

Verifikation kooperierender Systeme mittels schlichter Homomorphismen

P.Ochsenschläger

Kurzfassung

In der vorliegenden Arbeit wird für beschriftete Transitionssysteme eine Klasse von Sprachhomomorphismen definiert, die sicherstellen, daß Lebendigkeitseigenschaften des homomorphen Bildes einer Aktionsfolge auch in entsprechender Form für die Urbildmenge gelten. Die Sprachäquivalenz ist also in diesem Fall fein genug, nicht nur Sicherheits- sondern auch Lebendigkeitseigenschaften zu erhalten. Es zeigt sich, daß damit die Äquivalenz von Spezifikationen formal erfaßt werden kann, welche im intuitiven Sinne äquivalent sind, wo aber die bekannten Äquivalenzbegriffe versagen.

Unter dem im Titel dieser Arbeit benutzten Begriffs eines kooperierenden System wird ein verteiltes System verstanden, das sich durch die lose Kopplung und die Entscheidungsfreiheit der beteiligten Kooperationspartner auszeichnet. Der dadurch entstehende hohe Grad an nichtdeterministischem Verhalten läßt sich mit dem in dieser Arbeit definierten Begriff des schlichten Homomorphismus in adäquater Weise kontrollieren.

Inhaltsverzeichnis

Einleitung	3
Motivation	5
Schlichte Homomorphismen	17
Reguläre Sprachen	22
Ein Beispiel	31
Reduzierte Erreichbarkeitsgraphen	41
Literatur	49

Einleitung

Die Semantik vieler formaler Spezifikationsmethoden für verteilte Systeme basiert auf beschrifteten Transitionssystemen [4, 10, 12, 18, 20]. Dabei wird der Standpunkt der sogenannten Interleaving-Semantik eingenommen, was bedeutet, daß die Dynamik eines Systems durch lineare Folgen von Aktionen beschrieben wird, welche durch ein beschriftetes Transitionssystem dargestellt sind. Die Nebenläufigkeit von Aktionen wird darin nicht explizit ausgedrückt, sondern äußert sich in der Vertauschbarkeit ihres Auftretens in den Aktionsfolgen. Diese Betrachtungsweise erfaßt wichtige Aspekte verteilter Systeme, wie z.B. Nichtdeterminismus, Deadlockfreiheit, Livelockfreiheit, Terminierung, etc. [20].

Verifikation formaler Spezifikationen beruht in diesem Kontext auf dem Vergleich beschrifteter Transitionssysteme von Spezifikationen unterschiedlichen Abstraktionsniveaus unter bestimmten Homomorphismen [18]. Ein unmittelbar einleuchtendes notwendiges Vergleichskriterium ist dafür die Gleichheit der homomorphen Bilder der Mengen von Aktionsfolgen. Bekanntlich ist eine solche Sprachäquivalenz aber nicht deadlocksensitiv und berücksichtigt keine Lebendigkeitseigenschaften sondern lediglich Sicherheitseigenschaften des spezifizierten Systems. Deshalb wurden für beschriftete Transitionssysteme verschiedene Äquivalenzbegriffe definiert, die deadlocksensitiv und feiner als die Sprachäquivalenz sind [4, 12, 20]. Darüber hinaus werden zu Verifikationszwecken Lebendigkeitseigenschaften einer Spezifikation explizit mittels temporaler Logik [8] oder μ -Sprachen [1] untersucht.

In der vorliegenden Arbeit wird eine Klasse von Sprachhomomorphismen definiert, die sicherstellen, daß Lebendigkeitseigenschaften des homomorphen Bildes einer Aktionsfolge auch in entsprechender Form für die Urbildmenge gelten. Die Sprachäquivalenz ist also in diesem Fall fein genug, nicht nur Sicherheits- sondern auch Lebendigkeitseigenschaften zu erhalten. Es zeigt sich, daß damit die Äquivalenz von Spezifikationen formal erfaßt werden kann, welche im intuitiven Sinne äquivalent sind, wo aber die bekannten Äquivalenzbegriffe versagen.

Im Titel dieser Arbeit wird nicht der Begriff des verteilten Systems, sondern der des kooperierenden Systems benutzt. Beides sind keine scharf abgegrenzten Begriffe. Unter einem kooperierenden System soll ein verteiltes System verstanden werden, das sich durch die lose Kopplung und die Entscheidungsfreiheit der beteiligten Kooperationspartner auszeichnet. Der dadurch entstehende hohe Grad an nichtdeterministischem Verhalten läßt sich mit dem Begriff des schlichten Homomorphismus in adäquater Weise kontrollieren.

Das folgende Kapitel erläutert an Hand eines einfachen Petrinetzes die Problemstellung. Danach wird der Begriff des schlichten Homomorphismus definiert und einige grundlegende Eigenschaften bewiesen. Im Kapitel über reguläre Sprachen wird u.a. gezeigt, daß bei endlichen

Transitionssystemen die Schlichtheit von Homomorphismen entscheidbar ist. Darüber hinaus wird noch ein einfaches hinreichendes Kriterium für diese Eigenschaft bewiesen. Am Beispiel einer Produktnetzspezifikation /7/ eines Verbindungsauf- und Abbauprotokolls werden dann die Verifikationsmöglichkeiten demonstriert. Im letzten Kapitel wird gezeigt, wie die Schlichtheit von Projektionen /14/ und Modulhomomorphismen /15, 17/ mittels der entsprechenden reduzierten Erreichbarkeitsgraphen überprüft werden kann.

Motivation

Das folgende Petrinetz in Fig.1 beschreibt das Zusammenspiel zweier Kooperationspartner A und B. Der Partner A ist dabei durch die Transitionen T1, T2, T3, T4 und T5 modelliert und der Partner B durch die Transitionen T6, T7, T8, T9 und T10. Ausgehend von der Anfangsmarkierung, bei der sich je eine Marke auf den Stellen 6 und 9 befindet, kann A eine "Anfrage" an B richten (Schalten von T1). Nach Entgegennahme dieser Anfrage (Schalten von T6) gibt es für B drei Möglichkeiten: "positive" Beantwortung (Schalten von T7), "negative" Beantwortung (Schalten von T8) oder Rückfrage an A (Schalten von T9). Nachdem A eine mögliche Rückfrage beantwortet hat (Schalten von T5) kann B erneut zurückfragen (ggf. auch mehrfach) oder die ursprüngliche Anfrage "positiv" oder "negativ" beantworten. Nach Entgegennahme einer solchen Antwort durch A ist das System wieder im Ausgangszustand, und A kann erneut eine Anfrage an B richten.

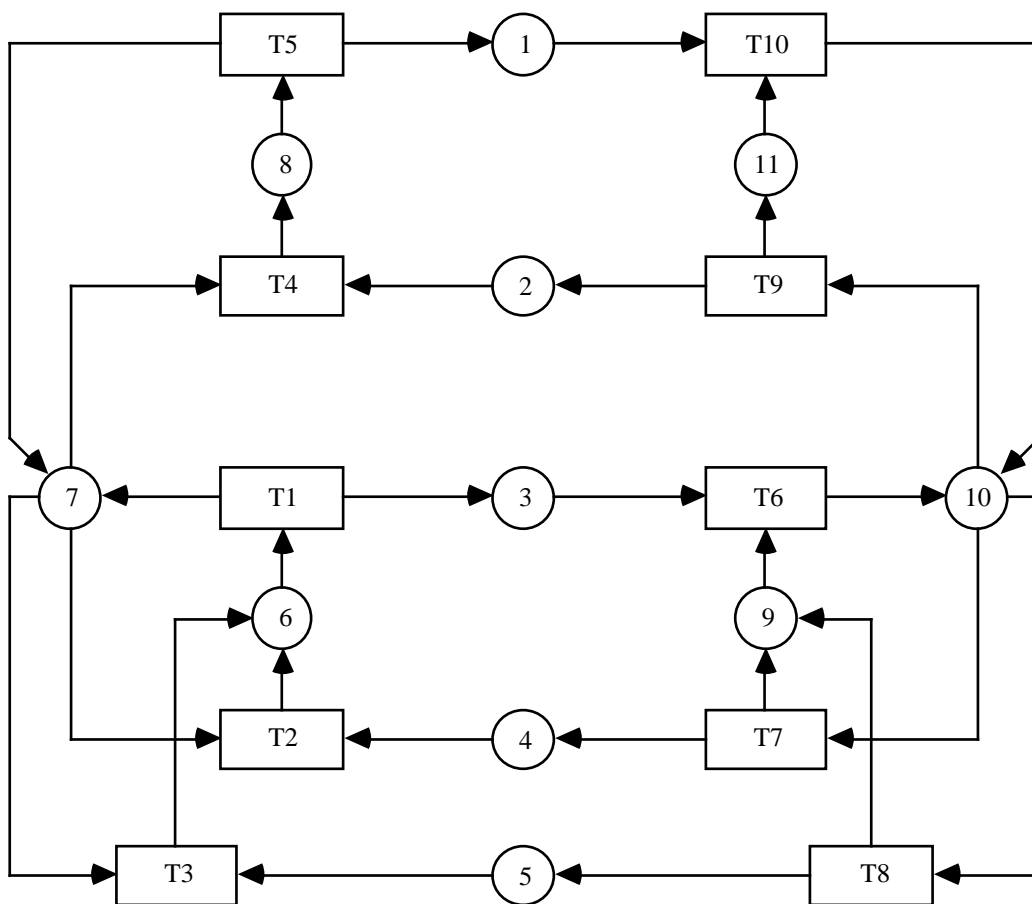


Fig.1

Formal lässt sich die Dynamik dieses Netzes durch den sogenannten Erreichbarkeitsgraphen beschreiben. Dieser gerichtete Graph enthält als Knoten die Markierungen des Netzes, welche ausgehend von der Anfangsmarkierung durch Schalten von Transitionen erzeugbar sind. Seine

Kanten beschreiben diese Schaltvorgänge, d.h. es führt genau dann eine mit T beschriftete Kante von einer Markierung M zu einer Markierung M', wenn die Transition T unter der Markierung M aktiviert ist und beim Schalten die Markierung M' erzeugt.

Der Erreichbarkeitsgraph des Beispielnetzes sieht folgendermaßen aus:

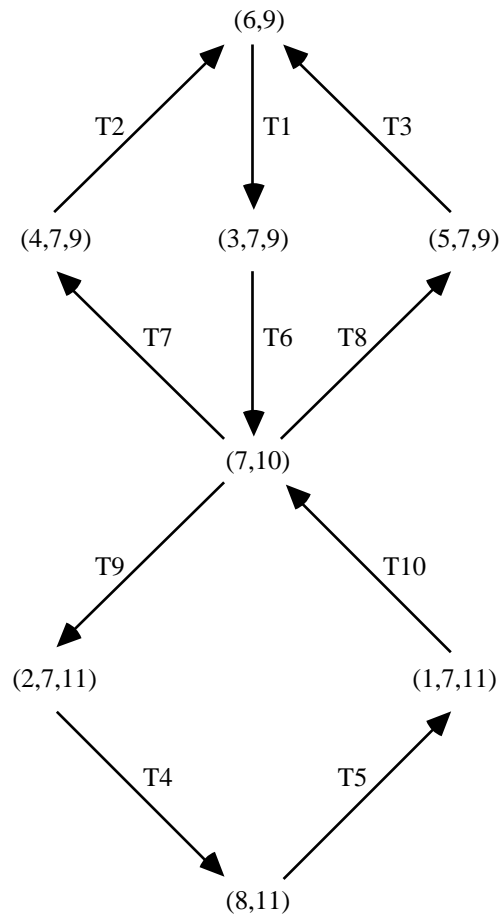


Fig.2

Hierbei sind die verschiedenen Markierungen durch Listen von Nummern der Stellen dargestellt, die je eine Marke enthalten. Das ist in diesem Fall möglich, da bei jeder Markierung auf den einzelnen Stellen maximal eine Marke liegt.

Eine andere Beschreibung der Dynamik eines Netzes ist die Menge aller Schaltfolgen /14/. Schaltfolgen sind Folgen von Schaltsschritten, die durch Tripel beschrieben werden. Die einzelnen Tripel sind dabei von der Form :

(Markierung, aktivierte Transition, Nachfolgemarkierung) .

Bei zwei benachbarten Schaltschritten einer Schaltfolge ist die letzte Komponente des ersten Schaltschrittes gleich der ersten Komponente des zweiten Schaltschrittes.

Die erste Komponente des ersten Schaltschrittes einer Schaltfolge heißt Startmarkierung der Schaltfolge, und die letzte Komponente des letzten Schaltschrittes heißt Zielmarkierung der Schaltfolge.

Beispiel :

((6,9),T1,(3,7,9))
 ((3,7,9),T6,(7,10))
 ((7,10),T7,(4,7,9))
 ((4,7,9),T2,(6,9))

ist eine Schaltfolge bei der Start- und Zielmarkierung gleich der Anfangsmarkierung des Netzes in Fig.1 sind.

\mathbb{M} bezeichne die Menge aller erreichbaren Markierungen und

$= \{(M,T,M') \mid M, M' \in \mathbb{M}, T \text{ ist unter } M \text{ aktiviert und erzeugt } M' \text{ als Nachfolgemarkierung}\}$

die Menge aller Schaltschritte.

ist eine endliche Menge, wenn der Erreichbarkeitsgraph endlich ist.

Die Schaltfolgen lassen sich jetzt als Worte über dem Alphabet betrachten. Die Menge aller Schaltfolgen, deren Startmarkierung gleich der Anfangsmarkierung des Netzes ist, ist dann eine formale Sprache, die Schaltfolgensprache L *.

* ist dabei die Menge aller Worte über dem Alphabet einschließlich des leeren Wortes ϵ .

Unter der Voraussetzung der Endlichkeit des Erreichbarkeitsgraphen ist die Schaltfolgensprache L eine reguläre Sprache, denn der Erreichbarkeitsgraph kann in gewissem Sinn als endlicher Automat betrachtet werden, der L erkennt.

Dazu definieren wir den deterministischen endlichen Automaten $\mathbb{A} = (S, \Sigma, q_0, F)$ mit

- der Zustandsmenge S mit $S = \mathbb{M}$
- dem Alphabet Σ ,
- der partiellen Zustandsüberföhrungsfunktion $\delta : S \times \Sigma \rightarrow S$ mit $\delta(M, (M, T, M')) = M'$ falls $(M, T, M') \in \mathbb{M}$ und ansonsten undefiniert,
- dem Anfangszustand $q_0 \in S$ mit $q_0 = M_1$,
- und der Menge der Endzustände F mit $F = S$.

\mathbb{A} ist i.a. ein unvollständiger Automat, da δ partiell ist.

Die Zustandsüberföhrungsfunktion wird in der üblichen Weise von Buchstaben auf Worte (buchstabenweise Verarbeitung des Wortes von links nach rechts) zu einer partiellen Funktion

* erweitert ; genauer :

* : $S \times \Sigma^* \rightarrow S$ mit

$$\delta^*(q, \epsilon) = q \quad \text{und} \quad \delta^*(q, wx) = (\delta^*(q, w), x)$$

$$\text{für } q \in S, x \in \Sigma \quad \text{und} \quad w \in \Sigma^*.$$

$\delta^*(q, wx)$ ist genau dann definiert, wenn sowohl $\delta^*(q, w)$ als auch $(\delta^*(q, w), x)$ definiert sind.
Ein Wort $w \in \Sigma^*$ wird genau dann von \mathbb{A} akzeptiert, wenn $\delta^*(q_0, w) \in F$ gilt.

Für ein Wort $w \in \Sigma^*$ ist jetzt der Ausdruck $\delta^*(q_0, w)$ genau dann definiert, wenn w eine Schaltfolge ist, die mit der Anfangsmarkierung beginnt. Da alle Zustände des Automaten als Endzustände ausgezeichnet wurden, führt ein Wort w genau dann vom Anfangszustand zu einem Endzustand, wenn $\delta^*(q_0, w)$ definiert ist, also wenn w eine Schaltfolge ist. Damit ist \mathbb{A} ein endlicher Automat der die Sprache L erkennt, also ist L eine reguläre Sprache.

Die Zustandsüberföhrungsfunktionen endlicher Automaten werden öblicherweise auch als Graphen dargestellt, den sogenannten Zustandsgraphen.

Die Knoten des Graphen sind die Zustände des Automaten, und die beschrifteten Kanten stellen die Zustandsübergänge dar, wobei im Graphen genau dann eine Kante vom Knoten q zum Knoten p föhrt und mit $a \in \Sigma$ beschriftet ist, wenn $\delta(q, a) = p$ gilt.

Der Unterschied zwischen dem Zustandsgraph des Automaten \mathbb{A} und dem Erreichbarkeitsgraphen besteht lediglich in der Kantenbeschriftung :

Im Erreichbarkeitsgraphen föhrt genau dann eine Kante von M nach M' und ist mit $T \in \Sigma$ beschriftet, wenn im Zustandsgraphen eine Kante von M nach M' föhrt und mit (M, T, M') beschriftet ist.

Eine Sichtweise der Dynamik des betrachteten Netzes, die vom "Innenleben" der Kooperationspartner abstrahiert, läßt sich jetzt durch spezielle Homomorphismen auf der Schaltfolgensprache, den sogenannten Projektionen /14/, beschreiben. Allgemein ist eine Projektion durch eine Aufteilung eines Netzes in "Module", d.h. durch eine Partition $\mathbb{P}T$ auf der Transitionsmenge, festgelegt. Eine solche Partition definiert eine spezielle Teilmenge $\mathbb{S}P$ der Stellenmenge, nämlich die Schnittstellenmenge, sowie eine spezielle Teilmenge $\mathbb{E}T$ der Transitionsmenge, nämlich die Menge der unsichtbaren Transitionen. Dabei ist eine Stelle Schnittstelle, wenn sie zu zwei Transitionen benachbart ist, die in unterschiedlichen Blöcken der Partition liegen. Eine Transition heißt unsichtbar, wenn sie zu keiner Schnittstelle benachbart ist.

Bei der Sichtweise der Projektion interessieren jetzt nicht mehr einzelne Schaltschritte, sondern nur noch die Art, wie die einzelnen Module die Markierungen der Schnittstellen verändern.

Für eine Markierung M bezeichne $M|_{\mathbb{S}P}$ die Einschränkung von M auf die Schnittstellenmenge $\mathbb{S}P$ /14/.

Eine Projektion ist jetzt ein Homomorphismus $p : \Sigma^* \rightarrow \Sigma^*$, der durch $p((M, T, M')) = (M|_{\mathbb{S}P}, T \in \mathbb{E}T, M')$ für $(M, T, M') \in \mathbb{A}$ und $T \in \mathbb{E}T$ sowie

$p((M,T,M')) = (M|\mathbb{S}\mathbb{P},A, M'|\mathbb{S}\mathbb{P})$ für (M,T,M') , $T \in \mathbb{E}\mathbb{T}$ und $T \in A$ mit $A \in \mathbb{P}\mathbb{T}$ definiert ist.

Dabei ist $\mathbb{L}' = \{ (M|\mathbb{S}\mathbb{P},A,M'|\mathbb{S}\mathbb{P}) \mid (M,T,M') , T \in \mathbb{E}\mathbb{T} \text{ und } T \in A \text{ mit } A \in \mathbb{P}\mathbb{T} \}$.

Das Bild $p(L)$ der Schaltfolgensprache unter der Projektion wird Projektionssprache genannt.

Im betrachteten Beispiel gilt $\mathbb{P}\mathbb{T} = \{A,B\}$ mit $A = \{T1,T2,T3,T4,T5\}$ und

$B = \{T6,T7,T8,T9,T10\}$ sowie $\mathbb{S}\mathbb{P} = \{1,2,3,4,5\}$ und $\mathbb{E}\mathbb{T} = \dots$

Mit den Abkürzungen $()$ bedeutet dabei die leere Markierung)

$A1 = ((1),A,(1))$ $A2 = ((2),A,())$ $A3 = ((),A,(3))$ $A4 = ((4),A,())$ $A5 = ((5),A,())$

$B1 = ((1),B,())$ $B2 = ((),B,(2))$ $B3 = ((3),B,())$ $B4 = ((),B,(4))$ $B5 = ((),B,(5))$

ergibt sich mit bekannten Methoden /6, 19/ aus dem Erreichbarkeitsgraphen unmittelbar der folgende Automat für die Projektionssprache $p(L)$:

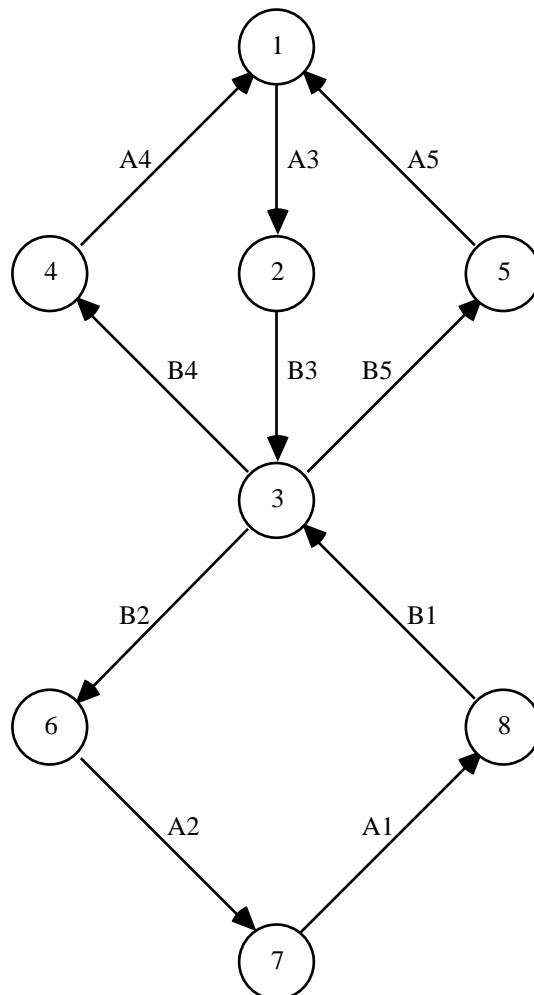


Fig.3

Dabei ist 1 der Anfangszustand, und alle Zustände sind Endzustände. Dieser Automat ist auch der Minimalautomat der Projektionssprache. I. allg., insbesondere dann, wenn $\mathbb{E}\mathbb{T} \dots$, ist der Minimalautomat der Projektionssprache wesentlich kleiner als der Erreichbarkeitsgraph /14/.

