

Erzeugung reduzierter Markovketten am Beispiel einer stochastischen Prozeßalgebra¹

Markus Siegle
Universität Erlangen-Nürnberg
IMMD VII

Kurzfassung: Markovmodelle spielen für die Leistungsbewertung paralleler und verteilter Systeme eine große Rolle, da ihnen eine verhältnismäßig einfache mathematische Theorie zugrunde liegt und effiziente Analyseverfahren bekannt sind. Bei der Modellierung komplexer nebenläufiger Systeme kann es aber zum Problem des unkontrollierten Anwachsens des Zustandsraums kommen (Zustandsraumexplosion), das die praktische Handhabung eines Modells unmöglich macht.

In bestimmten Situationen läßt sich eine erhebliche Reduktion des Zustandsraums eines Markovmodells erreichen, indem man Zustände nach dem Prinzip der sogenannten Lumpability zusammenfaßt. Wir zeigen am Beispiel einer stochastischen Prozeßalgebra, wie die Zusammenfaßbarkeit äquivalenter Zustände bereits während der Erzeugung der Markovkette aus einer prozeßalgebraischen Modellbeschreibung ausgenutzt werden kann. Diese Methode hat den Vorteil, daß als Endergebnis eine im Sinne der Lumpability reduzierte Markovkette vorliegt, ohne daß jemals unbeherrschbar große Zwischenergebnisse verarbeitet oder gespeichert werden müssen.

1. Idee der Matrix-Semantik

Für $TIPP^{MS}$, eine modifizierte Version der Prozeßalgebra TIPP [1], wurde von Rettelbach und Siegle eine sogenannte Matrix-Semantik entwickelt [2]. Die Modifikation der Sprache gegenüber TIPP besteht aus der Hinzunahme eines Replikationsoperators und aus dem Verzicht auf einen allgemeinen Paralleloperator und Ausblendeoperator. Um die Notation der im folgenden diskutierten semantischen Regeln so einfach wie möglich zu halten, sei zunächst angenommen, daß die Sprache nur über einen einzigen Aktionstyp, die Aktion a , verfügt. Die Erweiterung auf mehr als einen Aktionstyp ist auf einfache Weise möglich und wird in Abschnitt 3.1 angesprochen. Der Sprachumfang von $TIPP^{MS}$ wird durch die folgende Grammatik festgelegt

$$P ::= 0 \mid X \mid (a, r).P \mid P + P \mid recX : P \mid !^n_S P$$

Der neu in die Sprache $TIPP^{MS}$ aufgenommene Replikationsoperator $!^n_S$ beschreibt die n -fache Parallelschaltung eines Prozesses mit sich selbst, wobei S die Menge der synchronisiert stattfindenden Aktionen angibt. Er kann also mit Hilfe eines gewöhnlichen Paralleloperators \parallel_S wie folgt ausgedrückt werden.

$$!^n_S A = A \underbrace{\parallel_S A \parallel_S \dots \parallel_S A}_{n \text{ mal}}$$

¹ Fachgespräch "Modellierung und Bewertung Paralleler Systeme", Universität Erlangen-Nürnberg, IMMD III, Interner Bericht 2/96

Die Matrix-Semantik erlaubt ein direktes Übersetzen von Prozeßtermen in die Darstellung der zugehörigen Markovkette als Transitionsratenmatrix. Damit wird die aufwendige Erreichbarkeitsanalyse vermieden, welche auf dem üblichen Weg zur Gewinnung der Markovkette bei der Berechnung des beschrifteten Transitionssystems notwendig ist. Bei der Abbildung von Prozeßtermen auf die zugehörigen Ratenmatrizen wird angestrebt, daß eine erzeugte Matrix folgende wichtigen Eigenschaften erfüllt:

