^{k=x}$	129
6.6	Problematische Anwendung des Sequenzoperators – 1	130
6.7	Werte für $d_m^{k=x}$ in Abhängigkeit von d_3	132
6.8	Vergleich der Durchsätze der Transition T_0	132
6.9	Problematische Anwendung des Sequenzoperators – 2	133
6.10	Beispiel – Werte für $d_m^{k=2x}$	134
6.11	Vergleich der Erwartungswerte	135
6.12	Anwendung des Paralleloperators	136
6.13	Vergleich der Erwartungswerte und Wahrscheinlichkeiten	136
6.14	Werte für $d_m^{k=x}$ in Abhängigkeit von d_2	137
6.15	Vergleich der Erwartungswerte	138
6.16	Vergleich der Wahrscheinlichkeiten	138
6.17	Anwendung des Paralleloperators für mehrere Transitionen	139

6.18	Vergleich der Erwartungswerte und Wahrscheinlichkeiten	140
6.19	Fertigungssystemmodell	141
6.20	Aggregiertes Teilmodell des Schwenkarmroboters	142
6.21	Original- und aggregiertes Teilmodell des Bohrers	144
6.22	Teilmodell des Portalroboters	145
6.23	Aggregiertes Teilmodell des Portalroboters	145
6.24	Aggregiertes zusammengefaßtes Teilmodell des Bohrers, der Bearbeitung, der Fräse und der Transportbänder	146
6.25	Iterationsschritte für das Fertigungssystemmodell	148

Literaturverzeichnis

- [1] W. v. d. Aalst, A. Hirschall und E. Verbeek, An alternative way to analyze workflow graphs, in: *Proc. 13th Int. Conf. Advanced Information Systems Engineering, CAiSE* (2002, Springer-Verlag) .
- [2] M. Ajmone Marsan, G. Balbo, G. Chiola, G. Conte, S. Donatelli und G. Franceschinis, An Introduction to Generalized Stochastic Petri Nets, *Microelectronics and Reliability* **31** (4) (1991) 699–725.
- [3] M. Ajmone Marsan und G. Chiola, On Petri Nets with Deterministic and Exponentially Distributed Firing Times, in: G. Rozenberg, Ed., *Advances on Petri Nets, Lecture Notes on Computer Science*, Bd. 266 (Springer Verlag, 1987) 132–145.
- [4] M. Ajmone Marsan, G. Conte und G. Balbo, A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems, *ACM Transactions on Computer Systems* **2** (2) (May 1984) 93–122.
- [5] H. Ammar und S. Rezaul Islam, Time scale decomposition of a class of generalized stochastic Petri net models, *IEEE Transactions on Software Engineering* **15** (6) (June 1989) 809–820.
- [6] R. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Prado und F. Somenzi, Algebraic decision diagrams and their applications, *Formal Methods in System Design* **10** (2/3) (1997) 171–206.
- [7] G. Balbo, G. Chiola, G. Franceschinis und G. Molinar Roet, On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Nets, in: *2nd Int. Workshop on Petri Nets and Performance Models* (1987, IEEE-CS Press.) 136–145.
- [8] M. L. Balinski, An algorithm for finding all vertices of convex polyhedral sets, *SIAM* **9** (1961) 72–78.
- [9] G. Berthelot, Checking properties of nets using transformations, in: G. Rozenberg, Ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 222, (1985, Springer Verlag) 19–40.