Die Idee, vergrößernde Sichtweisen der Dynamik eines Netzes durch Schaltfolgenhomomorphismen zu beschreiben, ist nicht auf Projektionen beschränkt. Vielmehr bieten sich ganz allgemein alphabetische Homomorphismen zur Beschreibung von Abstraktionen an. Das sind solche Homomorphismen $h : \Sigma^* \rightarrow \Sigma'^*$, die durch eine Abbildung $h : \Sigma \rightarrow \Sigma'$ definiert sind, wobei Σ' eine beliebige Menge ist.

Vergrößernde Sichtweisen bieten die Möglichkeit, Spezifikationen unterschiedlichen Abstraktionsniveaus zu vergleichen. Betrachten wir dazu eine leichte Verfeinerung unseres Beispiels:

Nach Entgegennahmen einer Anfrage oder einer Antwort auf eine Rückfrage hat B die Möglichkeit eine Rückfrage an A zu richten. Angenommen es gibt unterschiedliche "lokale" Gründe für derartige Rückfragen, die zum Zweck einer "Implementationsspezifikation" jetzt unterschieden werden sollen. Ein Grund, beispielsweise eine "lokale Fehlerbehandlung", möge dazu führen, daß eine zweifache Rückfrage notwendig wird. Das folgende Netz in Fig.4 mit der gleichen Anfangsmarkierung wie im Netz von Fig.1 ist dann eine Formalisierung dieses Sachverhalts. Die zweifache Rückfrage ist dabei grau gekennzeichnet.

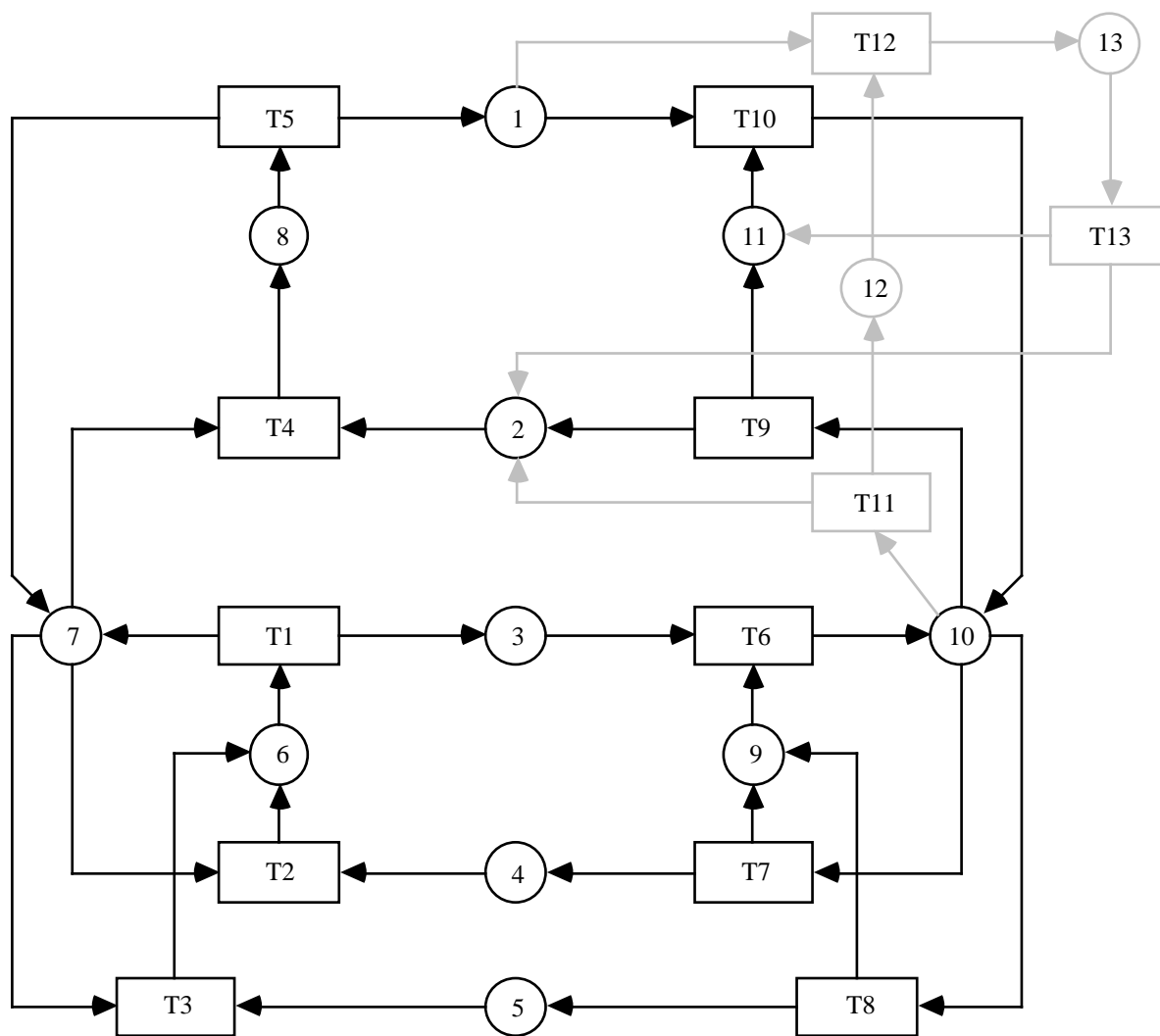


Fig.4

Die Dynamik dieses Netzes wird durch den Erreichbarkeitsgraphen in Fig.5 beschrieben. Auch hier ist die zweifache Rückfrage grau gekennzeichnet.

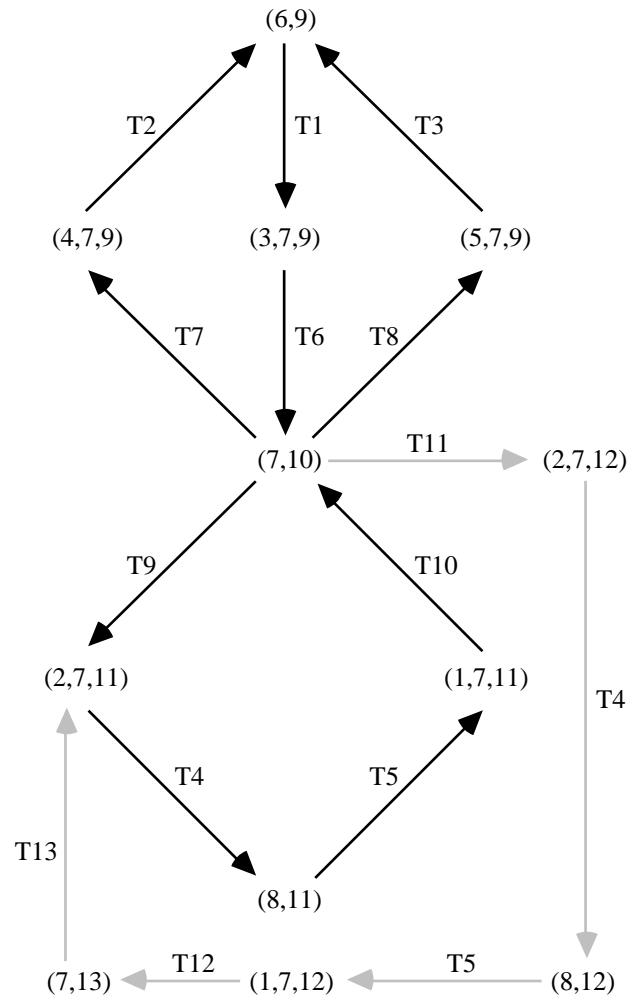


Fig.5

Betrachtet man in diesem Netz die Projektion, die durch $A = \{T1, T2, T3, T4, T5\}$ und $B = \{T6, T7, T8, T9, T10, T11, T12, T13\}$ definiert ist, dann entsteht die gleiche Projektionssprache wie im vorhergehenden Netz. Man könnte deshalb versucht sein, diese Gleichheit der Projektionssprachen als "Verifikation" des Netzes von Fig.4 zu bezeichnen, zumal es sich hier vom intuitiven Standpunkt aus sicher um eine "korrekte Verfeinerung" des ursprünglichen Netzes handelt. Allerdings ist bekannt, daß die Gleichheit solcher "Tracesprachen" zu Verifikationszwecken nicht ausreicht /4, 12 14, 18/.

Betrachten wir dazu in Fig.6 eine fehlerhafte Version des Netzes von Fig.4, wobei der Fehler darin besteht, daß die Kante von der Transition T13 nicht zur Stelle 11, sondern zur Stelle 12 führt.

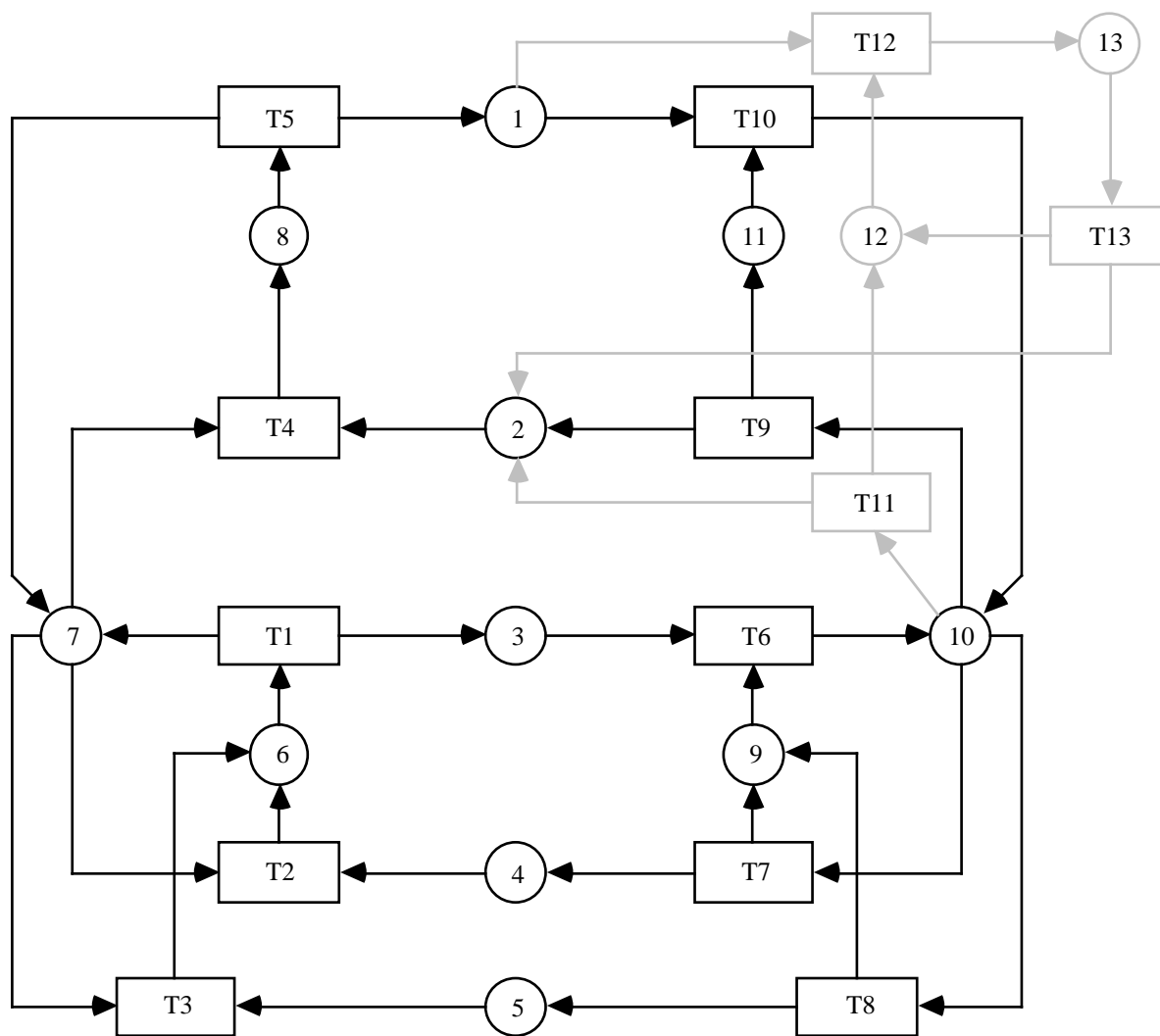


Fig.6

Die Dynamik dieses Netzes wird durch den folgenden Erreichbarkeitsgraphen in Fig.7 beschrieben.

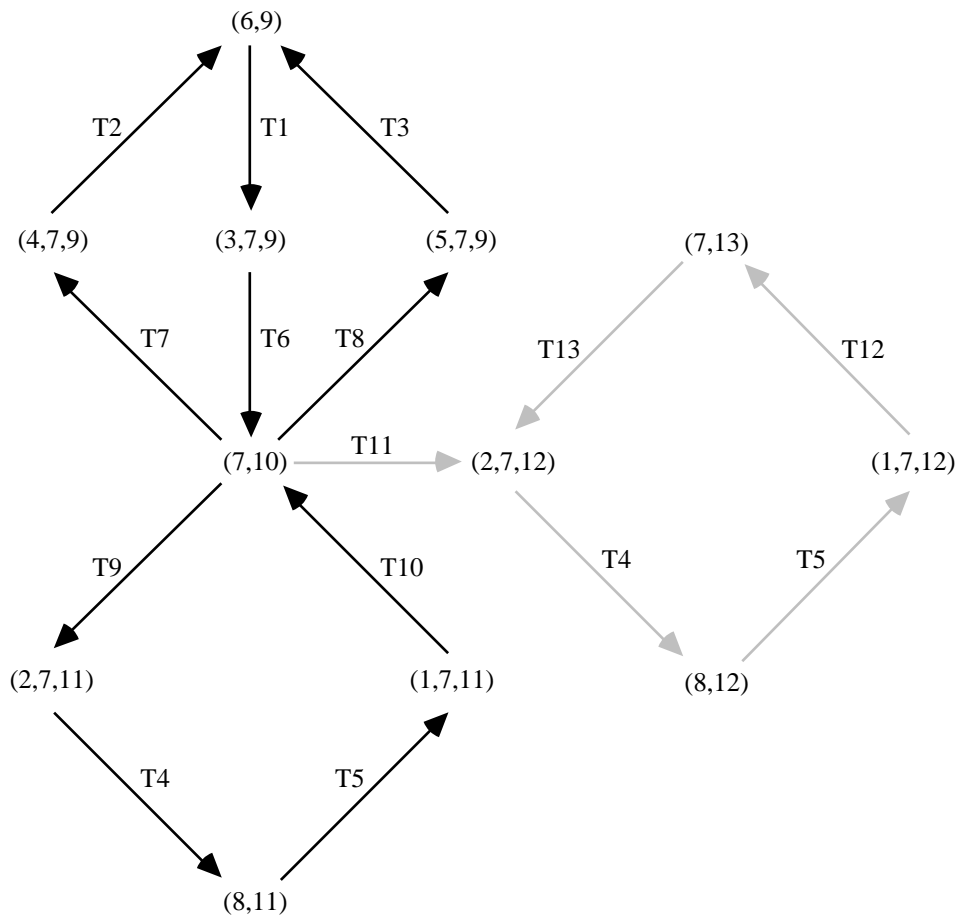


Fig.7

Wie der Erreichbarkeitsgraph zeigt, besitzt dieses Netz eine wesentlich andere Dynamik. Wird nämlich einmal die Alternative T11 gewählt, dann kommt B nie mehr in die Situation, die Anfrage mit T7 oder T8 beantworten zu können. D.h. das Netz von Fig.6 ist keine "korrekte Verfeinerung" des ursprünglichen Netzes. Betrachtet man allerdings die Projektion, wie sie für das Netz in Fig.4 definiert wurde, dann entsteht hier die gleiche Projektionssprache. D.h. mit der Projektionssprache alleine kann dieser Fehler, der sich auf das "zukünftige Verhalten" ab einer bestimmten Situation bezieht, nicht entdeckt werden. Solche Eigenschaften bezüglich des "zukünftigen Verhaltens" bezeichnet man üblicherweise als Lebendigkeitseigenschaften /3/. Im Gegensatz dazu stehen die sogenannten Sicherheitseigenschaften /3/, die sich auf das "bereits abgelaufene Verhalten" beziehen, und damit natürlich in der Projektionssprache (oder allgemein in einem entsprechenden homomorphen Bild der Schaltfolgensprache) eingefangen werden.

Zur Erfassung einer allgemeinen Lebendigkeitseigenschaft wurde in /14/ der Begriff der Deadlocksprache bezüglich eines Homomorphismus definiert. Eine Deadlocksprache besteht aus den homomorphen Bildern der Schaltfolgen, welche höchstens solche Fortsetzungen besitzen, die vom betrachteten Homomorphismus auf abgebildet werden, d.h. bezüglich der

vergrößernden Sichtweise unsichtbar sind. Mit den Deadlocksprachen werden also Verhaltensweisen erfaßt, die zu solchen Situationen führen können, ab denen unter einer vergrößernden Sichtweise nichts mehr geschieht. Man kann sich jetzt leicht überlegen, daß bezüglich der Projektionen der betrachteten drei Netze die Deadlocksprache jeweils leer ist. D.h., daß neben der Projektionssprache auch die Deadlocksprache nicht ausreicht, den Fehler des Netzes von Fig.6 aufzudecken.

Auf dem Gebiet der Prozeßalgebren gibt es verschiedene Äquivalenzbegriffe für sogenannte beschriftete Transitionssysteme /4, 12 20/; das sind beschriftete gerichtete Graphen mit einem ausgezeichneten Anfangsknoten, oder anders ausgedrückt: nichtdeterministische Automaten ohne ausgezeichnete Endzustände. Diese Äquivalenzbegriffe können auf Schaltfolgenhomomorphismen übertragen werden. Dazu werden in den Erreichbarkeitsgraphen die Kantenanschriften durch die Bilder der entsprechenden Schaltschritte ersetzt und die dadurch entstehenden beschrifteten Transitionssysteme betrachtet. Fig.8 zeigt das aus Fig.7 entstehende beschriftete Transitionssystem.

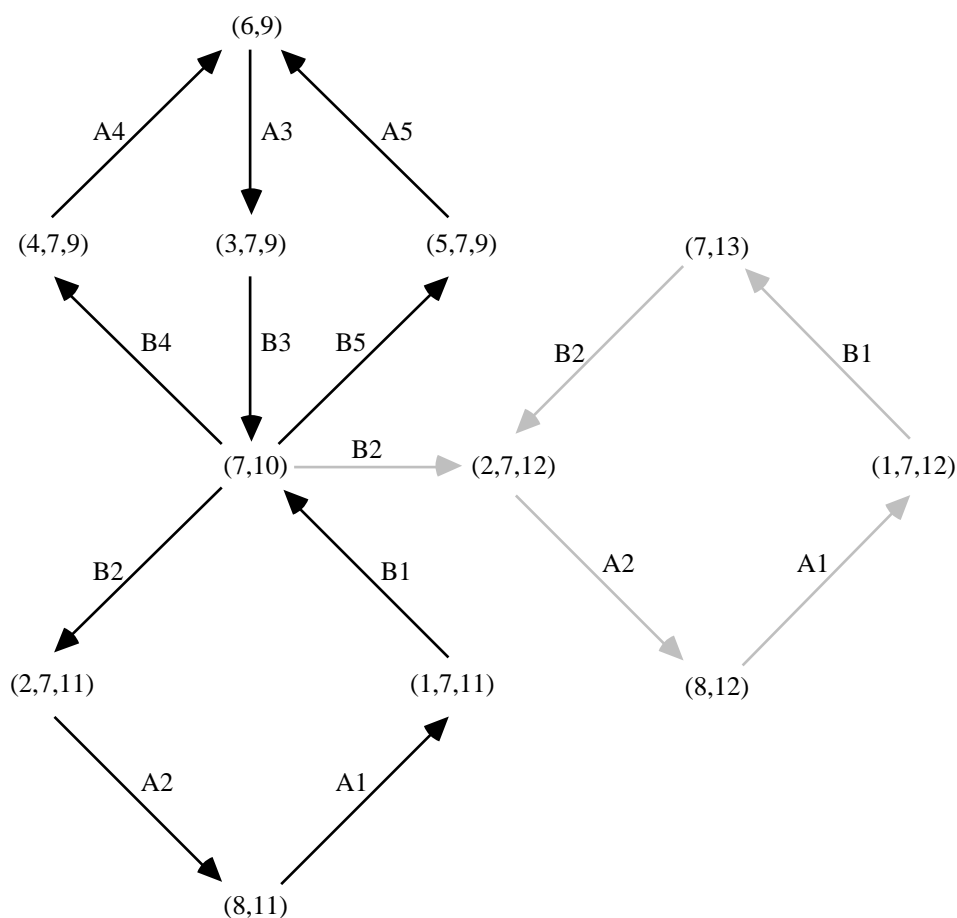


Fig.8

Bis auf die Benennung der Knoten zeigt Fig.3 das aus Fig.2 entstehende beschriftete Transitionssystem.

