Annales Univ. Sci. Budapest., Sect. Comp. 9 (1988) 201-226

MODELLIERUNG MIT PETRI-NETZEN

PETER H. STARKE

1.

Der Entwurf komplexer hardware- oder software-Systeme, z.B. zur Steuerung flexibler Produktionssysteme (CAM), zur Steuerung von Rechnerkommunikation (Kommunikationsprotokoll) oder zur Steuerung von Verkehrssystemen, beginnt stets mit einer formalen Spezifikation des Systems und dem Beweis der Korrektheit dieser Spezifikation. Natürlich hängen die beim Korrektheitsbeweis verwendeten Methoden in starkem Maße von der verwendeten Spezifikationssprache ab, insbesondere davon, ob dieses Begriffssystem durch eine mathematische Theorie gestützt ist, deren Resultate bei der Korrektheitsprüfung angewendet werden können.

In der Vergangenheit sind Probleme dieser Art hauptsächlich mit Hilfe von Begriffen und Methoden der Automatentheorie einerseits sowie speziellen Programmiersprachen andererseits – insbesondere den Spezifikationsteil betreffend – behandelt worden. Der Korrektheitsbeweis wurde vielfach nur durch Simulation des Modells auf der Basis der "Einsicht" des Entwerfers durchgeführt, was bei großen Systemen, d.h. solchen Systemen, deren relevante Details ein Mensch zu einer Zeit gar nicht überblicken kann, nicht möglich ist.

Ein weiterer Nachteil der automatentheoretischen und anderer Ansätze zur Systemspezifikation besteht darin, daß sie auf

eine strikt sequentielle Abarbeitung (und übrigens auch auf eine sequentielle Denkweise) orientiert sind, auf diese Weise Nebenläufigkeit, Parallelität sowie kausale Unabhängigkeit von Aktionen ignorieren; eine Folge der wesentlichen (expliziten oder impliziten) Verwendung des Zustandsbegriffs der Automatentheorie. Dadurch ist zum Beispiel eine partielle Unabhängigkeit in der Arbeit von Teilsystemen im Modell nicht mehr erkennbar bzw. es bedarf eines erheblichen Aufwandes, wie er bei der Dekomposition von Automaten betrieben wird, um sie wieder sichtbar zu machen.

Um das an einem Beispiel zu verdeutlichen, betrachten wir die automatentheoretische Spezifikation eines Systems mit 7 Zuständen $z_0, \ldots, z_6$ und 6 Aktivitäten $x_1, \ldots, x_6$ (die Zustandsänderungen bewirken) in Form der Überführungstabelle eines partiellen Automaten:

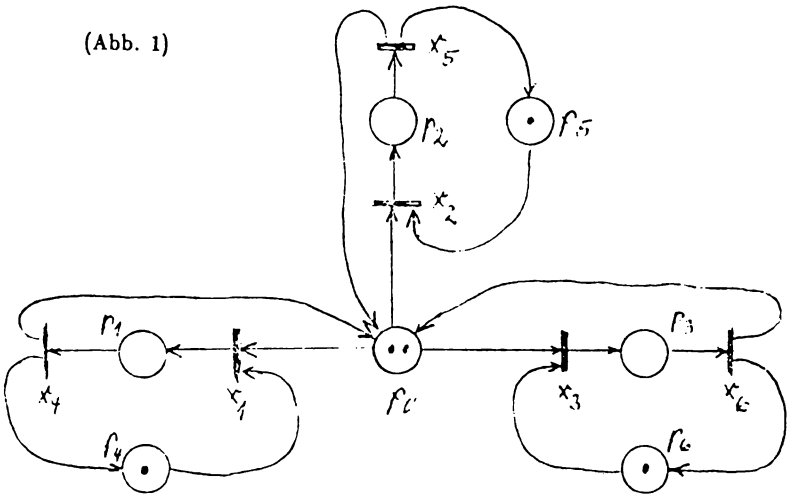
	z_0	z_1	z_2	z_3	z_4	z_5	z_6
x_1	z_1		z_4	z_5			
x_2	z_2	z_4		z_6			
x_3	z_3	z_5	z_6				
x_4		z_0			z_2	z_3	
x_5			z_0		z_1		z_3
x_5				z_0		z_1	z_3

Auch wenn man den Graph dieses Automaten aufzeichnet, ist es schwer, aus dieser Spezifikation mehr Information zu ersehen als die, daß die Aktivitäten x_i und $x_{i+3}(i = 1, 2, 3)$ invers zu einander sind, d.h. einander rückgängig machen.

Das beschriebene System besteht aus drei Teilsystemen, die für ihre Arbeit eine gemeinsame Ressource nutzen, die in nur zwei Exemplaren vorhanden ist. Man könnte etwa an zwei Transportroboter denken, die drei Produktionstationen zu beschicken haben. Eine Spezifikation desselben Systems durch ein Petri-Netz zeigt die Abbildung 1. Auch

(Abb. 1)

(Abb. 1)



ohne daß man etwas von Petri-Netzen versteht, erkennt man sofort, daß durch dieses Modell auch strukturelle Information widergespiegelt wird.

Informationen über die Struktur und kausale Abhängigkeiten bei der Spezifikation komplexer Systeme, d.h. der Konstruktion eines mathemathischen Modells, einzubeziehen, ist aber nicht nur notwendig, um das Modell transparent zu halten, sondern das gibt uns zusätzlich die Möglichkeit, eine Theorie zu entwickeln, die Struktur und Verhalten in Beziehung zueinander setzt. Während bei der Verifikation von Modellen auf automatentheoretischer Basis Methoden typisch sind, die auf einer Aufzählung aller möglichen Zustände und ihrer Prüfung hinsichtlich der gewünschten Eigenschaften gründen, können bei Petri-Netz-Modellen durch Anwendung entsprechender theoretischer Resultate oft aufwendige Aufzählungsverfahren vermieden werden.

Die Hauptvorteile der Verwendung von Petri-Netzen bei der Modellierung bzw. beim Entwurf komplexer Systeme bestehen darin, daß

(1) die graphische Darstellung an die Anschauung appelliert und auch dem Nichtfachmann verständlich ist,

(2) Netzmodelle auf ganz verschiedenen Abstraktionsniveaus anwendbar und die dabei entstehenden Modelle durch einfache (Verfeinerungs-bzw. Vergröberungs-) Operationen miteinander verbunden sind, also top-down bzw. bottom-up Spezifikation unter Beibehalten der Spezifikationssprache möglich ist,

(3) eine Theorie entwickelt worden ist, deren Resultate bei der Verifikation der Modelle, d.h. im Korrektheitsbeweis angewendet werden können und

(4) Programmpakete existieren, mit deren Hilfe Modellherstellung und Verifikation rechnergestützt erfolgen können.

Gegen den Vorteil (1) wird gelegentlich eingewendet, daß auch die Übersichtlichkeit eines Netzmodells bei großen Netzen stark leide. Die Tatsache, daß bei der Systembeschreibung ein Netz mit Hunderten von Knoten entstanden ist, zeigt aber nur, daß beim ersten Ansatz ein zu niedriges Abstraktionsniveau gewählt wurde, im allgemeinen bereitet es keine großen Schwierigkeiten, durch Abstraktion und entsprechende Vergröberung des Netzes die Übersichtlichkeit wieder herzustellen.