- Laut Konvention wird der erste Zustand als Startzustand angenommen, und alle Zustände müssen vom Startzustand aus *erreichbar* sein. Dies wird gefordert, um zu vermeiden, daß für unerreichbare Zustände Speicherplatz vergeudet wird.
- Es wird nicht vorausgesetzt, daß die Matrizen eine irreduzible Markovkette beschreiben. Vielmehr sind *reduzible* Markovketten, also solche mit absorbierenden Zuständen bzw. absorbierenden Teilmengen von Zuständen erlaubt.
- Die Matrix beschreibt eine Markovkette, die bezüglich des Konzepts der *Lumpability* [3] nicht weiter zusammenfaßbar ist, die also in diesem Sinne *minimal* ist.
- Da stochastische Prozeßalgebren auch die qualitative Analyse unterstützen sollen, sind Schleifentransitionen, d.h. Transitionen von einem Zustand zurück zu sich selbst, sinnvoll. Daher sind die verwendeten Matrizen nicht Generatormatrizen, sondern haben nichtnegative Diagonalelemente.

Die Matrix-Semantik enthält die beiden folgenden Definitionen, die die Zuweisung von den elementaren Prozeßtermen 0 (gestoppter Prozeß) und X (Prozeßvariable) zu elementaren 1×1 -Matrizen beschreiben:

$$Q_0 := (0)$$

$$Q_X := (X)$$

Mit Q_A wird allgemein die dem Prozeßterm A durch die Matrix-Semantik zugeordnete Matrix bezeichnet.

Die komplexen Prozeßtermen entsprechenden Transitionsmatrizen werden aus diesen elementaren Matrizen mit Hilfe von semantischen Regeln für die Operatoren der Sprache aufgebaut. Die Semantik eines Operators der Sprache beschreibt also die Art der Verknüpfung der den Operanden zugeordneten Matrizen. Daher spricht man auch von einer *kompositionellen Semantik*. Die Regeln für Präfixoperator, Auswahloperator, Rekursion und Replikation sind im folgenden kurz dargestellt.

2. Semantik der einzelnen Operatoren

2.1. Präfixoperator

Die Semantik für den Präfixoperator, also das Vorschalten einer Aktion (a, λ) vor einen Prozeß A , ist in Abb. 2.1 dargestellt. Dabei wird angenommen, daß die zum Prozeßterm A gehörende Matrix Q_A schon bekannt ist, d.h. in vorausgegangenen Schritten ermittelt wurde. Die Matrix für den Term $(a, \lambda).A$ ergibt sich aus Q_A durch Erweiterung um eine Zeile und eine Spalte, deren einziges von Null verschiedenes Element ein λ in der Position $(1, 2)$ ist. Dieses λ repräsentiert die Rate vom neuen Anfangszustand in den Nachfolgezustand, den alten Anfangszustand des Prozesses A . Wegen der momentanen Beschränkung der Sprache TIPP^{MS} auf einen einzigen

Aktionstyp (die Aktion a) tritt der Name der Aktion in den Matrizen nicht explizit auf, sondern steht implizit fest.

$$Q_{(a,\lambda).A} = \begin{array}{|c|c|} \hline 0 & \lambda \quad 0 \quad \cdots \quad 0 \\ \hline 0 & \boxed{Q_A} \\ \hline \vdots & \\ \hline 0 & \\ \hline \end{array}$$

Abbildung 2.1: Matrix-Semantik des Präfixoperators

Es ist möglich, daß die nach dieser einfachen Regel konstruierte Matrix eine zusammenfaßbare Markovkette beschreibt. Man kann leicht zeigen, daß dies genau dann der Fall ist, wenn der Prozeß A einen Zustand enthält, von dem aus ein Übergang nur in den Anfangszustand von A möglich ist, und zwar mit derselben Rate λ . Dieser Zustand kann dann mit dem neu hinzukommenden Zustand zusammengefaßt werden. In diesem Fall muß zur Erzeugung der Matrix $Q_{(a,\lambda).A}$ lediglich die Matrix Q_A permutiert werden, so daß der betreffende Zustand an die erste Stelle gelangt. Dies ist in Abb. 2.2 dargestellt.