Unter den verschiedenen Äquivalenzbegriffen für beschriftete Transitionssysteme ist die sogenannte Failure-Äquivalenz die grösste Äquivalenz, die feiner als die reine Sprachäquivalenz ist. Enthält ein beschriftetes Transitionssystem keine Kante, die mit a beschriftet ist, dann ist (w,a) ein Failure-Paar, wenn es ausgehend vom Anfangsknoten einen mit w beschrifteten Pfad zu einem Knoten gibt, von dem aus keine mit a beschriftete Kante ausgeht. Betrachtet man im beschrifteten Transitionssystem von Fig.8 den Knoten $(7,13)$, dann ist z.B. $(A3 B3 B2 A2 A1 B1, B4)$ ein Failure-Paar.

Zwei beschriftete Transitionssysteme heißen failure-äquivalent, wenn sie die gleichen Failure-Paare besitzen. Damit sind die Projektionen der Netze aus Fig.1 und Fig.6 nicht failure-äquivalent, denn das beschriftete Transitionssystem von Fig.3 besitzt nicht das Failure-Paar $(A3 B3 B2 A2 A1 B1, B4)$. Die Failure-Äquivalenz ist also fein genug, das Fehlverhalten des Netzes von Fig.6 aufzudecken. Sie ist allerdings für unsere Beispiele zu fein, denn auch die Projektion des Netzes von Fig.4, welches wir als eine "korrekte Verfeinerung" des Netzes von Fig.1 betrachtet haben, besitzt das Failure-Paar $(A3 B3 B2 A2 A1 B1, B4)$.

Schlichte Homomorphismen

Zur genaueren Betrachtung der Tatsache, daß die homomorphen Bilder von Schaltfolgensprachen keine Information über Lebendigkeitseigenschaften enthalten wird folgende allgemeine Definition benötigt:

Für eine formale Sprache $L \subseteq \Sigma^*$ und ein Wort $x \in \Sigma^*$ bezeichnet

$$x^{-1}(L) = \{ y \in \Sigma^* \mid xy \in L \}$$

die Menge der Fortsetzungen von x in L oder den Linksquotienten von L nach x [6, 19].

Bei einer vergrößernden Sichtweise, die durch einen alphabetischen Homomorphismus $h : \Sigma^* \rightarrow \Sigma'^*$ auf einer Schaltfolgensprache beschrieben ist, stecken die Lebendigkeitseigenschaften gewissermaßen in den Mengen $h(x^{-1}(L))$, mit $x \in L$. Das homomorphe Bild $h(L)$ der Schaltfolgensprache hingegen enthält aber nur Information über die Mengen $h(x)^{-1}(h(L))$. Es gilt zwar $h(x^{-1}(L)) \subseteq h(x)^{-1}(h(L))$ für jedes $x \in L$, aber die Inklusion ist i.allg. echt. Ist $h(x)$ Element der Deadlocksprache, dann gilt sogar $h(x^{-1}(L)) = \{ \}$.

Die Failure-Paare liefern zwar zusätzliche Information über die Mengen $h(x^{-1}(L))$; wie unsere Beispiele gezeigt haben, ist dies aber zu eng gefaßt, da jeweils nur die "mögliche nächste Aktion" betrachtet wird. Für die Korrektheit kooperierender Systeme hingegen ist wichtig, daß "irgendwann nach einer eingeschränkten Verhaltensweise das vollständige Verhaltensspektrum (wie es in $h(L)$ dargestellt ist) wieder möglich ist". Diese Eigenschaft wird mit dem Begriff des schlichten Homomorphismus erfaßt:

Für eine formale Sprache $L \subseteq \Sigma^*$ und ein Wort $x \in L$ heißt ein Homomorphismus

$h : \Sigma^* \rightarrow \Sigma'^*$ schlicht auf L in x , wenn es ein $u \in h(x)^{-1}(h(L))$ gibt mit

$$u^{-1}(h(x^{-1}(L))) = u^{-1}(h(x)^{-1}(h(L))). \quad (*)$$

Insbesondere gilt dann $u \in h(x^{-1}(L))$.

Der Homomorphismus h heißt schlicht auf L , wenn er schlicht auf L in jedem $x \in L$ ist.

Die Schlichtheit eines Homomorphismus garantiert gewissermaßen, daß die "impliziten Lebendigkeitseigenschaften" des homomorphen Bildes einer Schaltfolgensprache auch für die Schaltfolgensprache selbst, und damit auch für die Dynamik des Netzes gelten.

Für die Projektion p des Netzes in Fig.1 ist aus dem Erreichbarkeitsgraphen in Fig.2 und dem Minimalautomaten in Fig.3 unmittelbar ersichtlich, daß $p(x^{-1}(L)) = p(x)^{-1}(p(L))$ für jedes $x \in L$ gilt. Mit $u = \epsilon$ ist deshalb p schlicht.

Zum Beweis der Schlichtheit der Projektion des Netzes in Fig.4 wird die Schaltfolgensprache L in fünf disjunkte Klassen aufgeteilt: L_1 enthält alle Schaltfolgen, die zur Markierung (2,7,12) führen, L_2 diejenigen, welche zu (8,12) führen, L_3 diejenigen, welche zu (1,7,12) führen, L_4 diejenigen, welche zu (7,13) führen, und L_0 enthält die restlichen Schaltfolgen. Durch den

Vergleich von Fig.3 und Fig.5 ist die Schlichtheitsbedingung(*) leicht zu verifizieren mit $u =$ für $x \in L_0$, mit $u = A_2 A_1 B_1 B_2$ für $x \in L_1$, mit $u = A_1 B_1 B_2$ für $x \in L_2$, mit $u = B_1 B_2$ für $x \in L_3$ und mit $u = B_2$ für $x \in L_4$.

Bezüglich des Netzes in Fig.6 und der Schaltfolge

$x = ((6,9),T1,(3,7,9))((3,7,9),T6,(7,10))((7,10),T11,(2,7,12))$ enthält $p(x^{-1}(L))$ kein Wort, in dem ein A_3 vorkommt. Für jedes $u \in p(x)^{-1}(p(L))$ enthält aber $u^{-1}(p(x)^{-1}(p(L)))$ Worte, in denen A_3 vorkommt.

Also gibt es kein $u \in p(x)^{-1}(p(L))$ mit $u^{-1}(p(x^{-1}(L))) = u^{-1}(p(x)^{-1}(p(L)))$, und damit ist p nicht schlicht auf L im betrachteten x .

Die Untersuchung dieser drei Projektionen zeigt, daß hier der intuitive Korrektheitsbegriff von der Gleichheit der Projektionssprachen zusammen mit der Schlichtheit der Projektionen erfaßt wird. Wenn also zwei Spezifikationen unterschiedlichen Abstraktionsniveaus mittels Schaltfolgenhomomorphismen gegeneinander verifiziert werden sollen, und wenn einer dieser Homomorphismen schlicht ist, dann muß auch der andere Homomorphismus schlicht sein, und die homomorphen Bilder beider Schaltfolgensprachen müssen gleich sein. Dies garantiert dann nämlich, daß neben den Sicherheitseigenschaften auch die impliziten Lebendigkeitseigenschaften des homomorphen Bildes einer Schaltfolgensprache für beide Spezifikationen gelten.

Falls keiner der beiden Schaltfolgenhomomorphismen schlicht ist, dann ist keine Lebendigkeitsaussage möglich.

Die obige Definition der Schlichtheit gilt allgemein für Homomorphismen auf formalen Sprachen; sie ist also nicht auf Schaltfolgenhomomorphismen beschränkt. Im folgenden soll diese Allgemeinheit beibehalten werden, damit die Konzepte und Ergebnisse auch für allgemeine Mengen von "Aktionsfolgen" gelten, welche die Dynamik eines "Systems" beschreiben, das nicht notwendigerweise durch ein Netz spezifiziert ist. Solche allgemeine Mengen von "Aktionsfolgen" haben mit den Schaltfolgensprachen die Präfixstabilität gemeinsam; genauer:

Für ein Wort $x \in \Sigma^*$ ist die Präfixmenge $PR(x)$ definiert durch

$$PR(x) = \{ u \in \Sigma^* \mid \text{es existiert } v \in \Sigma^* \text{ mit } uv = x \}.$$

Es gilt insbesondere $x \in PR(x)$ und $\Sigma^* \in PR(x)$.

Der Begriff der Präfixmenge läßt sich in natürlicher Weise auf formale Sprachen $L \subseteq \Sigma^*$ erweitern:

$$PR(L) = \{ u \in \Sigma^* \mid \text{es existiert } v \in \Sigma^* \text{ mit } uv \in L \}$$

Aus der Definition folgt unmittelbar $L \subseteq PR(L)$ und $\Sigma^* \in PR(L)$, falls $L \neq \emptyset$.

Eine Sprache $L \subseteq \Sigma^*$ heißt präfixstabil, falls $L = PR(L)$.

Für den Rest des Papieres soll jetzt generell vorausgesetzt werden, daß die betrachteten

formalen Sprachen **präfixstabil** sind und, da vergrößernde Sichtweisen beschrieben werden, daß die Homomorphismen **alphabetisch** sind. Es sei noch erwähnt, daß diese Voraussetzungen aber nicht für jede Aussage notwendig sind.

Für ein Wort $u \in \Sigma^*$ und eine Sprache $L \subseteq \Sigma^*$ bezeichnet uL die Konkatenation von u mit L . Sie ist definiert durch $uL = \{ x \in \Sigma^* \mid \text{es existiert } v \in L \text{ mit } uv = x \}$.

Mit den Definitionen Linksquotient und Konkatenation lassen sich leicht folgende Aussagen für $u, v \in \Sigma^*$ und $L \subseteq \Sigma^*$ zeigen:

- (1) $u^{-1}(L) = L$
- (2) $(uv)^{-1}(L) = v^{-1}(u^{-1}(L))$
- (3) $u^{-1}(uL) = L$
- (4) $u(u^{-1}(L)) \subseteq L$
- (5) $u^{-1}(L') \subseteq u^{-1}(L)$ für $L' \subseteq L$
- (6) Ist $h : \Sigma^* \rightarrow \Sigma'^*$ ein Homomorphismus, dann gilt $h(u^{-1}(L)) \subseteq h(u)^{-1}(h(L))$.
- (7) $u^{-1}(L)$ genau dann, wenn $u \in L$.

Es sei bemerkt, daß die Inklusionen in (4) und (6) i. allg. echt sind.

Mit (2) und (4) folgt aus der Schlichtheitsbedingung (*)

$$u((h(x)u)^{-1}(h(L))) \subseteq h(x^{-1}(L)). \quad (**)$$

Umgekehrt folgt aus dieser Inklusion mit (3) $(h(x)u)^{-1}(h(L)) \subseteq u^{-1}(h(x^{-1}(L)))$ und daraus mit (2) $u^{-1}(h(x)^{-1}(h(L))) \subseteq u^{-1}(h(x^{-1}(L)))$. Wegen (6) und (5) gilt $u^{-1}(h(x^{-1}(L))) \subseteq u^{-1}(h(x)^{-1}(h(L)))$, woraus dann die Gleichung (*) folgt. Für die Definition der Schlichkeit von h auf L sind damit die Gleichung (*) und die Inklusion (**) äquivalent.

Bemerkung: Für jedes $L \subseteq \Sigma^*$ ist jeder Homomorphismus h auf Σ^* schlicht auf L in x ($u = \epsilon$).

Hilfssatz 1: Ist der Homomorphismus h schlicht auf L in x , dann ist er auch schlicht auf L in jedem $y \in PR(x)$.

Beweis:

Sei $y \in PR(x)$, dann gibt es ein $z \in \Sigma^*$ mit $x = yz$. Wegen (4) gilt $z(z^{-1}(y^{-1}(L))) \subseteq y^{-1}(L)$, und aus (2) folgt dann $z(x^{-1}(L)) \subseteq y^{-1}(L)$. Damit gilt $h(z)h(x^{-1}(L)) \subseteq h(y^{-1}(L))$.

Wegen der Schlichkeit von h auf L in x gibt es ein $u \in h(x)^{-1}(h(L))$ mit $u((h(x)u)^{-1}(h(L))) \subseteq h(x^{-1}(L))$, woraus dann $h(z)u((h(x)u)^{-1}(h(L))) \subseteq h(y^{-1}(L))$ folgt. Mit $v = h(z)u$ folgt daraus $v((h(y)v)^{-1}(h(L))) \subseteq h(y^{-1}(L))$, also die Schlichkeit von h auf L in y .

Wie schon im vorhergehenden Kapitel erwähnt wurde, ist für eine Sprache $L \subseteq \Sigma^*$ und einen Homomorphismus $h : \Sigma^* \rightarrow \Sigma'^*$ die Deadlocksprache DL definiert durch

$DL = \{ u \in \Sigma^* \mid \text{es existiert } x \in L \text{ mit } u = h(x), \text{ und } h(x^{-1}(L)) = \{ \} \} .$

Ist ein Wort u Element der Deadlocksprache, dann bedeutet dies, daß die "Verhaltensweise des Systems", die durch u beschrieben ist, unter der gröberen Sichtweise des Homomorphismus zu einem "Deadlock" führen kann, aber nicht notwendigerweise führen muß. Das Bild $h(L)$ liefert in diesem Fall noch zusätzliche Information. Wenn es nämlich in $h(L)$ kein Wort u' gibt, welches u als echten Präfix enthält, dann führt u zwangsweise zu einem "Deadlock" unter der gröberen Sichtweise. Solche "Deadlocks" sind auch schon alleine aus $h(L)$ erkennbar. Im anderen Fall ist unter der gröberen Sichtweise nicht erkennbar, ob ein "Deadlock" erreicht ist oder nicht.

Es läßt sich zeigen, daß bei schlichten Homomorphismen dieser zweite Fall nicht auftreten kann. Dazu sei die Terminierungssprache TL definiert durch

$TL = \{ u \in h(L) \mid u^{-1}(h(L)) = \{ \} \} .$

Ist $h(L)$ eine reguläre Sprache, dann besitzt der Minimalautomat für $h(L)$ höchstens einen Zustand, der keinen Nachfolgezustand besitzt (wegen der Minimalität). Durch Wahl dieses Zustands als einzigen Endzustand und Entfernen der "überflüssigen Zustände und Kanten" entsteht der Minimalautomat für TL .

Aus der Definition von TL folgt mit (6) und (7) $TL \subseteq DL$.

Wie Beispiel 3 in /14/ zeigt, ist diese Inklusion i. allg. echt.

Satz 1 : Ist der Homomorphismus h schlicht auf L , dann gilt $TL = DL$.

Beweis :

Wegen der obigen Inklusion muß nur noch $DL \subseteq TL$ gezeigt werden. Sei also $y \in DL$.

Dann existiert ein $x \in L$ mit $y = h(x)$, und $h(x^{-1}(L)) = \{ \} .$ Für jedes $u \in \Sigma^+$ gilt deshalb $u^{-1}(h(x^{-1}(L))) = \{ \} .$ Da aber $u^{-1}(h(x)^{-1}(h(L))) = \{ \} .$ für jedes $u \in h(x)^{-1}(h(L))$, folgt aus der Definition der Schlichtheit von h $u^{-1}(h(x)^{-1}(h(L))) = \{ \} .$

Damit gilt $h(x)^{-1}(h(L)) = \{ \} ,$ woraus $y \in TL$ folgt.

Für zwei Abbildungen $h : \Sigma_1^* \rightarrow \Sigma_2^*$ und $g : \Sigma_2^* \rightarrow \Sigma_3^*$ bezeichnet $g \circ h$ die Komposition der Abbildungen, die definiert ist durch :

$g \circ h : \Sigma_1^* \rightarrow \Sigma_3^*$ mit $(g \circ h)(x) = g(h(x))$ für alle $x \in \Sigma_1^* .$

Sind g und h Homomorphismen, dann folgt durch einfaches Nachrechnen, daß auch $g \circ h$ ein Homomorphismus ist. Unter dieser Voraussetzung gilt

Satz 2 : Ist h schlicht auf $L \subseteq \Sigma_1^*$ und g schlicht auf $h(L) \subseteq \Sigma_2^*$, dann ist $g \circ h$ schlicht auf L .

Beweis :

Wegen der Schlichtheit von h auf L gibt es zu jedem $x \in L$ ein $u \in h(x)^{-1}(h(L))$ mit $u \in (h(x)u)^{-1}(h(L)) \cap h(x^{-1}(L))$, woraus $g(u)g((h(x)u)^{-1}(h(L))) \subseteq g(h(x^{-1}(L)))$ folgt.

Wegen der Schlichtheit von g auf $h(L)$ gibt es zu $h(x)u \in h(L)$ ein $v \in g(h(x)u)^{-1}(g(h(L)))$ mit $v \in ((g(h(x)u)v)^{-1}(g(h(L)))) \cap g((h(x)u)^{-1}(h(L)))$, woraus dann mit der vorhergehenden Inklusion $g(u)v \in ((g(h(x)u)v)^{-1}(g(h(L)))) \cap g(h(x^{-1}(L)))$ folgt. Das bedeutet aber die Schlichtheit von $g \circ h$ auf L , da mit $v \in g(h(x)u)^{-1}(g(h(L)))$ auch $g(u)v \in g(h(x))^{-1}(g(h(L)))$ gilt.

Bezüglich g gilt auch die Umkehrung von Satz 2.

Satz 3: Ist $g \circ h$ schlicht auf L , dann ist g schlicht auf $h(L)$.

Beweis :

Sei $y \in h(L)$, dann existiert ein $x \in L$ mit $y = h(x)$. Aus (6) folgt $g(y^{-1}(h(L))) = g(h(x)^{-1}(h(L))) \cap g(h(x^{-1}(L)))$. Wegen der Schlichtheit von $g \circ h$ auf L gibt es ein $u \in g(h(x))^{-1}(g(h(L)))$ mit $g(h(x^{-1}(L))) \cap u \in ((g(h(x)u)^{-1}(g(h(L))))$. Damit gibt es zu jedem

$y \in h(L)$ ein $u \in g(y)^{-1}(g(h(L)))$ mit $g(y^{-1}(h(L))) \cap u \in ((g(y)u)^{-1}(g(h(L))))$.

Also ist g schlicht auf $h(L)$.

Reguläre Sprachen

Bei vielen Spezifikationen kooperierender Systeme gibt es Abstraktionsstufen, auf denen sich einerseits wichtige Systemeigenschaften ausdrücken lassen, und die andererseits einen endlichen Erreichbarkeitsgraph besitzen. Von daher ist es sinnvoll, speziell die Schlichtheit von Homomorphismen auf regulären Sprachen zu untersuchen.

Bemerkung :

Liegt eine Systemspezifikation in Form eines Petrinetzes mit Kantenvielfachheiten vor, dann ist die Endlichkeit des Erreichbarkeitsgraphen entscheidbar /9/. Bei ausdrucksstärkeren Spezifikationsmethoden, insbesondere solchen, die turingmächtig sind, wie z.B. Produktnetze /13/, ist es nicht entscheidbar, ob der Erreichbarkeitsgraph endlich ist. Bei der Definition der Produktnetze wurde deshalb besonderen Wert darauf gelegt, daß für jeden Knoten des Erreichbarkeitsgraphen die Menge der Nachfolgeknoten endlich ist und vollständig berechnet werden kann. Ausgehend von der Anfangsmarkierung kann damit der Erreichbarkeitsgraph schrittweise erzeugt werden, und es kann nach jedem Schritt festgestellt werden, ob der vollständige Graph erreicht ist.

Ein (nichtdeterministischer) endliche Automat $\mathbb{A} = (S, \Sigma, q_0, F)$ besteht aus

- einer endlichen Zustandsmenge S ,
- einem endlichen Alphabet Σ ,
- einer Zustandsüberföhrungsfunktion $\delta : S \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^S$,
- einem Anfangszustand $q_0 \in S$
- und einer Menge $F \subseteq S$ von Endzuständen .

Zur Definition der Arbeitsweise eines endlichen Automaten wird die Zustandsüberföhrungsfunktion von $S \times (\Sigma \cup \{\epsilon\})$ auf $S \times \Sigma^*$ fortgesetzt :

$\delta^* : S \times \Sigma^* \rightarrow 2^S$ mit

$\delta^*(q, \epsilon) = \{ p \in S \mid \text{es existieren } p_0, \dots, p_n \in S \text{ (} n \geq 0 \text{) mit } q = p_0, p = p_n \text{ und } p_i \in \delta^*(p_{i-1}, a) \text{ f\u00fcr alle } 1 \leq i \leq n \}$

und

$\delta^*(q, wa) = \delta^*(\delta^*(q, w), a)$ f\u00fcr $q \in S$, $a \in \Sigma$ und $w \in \Sigma^*$.