- [10] G. Berthelot, Transformations and decompositions of nets, in: W. Brauer, W. Reisig und G. Rozenberg, Ed., *Petri Nets: Central Models and Their Properties I, Lecture Notes in Computer Science*, Bd. 254, (1986, Springer Verlag) 359–376.
- [11] G. Berthelot, G. Roucairol und R. Valk, Reduction of nets and parallel programs, in: W. Brauer, Ed., *Net Theory and Applications, Lecture Notes in Computer Science*, Bd. 84, (1980, Springer Verlag) 277–290.
- [12] A. Blakemore, The cost of eliminating vanishing markings from generalized stochastic Petri nets, in: *3rd International Workshop on Petri Nets and Performance Models* (Kyoto, Japan, 1989) 85–92.
- [13] R. E. Bryant, Graph-based algorithms for boolean functions, *IEEE Transactions on Computers* **C-35** (12) (1986) 1035–1044.
- [14] R. E. Bryant, Symbolic boolean manipulation with ordered binary-decision diagrams, *ACM Computer Survey* **24** (3) (1992) 393–418.
- [15] P. Buchholz, *Die strukturierte Analyse Markovscher Modelle, Informatik Fachberichte*, Bd. 282 (Springer, 1991).
- [16] P. Buchholz, A Hierarchical View of GCSPNs and Its Impact on Qualitative and Quantitative Analysis, *Journal of Parallel and Distributed Computing* **15** (1992) 207–224.
- [17] P. Buchholz, Hierarchical high level Petri nets for complex system analysis, in: R. Valette, Ed., *15th Int. Conference on Application and Theory of Petri Nets, Lecture Notes on Computer Science*, Bd. 815, (1994, Springer Verlag) 119–138.
- [18] P. Buchholz, Structured Analysis Approaches for Large Markov Chains – A Tutorial, in: *Performance'96* (Lausanne, Schweiz, 1996) Tutorial.
- [19] P. Buchholz, G. Ciardo, S. Donatelli und P. Kemper, Complexity of Kronecker operations on sparse matrices with applications to solution of Markov models, Technischer Bericht 97-66, Institute for Computer Applications in Science and Engineering, 1997.
- [20] P. Buchholz und P. Kemper, Hierarchical reachability graph generation for Petri nets, Forschungsbericht 660, Universität Dortmund, 1997.
- [21] J. Burch, E. Clarke und D. Long, Symbolic model checking with partitioned transition relations, in: A. Halaas und P. Denyer, Ed., *International Conference on Very Large Scale Integration* (Edinburgh, Schottland, 1991) .
- [22] J. Burch, E. Clarke, K. McMillan, D. Dill und L. Hwang, Symbolic model checking: 10^{20} states and beyond, *Information and Computation* **98** (2) (1992) 142–170.

- [23] J. Campos, J. M. Colom, H. Jungnitz und M. Silva, A General Iterative Technique for Approximate Throughput Computation of Stochastic Marked Graphs, in: *Proc. 5th Int. Workshop on Petri Nets and Performance Models* (Toulouse, 1993) 138–147.
- [24] J. Campos, J. Colom, H. Jungnitz und M. Silva, Approximate Throughput Computation of Stochastic Marked Graphs, *IEEE Transactions on Software Engineering* **20** (7) (Juli 1994) 526–535.
- [25] W.-L. Cao und W. J. Stewart, Iterative Aggregation/Disaggregation Techniques for Nearly Uncoupled Markov Chains, *Journal of the ACM* **32** (3) (Juli 1985) 702–719.
- [26] G. Chiola, C. Dutheillet, G. Franceschinis und S. Haddad, On Well-Formed Coloured Nets and their Symbolic Reachability Graph, in: *11th International Conference on Application and Theory of Petri Nets* (1990) 387–410.
- [27] G. Chiola, C. Dutheillet, G. Franceschinis und S. Haddad, Stochastic well-formed coloured nets for symmetric modelling applications, *IEEE Transactions on Computers* **42** (11) (1993) 1343–1360.
- [28] G. Chiola, G. Franceschinis, R. Gaeta und M. Ribaud, GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets, *Performance Evaluation* **24** (1–2) (1995) 47–68.
- [29] G. Ciardo, What a structural world, in: R. German und B. Haverkort, Ed., *Petri Nets and Performance Models (PNPM'01)* (2001, IEEE CS Press.) 3–16.
- [30] G. Ciardo, R. German und C. Lindemann, A characterization of the stochastic process underlying a stochastic Petri net, *IEEE Transactions on Software Engineering* **20** (1994) 506–515.
- [31] G. Ciardo, G. Lüttgen und R. Siminiceanu, Efficient symbolic state-space construction for asynchronous systems, in: M. Nielson und D. Simpson, Ed., *21th International Conference on Application and Theory of Petri Nets, Lecture Notes on Computer Science*, Bd. 1825, (2000, Springer-Verlag) 103–122.
- [32] G. Ciardo, G. Lüttgen und R. Siminiceanu, Saturation: an efficient iteration strategy for symbolic state-space generation, in: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001)* (2001, Springer-Verlag) 328–342.
- [33] G. Ciardo und A. Miner, A data structure for the efficient Kronecker solution of GSPNs, in: P. Buchholz, Ed., *8th International Workshop on Petri Nets and Performance Models (PNPM'99)* (Zaragoza, Spanien, 1999) 22–31.