$$Q_A = \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline \lambda & 0 & \cdots & 0 \\ \hline & & & \\ \hline \end{array} \qquad Q_{(a,\lambda).A} = \begin{array}{|c|c|c|c|} \hline 0 & \lambda & \cdots & 0 \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

Abbildung 2.2: Semantik des Präfixoperators bei zusammenfaßbaren Zuständen

2.2. Auswahloperator

Dieser Operator beschreibt eine Auswahl zwischen zwei alternativen Verhalten A und B . Die erste stattfindende Aktion bestimmt, ob sich der Prozeß in der Folge wie A oder wie B verhält. Der Grundgedanke für die Matrix-Semantik des Auswahloperators ist in Abb. 2.3 dargestellt.

Im Anfangszustand sind alle Transitionen möglich, die in A oder in B stattfinden können. Daher ist die erste Zeile von Q_{A+B} aus den ersten Zeilen q_A und q_B der Operandenmatrizen zusammengesetzt. Die gesamte erste Spalte ist mit Nullelementen besetzt, da der Prozeß sich nach einmal erfolgter Entscheidung in der Folge entweder wie A oder wie B verhält und niemals zum Ausgangszustand zurückkehren kann.

Damit ist jedoch nur das Grundprinzip der Semantik für den Auswahloperator beschrieben. Die so erzeugte Matrix Q_{A+B} erfüllt nicht notwendigerweise die oben geforderten Eigenschaften der Erreichbarkeit und Minimalität. Es zeigt sich, daß die Anfangszustände der Prozesse A und B in der resultierenden Matrix Q_{A+B} unerreichbar sein können. Dies ist genau dann der Fall,

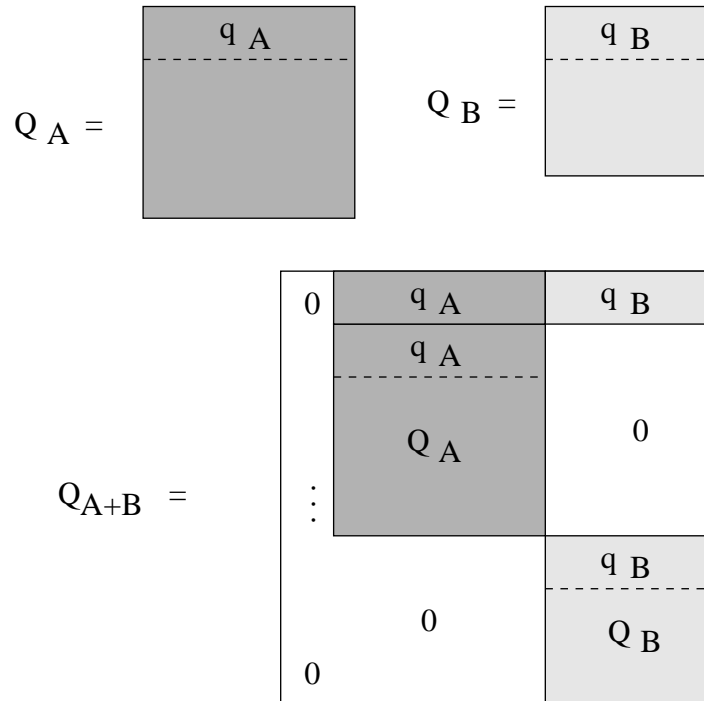


Abbildung 2.3: Grundidee der Matrix-Semantik für den Auswahloperator

wenn die erste Spalte der Matrix Q_A (Q_B) nur Nullelemente enthält. Trifft dies zu, so kann der vormalige Anfangszustand von A (B) eliminiert werden.