Im zweiten Teil der Definition werden beide Funktionen δ und δ^* in nat\u00fcrlicher Weise auch auf Teilmengen von S in der ersten Komponente betrachtet. Dabei gilt :

$\delta(A, a) = \{ q \in S \mid \text{es existiert } p \in A \text{ mit } q \in \delta(p, a) \}$ f\u00fcr $A \subseteq S$ und entsprechend f\u00fcr δ^* .

Ein Wort $w \in \Sigma^*$ wird genau dann von \mathbb{A} akzeptiert, wenn $\delta^*(q_0, w) \cap F \neq \emptyset$ gilt.

Eine Sprache heißt regulär, wenn ihre Worte von einem endlichen Automaten erkannt werden.

Im vorletzten Kapitel wurde schon der Begriff des deterministischen endlichen Automaten eingeführt. Dies sind spezielle nichtdeterministische Automaten, wobei für jedes $q \in S$

$(q, a) = \emptyset$ und für jedes $a \in \Sigma$ die Menge $\delta(q, a)$ höchstens ein Element enthält. Bei deterministischen Automaten kann die Zustandsüberföhrungsfunktion auch als partielle Funktion $\delta : S \times \Sigma \rightarrow S$ betrachtet werden. Entsprechend ist dann δ^* eine partielle Funktion $\delta^* : S \times \Sigma^* \rightarrow S$.

Zu jeder regulären Sprache L gibt es bekanntlich [6, 19] einen deterministischen endlichen Automaten mit minimaler Zustandszahl (Minimalautomat), der L erkennt.

Für $x \in L$ ist $x^{-1}(L)$ regulär. Wählt man nämlich in einem deterministischen endlichen Automaten, der L erkennt, den Zustand $\delta^*(q_0, x)$ als neuen Anfangszustand, dann entsteht ein deterministischer endlicher Automat, der $x^{-1}(L)$ erkennt.

Wegen der generellen Voraussetzung der Präfixstabilität der Sprachen, brauchen im folgenden nur solche endlichen Automaten betrachtet werden, deren Endzustandsmenge gleich der ganzen Zustandsmenge ist.

Die Zustandsüberföhrungsfunktionen endlicher Automaten werden üblicherweise auch als gerichtete Graphen mit beschrifteten Kanten dargestellt, den sogenannten Zustandsgraphen.

Die Knoten des Graphen sind die Zustände des Automaten, und die beschrifteten Kanten stellen die Zustandsübergänge dar, wobei im Graphen genau dann eine Kante vom Knoten q zum Knoten p führt und mit $a \in \Sigma$ beschriftet ist, wenn $p \in \delta(q, a)$ gilt.

Ein Wort $w \in \Sigma^*$ wird genau dann von \mathbb{A} akzeptiert, wenn es im Zustandsgraphen einen Pfad gibt, der vom Anfangszustand zu einem Endzustand führt und dessen zusammengesetzte Kantenbeschriftung das Wort w ergibt. Wir sagen dazu auch, daß der Pfad mit dem Wort w beschriftet ist.

Bekanntlich ist das homomorphe Bild einer regulären Sprache ebenfalls regulär. Ein endlicher Automat \mathbb{A}' , der das homomorphe Bild $h(L)$ akzeptiert, läßt sich jetzt leicht aus dem Automaten \mathbb{A} , der L akzeptiert, konstruieren, nämlich durch "Einfügen" der Abbildung h in die Zustandsüberföhrungsfunktion, genauer :

$\mathbb{A}' = (S, \Sigma, \delta', q_0, F)$ mit $\delta' = h(\delta)$, und für $x \in \Sigma^*$ und $q \in S$ ist die Zustandsüberföhrungsfunktion $\delta' : S \times (\Sigma \cup \{x\}) \rightarrow S$ definiert durch $\delta'(q, x) = \{ p \in S \mid \text{es existiert } y \in \Sigma^* \text{ mit } h(y) = x \text{ und } p \in \delta(q, y) \}$.

\mathbb{A}' heißt der zu \mathbb{A} homomorphe Automat.

Der Zustandsgraph von \mathbb{A}' entsteht aus dem Zustandsgraphen von \mathbb{A} durch Ersetzen der

Kantenanschriften durch ihre homomorphen Bilder und gegebenenfalls durch das Zusammenlegen von dadurch entstehenden Mehrfachkanten mit gleicher Anschrift.

Satz 4: Ist $L \subseteq \Sigma^*$ eine reguläre Sprache, $h : \Sigma^* \rightarrow \Sigma'^*$ ein Homomorphismus und $x \in L$, dann ist es entscheidbar, ob h schlicht auf L in x ist.

Beweis:

Ist A eine reguläre Sprache, dann ist $\{x^{-1}(A) \mid x \in A\}$ eine endliche Menge, denn die Elemente dieser Menge entsprechen gerade den Zuständen des Minimalautomaten von A . Für jedes $B \in \{x^{-1}(A) \mid x \in A\}$ sei die Sprache A_B definiert durch $A_B = \{x \in A \mid x^{-1}(A) = B\}$. Wählt man im Minimalautomaten von A den Zustand, der B entspricht, als einzigen Endzustand, dann entsteht ein endlicher Automat, der A_B erkennt. Damit ist A_B eine reguläre Sprache.

Mit L sind auch die Sprachen $h(x^{-1}(L))$ und $h(x)^{-1}(h(L))$ regulär.

Der Homomorphismus h ist genau dann schlicht auf L in x , wenn es ein

$B \in \{u^{-1}(h(x^{-1}(L))) \subseteq \Sigma'^* \mid u \in h(x^{-1}(L))\} \cap \{u^{-1}(h(x)^{-1}(h(L))) \subseteq \Sigma'^* \mid u \in h(x)^{-1}(h(L))\}$ gibt mit $h(x^{-1}(L))_B = (h(x)^{-1}(h(L)))_B$. Diese Bedingung ist entscheidbar.

Satz 5: Ist $L \subseteq \Sigma^*$ eine reguläre Sprache und $h : \Sigma^* \rightarrow \Sigma'^*$ ein Homomorphismus, dann ist es entscheidbar, ob h schlicht auf L ist.

Beweis:

Mit der Terminologie des obigen Beweises gilt:

Der Homomorphismus h ist genau dann schlicht auf L , wenn es zu jedem

$C \in \{x^{-1}(L) \subseteq \Sigma^* \mid x \in L\}$ und jedem $D \in \{v^{-1}(h(L)) \subseteq \Sigma'^* \mid v \in h(L)\}$ mit $h(L_C) = L_D$

ein $E \in \{u^{-1}(h(C)) \subseteq \Sigma'^* \mid u \in h(C)\} \cap \{u^{-1}(D) \subseteq \Sigma'^* \mid u \in D\}$ gibt mit $h(C)_E = D_E$. Diese Bedingung ist entscheidbar.

Diese beiden Entscheidungsverfahren sind sehr komplex, da in ihnen vielfach zu einem nichtdeterministischen Automaten ein äquivalenter deterministischer Automat konstruiert werden muß. Es ist daher sehr wichtig, einfacher zu entscheidende hinreichende Bedingungen für die Schlichtheit zu finden.

In Satz 6 wird eine solche Bedingung angegeben. Dazu werden einige Begriffe aus der Graphentheorie benötigt.

Ein gerichteter Graph heißt stark zusammenhängend, wenn von jedem Knoten zu jedem anderen Knoten ein Pfad existiert.

Die maximalen Teilgraphen eines gerichteten Graphen, die stark zusammenhängend sind, nennt man starke Zusammenhangskomponenten.

Ein gerichteter Graph heißt pseudo stark zusammenhängend, wenn er eine starke Zusammenhangskomponente besitzt, sodaß außerhalb dieser Komponente der Graph zyklensfrei ist und von jedem Knoten ein Pfad in diese Komponente führt. Diese starke

Zusammenhangskomponente nennen wir den Kern des Graphen.

Mit bekannten Graphenalgorithmien /2/ läßt sich diese Eigenschaft leicht entscheiden.

Die beiden folgenden Graphen sind pseudo stark zusammenhängend:

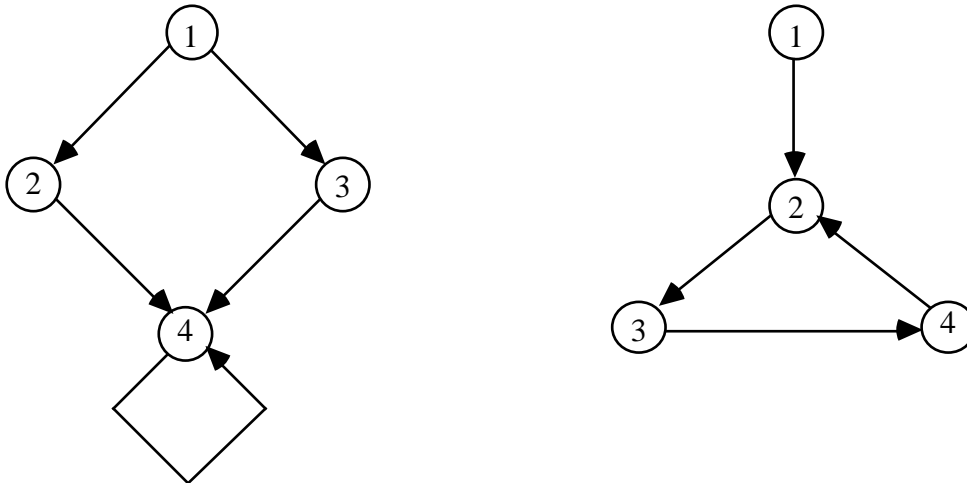


Fig.9

Im linken Graphen besteht der Kern nur aus dem Knoten 4 und der Kante von 4 nach 4 und im rechten Graphen aus dem Teilgraphen, der von den Knoten 2,3 und 4 erzeugt wird.

Der Graph

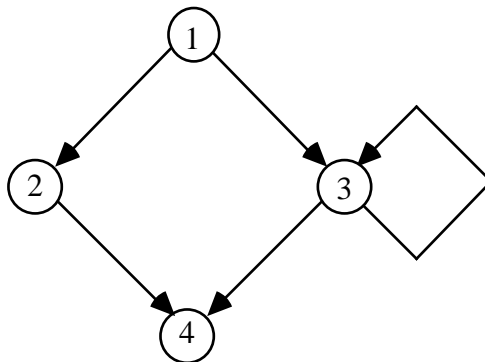


Fig.10

ist nicht pseudo stark zusammenhängend.

Die Bedeutung dieser Definition liegt in dem folgenden Satz und der Tatsache, daß die Erreichbarkeitsgraphen vieler realistischer Spezifikationen pseudo stark zusammenhängend sind.

Satz 6: Wird eine Sprache L von einem endlichen Automaten \mathbb{A} erkannt, dessen Zustandsgraph pseudo stark zusammenhängend ist, dann ist jeder Homomorphismus h schlicht auf L ,

für den $h(L)$ unendlich ist.

Die Unendlichkeitsbedingung ist genau dann erfüllt, wenn es im Kern des Zustandsgraphen eine Kante gibt, deren Anschrift durch den Homomorphismus nicht auf ϵ abgebildet wird.

Für den Beweis dieses Satzes werden noch drei Hilfssätze benötigt.

Hilfssatz 2: Sind A und B Teilmengen einer halbgeordneten Menge mit der Eigenschaft, daß es zu jedem $x \in A$ ein $y \in B$ mit $y \leq x$ und zu jedem $u \in B$ ein $v \in A$ mit $v \leq u$ gibt, dann sind die maximalen Elemente von A gleich denen von B .

Beweis :

Sei z maximal in A . Nach Voraussetzung gibt es ein $y \in B$ mit $y \leq z$ und ein $v \in A$ mit $v \leq y$, also $v \leq y \leq z$. Wegen der Maximalität von z in A folgt daraus $x = y = z$, also auch $z \in B$.

Sei $s \in B$ mit $s \leq z$, dann gibt es nach Voraussetzung ein $t \in A$ mit $t \leq s$, also $t \leq s \leq z$. Wegen der Maximalität von z in A folgt daraus $t = s = z$. Also ist auch z maximal in B .

Die Umkehrung läßt sich genauso beweisen.

Für einen endlichen Automaten und einen Zustand q aus seiner Zustandsmenge bezeichne L_q die Sprache, die der Automat erkennt, wenn an Stelle des ursprünglichen Anfangszustandes der Zustand q als Anfangszustand gewählt wird.

Hilfssatz 3: Seien p und q Zustände eines endlichen Automaten, dessen Zustandsgraph pseudo stark zusammenhängend ist. Liegen p und q im Kern des Zustandsgraphen, und u ist $u \in L_p$ mit $u^{-1}(L_p) \subseteq u^{-1}(L_q)$, dann existiert ein $x \in u^{-1}(L_p)$ mit $(ux)^{-1}(L_p) = (ux)^{-1}(L_q)$.

Beweis :

S bezeichne die Zustandsmenge des betrachteten endlichen Automaten. Für $x \in S$ und $y \in L_x$ gilt dann $y^{-1}(L_x) = L_{z_1} \dots L_{z_k}$ mit $\{z_1, \dots, z_k\} = \{z \in S \mid z \stackrel{*}{\rightarrow} (x, y)\}$.

Sei $\mathbb{X} = \{L_{x_1} \dots L_{x_n} \stackrel{*}{\rightarrow} \mid n \in \mathbb{N} \text{ und } x_i \in S \text{ für } 1 \leq i \leq n\}$.

(\mathbb{N} bezeichnet dabei die Menge der natürlichen Zahlen)

\mathbb{X} ist eine endliche Menge, und die Mengeninklusion ist eine Halbordnung auf \mathbb{X} . Die Mengen $\mathbb{X}_p = \{(ux)^{-1}(L_p) \stackrel{*}{\rightarrow} \mid x \in u^{-1}(L_p)\}$ und $\mathbb{X}_q = \{(ux)^{-1}(L_q) \stackrel{*}{\rightarrow} \mid x \in u^{-1}(L_q)\}$ sind dann Teilmengen von \mathbb{X} .

Wegen $u \in L_p$ gibt es im Zustandsgraphen des Automaten einen Pfad, der von p zu einem Zustand r führt und mit u beschriftet ist. Damit liegt auch r im Kern des Zustandsgraphen, und es gibt einen Pfad von r nach q . Dieser sei mit v beschriftet. Daraus folgt $v \in u^{-1}(L_p)$ und $(uv)^{-1}(L_p) \subseteq L_q$. Für jedes $y \in u^{-1}(L_q)$ gilt dann $(uvuy)^{-1}(L_p) \subseteq (uy)^{-1}(L_q)$.

Zu jedem Element von \mathbb{X}_q gibt es also ein größeres Element von \mathbb{X}_p bezüglich der Mengeninklusion. Entsprechend läßt sich eine Aussage beweisen, bei der die Rollen von p und

q vertauscht sind. Damit erfüllen \mathbb{X}_p und \mathbb{X}_q die Voraussetzungen von Hilfssatz 2 und besitzen deshalb die gleichen maximalen Elemente. Da die Mengen endlich sind, gibt es auch ein solches maximales Element $(ux)^{-1}(L_p) \in \mathbb{X}_p$.

Aus $u^{-1}(L_p) \subseteq u^{-1}(L_q)$ folgt $(ux)^{-1}(L_p) \subseteq (ux)^{-1}(L_q) \in \mathbb{X}_q$. Da $(ux)^{-1}(L_p)$ auch maximal in \mathbb{X}_q ist, folgt daraus $(ux)^{-1}(L_p) = (ux)^{-1}(L_q)$, also die Behauptung des Hilfssatzes 3.

Hilfssatz 4: Seien für $n \geq 1$ und für $1 \leq i \leq n$ p_i Zustände eines endlichen Automaten, dessen Zustandsgraph pseudo stark zusammenhängend ist. Liegen alle p_i im Kern des Zustandsgraphen, dann existiert ein $u \in L_{p_1}$, sodaß

$$u^{-1}(L_{p_1} \dots L_{p_k}) = u^{-1}(L_{p_1}) \text{ für jedes } 1 \leq k \leq n.$$

Induktionsbeweis über n :

Induktionsverankerung: Die Aussage gilt trivialerweise für $n = 1$.

Nach Induktionsvoraussetzung gibt es ein $v \in L_{p_1}$, sodaß $v^{-1}(L_{p_1} \dots L_{p_k}) = v^{-1}(L_{p_1})$ für jedes $1 \leq k \leq n$. Falls es ein $x \in v^{-1}(L_{p_1})$ gibt mit $x \in v^{-1}(L_{p_{n+1}})$, dann gilt

$$(vx)^{-1}(L_{p_{n+1}}) = v^{-1}(L_{p_1}) \cap v^{-1}(L_{p_{n+1}}).$$

Nach Hilfssatz 3 gibt es dann ein $x \in v^{-1}(L_{p_1})$ mit $(vx)^{-1}(L_{p_1}) = (vx)^{-1}(L_{p_{n+1}})$. Aus der Induktionsvoraussetzung folgt $(vx)^{-1}(L_{p_1} \dots L_{p_k}) = (vx)^{-1}(L_{p_1})$ für jedes $1 \leq k \leq n$.

In beiden Fällen gilt dann $(vx)^{-1}(L_{p_1} \dots L_{p_{n+1}}) = (vx)^{-1}(L_{p_1} \dots L_{p_n}) \cap (vx)^{-1}(L_{p_{n+1}}) = (vx)^{-1}(L_{p_1}) \cap (vx)^{-1}(L_{p_{n+1}}) = (vx)^{-1}(L_{p_1})$. Mit $u = vx$ ist dies die Behauptung.

Beweis des Satzes 6:

Sei \mathbb{A}' der zu \mathbb{A} homomorphe Automat. Seine Zustandsmenge, die gleich der von \mathbb{A} ist, sei S . Für jeden Zustand $z \in S$ bezeichne $L'z$ die Sprache, welche \mathbb{A}' erkennt, wenn nicht q_0 , sondern z als Anfangszustand gewählt wird. Damit gilt $h(Lz) = L'z$.

Da der Zustandsgraph von \mathbb{A} , und damit auch der von \mathbb{A}' , pseudo stark zusammenhängend ist, gibt es eine Zahl e , sodaß jeder Pfad, der vom Anfangszustand ausgeht und mindestens die Länge e besitzt, zu einem Knoten im Kern des Zustandsgraphen führt. Außerdem folgt aus der Voraussetzung des Satzes, daß es in \mathbb{A}' von jedem Knoten aus einen Pfad gibt, der mit einem Wort beschriftet ist, welches mindestens die Länge e besitzt.

Damit gibt es zu jedem $x \in L$ ein $y \in L$ mit $x \in PR(y)$ und $|h(y)| \geq e$.

(Für ein Wort w bezeichne dabei $|w|$ die Länge des Wortes w .)

Wegen Hilfssatz 1 genügt es also, die Schlichtheit von h auf L in den $y \in L$ zu zeigen, für die $|h(y)| \geq e$ gilt.

Es gilt jetzt $h(y^{-1}(L)) = h(Lz_1 \dots Lz_k) = L'z_1 \dots L'z_k$ mit

$$\{z_1, \dots, z_k\} = \{z \in S \mid z \in \text{Kern}(\mathbb{A}') \cap \text{Kern}(h) \}.$$

Aus $\{z \in S \mid z \in \text{Kern}(\mathbb{A}') \cap \text{Kern}(h)\} = \{z \in S \mid z \in \text{Kern}(\mathbb{A}') \cap \text{Kern}(h(y))\}$ folgt dann

$h(y)^{-1}(h(L)) = L'z_1 \dots L'z_n$ mit $\{z_1, \dots, z_n\} = \{z \in S \mid z \in \hat{*}(q_0, h(y))\}$, $n \leq k$, und die z_i liegen alle im Kern des Zustandsgraphen.

Hilfssatz 4, angewendet auf den Automaten $\hat{\mathbb{A}}$, liefert jetzt die Existenz eines $u \in h(y)^{-1}(h(L))$ mit $u^{-1}(h(y^{-1}(L))) = u^{-1}(h(y)^{-1}(h(L)))$, also die Schlichtheit von h auf L in y .

Der folgende Satz liefert eine weitere einfach zu überprüfende hinreichende Bedingung für die Schlichtheit eines Homomorphismus.

Satz 7: Sei $h : \hat{*} \rightarrow \hat{*}$ ein Homomorphismus und $L \subseteq \hat{*}$ eine Sprache, die von einem endlichen Automaten \mathbb{A} erkannt wird.

Ist der zu \mathbb{A} homomorphe Automat $\hat{\mathbb{A}}$ deterministisch, dann ist h schlicht auf L .

Beweis:

Mit $\hat{\mathbb{A}}$ ist auch \mathbb{A} deterministisch. Beachtet man, daß $\hat{\mathbb{A}}$ und \mathbb{A} die gleichen Zustandsmengen haben, dann gilt $\hat{*}(q_0, x) = \hat{*}(q_0, h(x))$ für jedes $x \in L$.

