Petri-Netze

Udo Kelter

16.03.2003

Zusammenfassung dieses Lehrmoduls

Petri-Netze sind Modelle für die Zustände eines Systems und die Übergänge zwischen diesen Zuständen. Petri-Netze eignen sich insbesondere für die Modellierung paralleler bzw. verteilter Systeme, die aus Komponenten bestehen, deren Zustand sich unabhängig weiterentwickelt. Dieses Lehrmodul stellt die Grundformen der Petri-Netze (Bedingungs-Ereignis-Netze und Stellen-Transitions-Netze) ausführlich vor und geht auf die Bildung, Interpretation und Analyse der Netze ein. Ferner stellen wir erweiterte Varianten (Prädikat-Transitions-Netze und hierarchische Netze) kurz vor.

Vorausgesetzte Lehrmodule:

obligatorisch:
- Vorgehensmodelle
- Zustandsübergangsdiagramme

Stoffumfang in Vorlesungsdoppelstunden: 1.0

Inhaltsverzeichnis

1 Motivation

Petri-Netze sind nach ihrem Erfinder, Carl Adam Petri, benannt. Sie sind ebenso wie Zustandsübergangsdiagramme Modelle, mit denen man die Zustände eines Systems und die Übergänge zwischen diesen Zuständen aufgrund äußerer Ereignisse modellieren kann. Petri-Netze eignen sich insbesondere für die Modellierung *paralleler bzw. verteilter* Systeme, die aus Komponenten bestehen, deren Zustand sich unabhängig weiterentwickelt.

Für derartige Modellierungsaufgaben sind Zustandsübergangsdiagramme weniger geeignet. Betrachten wir hierzu als Beispiel einen kleinen Teil einer automatisierten Fabrik, der aus drei Komponenten besteht (s. Bild 1):

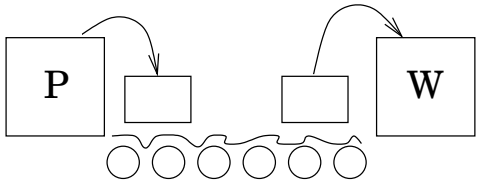

Abbildung 1: Transportband

1. einer Produktionseinheit, die Gegenstände eines bestimmten Typs produziert
2. einer Weiterverarbeitungseinheit, die jeweils einen der produzierten Gegenstände verbraucht
3. einem Förderband zwischen diesen beiden Einheiten, das die Gegenstände von der Produktionseinheit zur Weiterverarbeitungseinheit transportiert.

Die Produktionseinheit produziert jeweils einen Gegenstand und geht dann in einen Zustand über, in dem sie darauf wartet, daß auf dem Förderband Platz frei wird. Sobald dies der Fall ist, wird der produzierte Gegenstand auf das Band abgegeben, und es wird wieder produziert. Analog wartet die Weiterverarbeitungseinheit darauf, daß

ein Gegenstand auf dem Band vorliegt, dann nimmt sie ihn von dort und verarbeitet ihn. Das Förderband ist leider ziemlich kurz, es kann in unserem Beispiel maximal 2 Gegenstände aufnehmen; es hat daher 3 Zustände (0, 1, 2) gemäß der Zahl der vorhandenen Gegenstände. Die Steuerung der Bewegung des Bands betrachten wir hier nicht.