Schwerer wiegt die Tatsache, daß Q_{A+B} Teilmengen von zusammenfaßbaren Zuständen beinhalten kann. Dieser Fall tritt ein, wenn A und B gemeinsames Teilverhalten aufweisen. Um dieses Problem in den Griff zu bekommen, wird eine Normalform für Transitionsratenmatrizen definiert, die auf der "Inselstruktur" der Matrizen basiert. Als Inseln werden dabei Teilmengen gegenseitig erreichbarer Zustände bezeichnet. Die Verwendung dieser Normalform erleichtert das Lokalisieren gemeinsamen Teilverhaltens wesentlich. Für das Erkennen und Zusammenfassen des gemeinsamen Verhaltens von A und B gibt es einen Algorithmus, welcher die Normalformigenschaften ausnutzt.

2.3. Rekursionsoperator

Wir wollen nur den Grundgedanken der Matrix-Semantik für den Rekursionsoperator beschreiben. Um die zum Prozeßterm $recX : A$ gehörende Matrix zu bestimmen, wird angenommen, daß die Matrix Q_A für den Prozeß A bereits vorliegt. Uns interessieren hier nur die Fälle, in denen in A die Prozeßvariable X vorkommt, da andernfalls der Rekursionsoperator keinen Effekt hat. Die allgemeine Struktur der Matrix Q_A ist in Abb. 2.4 (links) dargestellt. Ohne Beschränkung der Allgemeinheit kann man annehmen, daß sich der Eintrag X in der letzten Position befindet. Mehr als einen X -Eintrag kann eine Matrix in Normalform nicht enthalten, da mehrere X -Einträge beim Aufbau von Q_A als identisches Verhalten erkannt und daher zusammengefaßt würden.

Der Rekursionsoperator bedeutet informell, daß immer dann, wenn innerhalb des Prozesses A die Prozeßvariable X auftaucht, der Prozeß in seinen Anfangszustand zurückspringt. Deshalb entsteht die in der rechten Hälfte der Abbildung wiedergegebene Matrix $Q_{recX:A}$ aus Q_A durch Aufaddieren der letzten Spalte q_A^X auf die erste Spalte q_A^1 und anschließendes Streichen der

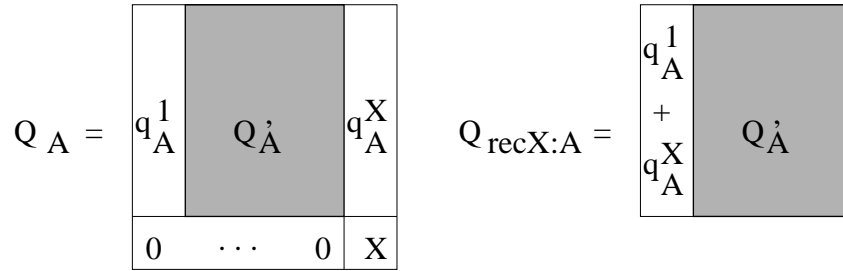


Abbildung 2.4: Grundidee der Matrix-Sematik für Rekursion

letzten Zeile und Spalte. Dies läßt sich so interpretieren, daß ein Übergang in den Zustand X gleichbedeutend ist mit einem Übergang in den Anfangszustand des Prozesses.

Auch hier sind über das dargestellte Grundprinzip hinaus weitergehende Überlegungen in bezug auf Minimalität und Normalform der entstehenden Matrix $Q_{recX:A}$ notwendig.

2.4. Replikationsoperator

Zur Ermittlung der Transitionsratenmatrix $Q_{S^m A}$ für den Replikationsoperator nimmt man zunächst an, daß keine Synchronisation stattfindet, daß also gilt $S = \emptyset$. Dann wird das Verhalten des n -fach replizierten Prozesses A durch die Matrix

$$Q = \underbrace{Q_A \oplus Q_A \oplus \dots \oplus Q_A}_{n \text{ mal}}$$

beschrieben. Dabei bezeichnet der Operator \oplus die Tensorsumme von Matrizen. Die so definierte Matrix Q beschreibt für $n > 1$ eine zusammenfaßbare Markovkette, ist also in diesem Sinne nicht minimal. Unter der Annahme, daß Q_A die Dimension s hat, folgt, daß Q von der Dimension s^n ist.