2.

Die Hauptunterschiede zwischen der Modellierung durch ein Petri-Netz und der Automaten-Modellierung bestehen darin, daß in einem Netzmodell der Begriff des (globalen) Zustandes keine Hauptrolle spielt und daß keine globale Zeitskala postuliert wird, in der sich die Aufeinanderfolge der Aktionen des modellierten Systems messen ließe. Vielmehr werden die Systemzustände dargestellt durch Markierungen von Plätzen, die das (quantitative) Erfülltsein von (eventuell lokal verteilten) Systembedingungen beschreiben, während sich die realisierbaren Folgen von Aktionen allein aus ihrer kausalen Abhängigkeit, die Kausalstruktur des System beschreibend, herleiten. Ein Netzmodell eines Sys-

tems besteht demnach aus Elementen, die den Systembedingungen (Teilsystemzuständen) zugeordnet sind - sie werden *Plätze* genannt - und deren Markierung den aktuellen Stand der Erfülltheit dieser Bedingungen in gegebenen Systemzustand anzeigt, aus Elementen, die den Aktivitäten (Änderungen eines Teilsystemzustands) zugeordnet sind - sie werden *Transitionen* genannt - , die unter gegebenen lokalen Bedingungen lokale Auswirkungen auf diese und andere Bedingungen haben, sowie aus Elementen, die eben diesen kausalen Zusammenhang zwischen dem Erfülltsein von Bedingungen und Aktivitäten beschreiben - sie werden *Bögen* genannt.

Bei der graphischen Repräsentation von Netzen verwendet man Kreise zur Darstellung von Plätzen, Rechtecke, die bis zu einem verdickten Strich entartet sein können, zur Darstellung von Transitionen und Pfeile zur Darstellung der Bögen. Ein Bogen verbindet stets einen Platz mit einer Transition oder umgekehrt, niemals zwei Plätze oder zwei Transitionen, denn direkte kausale Abhängigkeiten zwischen Bedingungen bzw. Aktivitäten gibt es nicht. Ein Bogen von einem Platz p zu einer Transition t (p ist dann ein *Vorplatz* von t) zeigt an, daß die Erfülltheit der p zugeordneten Systembedingung eine Voraussetzung für die Ausführbarkeit der t entsprechenden Aktivität (eine Vorbedingung) ist; ein Bogen von einer Transition t zu einem Platz p (p ist dann *Nachplatz* von t) bedeutet, daß das Durchführen der Aktivität, die von t modelliert wird, dazu führt, daß die Erfülltheit der p zugeordneten Bedingung zunimmt.

Die Kompliziertheit der Ausdrucksweise reflektiert die Tatsache, daß wir beim Erfülltsein von Bedingungen nicht von einer zweiwertigen Logik, sondern davon ausgehen, daß eine Bedingung wie "Transportroboter verfügbar" auch doppelt erfüllt sein kann (wenn zwei Roboter frei sind). In der Abbildung 1 repräsentiert der Platz p_0 die Bedingung "Transportroboter frei" und die beiden Punkte im p_0 entsprechenden Kreis (sogenannte *Marken*) bedeuten, daß im dargestellten Systemzustand zwei Roboter verfügbar sind.

In mathematischer Terminologie ist ein Petri-Netz ein paarer Graph, bei dem bestimmte Elemente (nämlich die Plätze bei gewöhnlichen Petri-Netzen, im allgemeinen Fall Plätze und Bögen) mit natürlichen Zahlen bewertet sind.

DEFINITION. Das Quadrupel $N = (P, T, F, m_0)$ wird *gewöhnliches Petri-Netz* genannt, wenn

(1) P und T endliche disjunkte Mengen sind,

(2) $F \subseteq (P \times T) \cup (T \times P)$ eine binäre Relation mit $dom(F) \cup cod(F) = P \cup T$ ist und

(3) m_0 eine Abbildung von P in die Menge IN der natürlichen Zahlen (einschließlich 0) ist.

Hierbei ist

$$dom(F) := \{x \mid \exists y(x,y) \in F\},$$

$$cod(F) := \{y \mid \exists x(x,y) \in F\}.$$

P ist die Menge der Plätze, T jene der Transitionen und F die Menge der Bögen des Netzes. Ein Platz, der von keinem Bogen berührt wird, repräsentiert eine Bedingung, die nicht verändert wird; eine Transition bei der kein Bogen entspringt oder einmündet, verändert nichts, beides wird ausgeschlossen.

Bisher haben wir nur die statischen Eigenschaften von Netzmodellen besprochen, wir wenden uns nun der Dynamik zu, d.h. wir geben an, wie das *Schalten* (auch *Feuern* genannt) von Transitionen die Markierung des Netzes verändert, mit anderen Worten, welche Veränderungen von Systembedingungen beim Ausführen von Systemaktivitäten beschreibbar sind.

Es sei $N = (P, T, F, m_0)$ ein gewöhnliches Petri-Netz, $m : P \to IN$ eine Markierung und $t \in T$ eine Transition. Wir sagen, daß t bei m *Konzession* hat (*schaltfähig* bzw. *feuerbar* ist), wenn für jeden Vorplatz p von t, also jedes p mit $(p,t) \in F$, gilt $m(p) \geq 1$ d.h. p enthält wenigstens eine Marke (p ist *markiert*):

$$Konz(t,m) :\leftrightarrow \forall p((p,t) \in F \to m(p) \geq 1).$$

Eine Aktivität kann also genau dann stattfinden, wenn alle ihre Vorbedingungen (wenigstens einfach) erfüllt sind. Im Netz der Abbildung 1 haben bei der dargestellten Markierung genau die Transitionen x_1, x_2 und x_3 Konzession.

Wenn eine Transition t bei einer Markierung m Konzession hat, so kann sie schalten (feuern). Die Wirkung des Schaltens von t besteht darin, daß von jedem Vorplatz von y eine Marke entnommen und auf jeden Nachplatz von t eine Marke zusätzlich aufgebracht wird; es entsteht also durch das Feuern von t eine neue Markierung m' mit

$$m'(p) := \begin{cases} m(p) - 1, & \text{falls } (p,t) \in F \ \& \ (t,p) \notin F, \\ m(p) + 1, & \text{falls } (p,t) \notin F \ \& \ (t,p) \in F, \\ m(p), & \text{sonst.} \end{cases}$$

Markierung werden, wenn die Menge P durchnumeriert ist, im allgemeinen als Zeilenvektoren von natürlichen Zahlen angegeben, wobei die i-te Stelle der Wert $m(p_i)$ ist ($bzw. m(p_{i-1})$, wenn beim Numerieren mit 0 begonnen wurde). Im Netz von Abbildung 1 ist also $m_0 = (2,0,0,0,1,1,1)$. Wie erwähnt, gilt Konz (x_1, m_0). Durch Schalten von x_1 entsteht die Markierung
$m_1 = (1,1,0,0,0,1,1)$.

3.