- [34] G. Ciardo und M. Tilgner, On the use of Kronecker operators for the solution of generalized stochastic Petri nets, Technischer Bericht 96-35, Institute for Computer Applications in Science and Engineering, 1996.
- [35] G. Ciardo und K. Trivedi, A decomposition approach for stochastic Petri net models, in: *4th Int. Workshop on Petri Nets and Performance Models* (1991, IEEE-CS Press.) 74–83.
- [36] E. Clarke, E. Emerson und A. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, *ACM Transactions on Programming Languages and Systems* **8** (2) (1986) 244–263.
- [37] E. Clarke, O. Grumberg und D. Long, Model checking and abstraction, in: *19th Ann. ACM Symposium on Principles of Programming Languages* (1992) .
- [38] J. Coleman, W. Henderson und P. Taylor, A convolution algorithm for calculating exact equilibrium distributions in resource allocation problems with moderate user interference, *IEEE Transactions on Communications* **42** (2/3/4) (February/March/April 1994) 1106–1111.
- [39] J. Colom und M. Silva, Improving the Linearly Based Characterization of P/T Nets, in: G. Rozenberg, Ed., *Advances in Petri Nets 1990, Lecture Notes in Computer Science*, Bd. 483, (1990, Springer-Verlag) 113–146.
- [40] J. M. Colom und M. Silva, Conex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows, in: *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 483, (1990, Springer-Verlag) .
- [41] F. Commoner, Deadlocks in Petri nets, Technischer Bericht CA 7206-2311, Applied Data Research Inc., Wakefield, Massachusetts, 1972.
- [42] P. Courtois, *Decomposability. Queueing and computer system Applications*, ACM Monograph Series (Academic Press, 1977).
- [43] G. B. Dantzig, *Linear programming and extensions* (Princeton University Press, 1963).
- [44] M. Davio, Kronecker Products and Shuffle Algebra, *IEEE Transactions on Computers* **C-30** (2) (1981) 1099–1109.
- [45] D. Deavours und W. Sanders, An efficient Disk-based tool for solving large Markov models, *Performance Evaluation* **33** (1998) 67–84.
- [46] D. Deavours und W. Sanders, „On-the-fly“ solution techniques for stochastic Petri nets and extensions, *IEEE Transactions on Software Engineering* **24** (10) (1998) 889–902.

- [47] J. Dehnert, J. Freiheit und A. Zimmermann, Modeling and Performance Evaluation of Workflow Systems, in: *4th. World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000)*, Bd. VIII (Orlando, USA, 2000) 632 – 637.
- [48] J. Dehnert, J. Freiheit und A. Zimmermann, Workflow Modeling and Performance Evaluation with Colored Stochastic Petri Nets, in: *AAAI Spring Symposium - Bringing Knowledge to Business Processes* (Stanford, USA, 2000) 139 – 141.
- [49] J. Dehnert, J. Freiheit und A. Zimmermann, Modelling and evaluation of time aspects in business processes, *The Journal of the Operational Research Society* (2002). zur Veröffentlichung akzeptiert.
- [50] J. Desel, Basic linear algebraic techniques for place/transition nets, in: *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. I: Basic Models, (1990, Springer-Verlag) .
- [51] J. Desel, *Petrinetze, lineare Algebra und lineare Programmierung* (Teubner Verlag, 1998).
- [52] E. Dijkstra, Hierarchical ordering of sequential processes, *Acta Informatica* **1** (1971) 115–138.
- [53] S. Donatelli, Superposed generalized stochastic Petri nets: definition and efficient solution, in: R. Valette, Ed., *Application and Theory of Petri Nets, Lecture Notes in Computer Science*, Bd. 815, (1994, Springer-Verlag) 258–277.
- [54] S. Donatelli und P. Kemper, Integrating synchronization with priority into a Kronecker representation, *Performance Evaluation* **44** (2001) 1–4.
- [55] J. Esparza und S. Melzer, Checking system properties via integer programming, in: Nielson und H. Riis, Ed., *ESOP'96, Lecture Notes in Computer Science*, Bd. 1058, (1996, Springer-Verlag) 250–264.
- [56] J. Ezpeleta, J. Couvreur und M. Silva, A new technique for finding a generating family of siphons, traps and s/t-components. Application to colored Petri nets, in: G. Rozenberg, Ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 674, (1993, Springer-Verlag) 126–147.
- [57] J. Freiheit und A. Zimmermann, Extending a response time approximation technique to colored stochastic Petri nets, in: *4th Int. Workshop on Performance Modeling of Computer and Communication Systems (PMCCS 4)* (Williamsburg, VA, USA, 1998) 67–71.