Zur Identifizierung äquivalenter Zustände betrachten wir nun eine Tupeldarstellung. Die s^n Zustände von Q können wie folgt mit n -Tupeln aus den Ziffern $\{0, 1, \dots, s-1\}$ in aufsteigender lexikographischer Reihenfolge durchnummeriert werden:

$$\begin{aligned} \text{Zustand } 0 &= (0, 0, \dots, 0) \\ \text{Zustand } 1 &= (0, 0, \dots, 1) \\ &\vdots \\ \text{Zustand } s^n - 1 &= (s-1, s-1, \dots, s-1) \end{aligned}$$

Im allgemeinen bezeichnet das Tupel (i_1, i_2, \dots, i_n) einen Zustand, in dem sich das k -te Replikat des Prozesses A im Zustand i_k befindet, $k = 1, \dots, n$. Aus Symmetriebetrachtungen ist bekannt, daß alle Zustände, deren Tupeldarstellungen auseinander durch beliebige Permutation hervorgehen, eine Teilmenge äquivalenter (und deshalb zusammenfaßbarer) Zustände darstellen. Insgesamt existieren $\binom{n+s-1}{s-1}$ solche Teilmengen äquivalenter Zustände.

Daher hat die minimale Matrix $Q_{\emptyset^m A}$, die das Verhalten von ${}^m_{\emptyset} A$ beschreibt, $\binom{n+s-1}{s-1}$ Zustände. Jeder Zustand von $Q_{\emptyset^m A}$ repräsentiert eine Teilmenge von äquivalenten, also zusammenfaßbaren Zuständen der Matrix Q . Als Repräsentant für eine solche Teilmenge wird das lexikographisch kleinste Element ausgewählt. Zum Beispiel ist das Tupel 012 der Repräsentant für die Teilmenge

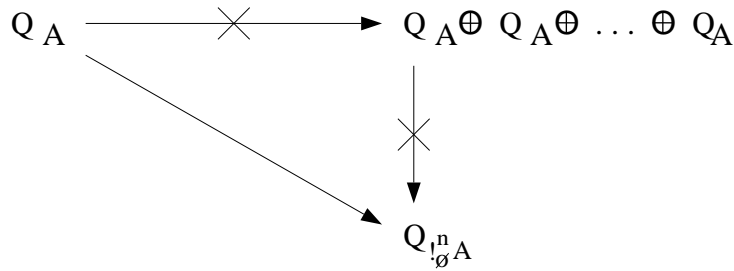


Abbildung 2.5: Zwei Wege zur Konstruktion der Matrix $Q_{!_{\emptyset}^n A}$

$\{012, 021, 102, 120, 201, 210\}$. Man kann sich zwei verschiedene Wege zur Konstruktion der Matrix $Q_{!_{\emptyset}^n A}$ aus Q_A vorstellen, siehe Abb. 2.5.

Das explizite Aufstellen der Tensorsumme und anschließendes Zusammenfassen der äquivalenten Zustände wäre extrem aufwendig. Wir haben eben gesehen, daß die Zwischenmatrix Q die Dimension s^n hat! Daher schlagen wir einen Algorithmus vor, der die reduzierte Matrix $Q_{!_{\emptyset}^n A}$ direkt aus der Matrix Q_A berechnet. Dieser Weg entspricht dem direkten Pfeil in Abb. 2.5.