Eines der in der Literatur beliebtesten Beispiele für den Entwurf eines nebenläufigen verteilten Systems bildet DIJKSTRA's "Five Philosophers Dining Problem":

Fünf Philosophen sitzen an einem runden Tisch, um Fisch zu essen oder zu denken. Um zu essen braucht jeder zwei Gabeln, aber es sind nur fünf Gabeln vorhanden, zwischen je zwei Philosophen liegt eine Gabel auf dem Tisch. Jeder Philosoph darf nur die Gabeln aufnehmen, die unmittelbar rechts und links neben seinem Teller liegen und muß sie nach Benutzung gesäubert dorthin wieder ablegen, jede Gabel kann also mehrmals von den ihr benachbarten Philosophen benutzt werden. Es soll eine verteilte

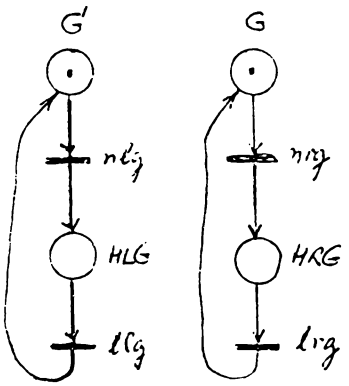
Steuerung entworfen werden, die sichert, daß jeder Philosoph immer wieder die Möglichkeit hat zu essen und die die Ressourcen (Gabeln) möglichst gut ausnutzt.

Daß wir eine verteilte Steuerung suchen, bedeutet, daß wir nicht einen zentralen Mechanismus entwerfen wollen, der die Ressourcenkonflikte der Philosophen löst, in dem er ihnen (einzeln oder paarweise) innerhalb gegebener bzw. erzeugter Zeitintervalle zu essen befiehlt, sondern ein einziges Verhaltensmuster für jeden Philosophen suchen, dessen Einhalten die genannten Ziele sichert, wobei aber die Dauer des Essens oder der Zeitpunkt, zu dem ein Philosoph hungrig wird, ohne Bedeutung sind.

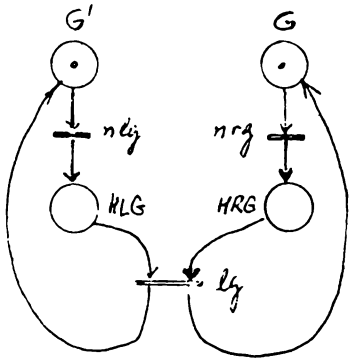
Wir setzen voraus, daß jeder Philosoph ißt, sobald er seine Gabeln in den Händen hat und sich anderenfalls seinen Gedanken hingibt. Ein Philosoph ist also ein Teilsystem mit vier Aktivitäten, nämlich nrg('nimmt die rechte Gabel'), *nlg* ('nimmt die linke Gabel'), *lrg* ('legt die rechte Gabel zurück'), *llg* ('legt die linke Gabel zurück') und mit zwei Bedingungen *HRG* ('hat die rechte Gabel') und *HLG* ('hat die linke Gabel'). Sind beide Bedingungen erfüllt, so ist das Teilsystem im Zustand 'essend', anderenfalls im Zustand 'denkend'. Vorbedingung G(*bzw.*G') für nrg (*bzw.nlg*) ist natürlich, daß die rechte (*bzw.linke*) Gabel auf dem Tisch liegt, und da diese Gabel zugleich die linke (*bzw.rechte*) Gabel des rechten (*bzw.linken*) Nachbarn ist, sind die Teilsysteme über die Gabel-Bedingungen miteinander verkoppelt.

Das Netzmodell eines einzelnen Philosophen sieht also aus wie in der Abbildung 2 dargestellt. Es beschreibt in der Tat alle Möglichkeiten, die ein Philosoph hat, etwa die, eine Gabel aufzunehmen und sie sofort wieder abzulegen.

Eine Steuerung dieses Teilsystems vorzunehmen bedeutet, seine Verhaltensmöglichkeiten einzuschränken. Wir können das sinnlose Aufnehmen und Ablegen derselben Gabel verhindern, in dem wir die Aktivitäten *llg* und *lrg synchronisieren*, d.h. zu einer Aktion *lg* ('legt beide Gabeln zurück') verschmelzen. Hat danach ein Philosoph eine Gabel, so muß er sie behalten und warten, bis er die andere bekommen kann, um dann zu essen und beide Gabeln abzulegen. Das Netz eines Philosophen sieht jetzt wie in der Abbildung 3 dargestellt aus.



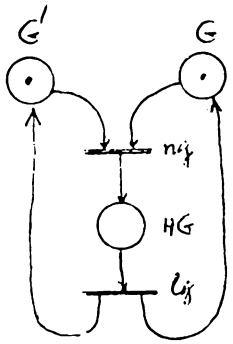
(Abb. 2)



(Abb. 3)

Diese Synchronisierung, die unerwünschtes Verhalten ausschließt, hat aber unangenehme Auswirkungen auf das Gesamtsystem: es kann in einen Deadlock geraten, d.h. in einen Zustand, bei dem keinerlei Aktivität mehr möglich ist, also ein totaler Systemstillstand erreicht ist. Das passiert zum Beispiel, wenn jeder der fünf Philosophen seine rechte Gabel aufnimmt.

Um diesen Deadlock zu vermeiden, synchronisieren wir auch die Aktivitäten nlg und nrg zu einer Aktivität ng und verschmelzen die Bedingungen HLG und HRG zu einer Bedingung HG ('hat Gabeln'). Die Abbildung 4 zeigt das entsprechende Netz.



Damit ist eine Lösung des Problems erreicht. Das System ist deadlockfrei und jeder Philosoph erhält immer wieder die Möglichkeit, seine Gabeln zu nehmen. Die Ressourcen werden maximal

genutzt, denn jederzeit haben zwei Philosophen Gabeln oder die Möglichkeit, sie aufzunehmen.

4.

Wenn wir das Erfülltsein von Bedingungen quantitativ durch Markenanzahlen auf Plätzen modellieren, ist es nur konsequent, wenn wir gestatten, daß eine einzelne Aktivität (wie z.B. das Belegen von zwei Magnetbandeinheiten in der Peripherie eines Rechners) die Markierung eines Platzes um mehr als eine Marke verändert. Zu diesem Zweck ordnen wir jedem Bogen eine positive natürliche Zahl zu, die *Vielfachheit* des Bogens genannt wird. Bei einem gewönlichen Petri-Netz sind also alle Vielfachheiten gleich 1.

DEFINITION. Das Quintupel $N = (P, T, F, V, m_0)$ wird *Petri-Netz* genannt, wenn

(1) (P, T, F, m_0) ein gewönliches Petri-Netz ist und

(2) V eine Abbildung von F in die Menge IN^+ der positiven natürlichen Zahlen ist.

Jeder Transition t eines Petri-Netzes $N = (P, T, F, V, m_0)$ ordnen wir zwei Markierungen t^-, t^+ wie folgt zu:

$$t^-(p) := \begin{cases} V(p,t), & \text{falls } (p,t) \in F, \\ 0, & \text{sonst;} \end{cases}$$

$$t^+(p) := \begin{cases} V(t,p), & \text{falls } (t,p) \in F, \\ 0, & \text{sonst} \end{cases}$$

(für alle p aus P). Die Markierung t^- gibt also für jeden Platz p an, wieviele Marken beim Feuern von t von p abgezogen werden, t^+ die Zahl der Marken, die dabei auf p aufgebracht werden.

Wir rechnen mit Markierungen wie mit Vektoren, also komponentenweise. Die Relation $\leq$ gilt für zwei Markierungen, wenn sie komponentenweise (platzweise) gilt.