- [58] J. Freiheit und A. Zimmermann, Dekompositionelle Leistungsbewertung großer stochastischer Petrinetze, in: E. Schnieder, Ed., *Engineering komplexer Automatisierungssysteme (EKA 2001)* (Braunschweig, 2001) 311–323.
- [59] J. Freiheit und A. Zimmermann, A Divide and conquer approach for the performance evaluation of large stochastic Petri nets, in: R. German und B. Haverkort, Ed., *9th Int. Workshop on Petri Nets and Performance Models (PNPM'01)*, Bd. Aachen (2001, IEEE CS Press) 91–100.
- [60] M. Fujita, P. McGeer und J.-Y. Yang, Multi-terminal binary decision diagrams: an efficient data structure for matrix representation, *Formal Methods in System Design* **10** (2/3) (1997) 149–169.
- [61] K. Fukuda. cdd.c: C-implementation of the double description method. <ftp://ftp.ifor.math.ethz.ch/pub/fukuda/cdd/>.
- [62] K. Fukuda und A. Prodon, Double description method revisited, Technischer bericht, Department of Mathematics, Swiss Federal Institut of Technology, Lausanne, Switzerland, 1995.
- [63] R. German, *Analysis of stochastic Petri nets with non-exponentially distributed firing times*, Dissertation, Technische Universität Berlin, 1994.
- [64] R. German, *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets* (John Wiley and Sons, 2000).
- [65] R. German und C. Lindemann, Analysis of stochastic Petri nets by the method of supplementary variables, *Performance Evaluation* **20** (1994) 317–335.
- [66] J. Giglmayr, Analysis of stochastic Petri nets by the decomposition of the transition rate matrix, *ntzArchiv* **9** (5/6) (May/June 1987) 115–120, 147–151.
- [67] GreatSPN. <http://www.di.unito.it/~greatspn>.
- [68] G. Havas, B. Majewski und K. Matthews, Extended gcd and Hermite normal form algorithms via lattice basis reduction, *Experimental Mathematics* **7** (1998) 125–136.
- [69] A. Heindl, *Traffic-based decomposition of general queueing networks with correlated input processes*, Dissertation, Technische Universität Berlin, 2001.
- [70] M. Heiner und P. Deussen, A case study in design and validation of reactive systems by means of Petri nets, in: *Computational Engineering in Systems Applications, CESA'96* (1996) 121–126.
- [71] W. Henderson, Finding and using exact equilibrium distributions for stochastic Petri nets, *Computer Networks and ISDN Systems* **25** (1993) 1143–1153.