Wir erläutern den Algorithmus anhand eines Beispiels. In Abb. 2.6 ist eine Matrix Q_A der Dimension 3 vorgegeben. Außerdem sind auch die Matrizen $Q_{!_{\emptyset}^{12} A}$ und $Q_{!_{\emptyset}^{13} A}$ dargestellt. Man kann beobachten, daß die Einträge der beiden letztgenannten Matrizen auf einfache Weise aus den Einträgen von Q_A entstehen. Zur Verdeutlichung des Vorgehens wurden für die von Null verschiedenen Einträge der Ausgangsmatrix Q_A verschiedene natürliche Zahlen verwendet, so daß für die Einträge der zusammengesetzten Matrizen $Q_{!_{\emptyset}^{12} A}$ und $Q_{!_{\emptyset}^{13} A}$ ersichtlich wird, aus welchen Einträgen der Ausgangsmatrix sie entstanden sind.

Um die Elemente der Matrix $Q_{!_{\emptyset}^n A}$ allgemein formal zu spezifizieren, definieren wir zunächst die Hilfsfunktion $\text{ndiff}(t_1, t_2)$, welche die Anzahl der Ziffern zurückgibt, in denen sich zwei Tupel t_1 und t_2 unterscheiden. Die Position einer Ziffer innerhalb des Tupels ist bei dieser Betrachtung irrelevant, da Tupel, die durch Permutation auseinander hervorgehen, äquivalente Zustände bezeichnen. Genau hier vollzieht sich also der Übergang von der Betrachtung der individuellen Replikate zu einer abstrakteren Sicht, die nur die Anzahl der sich in einem bestimmten Zustand befindlichen Replikate berücksichtigt.

Als Beispiel für die Funktion ndiff ergibt sich $\text{ndiff}(000, 001) = 1$, $\text{ndiff}(011, 112) = 1$ und $\text{ndiff}(011, 122) = 2$. Außerdem definieren wir die Gewichtsfunktion $w_i(t)$, welche die Häufigkeit des Vorkommens der Ziffer i im Tupel t angibt. Z.B. ergibt sich $w_0(001) = 2$ und $w_1(001) = 1$. Seien nun o und t die beiden n -Tupel, welche den Ausgangs- und den Zielzustand eines Übergangs bezeichnen. Dann läßt sich die Transitionsrate $r_{o,t}$ vom Ausgangs- in den Zielzustand wie folgt allgemein schreiben:

$$r_{o,t} = \begin{cases} \sum_{i \in o} w_i(o) a_{ii} & \text{falls } \text{ndiff}(o, t) = 0 \\ w_i(o) a_{ij} & \text{falls } \text{ndiff}(o, t) = 1 \\ 0 & \text{sonst} \end{cases} \quad (1)$$

Der erste der drei Fälle bezieht sich auf die Einträge auf der Diagonalen von $Q_{!_{\emptyset}^n A}$. Diese beschreiben den Fall einer Schleifentransition eines Replikats vom Zustand i in den Zustand i . Dafür wird die Rate a_{ii} mit dem Gewicht der Ziffer i im Ausgangszustand, also der Anzahl der sich im Zustand i befindlichen Replikate, multipliziert.

Im zweiten Fall unterscheiden sich Ausgangs- und Zielzustand um genau eine Ziffer (wobei wiederum die Position einer Ziffer innerhalb des Tupels irrelevant ist). Wir nehmen an, daß eine

$$Q_A = \begin{array}{c} \begin{array}{ccc} & 0 & 1 & 2 \\ 0 & 1 & & 2 \\ 1 & & 3 & 4 \\ 2 & & 5 & \end{array} \end{array} \quad Q_{i_2^3 \emptyset A} = \begin{array}{c} \begin{array}{cccccc} & 00 & 01 & 02 & 11 & 12 & 22 \\ 00 & 2x1 & & 2x2 & & & 0 \\ 01 & & 1+3 & 4 & & & 2 \\ 02 & & 5 & 1 & & & 2 \\ 11 & & & & 2x3 & 2x4 & \\ 12 & & 0 & & 5 & 3 & 4 \\ 22 & & & & & & 2x5 \end{array} \end{array}$$