Mit der vor Hilfssatz 3 eingeführten Notation L_q gilt dann bezüglich des Automaten $\hat{\mathbb{A}}$ $h(x^{-1}(L)) = L \cap \hat{*}(q_0, h(x)) = h(x)^{-1}(h(L))$. Also ist h schlicht auf L ($u = \epsilon$).

Bei den Beweisen der letzten Sätze wurde fast nur auf der Ebene der homomorphen Automaten argumentiert. Das legt es nahe, den Begriff der Schlichtheit auch für Automaten zu definieren.

Die Kanten des Zustandsgraphen eines Automaten lassen sich als Tripel

$(p, a, q) \in S \times (\Sigma \cup \{\epsilon\}) \times S$ darstellen. Die Menge aller Kanten sei mit

$S \times (\Sigma \cup \{\epsilon\}) \times S$ bezeichnet.

Jeder Pfad im Zustandsgraphen ist damit ein Element von $\hat{*}$. Die Menge aller Pfade, die vom Anfangszustand q_0 ausgehen, sei mit $P \subseteq \hat{*}$ bezeichnet. Ist $h : \hat{*} \rightarrow \hat{*}$ der Homomorphismus, der durch $h((p, a, q)) = a$ für $(p, a, q) \in K$ definiert ist, und ist L die Sprache, die vom betrachteten Automaten erkannt wird, dann gilt $L = h(P)$.

Ersetzt man im Automaten die Anschrift a einer Kante, die von p nach q führt, durch die Anschrift (p, a, q) , dann entsteht ein deterministischer endlicher Automat, der die Sprache P erkennt.

Ein endlicher Automat heißt schlicht, wenn der Homomorphismus $h : \hat{*} \rightarrow \hat{*}$ schlicht auf P ist.

Aus den Sätzen 5, 6 und 7 folgen unmittelbar die folgenden drei Korollare :

Korollar 1 : Es ist entscheidbar, ob ein endlicher Automat schlicht ist.

Korollar 2 : Ist der Zustandsgraph eines endlichen Automaten pseudo stark zusammenhängend, und enthält sein Kern eine Kante, die nicht mit ϵ beschriftet ist, dann ist der Automat schlicht.

Korollar 3 : Jeder deterministische Automat ist schlicht.

Die Definition der Schlichtheit von Automaten wird durch den folgenden Satz gerechtfertigt.

Satz 8 : Sei $h : \Sigma^* \rightarrow \Sigma'^*$ ein Homomorphismus, \mathbb{A} ein schlichter Automat, der die Sprache $L \subseteq \Sigma^*$ erkennt und \mathbb{A}' der zu \mathbb{A} homomorphe Automat.

Der Homomorphismus h ist genau dann schlicht auf L , wenn \mathbb{A}' ein schlichter Automat ist.

Beweis :

Sei $E = S \times (\Sigma \cup \{ \epsilon \}) \times S$ die Kantenmenge von \mathbb{A}' , $P' \subseteq \Sigma'^*$ die Menge aller Pfade in \mathbb{A}' , die vom Anfangszustand ausgehen, $h : \Sigma^* \rightarrow \Sigma'^*$ der entsprechende Homomorphismus mit $h(L) = P'$ und $\alpha : \Sigma^* \rightarrow \Sigma'^*$ der Homomorphismus, der durch $((p,a,q) = (p,h(a),q))$ für $(p,a,q) \in E$ definiert ist.

Damit gilt $P' = \alpha(P)$ und $h \circ \alpha = \alpha \circ h$. Da der entsprechende Automat für P' deterministisch ist, folgt aus Satz 7 die Schlichtheit von α auf P . Wegen der vorausgesetzten Schlichtheit von h folgt aus der Gleichung $h \circ \alpha = \alpha \circ h$ und den Sätzen 2 und 3 die Aussage, daß h genau dann schlicht auf L ist, wenn α schlicht auf P' ist. Das ist gerade die Behauptung des Satzes.

Bekanntlich läßt sich zu jedem endlichen Automaten \mathbb{A} ein äquivalenter endlicher Automat \mathbb{A}_ϵ konstruieren, der keine ϵ -Übergänge enthält. Für dessen Zustandsüberföhrungsfunktion gilt also $(q, \epsilon) = (q, \epsilon)$ für jedes $q \in S$. Nach der üblichen Konstruktion ist \mathbb{A}_ϵ definiert durch $(q,a) = (\epsilon^*(q, \epsilon), a)$ für jedes $q \in S$ und $a \in \Sigma$. \mathbb{A}_ϵ heißt der ϵ -freie Automat von \mathbb{A} .

Satz 9 : Sei \mathbb{A} ein endlicher Automat und $L \subseteq \Sigma^*$ die Sprache, die er erkennt.

Bezüglich des Homomorphismus $\alpha : \Sigma^* \rightarrow \Sigma'^*$ und der Sprache $P \subseteq \Sigma'^*$ bezeichne DP die Deadlocksprache und TP die Terminierungssprache.

\mathbb{A} ist genau dann schlicht, wenn \mathbb{A}_ϵ schlicht ist, und wenn $DP = TP$.

Beweis :

Sei $E = S \times \Sigma \times S$ die Kantenmenge von \mathbb{A} , $P \subseteq \Sigma^*$ die Menge aller Pfade in \mathbb{A} , die vom Anfangszustand ausgehen und $\alpha : \Sigma^* \rightarrow \Sigma'^*$ der entsprechende Homomorphismus mit $L = \alpha(P)$.

(1) Sei \mathbb{A} ein schlichter Automat.

Nach Satz 1 gilt dann $DP = TP$.

Nach Konstruktion des Automaten \mathbb{A}_ϵ gibt es zu jedem $x \in P$ ein $y \in P$ mit $\alpha(x) = \alpha(y)$ und $\alpha(x^{-1}(P)) = \alpha(y^{-1}(P))$. Wegen $\alpha(P) = L = \alpha(P)$ folgt dann aus der Schlichtheit von α auf P die Schlichtheit von α auf P . Also ist \mathbb{A}_ϵ ein schlichter Automat.

(2) Sei \mathbb{A}_ϵ ein schlichter Automat und $DP = TP$.

Ist $y \in P$ mit $\alpha(y^{-1}(P)) = \{ \}$, dann ist $\alpha(y) \in DP$, und damit auch $\alpha(y) \in TP$, also $\alpha(y)^{-1}(\alpha(P)) = \{ \}$. Daraus folgt, daß α schlicht auf P in y ist ($u = y$).

Ist $y \in P$ mit $(y^{-1}(P)) \neq \{ \}$, dann existiert ein $z \in y^{-1}(P)$ minimaler Länge mit $(z) \neq \{ \}$.
 Nach Konstruktion des Automaten \mathbb{A} gibt es dann ein $x \in P$ mit $(x) = (yz)$ und
 $(x^{-1}(P)) = ((yz)^{-1}(P))$. Wegen $(P) = (P)$ folgt dann aus der Schlichtheit von (z) auf
 P in x die Schlichtheit von (z) auf P in yz . Mit Hilfssatz 1 ist dann auch (z) schlicht auf P in y .

Ein Beispiel

Wir betrachten in diesem Kapitel eine einfache Version eines Verbindungsauf- und -abbauprotokolls [5, 10,11]. Dieses Protokoll beschreibt das Zusammenspiel eines Senders A, eines Empfängers B und eines Transportsystems T. Die Grobstruktur des Systems ist in Abb. 15.11 dargestellt:

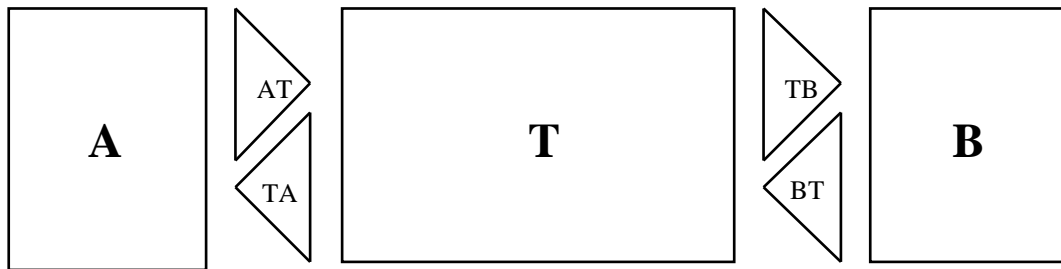


Abb. 15.11

A und B sind jeweils über zwei FIFO-Kanäle mit dem Transportsystem T gekoppelt.

Wenn A unter Benutzung von T eine Verbindung zu B aufbauen möchte, dann teilt er dies mit einem CR (connect request) dem Transportsystem T mit. T kann diesen Wunsch mit einem D (disconnect) ablehnen, falls es z.B. momentan über keine freien Betriebsmittel für diese Verbindung verfügt, oder es leitet den Wunsch an B weiter. Auch B kann den Wunsch mit einem D ablehnen oder mit einem CC (connect confirm) annehmen. In beiden Fällen übermittelt T die Antwort an A. Im positiven Fall befinden sich dann A und B in der sogenannten Datenphase, d.h. B ist nach Absenden eines CC bereit, Daten zu empfangen, und A kann nach Empfang eines CC Daten senden.

Alle drei Kooperationspartner A,B und T besitzen unabhängig voneinander jederzeit, d.h. sowohl in der Aufbau- als auch in der Datenphase die Entscheidungsfreiheit zu einem Abbau der Verbindung. Jeder Kooperationspartner besitzt also folgendes in Abb. 15.12 dargestelltes Verhaltensschema:

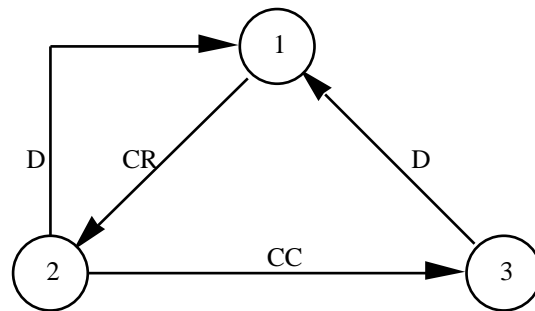


Abb. 15.12

Hierbei besteht allerdings ein grundsätzlicher Unterschied zwischen den Aktionen CR und CC einerseits und D andererseits. Bezüglich CR und CC steht nämlich für jede Schnittstelle und jeden Kooperationspartner fest, ob es sich um eine Sende- oder um eine Empfangsaktion handelt. Die Aktion D hingegen kann sowohl Sende- als auch Empfangsaktion sein, da ein "Disconnect" an jeder Schnittstelle von beiden Seiten ausgelöst werden kann. Sie wird deshalb zu einem sogenannten symmetrischen Handshake verfeinert.

Die Verhaltensweisen von A, B und T müssen jetzt so aufeinander abgestimmt werden, daß Verbindungen "konsistent" auf- und abgebaut werden. Es muß also garantiert werden, daß sich nicht ein Partner noch in einer "alten" Datenphase befindet, während der andere schon in einer "neuen" Datenphase ist. Dies wird mit dem folgenden Produktnetz formal spezifiziert. Zur übersichtlichen Strukturierung ist es aus 7 Teilnetzen (Abb. 15.13 - 15.19) zusammengesetzt. Die einzelnen Teilnetze sind dabei über gemeinsame Stellen miteinander "verklebt".

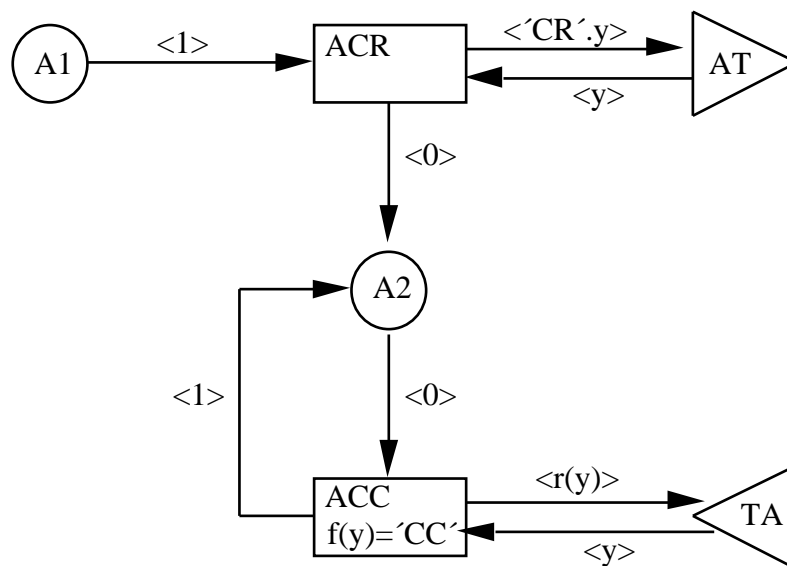


Abb. 15.13

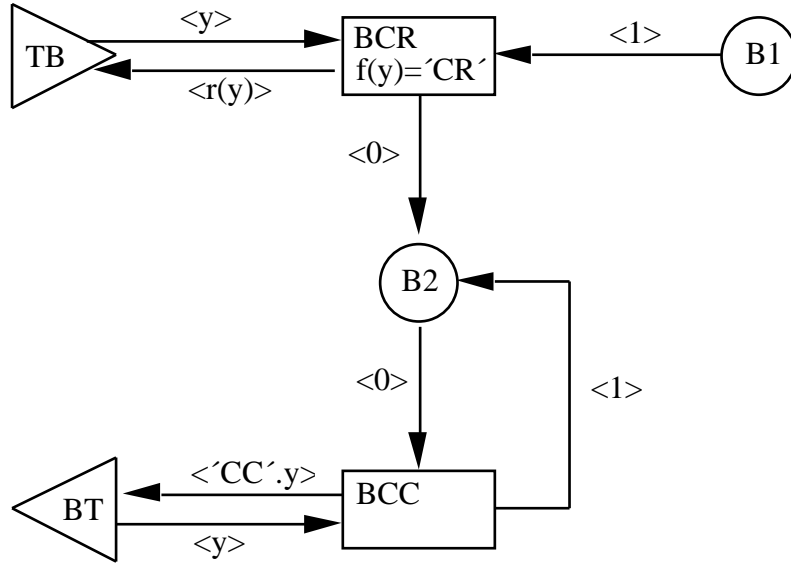


Abb. 15.14

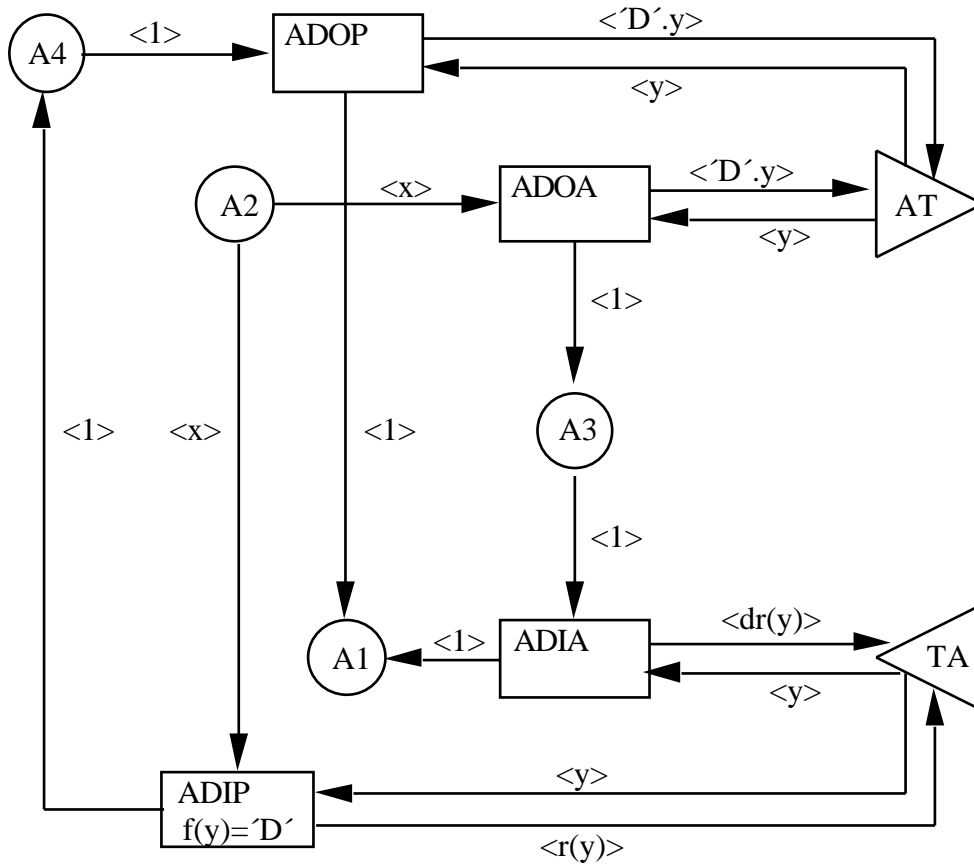


Abb. 15.15

Abb. 15.16

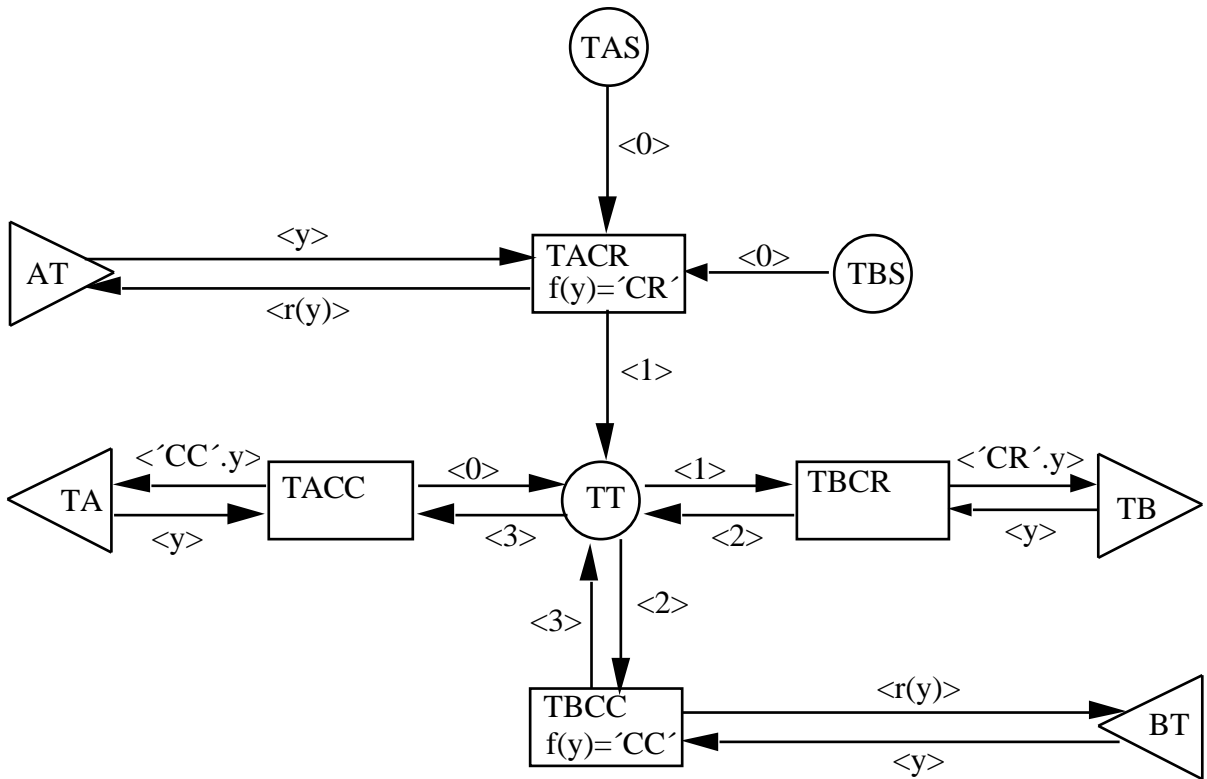
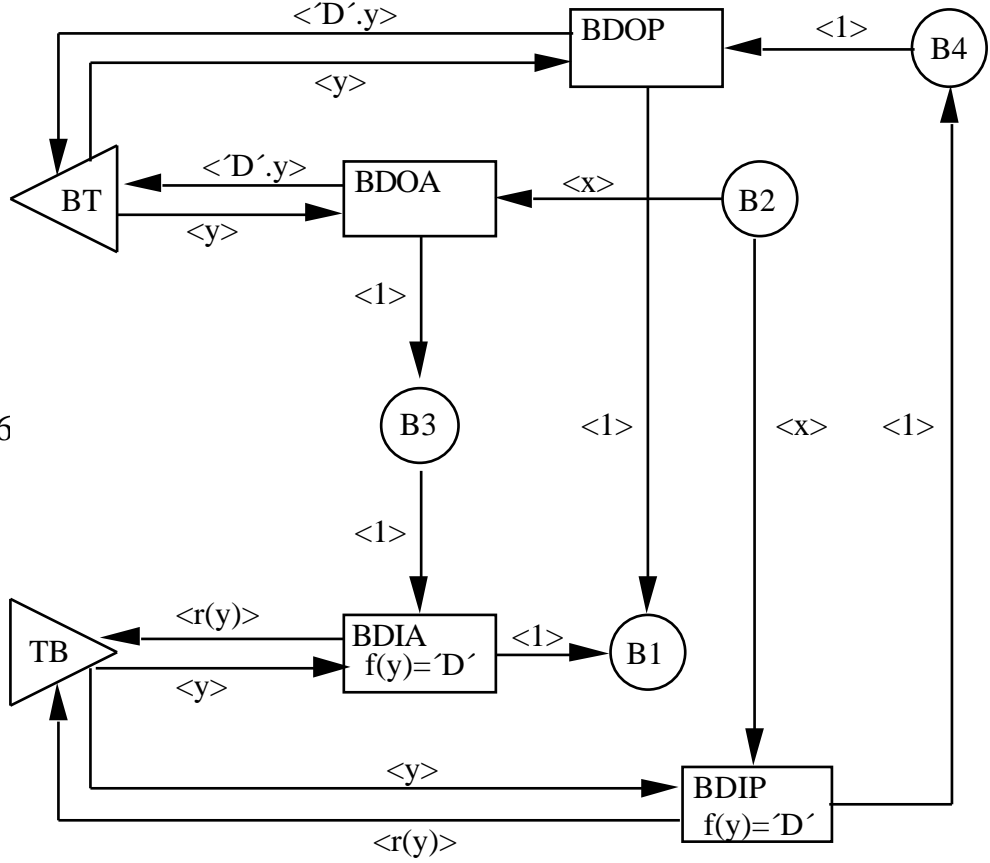