Eine Transition t hat Konzession bei der Markierung m (in N), wenn $t^- \leq m$ gilt. In diesem Fall entsteht durch Feuern von t die Markierung m' mit $m' := m - t^- + t^+$.

Jede Markierung des Petri-Netzes beschreibt einen denkbaren globalen Zustand des modellierten Systems, die *Anfangsmarkierung* m_0 den Initialzustand. Aus einem Netzmodell kann man sofort ein Automatenmodell des betrachteten Systems wie folgt erhalten: Systemzustände sind dann relevant, wenn sie durch Aktivitäten des Systems aus seinem Initialzustand entstehen können, im Netzmodell beschrieben von Markierungen, die durch das Schalten von Transitionen aus der Anfangsmarkierung hervorgehen können. Solche Markierungen heißen *erreichbar von* m_0. Die Aktivitäten des Systems (Transitionen des Netzes) erscheinen im Automatenmodell als Eingabesignale, die das Durchführen der entsprechenden Aktivität auslösen. Wir definieren dementsprechend zu jedem Petri-Netz N eine (partielle) Überführungsfunktion δ_N wie folgt:

$$\delta_N(m,t) := \begin{cases} m - t^- + t, & \text{falls} \quad t^- \leq m, \\ \text{nicht definiert}, & \text{sonst.} \end{cases}$$

$(m \in IN^P, t \in T)$

Diese Abbildung wird auf Wörter (endliche Folgen) von Transitionen induktiv fortgesetzt (e bezeichnet das leere Wort):

$$\delta_N(m,e) := m,$$

$$\delta_N(m,wt) := \begin{cases} \delta_N(\delta_N(m,w),t), & \text{falls } \delta_N(m,w) \text{ definiert} \\ & \& \ \delta_N(m,w) \leq t^-, \\ \text{nicht definiert}, & \text{sonst.} \end{cases}$$

Ist w eine endliche Folge von Transitionen und $\delta_N(m,w)$ definiert, so ist $\delta_N(m,w)$ die Markierung, die in N entsteht, wenn bei der Markierung m beginnend die Transitionen in w in der durch w gegebenen Reihenfolge gefeuert werden.

Bezeichnet $W(T)$ die Menge aller Wörter über der Menge T als Alphabet, dann ist die Menge der in N von der Markierung m aus erreichbaren Markierungen also

$$R_N(m) := \{\delta_N(m,w) \mid w \in W(T) \& \delta_N(m,w) \text{definiert}\}.$$

Der zum Netz N gehörige partielle Automat (das Automatenmodell des von N modellierten Systems) ist durch $A_N = (T, R_N(m_0), \delta_N, m_0)$ gegeben, wobei also T als Menge der Eingabesignale, $R_N(m_0)$ als Zustandsmenge, δ_N als Überführungsfunktion und m_0 als Anfangszustand figuriert.

Ist ein System entworfen, d.h. ein Netzmodell N hergestellt, so besteht eine der ersten Verifikationsfragen darin, ob überhaupt ein System mit endlich vielen globalen Zuständen entworfen wurde, denn nur solche Systeme sind realisierbar. Das ist die Frage, ob die Menge $R_N(m_0)$ endlich ist, oder, wie wir sagen, ob das Netz N *beschränkt* ist.

DEFINITION. Es sei $N = (P, T, F, V, m_0)$ ein Petri-Netz, m eine Markierung, $k \in IN, p \in P$. Der Platz p heißt *k-beschränkt bei* m, wenn für alle $m' \in R_N(m)$ gilt $m'(p) \leq k$. Wenn ein solches k existiert, wird p *beschränkt bei* m genannt. Das Netz N wird als k-beschränkt (bzw. beschränkt) bezeichnet, wenn alle seine Plätze k-beschränkt (bzw. beschränkt) bei m_0 sind. Statt '1-beschränkt' sagt man auch 'sicher'.

Folgerung. Ein Petri-Netz N ist beschränkt genau dann, wenn $R_N(m_0)$ endlich ist.

Wenn ein Petri-Netz sicher ist, so kann bei keiner erreichbaren Markierung ein Platz mehr als eine Marke tragen. Das Überprüfen dieser Eigenschaft ist dann wichtig, wenn die den Plätzen des Netzes zugeordneten Systembedingungen rein logischer Natur sind, also nur erfüllt sein können (wie z.B. die Bedingungen im Philosophenproblem). Wenn in solch einem Fall das Netz nicht sicher ist, liegt sicher ein Entwurfsfehler vor.

Satz. Es gibt einen Algorithmus, der für jedes Petri-Netz entscheidet, ob es beschränkt ist.

Zum Beweis dieses Satzes wurde ein Algorithmus angegeben, der aus dem interessierenden Petri-Netz N den sogenannten Überdeckbarkeitsgraphen konstruiert. Neben der Frage, ob das Netz beschränkt ist, kann durch Inspektion dieses stets endlichen Graphen entschieden werden, welche Plätze unbeschränkt sind, welche Platzmengen $Q \subseteq P$ *simultan unbeschränkt* sind, d.h. ob gilt

$$\forall k(k \in IN \rightarrow \exists m(m \in R_N(m_0) \ \& \ \forall p(p \in Q \rightarrow m(p) > k))),$$

und ob eine gegebene Markierung m in N *überdeckbar* ist, d.h. ob es eine erreicbare Markierung $m' \in R_N(m_0)$ mit $m' \geq m$ gibt. Die Frage nach der Überdeckbarkeit einer Markierung stellt sich auf der Systemebene als Frage nach der (eventuell unerwünschten) Übererfüllbarkeit gewisser Systembedingungen. Wie wir sehen werden, lassen sich viele andere Fragen (eventuell durch Netzmodifikation) auf Überdeckbarkeitsprobleme reduzieren.

Die Frage nach der simultanen Erfüllbarkeit bzw. Nichterfüllbarkeit bestimmter Systembedingungen, d.h. nach der Herstellbarkeit bestimmter (eventuell unerwünschter) Systemzustände stellt sich auf der Netzebene als Frage nach der Erreichbarkeit gewisser (endlich vieler) *Teilmarkierungen* (das sind Abbildungen von einer Teilmenge Q von P in IN). Diese Frage läßt sich durch Modifikation des Netzes (Hinzunehmen weiterer Transitionen und Plätze) auf die Frage nach der Erreichbarkeit einer einzigen (totalen) Markierung algorithmisch reduzieren. Für beschränkte Netze ist dieses Problem, das sogenannte *Erreichbarkeitsproblem* trivial durch Inspektion der Menge $R_N(m_0)$ lösbar, aber es gilt sogar der

Satz. Es gibt einen Algorithmus, der für jedes Petri-Netz N und jede Markierung m entscheidet, ob m in der Menge $R_N(m_0)$ liegt.

Das Erreichbarkeitsproblem ist von sehr hoher algorithmischer Komplizierheit. Leider geben die derzeit bekannten Algorithmen zu seiner Lösung im allgemeinen Fall keinen Hinweis darauf, wie man das im beschränkten Fall anwendbare Inspektionsverfahren verbessern könnte.