$$Q_{i_3^3 \emptyset A} = \begin{array}{c} \begin{array}{cccccccc} & 000 & 001 & 002 & 011 & 012 & 022 & 111 & 112 & 122 & 222 \\ 000 & 3x1 & & 3x2 & & 0 & & & & & 0 \\ 001 & & 2x1 & & 4 & & 2x2 & & & & 0 \\ 002 & & 5 & 2x1 & & & 2x2 & & & & \\ 011 & & & & 1+ & 2x3 & 2x4 & & & & 2 \\ 012 & & & & 5 & 1+3 & 4 & & & & 2 \\ 022 & & & & 2x5 & 1 & & & & & 2 \\ 111 & & & & & & & 3x3 & 3x4 & & \\ 112 & & 0 & & & & & 5 & 2x3 & 2x4 & \\ 122 & & & & & & & 2x5 & 3 & 4 & \\ 222 & & & & & & & & & & 3x5 \end{array} \end{array}$$

Abbildung 2.6: Beispiel für die Konstruktion der Matrix $Q_{i_!^n \emptyset A}$

Ziffer i durch ein j ersetzt wurde. Dann ergibt sich die Transitionsrate als a_{ij} multipliziert mit dem Gewicht der Ziffer i im Ausgangszustand.

Im dritten Fall unterscheiden sich Ausgangs- und Zielzustand um mehr als eine Ziffer. Da simultane Transitionen unter Markovschen Bedingungen mit Wahrscheinlichkeit 0 auftreten, ergibt sich die Transitionsrate in diesem Fall als 0.

Bei den eben angestellten Betrachtungen waren wir davon ausgegangen, daß keine Synchronisation zwischen den Replikaten existierte, daß also $S = \emptyset$. Was geschieht nun im Fall $S \neq \emptyset$? Unter der Annahme, daß alle durch die Matrix Q_A beschriebenen Aktionen vom Typ a sind und daß über a synchronisiert wird, ist in obiger Definition für die Matrix Q die Tensorsumme durch das Tensorprodukt zu ersetzen, man erhält also

$$Q = \underbrace{Q_A \otimes Q_A \otimes \dots \otimes Q_A}_{n \text{ mal}}$$

Die Teilmengen zusammenfaßbarer Zustände sind hier dieselben wie im unsynchronisierten Fall und es läßt sich ein entsprechender Algorithmus angeben.

Wenn die Ausgangsmatrix Q_A reduzibel ist, müssen die Zustände des replizierten Modells in einer Reihenfolge entsprechend der Inselstruktur von Q_A generiert werden, damit die resultierende Matrix Q_{\emptyset}^n wiederum Normalform besitzt.

3. Erweiterungen der Methode und Zusammenfassung

3.1. Mehrere Aktionstypen

Um zwischen verschiedenen Aktionen unterscheiden zu können ist es notwendig, die Sprache $TIPP^{MS}$ so zu erweitern, daß neben der Aktion a auch andere Aktionstypen zulässig sind. Das in diesem Abschnitt dargestellte Vorgehen kann dazu wie folgt ausgebaut werden: Die Matrixeinträge der einem Prozeßterm A zugeordneten Matrix Q_A werden durch Tupel ersetzt, und zwar so, daß jede Komponente des Tupels die Übergänge bezüglich eines bestimmten Aktionstyps beschreibt. Beim Überprüfen der Äquivalenz von Zuständen muß nun die Äquivalenz bezüglich aller Aktionstypen überprüft werden, d.h. zwei Zustände können nur dann als äquivalent betrachtet werden, wenn äquivalentes Verhalten bezüglich aller Aktionstypen vorliegt.

Bei der Anwendung der semantischen Regeln für den Replikationsoperator ist ein differenziertes Vorgehen notwendig, wenn von der Replikation sowohl unsynchronisierte als auch synchronisierte Aktionen betroffen sind. Für jeden Aktionstyp, d.h. für jede einzelne Komponente eines Tupels, ist dann zu entscheiden, ob die Regel für unsynchronisierte oder die Regel für synchronisierte Aktionen anzuwenden ist.