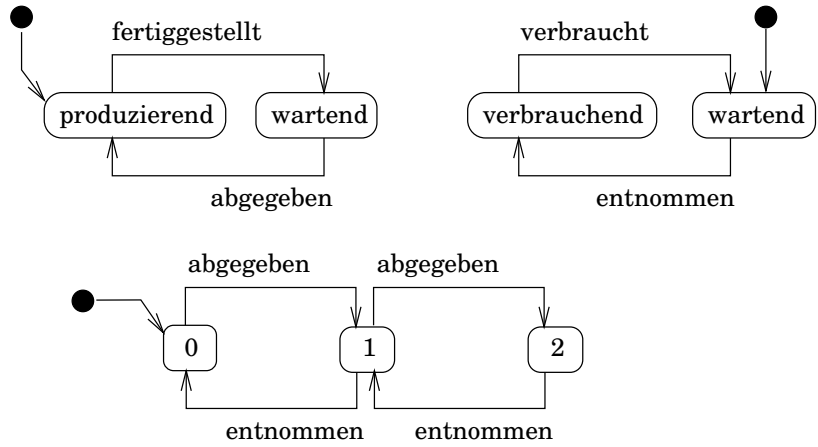

Abbildung 2: isolierte ZÜD

Bild 2 zeigt drei ZÜD, jeweils eines für jede Komponente. Man beachte, daß ein einzelnes Ereignis, z.B. die Abgabe eines Gegenstands auf das Förderband, in zwei Diagrammen zu einem Zustandsübergang führt. Die drei modellierten Systeme sind also nicht unabhängig voneinander. Die drei einzelnen Diagramme bilden insgesamt kein korrektes ZÜD, weil in einem ZÜD ein Knoten immer einen Gesamtzustand repräsentieren muß (und dementsprechend auch nur ein einziger Zustand als Anfangszustand markiert sein darf).

Ein globaler Zustand des Gesamtsystems läßt sich in unserem Beispiel am besten als ein Tripel darstellen, das die lokalen Zustände der Teilsysteme enthält. Wir erhalten insgesamt 12 Kombinationen. Bild 3 zeigt das resultierende ZÜD. Aus Platzgründen sind darin jeweils nur die Anfangsbuchstaben der Zustandsbezeichnungen verwendet ([pw]/[012]/[vw]).

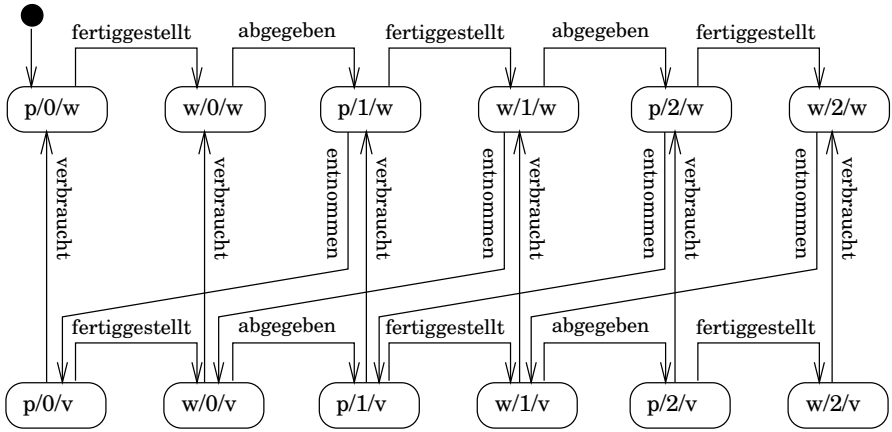

Abbildung 3: ZÜD mit zusammengesetzten Zuständen

Man erkennt schon bei diesem einfachen Beispiel unschwer die Probleme, die ZÜD bei Systemen mit verteiltem Zustand aufwerfen:

- Der Zustandsraum "explodiert" kombinatorisch.
- Gleichartige Zustandsübergänge müssen mehrfach eingezeichnet werden. So kann in unserem Beispiel die Produktionseinheit unabhängig davon, in welchem Zustand sich das Förderband oder die Weiterverarbeitungseinheit befinden, die Produktion eines Gegenstands beenden und in den Zustand 'wartend' übergehen. Der mit dem Ereignis 'fertiggestellt' verbundene Zustandsübergang muß daher 6 Mal (!) eingezeichnet werden.
- Die Übersichtlichkeit leidet infolge der vorstehenden Punkte erheblich.

2 Grundkonzepte

Petri-Netze haben im Gegensatz zu ZÜD prinzipiell einen strukturierten Zustandsraum, in dem "lokale" Zustandsübergänge möglich sind. Parallele Systeme sind daher sehr gut modellierbar.

Petri-Netze sind zum einen – analog zu endlichen Automaten – abstrakte mathematische Modelle, andererseits wurde von vorneherein eine graphische Notation eingeführt, die sich bis heute weitgehend erhalten hat.