- [72] W. Henderson und D. Lucic, Exact Results in the Aggregation and Disaggregation of Stochastic Petri Nets, in: *Proc. 4th Int. Workshop on Petri Nets and Performance Models* (Melbourne, Australia, 1991) 166–175.
- [73] W. Henderson und P. Taylor, Aggregation methods in exact performance analysis of stochastic Petri nets, in: *3rd Int. Workshop on Petri Nets and Performance Models* (1989, IEEE CS Press.) 12–18.
- [74] H. Hermanns, J. Meyer-Kayser und M. Siegle, Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains, in: B. Plateau, W. Stewart und M. Silva, Ed., *Numerical Solutions of Markov Chains* (Zaragoza, Spanien, 1999) 188–207.
- [75] A. Huck, Strukturelle Analyse und Zustandsraumabschätzung für Petri-netze mit Hilfe von Methoden der konvexen Geometrie, Diplomarbeit, Technische Universität Berlin, 1999.
- [76] A. Huck, J. Freiheit und A. Zimmermann, Convex Geometry Applied to Petri Nets: State Space Size Estimation and Calculation of Traps, Siphons, and Invariants, Forschungsbericht 2000-6, Technische Universität Berlin, 2000.
- [77] K. Jensen, Coloured Petri nets, in: W. Brauer, W. Reisig und G. Rozenberg, Ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 254, (1987, Springer-Verlag) 248–299.
- [78] K. Jensen, *Coloured Petri nets. Basic concepts, analysis methods and practical use*, EATCS Monographs in Theoretical Computer Science (Springer-Verlag, 1992).
- [79] K. Jensen, *Coloured Petri nets. Volume 2, Analysis methods*, EATCS Monographs in Theoretical Computer Science (Springer-Verlag, 1995).
- [80] R. Johnsonbaugh und T. Murata, Additional methods for reduction and expansion of marked graphs, *IEEE Transactions on Circuits and Systems* **CAS-28** (10) (1981) 1009–1014.
- [81] H. Jungnitz und A. Desrochers, Flow Equivalent Nets for the Performance Analysis of Generalized Stochastic Petri Nets, in: *Proc. IEEE Int. Conf on Robotics and Automation* (Sacramento, California, 1991) 122–127.
- [82] H. Jungnitz, B. Sánchez und M. Silva, Response Time Approximation for the Performance Analysis of Manufacturing Systems Modeled with Stochastic Marked Graphs, in: *Proc. Int. Conf. on Robotics and Automation* (Nice, 1992) 1000–1005.

- [83] R. Kannan und A. Bachem, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix, *SIAM Journal on Computing* 8 (1979) 499–507.
- [84] P. Kemper, Numerical analysis of superposed GSPNs, in: *Proc. of the 6th Int. Workshop on Petri Nets and Performance Models* (1995, IEEE CS-Press) 52–61.
- [85] P. Kemper, Reachability analysis based on structured representations, in: *17th International Conference Application and Theory of Petri Nets, Lecture Notes in Computer Science*, Bd. 1091, (1996, Springer) 269 – 288.
- [86] P. Kemper, Reachability analysis based on structured representations, in: J. Billington und W. Reisig, Ed., *Application and Theory of Petri Nets, Lecture Notes on Computer Science*, Bd. 1091, (1996, Springer Verlag) 269–288.
- [87] F. Krückeberg und M. Jaxy, Mathematical methods for calculating invariants in Petri nets, in: *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 266, (1987, Springer-Verlag) 104–131.
- [88] Y. Kwong, On reduction of asynchronous systems, *Theoretical Computer Science* 5 (1977) 25–50.
- [89] C. Lakos, On the Abstraction of Coloured Petri Nets, in: P. Azema und G. Balbo, Ed., *18th Int. Conference on Application and Theory of Petri Nets, Lecture Notes on Computer Science*, Bd. 1248, (1997, Springer Verlag) 42–61.
- [90] K. Lautenbach und H. A. Schmidt, Use of Petri nets for proving correctness of concurrent process systems, in: *IFIP 74* (1974, North Holland Pub. Co.) 187–191.
- [91] Y. Li und M. Woodside, Iterative Decomposition and Aggregation of Stochastic Marked Graph Petri Nets, in: *14th Int. Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science*, Bd. 674, (1993, Springer Verlag) 325–349.
- [92] C. Lindemann und A. Zimmermann, An adaptive algorithm for the efficient generation of the tangible reachability graph of a stochastic Petri net, *Technischer Bericht des Fachbereichs Informatik 8*, Technische Universität Berlin, 1994.
- [93] M. Lindquist, Parameterized reachability trees for predicate/transition nets, in: G. Rozenberg, Ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 674, (1993, Springer-Verlag) 301–324.
- [94] M. Lübbecke, Algorithmen zur Enumeration der Ecken und Facetten konvexer Polyeder, Diplomarbeit, Technische Universität Braunschweig, 1996, Diplomarbeit.