3.2. Vorteile der Matrix-Semantik

Die bedeutenden Vorteile der Matrix-Semantik für die Sprache $TIPP^{MS}$ gegenüber dem herkömmlichen Vorgehen bei der Generierung der Markovkette sind nicht zu übersehen. Die normalerweise notwendige aufwendige dynamische Erreichbarkeitsanalyse, bei der die in den Prozeßtermen steckende Strukturinformation nicht genutzt werden kann, entfällt zugunsten eines strukturierten Verfahrens, welches direkt zur minimalen Darstellung der zugehörigen Markovkette führt. Während das Auffinden und Zusammenfassen äquivalenter Zustände in einem unter Umständen riesigen Zustandsraum praktisch kaum durchführbar ist, leistet die neue Matrix-Semantik dies durch ein geordnetes kompositionelles Vorgehen, bei dem schrittweise minimale Matrizen aufgebaut werden.

Die eingeschränkte Sprache $TIPP^{MS}$, anhand derer das Konzept der Matrix-Semantik eingeführt wurde, erfüllt die Anforderungen und Wünsche der Benutzer noch nicht. Beispielsweise ist eine Erweiterung um einen allgemeinen Paralleloperator für die Praxis unabdingbar. Ebenso wünscht man sich zur Abstraktion vom internen Verhalten eines Prozesses einen Ausblendeoperator. Die grundsätzliche Vorgehensweise bei der Matrix-Semantik, die Vorteile und die zu lösenden Probleme sind jedoch klar erkennbar.

3.3. Schlußbemerkung

Neuere Erkenntnisse zur Minimalität der von der Matrix-Semantik erzeugten Matrizen finden sich in der Diplomarbeit von Bohnenkamp [4]. Dort wird nachgewiesen, daß die Semantik nicht immer minimale Matrizen erzeugt. Insbesondere gibt es eine Klasse von Prozeßtermen, aus denen der Replikationsoperator Matrizen generiert, die auch nach dem Zusammenfassen aller auseinander durch Permutation hervorgehenden Zustände noch äquivalente Zustände beinhalten,

wo also eine weitere Reduktion möglich ist. Man kann eine Charakterisierung der Prozeßterme angeben, für die dieser Fall eintritt.

In der Dissertation von Siegle wird eine Darstellung der Zustandsraumreduktion für strukturierte Markovmodelle in einem allgemeineren Rahmen als dem der stochastischen Prozeßalgebren gegeben [5].

Literatur

- [1] M. Rettelbach, N. Götz, and U. Herzog. TIPP – Einführung in die Leistungsbewertung von Verteilten Systemen mit Hilfe von Prozeßalgebren. In H. Wedekind, editor, *Verteilte Systeme – Grundlagen und zukünftige Entwicklung*. BI Wissenschaftsverlag, Mannheim, 1994.
- [2] M. Rettelbach and M. Siegle. Compositional Minimal Semantics for the Stochastic Process Algebra TIPP. In U. Herzog and M. Rettelbach, editors, *Proc. of the 2nd Workshop on Process Algebras and Performance Modelling*, pages 89–106, Regensburg/Erlangen, July 1994. Arbeitsberichte des IMMD, Universität Erlangen-Nürnberg.
- [3] P. Buchholz. Exact and Ordinary Lumpability in Finite Markov Chains. *Journal of Applied Probability*, (31):59–75, 1994.
- [4] H. Bohnenkamp. Kompositionelle Semantiken stochastischer Prozeßalgebren zur Erzeugung reduzierter Transitionssysteme. Diplomarbeit, Universität Erlangen–Nürnberg, IMMD VII, 1995.
- [5] M. Siegle. *Beschreibung und Analyse von Markovmodellen mit großem Zustandsraum*. Dissertation, Universität Erlangen–Nürnberg, 1995.