Abb. 15.17

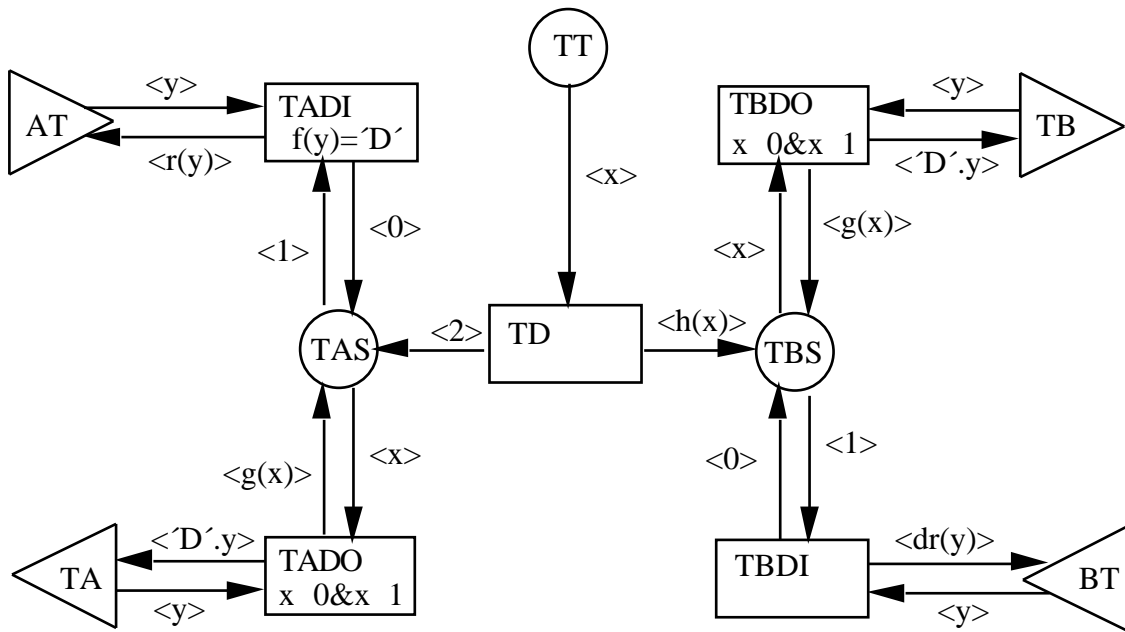


Abb. 15.18

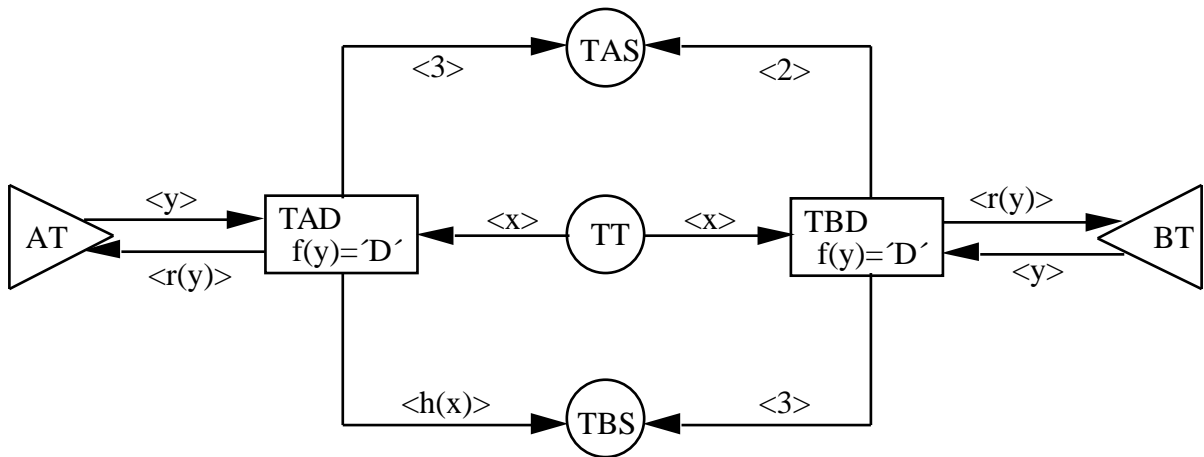


Abb. 15.19

Die Anfangsmarkierung dieses Produktnetzes besteht aus je einer 1 als Marke auf den Stellen A1 und B1, je einer 0 auf TA und TB sowie aus je einer leeren Folge als Marke auf den Stellen AT, TA, BT und TB, welche die FIFO-Kanäle zwischen A und T bzw. zwischen B und T darstellen.

Die in den Kantenanschriften und Transitionsinschriften benutzten Funktionen f, r, dr, g und h sind folgendermaßen definiert:

$$M = \{ CR, CC, D \}$$

$$Q = M^*$$

$$Q1 = \{ x \in Q \mid x \neq \epsilon \}$$

$$Q2 = \{ x \in Q \mid \text{es existieren } y \in Q \text{ und } z \in M \text{ mit } x = yD \text{ oder } x = yDz \}$$

$$f: Q1 \rightarrow M \text{ mit } f(x) = z \text{ für } x = yz \text{ und } y \in Q \text{ sowie } z \in M$$

$$r: Q1 \rightarrow Q \text{ mit } r(x) = y \text{ für } x = yz \text{ und } y \in Q \text{ sowie } z \in M$$

$$dr: Q2 \rightarrow Q \text{ mit } dr(x) = y \text{ für } x = yD \text{ oder } x = yDz \text{ und } y \in Q \text{ sowie } z \in M \setminus \{D\}$$

$$g: \text{NAT}_0 \rightarrow \text{NAT}_0 \text{ mit } g(x) = 0 \text{ für } x \geq 2 \text{ und } g(2) = 1$$

$$h: \text{NAT}_0 \rightarrow \text{NAT}_0 \text{ mit } h(x) = 2 \text{ für } x \geq 1 \text{ und } h(1) = 0$$

Die so gegebenen Definitionen entsprechen noch nicht dem Vorspannkalkül, sie lassen sich aber dahingehend transformieren.

Die Elemente der Menge $Q1$ beschreiben die nicht leeren Inhalte der FIFO-Kanäle. Dabei bestimmt die Funktion f die erste ankommende Nachricht und r den Rest. Die Menge $Q2$ besteht aus den Inhalten der FIFO-Kanäle, deren erste oder zweite ankommende Nachricht ein D ist. Die Funktion dr bestimmt den Rest, der sich nach diesem D im FIFO-Kanal befindet.

Die Stellen AT , TA , BT und TB haben den Definitionsbereich Q , für alle übrigen Stellen ist der Definitionsbereich gleich NAT_0 .

Das Teilnetz in Abb. 15.13 beschreibt ausgehend vom Grundzustand ($\langle 1 \rangle$ auf $A1$) den Verbindungsaufbau in A bis zum Erreichen der Datenphase ($\langle 1 \rangle$ auf $A2$). In Abb. 15.15 ist der symmetrische Disconnect-Handshake von A spezifiziert. Mit irgend einer Marke auf $A2$, d.h. sowohl in der Aufbauphase als auch in der Datenphase, kann A ein D abschicken ($ADOA$) oder empfangen ($ADIP$). Nach Beendigung des Handshakes ($ADIA$ bzw. $ADOP$) erreicht A wieder den Grundzustand. Abb. 15.14 und Abb. 15.16 beschreiben die entsprechende Funktionalität bei B .

Abb. 15.17 stellt die Aktionen von T dar, welche zum Verbindungsaufbau notwendig sind. Der Grundzustand von T ist dabei durch je eine $\langle 0 \rangle$ auf den Stellen TAS und TBS und die Datentransportphase durch eine $\langle 0 \rangle$ auf TT beschrieben. Es sei hier bemerkt, daß der eigentliche Datentransport in diesem Modell nicht explizit spezifiziert ist.

In Abb. 15.18 - 15.19 ist die Disconnect-Phase von T beschrieben. Sie wird aktiv durch TD oder passiv durch TAD bzw. TBD initiiert. Die Vervollständigung der Disconnect-Handshakes mit A und B bis zum Erreichen des Grundzustandes sind in Abb. 15.18 dargestellt.

Diese Spezifikation wurde mit der Produktnetzmaschine entworfen und analysiert. Dabei wurde ein Erreichbarkeitsgraph mit 370 Markierungen und 1083 Schaltschritten berechnet. Ausgehend von diesem Erreichbarkeitsgraphen können jetzt spezielle Eigenschaften der Dynamik des Produktnetzes untersucht werden. Weiter oben wurde schon bemerkt, daß es wichtig ist, etwa zur konsistenten Zuordnung von Betriebsmitteln oder zur Beibehaltung eines "gemeinsamen Kontexts", daß aufeinanderfolgende Datenphasen in A und B nicht nur "lokal", sondern auch "global" voneinander getrennt sind (Phasentrennung).

Beachtet man, daß A bzw. B jeweils durch ACC bzw. BCC in eine neue Datenphase eintritt und diese durch ADOA oder ADIP bzw. BDOA oder BDIP verläßt, falls auf der Stelle A₂ bzw. B₂ die Marke <1> liegt, dann kann diese Eigenschaft mit dem folgenden Schaltfolgenhomomorphismus $s : * \rightarrow *$ ($\Sigma = \{AC, AD, BC, BD\}$) untersucht werden :

$$\begin{aligned}
 s((M, ACC, M')) &= AC, \quad s((M, BCC, M')) = BC, \\
 s((M, ADOA, M')) &= AD = s((M, ADIP, M')) \text{ falls } M_{A_2} = \langle 1 \rangle, \\
 s((M, BDOA, M')) &= BD = s((M, BDIP, M')) \text{ falls } M_{B_2} = \langle 1 \rangle \text{ und} \\
 s((M, X, M')) &= \text{ falls } X \in \{ACC, ADOA, ADIP, BCC, BDOA, BDIP\} \text{ oder} \\
 X &\in \{ADOA, ADIP\} \text{ und } M_{A_2} = \langle 1 \rangle \text{ oder} \\
 X &\in \{BDOA, BDIP\} \text{ und } M_{B_2} = \langle 1 \rangle.
 \end{aligned}$$

Für das Bild $L' = s(L)$ der Schaltfolgensprache L unter diesem Homomorphismus kann mit der Produktnetzmaschine der Minimalautomat in Abb. 15.20 berechnet werden. Dabei ist 1 der Anfangszustand, und alle Zustände sind Endzustände.

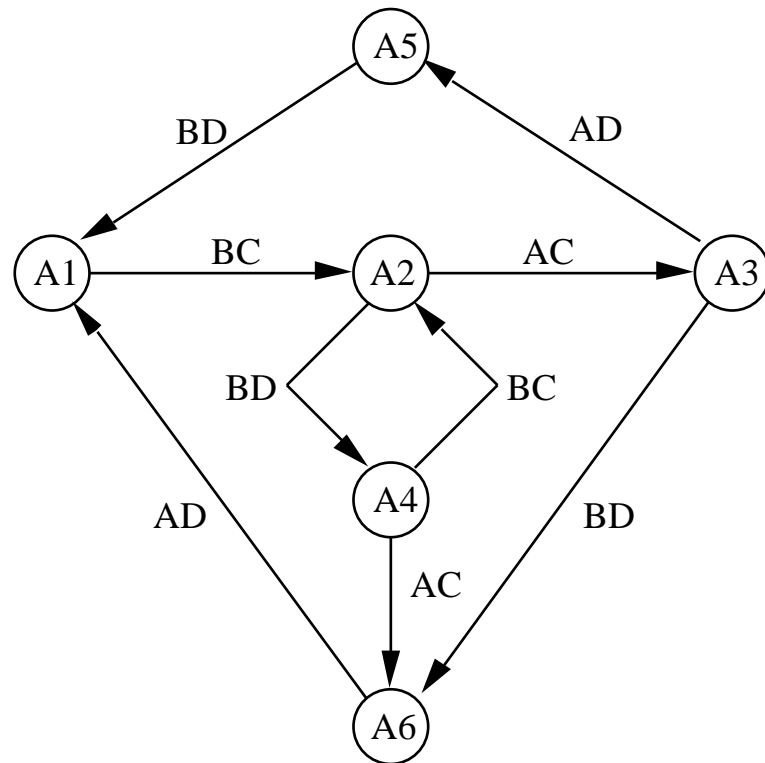


Abb. 15.20

An diesem Automaten kann leicht $\{y \in (\{AD\}^* \mid \text{es existiert } x \in \{A\}^* \text{ mit } xACy \in L') = \{\epsilon, BD\}$ sowie $\{y \in (\{BD\}^* \mid \text{es existiert } x \in \{A\}^* \text{ mit } xBCy \in L') = \{\epsilon, AC, ACAD\}$ nachgewiesen werden.

Aus der ersten Gleichung folgt, daß B in keine Datenphase eintreten kann, solange sich A in einer Datenphase befindet; B kann allerdings schon zuvor in eine Datenphase eingetreten sein. Insgesamt bedeutet dies, daß B in keine "zweite" Datenphase eintreten kann, während sich A in einer Datenphase befindet. Aus der zweiten Gleichung folgt dann noch, daß A höchstens eine Datenphase durchlaufen kann, während sich B in einer Datenphase befindet. Beide Eigenschaften zusammen bedeuten gerade die "globale" Trennung aufeinanderfolgender Datenphasen in A und B (Verifikation der Phasentrennung).

Es sind Eigenschaften, die sich auf "abgelaufene" Schaltfolgen beziehen, also Sicherheitseigenschaften. Solche können aus entsprechenden Eigenschaften der Bildsprache L' für Schaltfolgen aus L gefolgert werden. Sie gelten damit auch für andere Schaltfolgensprachen, die unter einem entsprechenden Homomorphismus auf die gleiche Bildsprache L' abgebildet werden.

Betrachten wir dazu eine "Verfeinerung" unseres Protokolls. Wie oben beschrieben wurde, kann das Transportsystem T schon in der Aufbauphase einen Disconnect initiieren. Die Gründe dazu wurden nicht spezifiziert. Ein Grund kann z.B. darin bestehen, daß T, wenn es sich im

Grundzustand befindet, für eine andere Aufgabe reserviert werden kann und erst nach Beendigung dieser Aufgabe wieder für eine Verbindung zwischen A und B zur Verfügung steht. Abb. 15.21 beschreibt den Teil dieses Vorgangs, der für Arelevant ist.

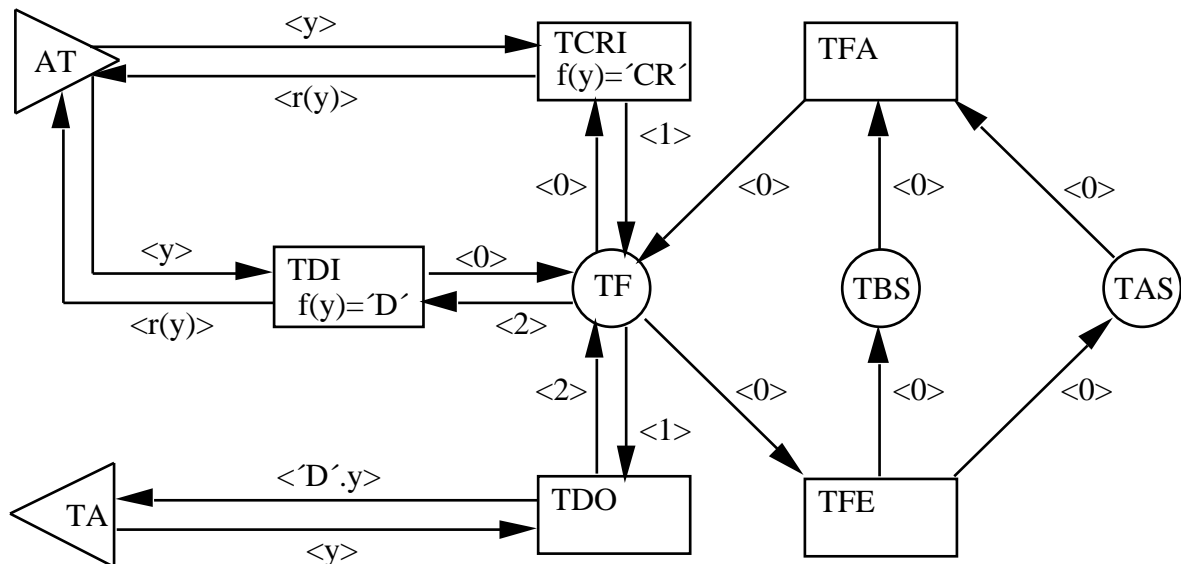


Abb. 15.21

Nach dem Schalten von TFA kann T ein ankommendes CR nur mit einem Disconnect-Handshake beantworten. Diese eingeschränkte Verhaltensweise von T wird durch TFE wieder beendet.

Das um Abb. 15.21 erweiterte Produktnetz besitzt einen Erreichbarkeitsgraphen mit 396 Markierungen und 1154 Schaltschritten. Definiert man auch dafür den Schaltfolgenhomomorphismus s wie oben, dann entsteht die gleiche Bildsprache L' . Damit ist auch in dieser Spezifikation eine "globale" Trennung aufeinanderfolgender Datenphasen gewährleistet.

Werden bei dieser "Verfeinerung" durch einen Fehler die Transition TFE und ihre benachbarten Kanten vergessen, dann berechnet die Produktnetzmaschine für den Homomorphismus s den gleichen Minimalautomaten wie oben. Das bedeutet, daß dieser Spezifikationsfehler keine Auswirkungen auf die Sicherheitseigenschaften bezüglich s hat. Dies ist auch unmittelbar einleuchtend, da das Fehlen von TFE lediglich bewirkt, daß eine einmal gewählte eingeschränkte Verhaltensweise von T nie mehr verlassen wird, aber keine "fehlerhaften" Verhaltensweisen erzeugt. Ein wesentlicher Unterschied zur korrekten Spezifikation wird allerdings durch eine weitere Eigenschaft der Bildsprache L' aufgedeckt.

Am Minimalautomaten in Abb. 15.20 kann leicht nachgewiesen werden, daß zu jedem $x \in L'$

ein $y \in L^*$ mit $x \in BCAC \cap L'$ existiert. Das ist eine Lebendigkeitseigenschaft der Bildsprache L' . Im Gegensatz zu den oben betrachteten Sicherheitseigenschaften kann hier nicht ohne weiteres auf eine entsprechende Eigenschaft der Schaltfolgensprache geschlossen werden, wie schon zu Beginn des Kapitels über schlichte Homomorphismen erläutert wurde. Falls allerdings der Homomorphismus s schlicht ist, dann sind Rückschlüsse auf die Schaltfolgensprache möglich (Verifikation von Lebendigkeitseigenschaften).

Sowohl der Erreichbarkeitsgraph des ursprünglichen als auch der des um Abb. 15.21 erweiterten Produktnetzes sind stark zusammenhängend. Nach Satz 6 ist dann in beiden Fällen der Homomorphismus s schlicht auf L , da L' unendlich ist.

Damit gibt es zu jedem $u \in L$ ein $v \in s(u)^{-1}(L')$ mit $v^{-1}(s(u^{-1}(L))) = v^{-1}(s(u)^{-1}(L')) = (s(u)v)^{-1}(L')$. Mit $x = s(u)v$ gibt es dann wegen der betrachteten Lebendigkeitseigenschaft von L' ein $y \in L^*$ mit $y \in BCAC \cap (s(u)v)^{-1}(L') = v^{-1}(s(u^{-1}(L)))$.

Daraus folgt $v \in BCAC \cap s(u^{-1}(L))$. Damit gibt es ein $w \in u^{-1}(L)$ mit $s(w) = v \in BCAC$. Es existieren also zu jedem $u \in L$ ein $w' \in L^*$ und ein $w'' \in L^*$ mit $uw'(N, BCC, N')w''(M, ACC, M') \in L$ und $s(w') = s(w'') = \cdot$. Das bedeutet aber, daß immer wieder, egal was geschehen ist, eine Situation erreichbar ist, in der sich sowohl B als auch A in einer Datenphase befinden.