Eine weitere erwünschte Systemeigenschaft ist die, daß alle (eventuell mit Ausnahme der Initialisierungs-) Aktivitäten, gleich was geschieht, immer wieder ausgeführt werden können (sofern es sich nicht um ein Wegwerf-System handelt), wie in unserem Philosophenbeispiel. Auf der Netzebene spiegelt sich diese Eigenschaft in der Lebendigkeit der Transitionen bzw. des Netzes wieder:

DEFINITION. Es sei $N = (P, T, F, V, m_0)$ ein Petri-Netz, m eine Markierung von N und t eine Transition. t heißt *potentiell feuerbar* bei m in N, wenn eine von m aus erreichbare Markierung m' existiert, bei der t Konzession hat:

$$\exists m'(m' \in R_N(m) \;\&\; t^- \leq m').$$

t wird *lebendig bei* m genannt, wenn t bei jeder von m aus erreichbare Markierung m'' potentiell feuerbar ist:

$$\forall m''(m'' \in R_N(m) \rightarrow \exists m'(m' \in R_N(m'') \;\&\; t^- \leq m')).$$

t heißt *lebendig in* N, wenn t bei m_0 lebendig ist, und das Netz N wird als *lebendig* bezeichnet, wenn alle seine Transitionen lebendig sind.

Durch Reduktion auf das Erreichbarkeitsproblem konnte bewiesen werden:

Satz. Es gibt einen Algorithmus, der für jedes Petri-Netz N und jede Transition t entscheidet, ob t lebendig in N ist.

Ob eine Transition t bei m potentiell feuerbar ist, ist offenbar ein Überdeckbarkeitsproblem für die Markierung t^-, also ebenfalls entscheidbar.

Für beschränkte Netze N kann die Lebendigkeit natürlich durch Inspektion des Erreichbarkeitsgraphen, d.h. des Graphen des Automaten A_N entschieden werden, wobei die Markierungen keine explizite Rolle spielen, es gilt nämlich

Folgerung. Eine Transitionen t ist genau dann lebendig, wenn

$$\forall w(w \in L_N(m_0) \rightarrow \exists u(u \in W(T) \;\&\; wut \in L_N(m_0))).$$

Dabei bezeichnet $L_N(m)$ die Menge aller Transitionswörter w derart, daß $\delta_N(m, w)$ definiert ist. Eine Transition t ist also genau dann lebendig, wenn jedes bei m_0 feuerbare Transitionswort w zu einem bei m_0 feuerbaren Wort wut fortgesetzt werden kann.

Für spezielle Netzklassen, insbesondere Klassen gewöhnlicher Netze, sind Lebendigkeitskriterien bekannt, deren Prüfung eine Aufzählung aller erreichbaren Markierungen, d.h. die Konstruktion des Erreichbarkeitsgraphen, nicht erfordert.

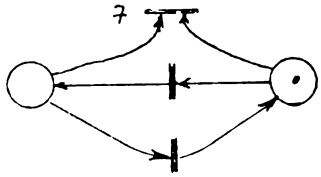
Neben der Wiederholbarkeit der Systemaktivitäten, d.h. der Lebendigkeit der entsprechenden Transitionen, interessiert häufig auch die Wiederholbarkeit von Systembedingungen, d.h. die Platzlebendigkeit.

DEFINITION. Es sei $N = (P, T, F, V, m_0)$ ein Petri-Netz, m eine Markierung und $p \in P$. Der Platz p heißt *markierbar* (bei m_0), wenn gilt

$$\exists m'(m' \in R_N(m_0) \ \& \ m'(p) > 0),$$

und p wird *lebendig* genannt, wenn p bei jeder von m_0 aus erreichbaren Markierung markierbar ist.

Markierbarkeit ist offensichtlich ein Überdeckungsproblem, also entscheidbar. Nehmen wir zum Netz N eine neue Transition t_p hinzu und zeichnen in N je einen einfachen Bogen von p zu t_p und zurück ein, so ist in dem erhaltenen Netz N' die Transition t_p genau dann lebendig, wenn p in N lebendig ist. Damit reduziert sich das Platzlebendigkeitsproblem auf das Lebendigkeitsproblem für Transitionen, d.h. es ist ebenfalls entscheidbar. Wenn ein Netz lebendig ist, so sind alle seine Plätze lebendig, weil jeder Platz Vorplatz oder Nachplatz einer Transitio ist; die Umkehrung gilt i.a. nicht, wie das Beispiel in der Abbildung 5 zeigt. In diesem Beispiel ist die



(Abb. 5)

Transition t sogar *tot*, d.h. t hat bei keiner erreichbaren Markierung Konzession. Entspricht eine solche Transition einer Systemaktivität, so deutet sich eventuell ein Entwurfsfehler an. Andererseits stellt die Tatsache, daß t tot ist, aber eine Systeminvariante dar, nämlich die, daß die entsprechende Aktivität bei der Arbeit des Systems niemals ausgeführt werden kann, was ja, wenn es sich um ein Schadensereignis handelt, durchaus von Interesse ist. Dementsprechend verwendet man beim Systementwurf erwünscht tote Transitionen (sie werden *Fakt* genannt) zur Spezifikation von Systemaktivitäten, die ausgeschlossen werden sollen. Ob eine Transition tot (bzw. Fakt), also nicht potentiell feuerbar ist, ist offensichtlich entscheidbar.

Einem absoluten Systemstillstand, einem Deadlock also, d.h. einem Zustand, den das System von sich aus nicht mehr verlassen kann, weil keinerlei Aktivität mehr möglich ist, entspricht auf der Netzebene eine Markierung, bei der keine Transition Konzession hat, solche Markierungen heißen *tot*. Wenn eine tote Markierungen erreichbar ist, dann ist keine Transition lebendig. Die Frage, ob eine tote Markierung erreichbar ist, kann auf das Erreichbarkeitsproblem reduziert werden, weil eine Markierung genau dann tot ist, wenn sie mit einer toten Teilmarkierung $m^* : Q \to IN$ mit minimalem Träger Q auf der Menge Q übereinstimmt und es nur endlich viele solche Teilmarkierungen gibt. Dabei ist $m^* :\to$ IN eine tote Teilmarkierung mit minimalem Träger, wenn gilt:

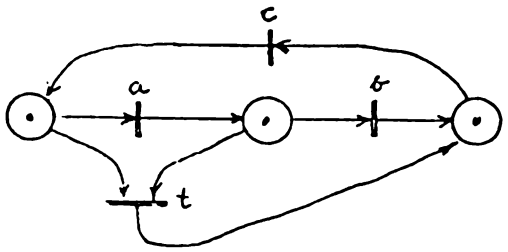
(1) Jedes m, das auf Q mit m^* übereinstimmt, ist tot in N;

(2) Zu jedem $p \in Q$ existiert eine Markierung m', die mit m^* auf der Menge $Q - \{p\}$ übereinstimmt, aber nicht tot in N ist.

Für viele Netzklassen sind hinreichende Bedingungen dafür bekannt, daß in Netzen aus diesen Klassen keine toten Markierungen erreichbar sind (z.B. die sog. Deadlock-Falle-Eigenschaft für gewöhnliche Netze).

Daß in einem System kein totaler Stillstand eintreten kann, d.h. im zugehörigen Petri-Netz keine tote Markierung erreichbar ist, bietet keine Garantie für die Lebendigkeit des Netzes. Im Beispiel der Abbildung 6 sind alle Transitionen bei der An-

fangsmarkierung konzessioniert, das Netz ist platzlebendig, die Transitionen a,b,c sind lebendig und t ist nicht lebendig. Bei dem hier beschriebenen System ist nach einer Aktivität von t der Anfangszustand nicht wieder herstellbar (ohne äußeren Eingriff), was auf der Netzebene bedeutet, daß das Netz nicht rücksetzbar (resetable) ist.