- [95] M. Manas und J. Nedoma, Finding all vertices of a convex polyhedron, *Numer. Math.* **12** (1968) 226–229.
- [96] J. Martinez und M. Silva, A simple and fast algorithm to obtain all invariants of a generalised Petri net, in: *Application and Theory of Petri Nets*, Informatik Fachberichte Nr.52 (1982, Springer-Verlag) .
- [97] T. H. Mattheiss und D. S. Rubin, A survey and comparison of methods for finding all vertices of conex polyhedral sets, *Mathematics of Operations Research* **5** (2) (1980) 167–185.
- [98] K. McMillan, *Symbolic model checking: an approach to the state explosion problem*, Dissertation, Carnegie Mellon University, 1992.
- [99] A. Miner, Efficient solution of GSPNs using canonical matrix diagrams, in: R. German und B. Haverkort, Ed., *8th International Workshop on Petri Nets and Performance Models (PNPM'01)* (Aachen, 2001) 101–110.
- [100] Möbius. <http://www.crhc.uiuc.edu/PERFORM/index.html>.
- [101] M. Molloy, *On the integration of delay and throughput measures in distributed processing models*, Dissertation, University of California, Los Angeles, 1981.
- [102] M. Molloy, Performance Analysis Using Stochastic Petri Nets, *IEEE Transactions on Computers* **31** (9) (Sep 1982) 913–917.
- [103] T. S. Motzkin, H. Raiffa, G. L. Thompson und R. M. Thrall, The double description method, in: H. W. Kuhn und A. W. Tucker, Ed., *Contributions to the Theory of Games* Nummer 28 in *Annals of Math. Studies*, (Princeton University Press, 1953) 81–103.
- [104] T. Murata, Petri nets: properties, analysis and applications, *Proceedings of the IEEE* **77** (4) (1989) 541–580.
- [105] T. Murata und J. Koh, Reduction and expansion of live and safe marked graphs, *IEEE Transactions on Circuits and Systems* **CAS-27** (1) (1980) 68–70.
- [106] S. Natkin, *Les Reseaux de Petri Stochastiques et leur Application a L'evaluation des Syst'emes Informatiques*, Dissertation, Conservatoire National des Arts et Metiers (CNAM), Paris, 1980.
- [107] E. Pastor und J. Cortadella, Efficient encoding schemes for symbolic analysis of Petri nets, in: *Design Automation and Test in Europe* (1998) .
- [108] E. Pastor und J. Cortadella, Structural methods applied to the symbolic analysis of Petri nets, in: *IEEE/ACM International Workshop on Logic Synthesis* (1998) .

- [109] E. Pastor, O. Roig, J. Cortadella und R. Badia, Petri net analysis using boolean manipulation, in: R. Valette, Ed., *15th Interanational Conference on Applications and Theory of Petri Nets, Lecture Notes on Computer Science*, Bd. 815, (1994, Springer-Verlag) 416–435.
- [110] C. J. Pérez-Jiménez und J. Campos, A response time approximation technique for stochastic general P/T systems, in: *Proc. Symp. Industrial and Manufacturing Systems, 2nd IMACS Int. Multiconf. on Computational Engineering in Systems Applications (CESA '98)* (Nabeul-Hammamet, Tunisia, 1998) .
- [111] C. J. Pérez-Jiménez und J. Campos, On State Space Decomposition for the Numerical Analysis of Stochastic Petri Nets, in: *Proceedings of the 8th International Workshop on Petri Nets and Performance Models* (Zaragoza, Spain, 1999) 32–41.
- [112] C. J. Pérez-Jiménez, J. Campos und M. Silva, State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes, in: *1996 IEEE International Conference on Robotics and Automation* (1996) 1159–1165.
- [113] C. Pixley, S. Jeong und G. Hachtel, Exact calculation of synchronization sequences based on binary decision diagrams, in: *29th Design Automation Conference* (1992) 620–623.
- [114] B. Plateau, On the stochastic structure of parallelism and synchronisation models for distributed algorithms, in: *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems* (Austin, Texas, USA, 1985) 147–154.
- [115] W. Reisig, *Petri Nets, EATCS Monographs on Theoretical Computer Science* **4** (1985).
- [116] W. Reisig, *Elements of distributed algorithms: modeling and analysis with Petri nets* (Springer-Verlag, 1998).
- [117] S. Rönisch, *Steuerung einer Modell-Fertigungsanlage mit speziellen farbigen Petri-Netzen*, Diplomarbeit, Technische Universität Berlin, 1998.
- [118] W. Sadiq und M. E. Orlowska, Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Models, in: M. Jarke und A. Oberweis, Ed., *Proc. 11th Int. Conf. Advanced Information Systems Engineering, CAiSE*, Nummer 1626 in *Lecture Notes on Computer Science* (1999, Springer-Verlag) .
- [119] M. Silva, E. Teruel und J. M. Colom, Linear algebraic and linear programming techniques for the analysis of place/transition net systems, in: *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. I: Basic Models, (1998, Springer-Verlag) .