Es sind diverse Varianten von Petri-Netzen vorgeschlagen und sehr intensiv untersucht worden. Wir stellen in diesem Abschnitt zunächst die Grundform vor.

2.1 Plätze und Transitionen

Ein **Petri-Netz** wird definiert durch

- eine endliche Menge **P** von **Plätzen** (oder synonym hierzu **Stellen**)
- endliche Menge **T** von **Transitionen**; jede Transition t hat:
  - eine Menge von **Eingangsplätzen** E_t
  - eine Menge von **Ausgangsplätzen** A_t, $A_t \cap E_t = \emptyset$ a

Graphisch dargestellt werden diese Strukturen wie folgt:

- Ein Platz wird durch einen Kreis dargestellt.
- Eine Transition wird durch einen kleinen schwarzen Balken dargestellt[1]. Von jedem Eingangsplatz der Transition führt ein Pfeil zu diesem Balken, von dem Balken aus führt ein Pfeil zu jedem Ausgangsplatz der Transition.

Bei der Modellierung werden Ereignisse durch Transitionen dargestellt. Die Eingangsplätze repräsentieren Bedingungen oder Ressourcen, die erfüllt oder vorhanden sein müssen, damit das Ereignis eintritt.

Bild 4 zeigt drei kleine Petri-Netze, die die drei Komponenten unseres Systems einzeln modellieren. Bild 5 zeigt ein einziges Petri-Netz, das das Gesamtsystem modelliert.

2.2 Markierungen

Auf einigen Plätzen in den Beispielen sind schwarze Punkte eingezeichnet; diese stellen den Zustand des Systems dar. Vergleichen kann

[1] Anstelle des Balkens wird vielfach auch ein Quadrat benutzt.

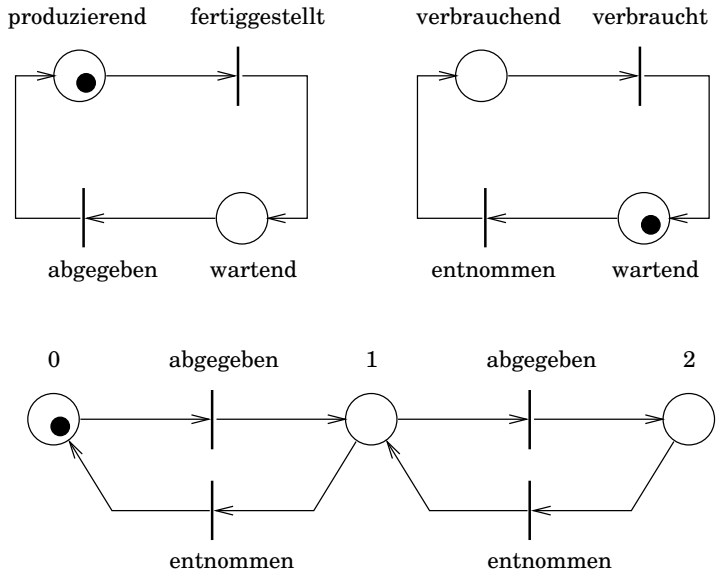

Abbildung 4: Beispiele für Petri-Netze

man dies mit Spielsteinen auf einem Brettspiel: Die Spielsteine stehen auf den Plätzen, bewegt werden sie infolge von Ereignissen über die Transitionen. Statt von Spielsteinen spricht man bei Petri-Netzen von **Marken**. Eine aktuell vorhandene Konfiguration von Spielsteinen bzw. Marken nennt man **Markierung**. Das Netz stellt also die statische Struktur des Systems dar, ein Ablauf innerhalb dieser Struktur entspricht einer Folge von Markierungen.