(Abb. 6)

DEFINITION. Das Petri-Netz N wird rücksetzbar genannt, wenn die Anfangsmarkierung m_0 von jeder erreichbaren Markierung aus erreichbar ist.

Man überlegt sich, daß ein rücksetzbares Netz, bei dem jede Transition bei der Anfangsmarkierung potentiell feuerbar ist, stets sogar lebendig ist. Die Rücksetzbarkeit des Netzes N bedeutet, daß sein Erreichbarkeitsgraph stark zusammenhängend ist, was bei beschränkten Netzen leicht zu testen ist.

DEFINITION. Zwei Transitionen t_1, t_2 heißen *nebenläufig* bei der Markierung m im Netz N, wenn beide bei m Konzession haben und beide simultan gefeuert werden können, d.h. wenn $t_1^- + t_2^- \leq m$ ist.

In dem in Abbildung 1 dargestellten Netz sind also x_1, x_2 bei der Anfangsmarkierung nebenläufig. Nebenläufigen Transitionen entsprechen Aktivitäten, die in gewissen Systemzuständen kausal unabhängig voneinander ausführbar sind. Folglich können, beim gegeben Zustand, nebenläufige Aktivitäten in beliebiger Reihenfolge stattfinden, d.h. eine zeitliche Anordnung braucht auf der vorliegenden Spezifikationsebene noch nicht vorgenommen zu werden. Umgekehrt sind jedoch Aktivitäten, die in beliebiger Reihenfolge ausgeführt werden können, nicht notwendig nebenläufig, sie

können sogar in einem Konflikt stehen, etwa wenn beide dieselbe Ressource benötigen.

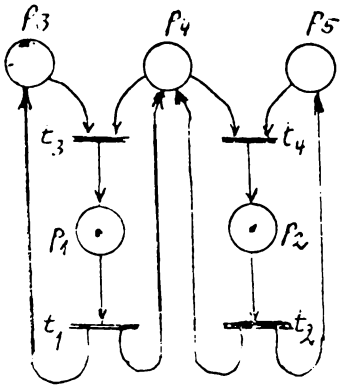
Das Erkennen und Modellieren (zeitweiliger) kausaler Unabhängigkeiten, d.h. von Nebenläufigkeit, beim Systementwurf bringt den Vorteil mit sich, daß man sich Entscheidungen über die zeitliche Anordnung von Aktivitäten, sofern sie überhaupt getroffen werden müssen, bis zur letzten Etappe vorbehalten und damit zur Optimierung nutzen kann.

DEFINITION. Wir sagen, daß Transitionen t_1, t_2 des Netzes N bei der Markierung m in einem dynamischen *Konflikt* stehen, wenn beide Transitionen bei m Konzession haben, aber wenigstens eine beim Feuern der anderen ihre Konzession verliert.

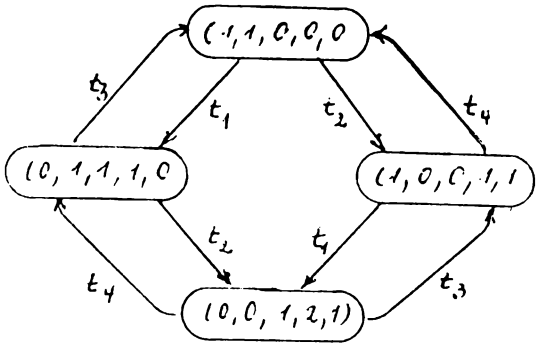
Im Beispielnetz von Abbildung 6 sind bei der Anfangsmarkierung die Transitionen a und t einerseits, sowie b und t andererseits in einem Konflikt, nicht aber a und b (a und b sind nebenläufig). Die Konfliktrelation ist also nicht transitiv.

Ein Konflikt in einem Netzmodell weist stets auf eine eventuelle Unzulänglichkeit bei der Systemspezifikation (Unterspezifikation) hin, auf eine Unbestimmtheit im Systemverhalten. Das kann auf der gegebenen Abstraktionsebene erwünscht bzw. zulässig bzw. sogar wesentlich sein, muß aber beachtet und kontrolliert werden, wenn ein System mit vorhersagbarem Verhalten entstehen soll. Überläßt man die Entscheidung der Konflikte dem Zufall, so kann es geschehen, daß gewisse Aktivitäten, denen im Netz sogar lebendige Transitionen entsprechen, durch einseitige Konfliktentscheidung blockiert werden. In unserem Philosophenbeispiel kann es z.B. vorkommen, daß ein Philosoph verhungert, weil die Konflikte um seine rechte bzw. linke Gabel stets zu seinen Ungunsten (unfair) entschieden werden.

Man spricht von einem *statischen Konflikt* zwischen zwei Transitionen, wenn diese einen Vorplatz gemeinsam haben. Das Vorliegen eines statischen Konflikts ist notwendig für das Auftreten eines dynamischen Konfliktes, aber nicht hinreichend, wie das Netz in der Abbildung 7 zeigt.



(Abb. 7)



(Abb. 8)

Eine andere Möglichkeit für eine wiederholbare Systemaktivität zu "verhungern" wird als 'Livelock' bezeichnet. Wenn es einen Kreis im Zustandsgraphen des Systems gibt, bei dessen Durchlaufen eine gewisse Aktivität nicht vorkommt, dann ist diese Aktivität aus dem Systemverhalten ausgeschlossen, solange das System in diesem zyklischen Verhalten verharrt, sie kommt also eventuell niemals zur Ausführung.

DEFINITION. Eine lebendige Transition t des Netzes N heißt im *Livelock* in bezug auf einen Kreis im Erreichbarkeitsgraphen von N, wenn t beim Durchlaufen dieses Kreises nicht gefeuert wird.

In unserem letzten Beispiel sind alle vier Transitionen lebendig. Der Erreichbarkeitsgraph dieses Netzes ist in der Abbildung 8 dargestellt. Offensichtlich existiert zu jeder Transition ein Kreis, in dem sie nicht vorkommt, d.h. für alle Systemaktivitäten besteht die Gefahr zu "verhungern".

Es sei $N = (P, T, F, V, m_0)$ ein Petri-Netz mit $P = \{p_1, \ldots, p_n\}$ und $T = \{t_1, \ldots, t_k\}$. Wir setzen für $t \in T$

$$\triangle t := t^+ - t^- .$$

Offenbar beschreibt $\triangle t$ die durch Feuern von t bewirkte Änderung der Markierung, d.h. wenn $t^- \leq m$ ist, so ist $\delta_N(m, t) = m + \triangle t$. Die Abbildungen $\triangle t : P \to Z$ schreiben wir als (ganzzahlige) Spaltenvektoren und bilden die *Inzidenzmatrix* C von N:

$$C = (c_{ij})_{n,k} := (\triangle t_1, ldots, \triangle t_k),$$

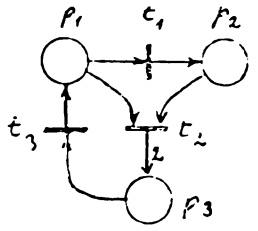
also ist $c_{ij} = \triangle t_j(p_i)$. Das Netz in der Abbildung 7 hat die Inzidenzmatrix

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

Man erkennt sofort, daß $\triangle t_1 + \triangle t_3 = 0$ ist, d.h. der Spaltenvektor y mit $y^T = (1,0,1,0)$ ist eine (nichtnegative) ganzzahlige Lösung des Gleichungssystems $C \cdot y = 0$.