- [120] SMART. <http://www.cs.wm.edu/~ciardo/SMART/index.html>.
- [121] J. Solberg, A graph theoretic formula for the steady state distribution of finite Markov processes, *Management Science* **21** (9) (1975) 1040–1048.
- [122] P. H. Starke. INA–Integrated net analyzer. <http://www.informatik.hu-berlin.de/lehrstuehle/automaten/ina/>.
- [123] P. H. Starke, *Analyse von Petri-Netz-Modellen*, Leitfäden und Monographien der Informatik (Teubner, Stuttgart, 1990).
- [124] W. Stewart, *Introduction to the numerical solution of Markov chains* (Princeton University Press, 1994).
- [125] F. Symons, *Modeling and analysis of communication protocols using numerical Petri nets*, Dissertation, University of Essex, 1978.
- [126] R. Valette, Analysis of Petri nets by stepwise refinements, *Journal of Computer and System Sciences* **18** (1979) 35–46.
- [127] A. Valmari, Stubborn sets for reduced state space generation, in: *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 483, (1990, Springer-Verlag) 491–515.
- [128] A. Valmari, The state space explosion problem, in: W. Reisig und G. Rozenberg, Ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Bd. 1491, (1996, Springer-Verlag) 429–528.
- [129] M. Weber, R. Walter, H. Völzer, T. Vesper, W. Reisig, S. Peuker, E. Kindler, J. Freiheit und J. Desel, DAWN: Petrinetzmodelle zur Verifikation verteilter Algorithmen, Informatik-Berichte 88, Humboldt-Universität zu Berlin, 1997.
- [130] D. Wilde, A library for doing polyhedral operations, Diplomarbeit, Oregon State University, 1993, <http://icps.u-strasbg.fr/PolyLib>.
- [131] P. Ziegler und H. Szczerbicka, A Structure Based Decomposition Approach for GSPN, in: *6th Int. Workshop on Petri Nets and Performance Models* (1995, IEEE-CS Press.) 261–270.
- [132] A. Zimmermann, *Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen*, Dissertation, Technische Universität Berlin, 1997.
- [133] A. Zimmermann und J. Freiheit, TimeNET-MS - An Integrated Modeling and Performance Evaluation Tool for Manufacturing Systems, in: *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC '98)* (San Diego, USA, 1998) 535–540.

- [134] A. Zimmermann, J. Freiheit, R. German und G. Hommel, Petri Net Modelling and Performability Evaluation with TimeNET 3.0, in: Haverkort, Bohnenkamp und Smith, Ed., *11th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation, Lecture Notes in Computer Science*, Bd. 1786, (Schaumburg, Illinois, USA, 2000, Springer-Verlag) 188–202.
- [135] A. Zimmermann, J. Freiheit und G. Hommel, Fertigungssysteme Modellieren, Bewerten und Steuern mit TimeNETMS, in: E. Schnieder, Ed., *Engineering komplexer Automatisierungssysteme (EKA 1999)* (Braunschweig, 1999) 31–48.
- [136] A. Zimmermann, J. Freiheit und A. Huck, A Petri net based design engine for manufacturing systems, *International Journal of Production Research* **39** (2) (2001) 225–253.
- [137] A. Zimmermann, R. German, J. Freiheit und G. Hommel, TimeNET 3.0 Tool Description, in: *Int. Conf. on Petri Nets and Performance Models (PNPM 99)* (Zaragoza, Spain, 1999) (tool descriptions).
- [138] A. Zimmermann und G. Hommel, Modelling and Evaluation of Manufacturing Systems Using Dedicated Petri Nets, *Int. Journal of Advanced Manufacturing Technology* **15** (1999) 132–137.