Zur Verdeutlichung vergleichen wir Petri-Netze mit ZÜD: In einem ZÜD wird der aktuelle Zustand überhaupt nicht graphisch dargestellt; man könnte ihn aber mit Hilfe einer Marke veranschaulichen. Bei einem ZÜD repräsentieren die Knoten die Zustände des *Gesamt*systems, daher könnte in einem ZÜD immer nur *genau eine* Marke, die den aktuellen Zustand markiert, vorhanden sein. Im Gegensatz dazu können in einem Petri-Netz *beliebig viele* Marken unterwegs sein, dies ist der entscheidende Unterschied zu einem ZÜD. Ein Platz in einem Petri-Netz modelliert daher auch nicht, im Gegensatz zu ZÜD, einen kompletten

Systemzustand.

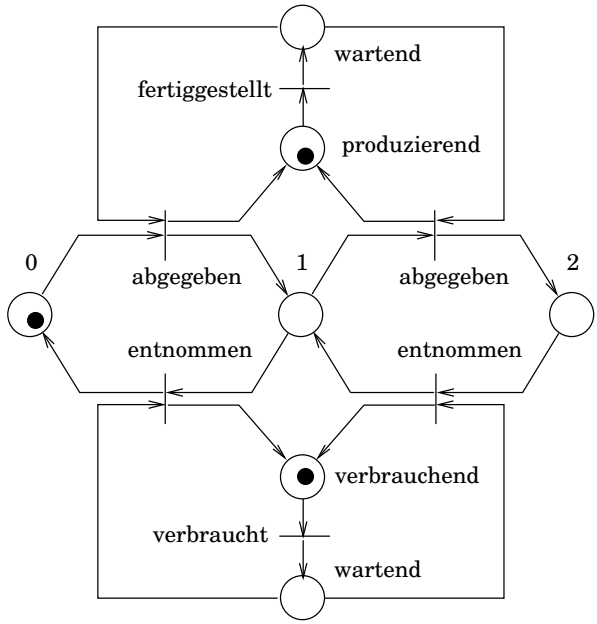

Abbildung 5: Petri-Netz für das Gesamtsystem

Die unterschiedlichen Varianten der Petri-Netze unterscheiden sich vor allem durch die zulässigen Markierungen:

- Bei einem **Booleschen Petri-Netz** (oder, synonym hierzu, **Bedingungs-Ereignis-Netz**) kann maximal eine Marke auf einem Platz stehen.
- Bei einem **Stellen-Transitions-Netz** (oder, synonym hierzu, **Nat-Netz**) können beliebig viele Marken auf einem Platz stehen.

Formal behandelt man einen Systemzustand als eine Abbildung, die jedem Platz die Zahl der dort liegenden Marken zuordnet. Diese Abbildung bezeichnet man auch als **Markierung**. Eine Markierung ist also eine Funktion

$$M : P \rightarrow Boolean$$

bei Booleschen Petri-Netzen und

$$M : P \to \mathbb{N}$$

bei Stellen-Transitions-Netzen. Dargestellt werden kann eine Markierung in den Diagrammen durch eine Zahl in jedem Platz oder eine entsprechende Anzahl von schwarzen Punkten.

3 Bedingungs-Ereignis-Netze

3.1 Zustandsübergänge

Ein Petri-Netz geht analog zu einem ZÜD durch das Schalten (oder "Feuern") einer Transition in einen neuen Zustand über. Während in einem ZÜD bildlich gesprochen nur an einer Stelle etwas passieren kann, nämlich an dem Symbol für den Zustand, in dem sich das System gerade befindet, können in einem Petri-Netz an verschiedenen Stellen Zustandsübergänge eintreten. Wir müssen deshalb begrifflich trennen zwischen der Schaltbereitschaft und dem eigentlichen Schalten:

Eine Transition t in einem Bedingungs-Ereignis-Netz ist bei einer Markierung M **schaltbereit** (oder **feuerbereit**), wenn

1. auf jedem Eingangsplatz $e \in E_t$ eine Marke vorhanden ist, also $M(e) = 1$
2. auf jedem Ausgangsplatz $a \in A_t$ keine Marke vorhanden ist, also $M(a) = 0$

In Bild 4 ist z.B. im Petri-Netz oben links die Transition 'fertiggestellt' schaltbereit, denn auf ihrem einzigen Eingangsplatz liegt eine Marke und der Ausgangsplatz ist leer.