DEFINITION. Die ganzzahligen Lösungen des Gleichungssystems $C \cdot y = 0$ werden *Transitions-Invarianten* (kurz: T-Invarianten) genannt. Das Netz N heißt *von T-Invarianten überdeckbar*, wenn es eine T-Invariante y mit $y(t) > 0$ für alle $t \in T$ gibt.

Jedes beschränkt und lebendig markierbare Petri-Netz ist mit T-Invarianten überdeckt (in unserem Beispiel leistet y mit $y^T = (1,1,1,1)$ das Verlangte), die Umkehrung gilt nicht, wie das Netz in der Abbildung 9 zeigt. Hier ist y mit $y^T = (1,1,2)$ eine



(Abb. 9)

T-Invariante, die N überdeckt, N besitzt aber keine lebendige Markierung. Offenbar wird aber die Markierung $m = (2,0,0)$ durch das Wort $u = t_1 t_2 t_3 t_3$ reproduziert, d.h. es ist $\delta_N(m,u) = m$, und für alle t ist $y(t)$ gleich der Zahl der Stellen in u, an denen t steht. Dieser Zusammenhang gilt allgemein, d.h. ist in einem beliebigen Petri-Netz $N\delta_N(m,u) = m$ und für alle t ist $y(t)$ die Anzahl der Stellen in u, die mit t besetzt sind, dann ist y eine T-Invariante. Neben notwendigen Bedingungen für die Lebendigkeit

liefert eine Untersuchung der T-Invarianten also auch Hinweise auf die Existenz reproduzierbarer Markierungen und damit auf eventuelle Kreise im Erreichbarkeitsgraphen.

DEFINITION. Die ganzzahligen Lösungen des Gleichungssystems $x \cdot C = 0^T$ werden *Platz-Invarianten (S-Invarianten)* genannt. Das Netz N heißt *von S-Invarianten überdeckbar*, wenn es eine S-Invariante x mit $x(p) > 0$ für alle $p \in P$ gibt.

In unserem Beispiel ist $x = (2,2,1,1,1)$ eine S-Invariante, die N überdeckt, weitere nichtnegative S-Invarianten sind $x_1 = (2,0,1,1,0)$ und $x_2 = (0,2,0,1,1)$.

Ist x eine nichtnegative S-Invariante des Petri-Netzes N, so gilt für alle $m \in R_N(m_0)$

$$x * m := \sum_{p \in P} x(p)m(p) = x * m_0,$$

folglich ist jedes von S-Invarianten überdeckbare Netz bei jeder Anfangsmarkierung beschränkt (dafür sagt man auch, es sei *strukturell* beschränkt). Da die Anfangsmarkierung in die Invariantenberechnung nicht eingeht, kann ein Netz beschränkt sein, ohne daß es mit S-Invarianten überdeckbar ist. Neben hinreichenden Bedingungen für die Beschränktheit des Netzes liefern S-Invarianten über ihre Interpretation durch Systemeigenschaften Informationen über Systeminvarianten, die bei der Verifikation des Entwurfs ausgenutzt werden können.

5.

Zur Vereinfachung der Modelle, insbesonders zur Komprimierung der Netze, werden häufig neben Petri-Netzen auch Netze höherer Typen, z.B. gefärbte Netze, Relationennetze, Prädikat-Transitionsnetze, bei der Modellierung angewendet. Allen diesen Netztypen ist gemeinsam, daß die Marken, die den aktuellen Zustand des Netzes bzw. des modellierten Systems anzeigen, nicht mehr ununterscheidbar sind, sondern Individualität besitzen. Alle Netze dieser Typen können aber im beschränkten Fall durch lokale Konstruktionen äquivalent in Petri-Netze überführt werden.

Während ein Platz in einem Petri-Netz als ein Zähler für Marken aufgefaßt werden kann, stellt ein Platz in einem 'high-level' Netz ein Prädikat variabler Extension dar, das auf jene Elemente aktuell zutrifft, mit denen der Platz gerade markiert ist.

Der Kürze halber werden wir hier nur auf einen der höheren Netztypen eingehen, die gefärbten Netze.

Ist X eine Menge, so wird jede Abbildung $b : X \to IN$ als *Multimenge* (oder auch als IN-wertiges Prädikat) über X bezeichnet. Die Zahl $b(x)$ können wir als Zahl des Vorkommens von x in b interpretieren. Nimmt b nur die Werte 0 und 1 an, so ist b die charakteristische Funktion einer gewönlichen Menge. Wir rechnen mit Multimengen wie mit Markierungen (elementweise), eine Markierung in einem Petri-Netz ist ja eine Multimenge über P. Wir notieren Multimengen auch als formale Summen: $b = \sum_{x \in X} b(x)x$. $2x + y$ ist also die Multimenge b mit $b(x) = 2, b(y) = 1$ und $b = 0$ sonst.

DEFINITION. $N = (P, T, F, C, V, m_0)$ ist ein *gefärbtes Netz*, wenn

(1) P und T endliche disjunkte Mengen sind,

(2) $F \leq (P \times T) \cup (T \times P)$ mit $\mathrm{dom}(F) \cup \mathrm{cod}(F) = P \cup T$ ist,

(3) C eine Abbildung ist, die jedem Element von $P \cup T$ eine nichtleere Menge zuordnet,

(4) V eine Abbildung ist, die jedem Bogen (p, t) bzw. (t, p) aus F eine Abbildung von $C(t)$ in die Menge aller Multimengen über $C(p)$ zuordnet, die nicht identisch 0 ist, und

(5) m_0 eine Abbildung ist, die jedem $p \in P$ eine Multimenge über $C(p)$ zuordnet.

Die Elemente von P bzw. T bzw. F sind die Plätze bzw. Transitionen bzw. Bögen des Netzes N.

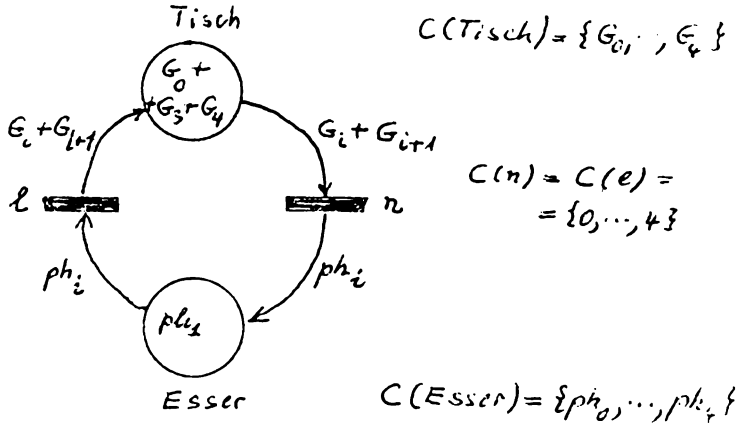
Für jeden Platz $p \in P$ ist $C(p)$ die Menge der Sorten (genannt *Farben*) von Individuen, mit denen der Platz p markiert sein kann. Eine Markierung m eines gefärbten Netzes ordnet jedem Platz p eine Multimenge über $C(p)$ zu, für $x \in C(p)$ ist $m(p)(x)$ die Zahl der Individuen der Farbe x, die sich bei m auf dem Platz p befinden. Individuen gleicher Farbe sind ununterscheidbar.