Für die fehlerhafte Erweiterung des ursprünglichen Produktnetzes kann diese Schlußfolgerung nicht gezogen werden. In diesem Fall ist der Erreichbarkeitsgraph nicht stark zusammenhängend. Er enthält vielmehr u.a. eine starke Zusammenhangskomponente, von der aus keine weitere starke Zusammenhangskomponente erreichbar ist, und die keine Kante enthält, deren Beschriftung auf AC abgebildet wird.

Dieser Defekt läßt sich gut durch den Homomorphismus $t: L^* \rightarrow L^*$ beschreiben, wobei $L'' = \{AC\}$ und $t((M, ACC, M')) = AC$ sowie $t((M, X, M')) = \cdot$ falls $X \neq ACC$. Mit der Produktnetzmaschine kann für t der Deadlockautomat in Abb. 15.22 berechnet werden. Dabei ist 1 Anfangs- und Endzustand.

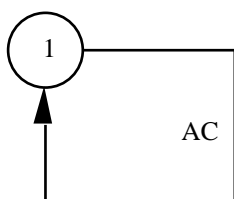


Abb. 15.22

Daraus folgt, daß es Schaltfolgen gibt, die keine Fortsetzung der Form $w(M, ACC, M')$ besitzen, d.h. es können Situationen auftreten, nach denen A nie mehr in eine Datenphase eintreten kann.

Der Automat in Abb. 15.22 ist auch der Minimalautomat der Bildsprache $t(L)$. Damit ist die Terminierungssprache bezüglich t leer und somit ungleich der Deadlocksprache. Wegen Satz 1 ist dann t nicht schlicht auf L . Ist der Homomorphismus $t' : \Sigma^* \rightarrow \Sigma'^*$ definiert durch $t'(AC) = AC$ und $t'(X) = \epsilon$ falls $X \in ACC$, dann gilt $t = t' \circ s$. Damit kann in diesem Fall wegen Satz 2 s nicht schlicht auf L sein, denn t' ist wegen Satz 6 schlicht auf L' , da der Minimalautomat von L' stark zusammenhängend und $t'(L') = t(L)$ unendlich ist.

Die soweit angestellten Lebendigkeitsüberlegungen haben zwar für das ursprüngliche sowie für das um Abb. 15.21 erweiterte Produktnetz gezeigt, daß immer wieder eine Situation erreichbar ist, in der sich sowohl B als auch A in einer Datenphase befinden, lassen aber nicht den Schluß zu, daß dann eine Verbindung vollständig aufgebaut ist, da der Homomorphismus s die Aktionen des Transportsystems T vollständig ignoriert. T kann nämlich mittels TD jederzeit, also auch dann, wenn es gerade ein CC an A abgeschickt hat, in eine Disconnectphase eintreten. Es kann also die oben beschriebene Situation erreicht sein, ohne daß eine Verbindung vollständig aufgebaut ist.

Diese Situation kann auf mehrere Arten untersucht werden. Eine besteht darin, den Homomorphismus s dahingehend zu verfeinern, daß die Schaltschritte in denen TD auftritt auch noch berücksichtigt, anstatt auf ϵ abgebildet zu werden. Bei diesem Vorgehen entsteht ein größerer Minimalautomat, bei dem in der Tat die gewünschten Eigenschaften nachgewiesen werden können.

Eine einfachere Möglichkeit liegt in der Untersuchung eines Homomorphismus s° , der die Markierungen der Stellen TT und B2 berücksichtigt, da diese anzeigen, ob sich T bzw. B in einer Datenphase befinden ($\langle 0 \rangle$ auf TT bzw. $\langle 1 \rangle$ auf B2).

Sei $s^\circ : \Sigma^* \rightarrow \Sigma'^*$ mit $\Sigma'^\circ = M_{TT} \times M_{B2}$ und $s^\circ((M, ACC, M')) = (M'_{TT}, M'_{B2})$ sowie $s^\circ((M, X, M')) = \epsilon$ falls $X \in ACC$.

Abb.15.23 stellt den Minimalautomaten der Bildsprache $s^\circ(L)$ dar, und zwar sowohl für das ursprüngliche als auch für das erweiterte Produktnetz. \emptyset stellt in den Kantenbeschriftungen die leere Markierung dar.

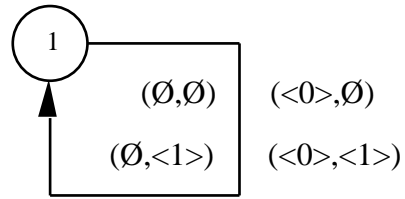


Abb. 15.23

Für jedes $x \in s^\circ(L)$ gilt damit $x(\langle 0 \rangle, \langle 1 \rangle) \in s^\circ(L)$. Da wegen Satz 6 s° schlicht auf List, folgt daraus, wie bei den Lebendigkeitsüberlegungen bzgl. s , daß immer wieder eine Situation erreichbar ist, in der A in eine Datenphase eintritt, während sich T und B schon in einer Datenphase befinden, daß also immer wieder eine Verbindung vollständig aufgebaut werden kann.

Reduzierte Erreichbarkeitsgraphen

Für spezielle Schaltfolgenhomomorphismen, nämlich Projektionen und Modulhomomorphismen, wurden in /14, 15,17/ sogenannte reduzierte Erreichbarkeitsgraphen definiert. Diese Graphen vernachlässigen gewisse Schaltschritte, welche bezüglich der entsprechenden Homomorphismen irrelevant sind, und enthalten trotzdem genügend Information zur Bestimmung der Projektions- bzw. Modulsprache. Im vorliegenden Kapitel wird gezeigt, daß an Hand dieser reduzierten Erreichbarkeitsgraphen auch die Schlichtheiteiner Projektion bzw. eines Modulhomomorphismus festgestellt werden kann.

Im folgenden bezeichne M_1 jeweils die Anfangsmarkierung des betrachteten Netzes.

Bezüglich der Definition einer Projektion sowie der automatentheoretischen Interpretation eines Erreichbarkeitsgraphen sei auf das erste Kapitel verwiesen. Die Definition des reduzierten Erreichbarkeitsgraphen $\mathbb{R}\mathbb{G}$ sowie die der Hilfsgraphen $\mathbb{G}(M,A)$ findet sich in /14/. Dort wurde der reduzierte Erreichbarkeitsgraph so definiert, daß er nicht nur für Projektionen, sondern auch noch für eine Klasse von feineren Homomorphismen, den sogenannten ET-PT-Homomorphismen benutzt werden kann. Zur Untersuchung der Schlichtheit von Projektionen genügt eine vereinfachte Form des reduzierten Erreichbarkeitsgraphen, wobei die Kanten nicht wie in /14/ mit Elementen aus Σ^* , sondern mit den entsprechenden p-Bildern aus Σ^p beschriftet sind. Die Menge der Knoten von $\mathbb{R}\mathbb{G}$ sei mit $\mathbb{M}\mathbb{R}$ bezeichnet. Sie ist eine Teilmenge der Menge \mathbb{M} aller erreichbaren Markierungen.

$\mathbb{R}\mathbb{G}$ kann als nichtdeterministischer Automat mit einer Zustandsüberföhrungsfunktion

$$\delta^* : \mathbb{M}\mathbb{R} \times \Sigma^* \rightarrow 2^{\mathbb{M}\mathbb{R}} \text{ betrachtet werden.}$$

Zur Formulierung der nächsten Definition werden noch einige Bezeichnungen benötigt:

Für ein $w \in \Sigma^*$ bezeichne $\mathbb{B}(w)$ die Menge aller Blöcke, die in w "vorkommen", also

$$\mathbb{B}(w) = \{X \in \mathbb{P}\mathbb{T} \mid \text{es existieren } u, v \in \Sigma^* \text{ sowie } M, M' \in \mathbb{M} \text{ mit } w = u(M|\mathbb{S}P, X, M'|\mathbb{S}P)v\}.$$

Für $L' \in \Sigma^*$ bezeichne $\mathbb{B}(L') = \{X \in \mathbb{P}\mathbb{T} \mid \text{es existiert } w \in L' \text{ mit } X \in \mathbb{B}(w)\}.$

Für $M \in \mathbb{M}\mathbb{R}$ bezeichne $\mathbb{R}M = \{w \in \Sigma^* \mid \delta^*(M, w) \neq \emptyset\}.$

Im Hilfsgraphen $\mathbb{G}(M,A)$ bezeichne $\mathbb{D}(M,A)$ die Menge aller Knoten der Form

$(M', 0) \in (M, 0)$, von denen aus kein Knoten der Form $(M'', 1)$ erreichbar ist.

$\mathbb{R}(M,A)$ bezeichne die Menge der Knoten von $\mathbb{G}(M,A)$, die nicht in $\mathbb{D}(M,A)$ liegen und ungleich $(M, 0)$ sind.

Ein Knoten $M \in \mathbb{M}\mathbb{R}$ heißt abgeschlossen in $\mathbb{R}\mathbb{G}$, wenn für jede nichtleere Teilmenge

$Z \subseteq \mathbb{P}\mathbb{T}$ eine der folgenden zwei Eigenschaften gilt :

- (1) Es gibt ein $M^\circ \in \mathcal{MR}$ und ein $v \in \mathcal{V}^*$ mit $M^\circ \in \mathcal{M}(M, v)$ und $Z \in \mathcal{B}(v) = \mathcal{B}(RM^\circ)$.
- (2) Zu jeder Familie von Knoten $((M_X, 0))_{X \in Z}$ mit $(M_X, 0) \in \mathcal{D}(M, X)$ für jedes $X \in Z$ gibt es einen Block $A \in Z$, eine Markierung $M^* \in \mathcal{MR}$, ein $u \in \mathcal{V}^*$ mit $M^* \in \mathcal{M}(M, u)$ und $Z \in \mathcal{B}(u) = \mathcal{B}(RM^*)$ sowie ein $(M', 0) \in \mathcal{R}(M^*, A)$ mit $M' \upharpoonright N(A \in \mathcal{ET}) = M_A \upharpoonright N(A \in \mathcal{ET})$.

Satz 10: Eine Projektion p ist genau dann schlicht auf L , wenn \mathcal{RG} schlicht ist und wenn jeder Knoten $M \in \mathcal{MR}$ abgeschlossen in \mathcal{RG} ist.

Beweis:

Die beschrifteten Kanten von \mathcal{RG} lassen sich als Tripel $(M, a, M') \in \mathcal{MR} \times \mathcal{V} \times \mathcal{MR}$ darstellen. Die Menge aller Kanten sei mit $\mathcal{K} \subseteq \mathcal{MR} \times \mathcal{V} \times \mathcal{MR}$ bezeichnet. Die Pfade in \mathcal{RG} sind damit Elemente von \mathcal{K}^* . Die Menge aller Pfade, die von der Anfangsmarkierung ausgehen, sei mit $\mathcal{PF} \subseteq \mathcal{K}^*$ bezeichnet. Ist $\alpha: \mathcal{K}^* \rightarrow \mathcal{V}^*$ der Homomorphismus, der durch $\alpha((M, a, M')) = a$ für $(M, a, M') \in \mathcal{K}$ definiert ist, dann ist die Sprache, die von \mathcal{RG} erkannt wird, gleich $\mathcal{L}(\mathcal{PF})$. Wie in /14/ gezeigt wurde gilt damit $p(L) = \mathcal{L}(\mathcal{PF})$.

(1) Sei p schlicht auf L .

Es soll zuerst gezeigt werden, daß \mathcal{RG} schlicht ist, d.h. daß $\mathcal{L}(\mathcal{PF})$ schlicht auf L ist.

Wie im Beweis zu Satz 1 in /14/ läßt sich zeigen, daß es zu jedem $x \in \mathcal{PF}$ ein $y \in L$ gibt mit $\alpha(x) = p(y)$ und $\alpha(x^{-1}(\mathcal{PF})) = p(y^{-1}(L))$. Da p schlicht auf L ist, gibt es ein $u \in p(y)^{-1}(p(L)) = \alpha(x)^{-1}(\mathcal{L}(\mathcal{PF}))$ mit $u^{-1}(p(y^{-1}(L))) = u^{-1}(p(y)^{-1}(p(L)))$, woraus $u^{-1}(\alpha(x^{-1}(\mathcal{PF}))) = u^{-1}(\alpha(x)^{-1}(\mathcal{L}(\mathcal{PF})))$ folgt. Damit ist $\mathcal{L}(\mathcal{PF})$ schlicht auf L , also \mathcal{RG} schlicht.

Bleibt noch zu zeigen, daß jeder Knoten $M \in \mathcal{MR}$ abgeschlossen in \mathcal{RG} ist.

Sei $M \in \mathcal{MR}$, $Z = \{A_1, \dots, A_k\}$ eine nichtleere Teilmenge von \mathcal{PT} und $((M_X, 0))_{X \in Z}$ eine Familie von Knoten mit $(M_X, 0) \in \mathcal{D}(M, X)$ für jedes $X \in Z$.

Mit den Methoden aus den Beweisen der Sätze 1 und 2 aus /14/ läßt sich dann eine Schaltfolge $wv_1 \dots v_k \in L$ mit folgenden Eigenschaften konstruieren:

$$\mathcal{M}(M, w) = M \quad \mathcal{M}(M, p(w)),$$

$$p(v_1 \dots v_k) = \epsilon,$$

die Transitionen der Schaltschritte in v_i liegen alle in $A_i \in \mathcal{ET}$ für $1 \leq i \leq k$,

$$\mathcal{M}(M, wv_1 \dots v_i) \upharpoonright N(A_i \in \mathcal{ET}) = M_{A_i} \upharpoonright N(A_i \in \mathcal{ET}) \text{ für } 1 \leq i \leq k \text{ und}$$

$$\mathcal{M}(M, wv_1 \dots v_{i-1}) \upharpoonright N(A_i \in \mathcal{ET}) = M \upharpoonright N(A_i \in \mathcal{ET}) \text{ für } 1 < i \leq k.$$

Für die Menge der Fortsetzungen der Schaltfolge $wv_1 \dots v_k$ gilt entweder

$$Z \in \mathcal{B}(p((wv_1 \dots v_k)^{-1}(L))) \quad \text{oder} \quad Z \in \mathcal{B}(p((wv_1 \dots v_k)^{-1}(L))) = \emptyset.$$

Im Fall $Z \in \mathcal{B}(p((wv_1 \dots v_k)^{-1}(L)))$ gibt es ein $A \in Z$ und ein $y \in (wv_1 \dots v_k)^{-1}(L)$ minimaler Länge mit $Z \in \mathcal{B}(p(y)) = A$. Durch geeignetes "Umsortieren" (Hilfssatz 1 /14/) und

Verkürzen der Schaltfolge $wv_1 \dots v_k y$ L kann eine Schaltfolge $wrstq$ L konstruiert werden mit folgenden Eigenschaften :

$$Z \quad \mathbb{B}(p(r)) = \quad ,$$

$$*(M,r) \quad *(M,p(r)) ,$$

die Transitionen der Schaltschritte in st liegen alle in $A \quad \mathbb{E}T$,

$$*(M,rs)|N(A \quad \mathbb{E}T) = M_A|N(A \quad \mathbb{E}T) \text{ und}$$

$$q \quad \text{mit } \mathbb{B}(p(q)) = A.$$

Daraus folgt $(*(M,rs),0) \quad \mathbb{R}(*(M,r),A)$. Mit $M^* = *(M,r)$, $M' = *(M,rs)$ und $u = p(r)$ ist dann die Eigenschaft (2) der Abgeschlossenheitsbedingung für M erfüllt.

Im Fall $Z \quad \mathbb{B}(p((wv_1 \dots v_k)^{-1}(L))) = \quad$ folgt aus der Schlichtheit von p auf L in $wv_1 \dots v_k$ die Existenz eines $v \quad p((wv_1 \dots v_k)^{-1}(L))$ mit

$$v^{-1}(p((wv_1 \dots v_k)^{-1}(L))) = v^{-1}(p(wv_1 \dots v_k)^{-1}(p(L))) .$$

Gilt $*(M1,y) \quad \mathbb{M}R$ für ein $y \quad L$, dann folgt aus dem Beweis des Satzes 1 aus /14/ $p(y^{-1}(L)) = \mathbb{R} \quad *(M1,y)$ ($*(M1,y)$ als Anfangsmarkierung betrachten). Da die Voraussetzung

für das betrachtete w erfüllt ist, gilt also $\mathbb{R}M = p(w^{-1}(L))$. Wegen $p(v_1 \dots v_k) = \quad$ gilt $p(w^{-1}(L)) \quad p(wv_1 \dots v_k)^{-1}(L)$ und damit $v \quad \mathbb{R}M$. Daraus folgt die Existenz eines

$$M^\circ \quad \mathbb{M}R \text{ mit } M^\circ \quad *(M,v) \text{ und } \mathbb{R}M^\circ \quad v^{-1}(\mathbb{R}M) = v^{-1}(p(w^{-1}(L))) \quad v^{-1}(p(w)^{-1}(p(L))) =$$

$$= v^{-1}(p(wv_1 \dots v_k)^{-1}(p(L))) = v^{-1}(p((wv_1 \dots v_k)^{-1}(L))) . \text{ Wegen } Z \quad \mathbb{B}(p((wv_1 \dots v_k)^{-1}(L))) =$$

folgt dann $Z \quad \mathbb{B}(v) = \quad = Z \quad \mathbb{B}(\mathbb{R}M^\circ)$, und damit ist die Eigenschaft (1) der Abgeschlossenheitsbedingung für M erfüllt.

(2) Sei $\mathbb{R}G$ schlicht und jeder Knoten $M \quad \mathbb{M}R$ abgeschlossen in $\mathbb{R}G$.

Für jedes $w \quad L$ ist jetzt zu zeigen, daß p schlicht auf L in w ist.

$$\text{Fall 2.1 : } *(M1,w) \quad *(M1,p(w))$$

In diesem Fall ist $*(M1,w) \quad \mathbb{M}R$ und deshalb, wie schon im ersten Teil des Beweises bemerkt wurde, $p(w^{-1}(L)) = \mathbb{R} \quad *(M1,w)$. Außerdem gibt es in $\mathbb{R}G$ einen Pfad $x \quad \mathbb{P}F$ mit

$$(x) = p(w) \text{ und } (x^{-1}(\mathbb{P}F)) = \mathbb{R} \quad *(M1,w) . \text{ Da } \text{schlicht auf } \mathbb{P}F \text{ in } x \text{ ist, gibt es ein}$$

$$u \quad (x^{-1}(\mathbb{P}F)) \text{ mit } u^{-1}((x^{-1}(\mathbb{P}F))) = u^{-1}((x)^{-1}(\mathbb{P}F)) . \text{ Wegen } p(L) = (\mathbb{P}F) \text{ folgt daraus}$$

$$u \quad p(w^{-1}(L)) \text{ und } u^{-1}(p(w^{-1}(L))) = u^{-1}(p(w)^{-1}(p(L))) , \text{ also die Schlichtheit von } p \text{ auf } L \text{ in } w.$$

$$\text{Fall 2.2 : } *(M1,w) \quad *(M1,p(w))$$

Ist p schlicht auf L in einem $x \quad L$ und gibt es ein $y \quad L$ mit $p(y) = p(x)$ und

$$*(M1,y) = *(M1,x) , \text{ dann ist } p \text{ auch schlicht auf } L \text{ in } y, \text{ denn aus}$$

$$u^{-1}(p(x^{-1}(L))) = u^{-1}(p(x)^{-1}(p(L))) \text{ folgt dann } u^{-1}(p(y^{-1}(L))) = u^{-1}(p(y)^{-1}(p(L))) , \text{ da mit}$$

$$*(M1,y) = *(M1,x) \text{ auch } y^{-1}(L) = x^{-1}(L) \text{ gilt.}$$

Die Idee des Beweises besteht nun darin, ausgehend von w eine endliche Folge (w_1, \dots, w_m)

mit folgenden Eigenschaften zu konstruieren:

$w_1 = w$,
 $w_i \in L$ für $1 \leq i \leq m$ und
 für $1 \leq i \leq m-1$ gibt es $x_i \in A_i$ mit $p(w_i x_i) = p(w_{i+1})$ und
 $*(M1, w_i x_i) = *(M1, w_{i+1})$.

Aus der obigen Bemerkung zusammen mit Hilfssatz 1 folgt dann aus der Schlichtheit von p auf L in w_m die Schlichtheit von p auf L in w .

Durch geeignetes "Umsortieren" von w entsteht im ersten Schritt die Schaltfolge w_2 mit den Eigenschaften :

$p(w_2) = p(w_1)$,
 $*(M1, w_2) = *(M1, w_1)$ und
 $w_2 = y_2 z_2$ mit
 $*(M1, y_2) = *(M1, p(y_2))$,
 $z_2 = z_{2,1} \dots z_{2,q}$,
 $p(z_2) = p(y_2)$,
 die Transitionen der Schaltschritte in $z_{2,i}$ liegen alle in einem Block $A_{2,i} \in \mathcal{PT}$,
 für $i < j$ ist $A_{2,i} \cap A_{2,j} = \emptyset$,
 $*(M1, y_2 z_{2,1}) = *(M1, y_2)$ und
 $*(M1, y_2 z_{2,1} \dots z_{2,i}) = *(M1, y_2 z_{2,1} \dots z_{2,i+1})$.