Wenn eine schaltbereite Transition t **schaltet**, hat dies folgenden Effekt:

1. auf jedem Eingangsplatz $e \in E_t$ wird eine Marke weggenommen ("verbraucht"), also $M(e) := M(e) - 1$
2. auf jedem Ausgangsplatz $a \in A_t$ wird eine Marke hinzugefügt ("erzeugt"), also $M(a) := M(a) + 1$

Der Schaltvorgang ist atomar, also zeitlos.

Wenn im Petri-Netz oben links in Bild 4 die Transition 'fertiggestellt' schaltet, wird eine Marke vom Eingangsplatz 'produzierend' weggenommen und eine Marke auf dem Ausgangsplatz 'wartend' hinzugefügt.

Bild 6 zeigt zwei Transitionen, die mehr Ausgangsplätze als Eingangsplätze haben und die daher bei jedem Schalten die Zahl der vorhandenen Marken vergrößern. Transition t0 ist, weil ohne Eingangsplätze, ein Sonderfall, sie erzeugt beliebig viele Marken. Gleiches gilt aber auch für Transition t2, die sozusagen in einer Endlosschleife liegt. Analog hierzu kann man mit Transitionen, die weniger Ausgangsplätze als Eingangsplätze haben, die Zahl der vorhandenen Marken verkleinern.

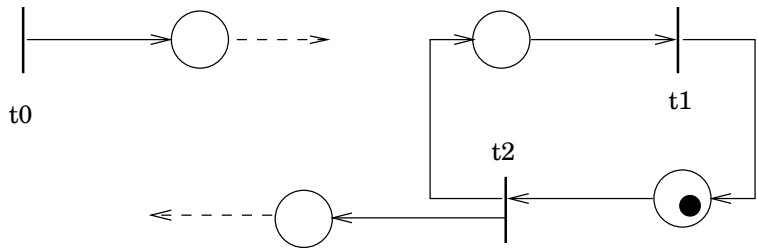

Abbildung 6: Beispiele für Petri-Netze

3.2 Transitionen und Ereignisse

In einem Petri-Netz können mehrere Transitionen zugleich schaltbereit sein. Wir betrachten hierzu noch einmal das Petri-Netz für das Förderband, allerdings in dem Zustand, wo nur der Platz '1' eine Marke enthält (s. Bild 7).

Offensichtlich kann jetzt entweder noch ein Gegenstand auf das Band gelegt werden oder einer entfernt werden; beides ist möglich. Schaltbereit ist dementsprechend sowohl die Transition 'abgegeben' oben rechts als auch die Transition 'entnommen' unten links.

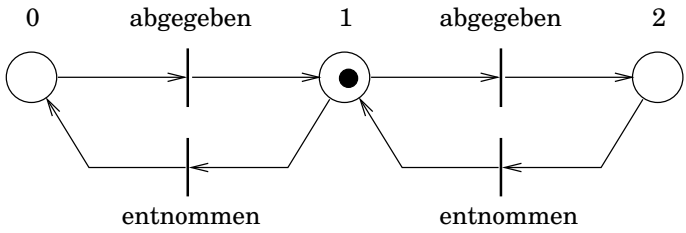

Abbildung 7: Konflikt zwischen Transitionen

Daß mehrere Transitionen schaltbereit sind, kann also – analog zu ZÜD – so interpretiert werden, daß in diesem Zustand mehrere *extern veranlaßte* Ereignisse zulässig sind und daß diejenige Transition schaltet, für die das zugehörige Ereignis zuerst eintritt. Das Petri-Netz spielt hier sozusagen eine passive Rolle.

Man kann aber umgekehrt dem Petri-Netz auch eine aktive Rolle zumessen, d.h. das Petri-Netz steuert irgendwelche externen Instanzen[2]. In diesem Fall unterstellt man, daß von mehreren schaltbereiten Transitionen eine, die schaltet, nichtdeterministisch ausgewählt wird.