Für jede Transition t ist $C(t)$ die Menge der Arten (Farben), auf die t eventuell gefeuert werden kann. Die Transition t hat *Konzession* bei der Markierung m in der Farbe $c \in C(t)$, wenn für jeden Vorplatz p von t gilt: $V(p,t)(c) \leq M(p)$. Wenn t bei m in der Farbe c Konzession hat, so kann t mit c feuern, dabei entsteht die Markierung m' mit:

$$m'(p) := \begin{cases} m(p) + V(t,p)(c) - V(p,t)(c), & \text{falls } (t,p) \in F \\ & \& \ (p,t) \in F, \\ m(p) + V(t,p)(c), & \text{falls } (t,p) \in F \\ & \& \ (p,t) \notin F, \\ m(p) - V(p,t)(c), & \text{falls } (t,p) \notin F \\ & \& \ (p,t) \in F, \\ m(p), & \text{sonst.} \end{cases}$$

Petri-Netz sind demnach gefärbte Netze, bei denen auf jedem Platz nur eine Sorte von Marken möglich und jeder Transition nur eine Farbe zugeordnet ist.

Als Beispiel modellieren wir unsere Lösung des Philosophenproblems durch ein gefärbten Netz. Die Symmetrie des Problems ausnutzend fassen wir alle Plätze, die das Liegen einer Gabel auf dem Tisch beschreiben, zu einem Platz 'Tisch' zusammen, dessen Markierung durch Elemente der Menge $C(\text{Tisch}) := \{G_0, \ldots, G_4\}$ andeuten soll, welche der Gabeln G_i sich jeweils auf dem Tisch befinden. Dabei soll G_i die linke und G_{i+1} die rechte Gabel des Philosophen ph_i (modulo 5) sein. Analog verfahren wir mit den Plätzen, die die Systembedingung 'ph_i ißt' modellieren. Es entsteht das in der Abbildung 10 gezeigte Netz, dessen Umfang unabhängig von der Zahl der Philosophen ist.



(Abb. 10)

Bei der eingezeichneten Markierung liegen die Gabeln G_0, G_3 und G_4 auf dem Tisch, die Transition n hat also Konzession in den Farben 3 und 4. Ein Feuern von n mit $i = 3$ würde die Gabeln G_3 und G_4 vom Platz 'Tisch' entfernen ($m'(\text{Tisch}) = G_0$) und auf den Platz 'Esser' das Element ph_3 aufbringen ($m'(\text{Esser}) = ph_1 + ph_3$).

Ein gefärbtes Netz als Systemmodell kann also um vieles einfacher und komprimierter als das entsprechende Petri-Netz sein, ein Vorteil, den alle höheren Netztypen bieten. Insbesonders der Invariantenkalkül ist für höhere Netztypen soweit entwickelt, daß die Resultate zur Verifikation grundlegender Eigenschaften herangezogen werden können. Diese Vorteile bezahlt man damit, daß bestimmte System- bzw. Modelleigenschaften weit weniger deutlich hervortreten. Beispielsweise erscheinen jetzt Konflikte im System (z.B. zwischen den Philosophen ph_3 und ph_4) verdeckt als Konflikte nicht zwischen Transitionen, sondern zwischen Farben (hier 3 und 4) derselben Transition. Die Gefahr wird deutlich, wenn man sich vergegenwärtigt, daß jedes Petri-Netz $N = (P, T, F, V, m_0)$ durch ein gefärbtes Netz $\underline{N} = (\{\underline{p}\}, \{\underline{t}\}, \{(\underline{p}, \underline{t}), (\underline{t}, \underline{p})\}, \underline{C}, \underline{V}, \underline{m}_0)$ modelliert werden kann, das nur einen Platz und nur eine Transition besitzt. Man setzt nämlich

$$\underline{C}(\underline{p}) := P, \qquad \underline{C}(\underline{t}) := T,$$

$$\underline{V}(\underline{p}, \underline{t})(t) := t^-, \quad \underline{V}(\underline{t}, \underline{p})(t) := t^+, \quad \underline{m}_0(\underline{p}) := m_0.$$

Dem Feuern von t in N entspricht dabei das Feuern von $\underline{t}$ in der Farbe t, auch die Markierungen beider Netze entsprechen einander eineindeutig.

Überdies ändert sich das zu modellierende System durch die Art der Modellierung nicht, insbesondere nicht seine Zustandszahl. Aus einer zu großen Zustandszahl resultierende Schwierigkeiten bei der Verifikation grundlegender Eigenschaften können also durch einen Wechsel des Modells nur insoweit behoben werden, als die theoretischen Grundlagen für eine Verifikation ohne Zustandsaufzählung für die betreffende neue Modellklasse erarbeitet worden sind.

Ein wichtiger Vorzug der Netzmodellierung besteht darin, daß zahlreiche Programme bzw. Programmpakete zur rechnergestützten Analyse und Verifikation von Netzmodellen existieren (vgl. Lecture Notes in Computer Sci., Vol. 222, Seite 203-223). Mit dem in MODULA-2 geschriebenen Paket 'Petri-Netz-Maschine', das an der Humboldt-Universität zu Berlin entwickelt wurde, können z.B. alle hier erwähnten und weitere Netzeigenschaften analysiert, Petri-Netze konstruiert und reduziert, sowie ihre Invarianten berechnet werden. Auf diese Weise können Entwurfsfehler aufgedeckt und kann der Korrektheitsbeweis erleichtert werden, insbesondere dann, wenn die theoretischen Resultate nicht ausreichen, um ohne die Inspektion von Tausenden von Zuständen auszukommen.

6.

Über die Theorie und die Anwendung von Petri-Netzen existiert eine umfangreiche Literatur von verwirrender Vielfalt. Für eine erste Orientierung halten wir die folgenden Monographien bzw. Sammelbände für geeignet.

References

[1] Net Theory and Applications Lecture Notes in Computer Sci., Vol. 84 (1980).

[2] STARKE, P. H., Petri-Netze Deutscher Verlag der Wissenschaften, (polnische Übersetzung in Vorbereitung) Berlin 1980.

[3] PETERSON, J. L., Petri net Theory and the modelling of systems. Prentice-Hall, Englewood Cliffs, 1981.

[4] REISIG, W., Petrinetze Springer-Verlag, Berlin-Heidelberg-New York, 1982. (englische Übersetzung: Petri Nets, 1984)

[5] Application and Theory of Petri Nets Informatik-Fachberichte, Bd. 52 (1982), Springer-Verlag.

[6] Application and Theory of Petri Nets Informatik-Fachberichte, Bd. 66 (1983), Springer-Verlag.

[7]

[8] Advances in Petri Nets 1984 Lecture Notes in Computer Sci., Vol. 188 (1985)

[9] Advances in Petri Nets 1985 Lecture Notes in Computer Sci., Vol. 222 (1986).

PETER H. STARKE
Sektion Mathematik der Humboldt-Universität
DDR-1086 Berlin, PSF 1297.

DDR