Diesen $z_{2,i}$ "entsprechen" Schaltfolgen $z'_{2,i}$ in den Hilfsgraphen $\mathbb{G}(*(M1, y_2), A_{2,i})$, die zu Markierungen $M_{2,i} \in \mathbb{R}(*(M1, y_2), 0)$ führen.

Falls es ein i gibt mit $M_{2,i} \in \mathbb{R}(*(M1, y_2), A_{2,i})$, dann besitzt $z'_{2,i}$ eine minimale Fortsetzung $r'_{2,i}$ mit $\mathbb{B}(p(z'_{2,i} r'_{2,i})) = \{A_{2,i}\}$. Durch geeignetes "Zusammensetzen" (Hilfssatz 2 in /14/) kann daraus eine Schaltfolge w_3 konstruiert werden, die wie w_2 aufgebaut ist, bei der aber z_3 nur noch $q-1$ Faktoren $z_{3,i}$ besitzt ("Vorziehen" von $z'_{2,i} r'_{2,i}$). Dieses Verfahren wird bis zu einem w_n fortgesetzt, für welches $z_n = z_{n,1} \dots z_{n,q}$ oder für welches es kein i gibt mit $M_{n,i} \in \mathbb{R}(*(M1, y_n), A_{n,i})$.

Im Fall $z_n = z_{n,1} \dots z_{n,q}$ ist mit $w_m = w_n$ und dem Fall 2.1 der Beweis abgeschlossen.

Im Fall $z_n = z_{n,1} \dots z_{n,q}$ liegen alle $M_{n,i}$ in den entsprechenden Mengen $\mathbb{D}(*(M1, y_n), A_{n,i})$. Da der Knoten $*(M1, y_n) \in \mathbb{MR}$ abgeschlossen ist, besitzt er die Eigenschaft (1) oder (2) der Abgeschlossenheitsbedingung.

Falls (1) für $Z = \{A_{n,1}, \dots, A_{n,1-n+2}\}$ nicht gilt, dann besitzt y_n gemäß (2) eine Fortsetzung r_n mit $*(M1, y_n r_n) = *(M1, p(y_n r_n))$. Den $z_{n,i}$ "entsprechen" jetzt Schaltfolgen $z'_{n,i}$ in den Hilfsgraphen $\mathbb{G}(*(M1, y_n r_n), A_{n,i})$, die zu Markierungen $M_{n,i} \in \mathbb{D}(*(M1, y_n r_n), 0)$ führen. Außerdem gibt es für ein $z'_{n,i}$ eine minimale Fortsetzung $r'_{n,i}$ mit $\mathbb{B}(p(z'_{n,i} r'_{n,i})) = A_{n,i}$. Durch geeignetes "Zusammensetzen" kann daraus die Schaltfolge w_{n+1} konstruiert werden, die

wie w_n aufgebaut ist, aber mit einem Faktor $z_{n+1,i}$ weniger. Dieses Verfahren wird bis zu einem w_m fortgesetzt, für welches $z_m = \dots$ oder für welches die Abgeschlossenheitseigenschaft (1) für den Knoten $(M1, y_m)$ bezüglich $Z = \{A_{m,1}, \dots, A_{m,l-m+2}\}$ gilt.

Im Fall $z_m = \dots$ ist wegen des Falls 2.1 der Beweis abgeschlossen.

Im Fall $z_m = \dots$ besitzt y_m wegen der Abgeschlossenheitseigenschaft (1) eine Fortsetzung r mit $(M1, y_m r) \xrightarrow{*} (M1, p(y_m r))$ und $Z \cap \mathbb{B}(p(r)) = \dots = Z \cap \mathbb{B}(R \cdot (M1, y_m r))$. Damit besitzt $w_m = y_m z_m$ eine Fortsetzung r' mit $p(r') = p(r)$ und $p((w_m r')^{-1}(L)) = p((y_m r)^{-1}(L))$. Wegen Fall 2.1 ist p schlicht auf L in $y_m r$. Deshalb gibt es ein $u \in p(y_m r)^{-1}(p(L)) = p(w_m r')^{-1}(p(L))$ mit $u^{-1}(p((w_m r')^{-1}(L))) = u^{-1}(p((y_m r)^{-1}(L))) = u^{-1}(p(y_m r)^{-1}(p(L))) = u^{-1}(p(w_m r')^{-1}(p(L)))$. Also ist p schlicht auf L in $w_m r'$ und damit wegen Hilfssatz 1 auch schlicht in w_m .

Damit ist Satz 10 vollständig bewiesen.

Während mittels der Projektionen das "Randverhalten" von Teilnetzen formal gefaßt wird, beschreiben die Modulhomomorphismen das interne Verhalten von Teilnetzen (Modulen).

Sei $\mathbb{T} = (\text{Modul})$, $B = \mathbb{T} \setminus A$ und $X = \mathbb{S} \setminus N(B)$, wobei $N(B)$ die Menge aller Stellen bezeichnet, die zu Transitionen aus B benachbart sind.

X ist die Menge der internen Stellen von A .

Die Menge \mathcal{C} der eingeschränkten Schaltschritte ist definiert durch

$$\mathcal{C} := \{ (M|X, T, M'|X) \mid T \in A, X = \mathbb{S} \setminus N(\mathbb{T} \setminus A) \text{ und } (M, T, M') \in \dots \}.$$

Der Modulhomomorphismus $m : \mathcal{C} \rightarrow \mathcal{C}$ ist definiert durch

$$m((M, T, M')) := (M|X, T, M'|X) \quad \text{für } T \in A$$

und

$$m((M, T, M')) := \dots \text{ sonst.}$$

Das Bild der Schaltfolgensprache unter einem Modulhomomorphismus wird Modulsprache genannt.

Wie bei den Projektionen wird auch bei den Modulhomomorphismen der reduzierte Erreichbarkeitsgraph mit $\mathbb{R}\mathbb{G}$ und die Menge seiner Knoten mit $\mathbb{M}\mathbb{R}$ bezeichnet. Die Definition von $\mathbb{R}\mathbb{G}$ sowie die der Hilfsgraphen $\mathbb{G}(M)$ findet sich in /17/. Die Menge der Knoten von $\mathbb{R}\mathbb{G}$ sei mit $\mathbb{M}\mathbb{R}$ bezeichnet.

Auch hier kann $\mathbb{R}\mathbb{G}$ als nichtdeterministischer Automat mit einer Zustandsüberföhrungsfunktion $\delta : \mathbb{M}\mathbb{R} \times \Sigma^* \rightarrow 2^{\mathbb{M}\mathbb{R}}$ betrachtet werden.

Für ein $w \in \Sigma^*$ bezeichne $\mathbb{T}(w)$ die Menge aller Transitionen, die in w "vorkommen", also $\mathbb{T}(w) = \{T \in \mathbb{T} \mid \text{es existieren } u, v \in \Sigma^* \text{ sowie } M, M' \in \mathbb{M} \text{ mit } w = u(M|X, T, M'|X)v\}$.

Für $L' \in \Sigma^*$ bezeichne $\mathbb{T}(L') = \{T \in \mathbb{T} \mid \text{es existiert } w \in L' \text{ mit } T \in \mathbb{T}(w)\}$.

Ein Knoten $Y \in \mathbb{M}\mathbb{R}$ mit $Y = M \in \mathbb{M}$ bzw. $Y = (M, M') \in \mathbb{M} \times \mathbb{M}$ heißt abgeschlossen in

$\mathbb{R}\mathbb{G}$, wenn eine der folgenden zwei Eigenschaften gilt :

- (1) Es gibt ein $Z \in \mathbb{M}\mathbb{R}$ und ein $v \in M^*$ mit $Z \in \mathcal{Z}^*(Y, v)$ und $R(A) \cap T(v) = R(A) \cap T(RZ)$.
- (2) Zu jedem Knoten M^* im Hilfsgraphen $\mathbb{G}(M)$ gibt es ein $Z \in \mathbb{M}\mathbb{R}$ und ein $u \in M^*$ mit $Z \in \mathcal{Z}^*(Y, u)$ und $R(A) \cap T(u) = \emptyset$ sowie $F^\circ(Z) \cap F^*(M^*) = \emptyset$.

Satz 11 : Ein Modulhomomorphismus m ist genau dann schlicht auf L , wenn $\mathbb{R}\mathbb{G}$ schlicht ist und wenn jeder Knoten $Y \in \mathbb{M}\mathbb{R}$ abgeschlossen in $\mathbb{R}\mathbb{G}$ ist.

Beweis :

(1) Sei m schlicht auf L .

Wie im Beweis des vorhergehenden Satzes läßt sich mit der Beweismethode des Satzes 1 in /17/ zeigen, daß $\mathbb{R}\mathbb{G}$ schlicht ist.

Bleibt noch zu zeigen, daß jeder Knoten $Y \in \mathbb{M}\mathbb{R}$ abgeschlossen in $\mathbb{R}\mathbb{G}$ ist.

Sei $Y \in \mathbb{M}\mathbb{R}$ mit $Y = M \in \mathbb{M}$ bzw. $Y = (M, M')$ $\in \mathbb{M} \times \mathbb{M}$ und M^* ein Knoten im Hilfsgraphen $\mathbb{G}(M)$. Mit den Hilfssätzen 1, 2 und 3 aus /17/ läßt sich dann eine Schaltfolge $wxy \in L$ mit folgenden Eigenschaften konstruieren :

$$\begin{aligned} \mathcal{Z}^*(M1, w) &= M \cap \mathcal{Z}^*(M1, m(w)) , \\ Y \cap \mathcal{Z}^*(M1, m(wx)) &= \emptyset , \\ x &= w \text{ falls } Y = M , \\ \mathcal{Z}^*(M, x) &= M' \text{ falls } Y = (M, M') , \\ \text{die Transitionen der Schaltschritte von } x &\text{ liegen alle in } A \setminus R(A) , \\ m(y) &= w \text{ und } \mathcal{Z}^*(M, xy) \cap N(B) = M^* \cap N(B) . \end{aligned}$$

Für die Menge der Fortsetzungen der Schaltfolge wxy gilt entweder

$$R(A) \cap T(m((wxy)^{-1}(L))) = \emptyset \quad \text{oder} \quad R(A) \cap T(m((wxy)^{-1}(L))) = M^* .$$

Im Fall $R(A) \cap T(m((wxy)^{-1}(L))) = M^*$ gibt es ein $z \in (wxy)^{-1}(L)$ minimaler Länge mit $R(A) \cap T(m(z)) = M^*$. Durch geeignetes "Umsortieren" (Hilfssatz 1 /17/) kann eine Schaltfolge $wxrstq \in L$ konstruiert werden mit folgenden Eigenschaften :

$$\begin{aligned} \text{Die Transitionen der Schaltschritte von } r &\text{ liegen alle in } A \setminus R(A) , \\ m(st) &= w , \\ \mathcal{Z}^*(M, xrs) \cap N(B) &= M^* \cap N(B) \text{ und} \\ q &\text{ mit } R(A) \cap T(m(q)) = M^* . \end{aligned}$$

Daraus folgt die Existenz eines $Z \in \mathbb{M}\mathbb{R}$ mit $Z \in \mathcal{Z}^*(Y, m(r))$ und $F^\circ(Z) \cap F^*(M^*) = \emptyset$. Mit $u = m(r)$ ist dann die Eigenschaft (2) der Abgeschlossenheitsbedingung für Y erfüllt.

Im Fall $R(A) \cap T(m((wxy)^{-1}(L))) = \emptyset$ folgt aus der Schlichtheit von m auf L in wxy die

Existenz eines $v \in m((wxy)^{-1}(L))$ mit $v^{-1}(m((wxy)^{-1}(L))) = v^{-1}(m(wxy)^{-1}(m(L)))$. Falls bei einer Schaltfolge $wx \in L$ die Transitionen der Schaltschritte von x alle in $A \setminus R(A)$ liegen, und falls $*(M1, w) = *(M1, m(w))$, dann läßt sich mit der Beweismethode des Satzes 1 in /17/ zeigen, daß es ein $Y \in *(M1, m(wx))$ gibt mit $m((wx)^{-1}(L)) = RY$. Wegen $m(y) = \text{gilt}$ $m((wx)^{-1}(L)) = m((wxy)^{-1}(L))$ und damit $v \in RY$. Daraus folgt die Existenz eines $Z \in *(Y, v)$ mit $RZ = v^{-1}(RY) = v^{-1}(m((wx)^{-1}(L))) = v^{-1}(m(wxy)^{-1}(m(L))) = v^{-1}(m((wxy)^{-1}(L)))$. Wegen $R(A) \cap \mathbb{T}(m((wxy)^{-1}(L))) = \emptyset$ folgt dann $R(A) \cap \mathbb{T}(v) = \emptyset = R(A) \cap \mathbb{T}(RZ)$, und damit ist die Eigenschaft (1) der Abgeschlossenheitsbedingung für Y erfüllt.

(2) Sei $\mathbb{R}\mathbb{G}$ schlicht und jeder Knoten $Y \in \mathbb{M}\mathbb{R}$ abgeschlossen in $\mathbb{R}\mathbb{G}$.

Für jedes $w \in L$ ist jetzt zu zeigen, daß m schlicht auf L in w ist.

Fall 2.1 : $w = w'x$ mit $*(M1, w') = *(M1, m(w'))$, und

die Transitionen der Schaltschritte von x liegen alle in $A \setminus R(A)$.

Wie oben bemerkt wurde, gibt es unter dieser Voraussetzung ein $Y \in *(M1, m(w))$ mit $m(w^{-1}(L)) = RY$. Die Schlichtheit von m auf L in w läßt sich dann wie im Fall 2.1 des vorhergehenden Beweises zeigen.

Fall 2.2 : w besitzt nicht die Darstellung wie in Fall 2.1.

Durch geeignetes "Umsortieren" von w läßt sich eine Schaltfolge $w'xy \in L$ mit folgenden Eigenschaften konstruieren :

$$m(w) = m(w'xy),$$

$$*(M1, w) = *(M1, w'xy),$$

$$*(M1, w') = *(M1, m(w')),$$

die Transitionen der Schaltschritte von x liegen alle in $A \setminus R(A)$ und

$$m(y) = \dots$$

Wie schon im letzten Beweis gezeigt wurde, genügt es jetzt wegen der ersten beiden Eigenschaften die Schlichtheit von m auf L in $w'xy$ zu zeigen.

Wegen der dritten und vierten Eigenschaft gibt es ein $Y \in *(M1, m(w'x))$, welches im Fall $x = \dots$ die Darstellung $Y = M \cup M'$ mit $M = *(M1, w')$ und im Fall $x = \dots$ die Darstellung $Y = (M, M') \cup M''$ mit $M = *(M1, w')$ und $M' = *(M, x)$ besitzt.

Nach Voraussetzung ist Y abgeschlossen in $\mathbb{R}\mathbb{G}$, d.h. es gilt (1) oder (2) der Abgeschlossenheitsbedingung.

Fall 2.2.1 : Abgeschlossenheitseigenschaft (1) gilt für Y .

Unter dieser Voraussetzung besitzt die Schaltfolge $w'x$ eine Fortsetzung r mit

$$R(A) \cap \mathbb{T}(m((w'xr)^{-1}(L))) = \emptyset$$

und die Transitionen der Schaltschritte von r liegen alle in $A \setminus R(A)$. Damit besitzt die Schaltfolge $w'xy$ eine Fortsetzung r' mit $m(r') = m(r)$ und $m((w'xyr')^{-1}(L)) = m((w'xr)^{-1}(L))$. Da die Schaltfolge $w'xr$ die Voraussetzung von Fall 2.1

erfüllt ist m schlicht auf L in $w'xr$.

Deshalb gibt es ein u mit $m(w'xr)^{-1}(m(L)) = m(w'xyr')^{-1}(m(L))$ mit

$$u^{-1}(m((w'xyr')^{-1}(L))) = u^{-1}(m((w'xr)^{-1}(L))) = u^{-1}(m(w'xr)^{-1}(m(L))) = u^{-1}(m(w'xyr')^{-1}(m(L)))$$

Also ist m schlicht auf L in $w'xyr'$ und damit wegen Hilfssatz 1 auch schlicht in $w'xy$.

Fall 2.2.2: Abgeschlossenheitseigenschaft (2) gilt für Y .

Der Schaltfolge y "entspricht" eine Schaltfolge y' im Hilfsgraphen $\mathbb{G}(M)$, die zu einer Markierung M^* führt. Wegen der Abgeschlossenheitseigenschaft (2) gibt es dann eine Fortsetzung s der Schaltfolge $w'x$, wobei die Transitionen der Schaltschritte von s alle in

$A \setminus R(A)$ liegen, und einen Knoten $Z \in MR$ mit $Z \in *(Y, m(s))$. Außerdem gibt es eine (ggf. leere) Fortsetzung der Schaltfolge y' im Hilfsgraphen $\mathbb{G}(M)$, die zu einer Kante in $\mathbb{R}\mathbb{G}$ beiträgt, die von Z ausgeht und deren Kantenbeschriftung eine Transition aus $R(A)$ enthält. Durch geeignetes "Umsortieren" und "Zusammensetzen" (Hilfssätze 1 und 2 aus /17/) können damit die Schaltfolgen $w'xys'r'q \in L$ und $w'xsy^\circ r'q \in L$ mit folgenden Eigenschaften konstruiert werden:

$$\begin{aligned} m(w'xys'r'q) &= m(w'xsy^\circ r'q), \\ *(M1, w'xys'r'q) &= *(M1, w'xsy^\circ r'q), \\ m(y^\circ r') &= \text{und} \\ q &\text{ mit } R(A) \cap T(m(q)) \text{ .} \end{aligned}$$

Wie schon oben bemerkt wurde, folgt wegen der ersten beiden Eigenschaften aus der Schlichtheit von m auf L in $w'xsy^\circ r'q$ auch die Schlichtheit in $w'xys'r'q$ und daraus die Schlichtheit in $w'xy$. Wegen der dritten und vierten Eigenschaft und der speziellen Form von s erfüllt die Schaltfolge $w'xsy^\circ r'q$ die Voraussetzung von Fall 2.1, woraus die Schlichtheit von m auf L in $w'xsy^\circ r'q$ folgt.

Damit ist Satz 11 vollständig bewiesen.

Literatur

- /1/ S.Aggerwal, C.Coucourbetis, P.Wolper :
 Adding Liveness Properties to Coupled Finite State Machines
 ACM Transactions on Programming Languages and Systems 12 1990

- /2/ A.V.Aho, J.E.Hopcroft, J.D.Ullman :
 The Design and Analysis of Computer Algorithms
 Addison -Wesley Publishing Company 1974

- /3/ B.Alpern, F.B.Schneider :
 Defining Liveness
 Information Processing Letters 24 1985

- /4/ J.C.M.Baeten, W.P.Weijland :
 Process Algebra
 Cambridge University Press 1990

- /5/ B.Baumgarten, P.Ochsenschläger, R.Prinoth :
 Building Blocks for Distributed System Design
 Fifth Int. Workshop on Protocol Specification, Testing, and Verification
 North Holland 1976

- /6/ J.Berstel :
 Finite automata and rational languages. An introduction
 LITP Spring School on Theoretical Computer Science 1988 LNCS 386

- /7/ H.J.Burkhardt, P.Ochsenschläger, R.Prinoth :
 Product Nets
 A Formal Description Technique for Cooperating Systems
 GMD - Studien 165 1989

- /8/ E.M.Clarke, E.A.Emerson, A.P.Sistla :
 Automatic Verification of Finite-State Concurrent Systems using Temporal Logic
 Specifications
 ACM TOPLAS 8 1986

- /9/ M.Hack :
Petri Net Languages
Laboratory for Computer Science MIT T.R.161 1976
- /10/ D.Hogrefe :
Estelle, LOTOS und SDL
Standard-Spezifikationsprachen für verteilte Systeme
Springer Verlag 1989
- /11/ W.Klug :
OSI-Vermittlungsdienst und sein Verhältnis zum ISDN-D-Kanalprotokoll
Spezifikation und Analyse mit Produktnetzen
Arbeitspapiere der GMD 676 1992
- /12/ R.Milner :
Operational and Algebraic Semantics of Concurrent Processes
Handbook of Theoretical Computer Science Vol. B 1990
- /13/ U.Nitsche :
Erreichbarkeitsanalyse von Produktnetzen
Arbeitspapiere der GMD 521 1991
- /14/ P.Ochsenschläger :
Projektionen und reduzierte Erreichbarkeitsgraphen
Arbeitspapiere der GMD 349 1988
- /15/ P.Ochsenschläger :
Modulhomomorphismen
Arbeitspapiere der GMD 494 1990
- /16/ P.Ochsenschläger :
Die Produktnetzmaschine
Petri Net Newsletter 39 1991
- /17/ P.Ochsenschläger :
Modulhomomorphismen II
Arbeitspapiere der GMD 597 1991

/18/ B.Pehrson :

Protocol Verification for OSI

Computer Networks and ISDN Systems 18 1989/90

/19/ D.Perrin :

Finite Automata

Handbook of Theoretical Computer Science Vol. B 1990

/20/ D.Taubner :

Finite Representations of CCS and TCSP Programs by Automata and Petri Nets

LNCS 369 1976