Im ersten Fall, also wenn man Transitionen als von externen Ereignissen veranlaßt ansieht, steht man vor dem Problem, daß dem gleichen Ereignis verschiedene Transitionen zugeordnet sein können. Bild 7 enthält zwei Transitionen mit gleicher Beschriftung 'entnommen' – offensichtlich ist dies nicht der (eindeutige) Name der Transition, sondern das zugehörige Ereignis. Das Petri-Netz in Bild 7 hat die Eigenschaft, daß zu einem Ereignis immer höchstens eine der zugehörigen Transitionen schaltbereit ist, d.h. die Reaktion auf ein Ereignis ist eindeutig bestimmt. Man kann leicht Netze konstruieren, die diese Eigenschaft nicht haben. Bei solchen Netzen müßte eine der schaltbereiten Transitionen nichtdeterministisch ausgewählt werden; ein derartiges Netz ist daher wahrscheinlich falsch entworfen.

[2]Hierzu muß man den Transitionen bzw. Zuständen eine aufzurufende Aktion zuordnen; dieser Aspekt wird in den Grundformen der Petri-Netze nicht behandelt.

3.3 Konflikte und wechselseitiger Ausschluß

Wenn in Bild 7 die Transition unten links schaltet, ist die bisher schaltbereite Transition oben rechts anschließend nicht mehr schaltbereit; solche Transitionen stehen in **Konflikt** zueinander.

Dieser Effekt kann ganz bewußt dazu ausgenutzt werden, einen wechselseitigen Ausschluß in einem Petri-Netz zu realisieren. Beim wechselseitigen Ausschluß geht es darum, daß mehrere parallele Prozesse eine bestimmte Ressource benötigen, die immer nur nacheinander exklusiv von jeweils einem Prozeß benutzt werden kann. Ein Beispiel hierfür sind mehrere Personen, die ein Buch aus einer Bibliothek oder eine CD mit einer bestimmten Software wiederholt benötigen. Bild 8 zeigt eine Musterlösung für dieses Problem. Der Zyklus aus den Transitionen t1, t2 und t3 soll einen Prozeß modellieren, der die Ressource wiederholt eine Zeitlang (zwischen t2 und t3, im sog. kritischen Bereich) benötigt. Die Transitionen t4, t5 und t6 sollen analog einen anderen Prozeß modellieren, der um die gleiche Ressource konkurriert.

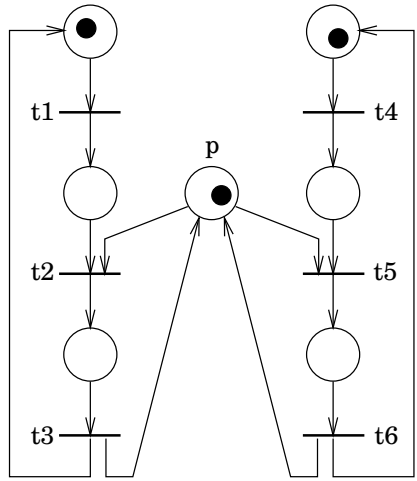

Abbildung 8: wechselseitiger Ausschluß

Der wechselseitige Ausschluß wird folgendermaßen erreicht: auf Platz p liegt initial eine Marke, sozusagen die “Eintrittskarte” in den

kritischen Bereich. Die Transition t2 benötigt diese "Eintrittskarte", um zu schalten. Wenn t2 geschaltet hat, schaltet später t3 und legt die "Eintrittskarte" wieder auf den Platz p zurück. Dort kann sie später von dem gleichen oder einem konkurrierenden Prozeß wieder entnommen werden.

Sofern die in Konflikt stehenden Transitionen t2 und t5 beide schaltbereit sind und t5 nicht zum Zuge kommt, würde man vielleicht erwarten, daß nach t3 und der erneuten Schaltbereitschaft von t5 nunmehr t5 bevorzugt wird, jedenfalls aber nicht der linke Prozeß den rechten überholt, indem ganz schnell t1 und wieder t2 schalten. Derartige Fairneß-Eigenschaften werden in Petri-Netzen nicht behandelt, d.h. die von dem Petri-Netz zugelassenen Schaltfolgen können beliebig unfair sein, der rechte Prozeß könnte also "verhungern". Erforderliche Fairneß-Eigenschaften müssen auf andere Weise spezifiziert werden.

4 Stellen-Transitions-Netze

Stellen-Transitions-Netze weisen gegenüber Bedingungs-Ereignis-Netzen zwei Erweiterungen auf:

- Stellen können nicht nur eine, sondern mehrere Marken enthalten. Individuell für jede Stelle kann eine Obergrenze für die Zahl der dort liegenden Marken angegeben werden; diese Zahl bezeichnet man als die **Kapazität** der Stelle. Notieren kann man sie z.B. in der Form "K = n" neben dem Platzsymbol.
- Jeder einzelne Ein- und Ausgangsplatz einer Transition bekommt ein **Gewicht** zugewiesen. Das Gewicht gibt an, wieviele Marken beim Schalten der Transition an diesem Platz verbraucht bzw. erzeugt werden. Dargestellt wird das Gewicht als Beschriftung an dem Pfeil, der den Platz und die Transition verbindet.

Kapazitäten können bei der Modellierung insb. dazu ausgenutzt werden, die Größe eines Lagers oder Puffers nachzubilden. Gewichte können den Verbrauch bzw. die Produktion von Ressourcen modellieren.

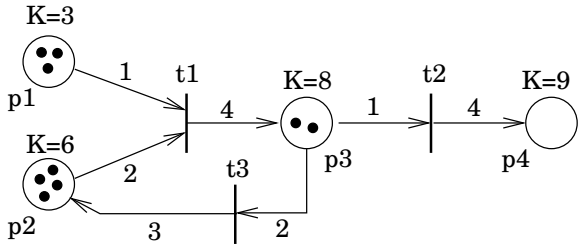

Abbildung 9: Beispiel für ein Stellen-Transitions-Netz

Die Definition der Schaltbereitschaft und des Effekts des Schaltens muß entsprechend angepaßt werden:

Wenn eine schaltbereite **Transition t schaltet**, hat dies folgenden Effekt:

1. auf jedem Eingangsplatz $e \in E_t$ werden $EG(t,e)$ Marken weggenommen, also $M(e) := M(e) - EG(t,e)$. **EG(t,e)** ist das Gewicht des Eingangsplatzes e für t.
2. auf jedem Ausgangsplatz $a \in A_t$ werden $AG(t,a)$ Marken hinzugefügt, also $M(a) := M(a) + AG(t,a)$. **AG(t,a)** ist das Gewicht des Ausgangsplatzes a für t.

Eine Transition t in einem Stellen-Transitions-Netz ist **schaltbereit**, wenn

1. auf jedem Eingangsplatz $e \in E_t$ (mindestens) soviele Marken vorhanden sind, wie das Eingabe-Gewicht EG(t,e) angibt, also $M(e) \geq EG(t,e)$
2. auf jedem Ausgangsplatz $a \in A_t$ genug Platz ist, um alle bei der Transition erzeugten Marken dort abzulegen, ohne die Kapazität des Platzes zu überschreiten. Die aktuell vorhandene Zahl von Marken zzgl. den bei der Transition erzeugten darf die Kapazität des Platzes nicht überschreiten, d.h. $M(a) + AG(t,a) \leq K(a)$.

Der Schaltvorgang ist auch hier atomar.

Bild 9 gibt ein Beispiel für ein Stellen-Transitions-Netz an. Im eingezeichneten Anfangszustand sind die Transitionen t1 und t2 schaltbereit. t3 ist nicht schaltbereit, weil die Kapazität des Ausgangsplatzes p2 überschritten werden würde.

Die folgende Tabelle zeigt einige Zustände an, die nach den angegebenen Schaltsequenzen erreicht werden.

Schaltsequenz (immer ausgehend vom Zustand in Bild 9)	Zahl der Marken in ..				danach
	p1	p2	p3	p4	schaltbereit
t1	2	2	6	0	t2, t3
t1;t3	2	5	4	0	t1, t2
t1;t3;t1	1	3	8	0	t2, t3
t1;t3;t1;t3	1	6	6	0	t2
t2	3	4	1	4	t1, t2
t2;t2	3	4	0	8	t1

5 Eigenschaften von Petri-Netzen

Eine Markierung M2 heißt **erreichbar** von einer Markierung M1 aus, wenn es ausgehend von M1 eine Folge von jeweils schaltbereiten Transitionen gibt, an deren Ende M2 entsteht. Oft interessiert man sich z.B. dafür, ob ausgehend von einem gegebenen Anfangszustand irgendwann ein Zustand erreicht wird, in dem ein bestimmter Platz eine Marke enthält oder eine bestimmte Transition schalten kann.

Im letzten Beispiel wird nach der Schaltsequenz t2;t2 ein Zustand erreicht, in dem die Transition t2 nie wieder schalten kann, in keinem noch erreichbaren Zustand wird t2 wieder schaltbereit; eine solche Transition nennt man **tot**.

Im Petri-Netz in Bild 9 sind irgendwann alle Transitionen tot. Wenn man das Petri-Netz als den Kontrollflußgraphen eines Algorithmus auffaßt, dann terminiert dieser Algorithmus. Ein Petri-Netz heißt **terminierend**, wenn ausgehend vom Startzustand nur endlich viele Transitionsschaltungen möglich sind.

Das Petri-Netz in Bild 8 ist nicht terminierend, die darin modellierten Prozesse sind zyklisch. Alle Transitionen in diesem Netz sind

lebendig, d.h. sie können immer wieder schalten. Die Lebendigkeit einer Transition kann bedeuten, daß in der realen Welt ein bestimmtes Ereignis immer wieder zulässig wird, obwohl es zwischendurch nicht zulässig ist. Beispielweise muß bei einer verkehrsgesteuerten Ampelanlage jede Ampel irgendwann wieder auf grün schalten. Bei einem Aufzug muß irgendwann jedes Stockwerk, in dem die Kabine angefordert worden ist, erreicht werden und die Kabinentür aufgehen. Wenn bei der Aufzugsteuerung irgendwie ein Zustand erreicht werden kann, bei dem die Transition, die dem Öffnen der Kabinentür entspricht, tot ist, liegt offenbar ein Konstruktionsfehler vor.

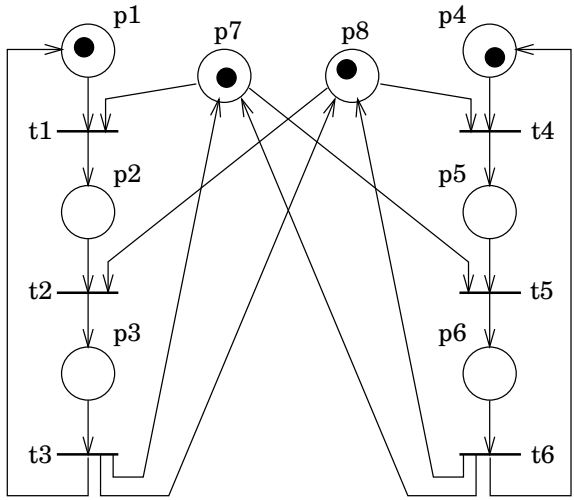

Abbildung 10: Deadlockgefährdetes Petri-Netz

Unerwünschte Totzustände sind insb. im Zusammenhang mit dem wechselseitigen Ausschluß möglich. Als Beispiel betrachten wir das Petri-Netz in Bild 10. Gegenüber dem früheren Beispiel in Bild 8 liegen hier zwei Ressourcen R1 und R2 vor, deren Verfügbarkeit durch eine Marke auf den Plätzen p7 bzw. p8 modelliert wird. Der linke Prozeß fordert zuerst R1, danach zusätzlich R2 an. Der rechte Prozeß fordert die Ressourcen in der umgekehrten Reihenfolge an. Solange beide

Anforderungen eines Prozesses hintereinander erfüllt werden, können die Prozesse beliebig viele Zyklen durchlaufen. Wenn allerdings R1 dem linken Prozeß zugeteilt wird und dann R2 dem rechten, entsprechend der Schaltfolge t1;t4, dann sind anschließend t2 und t5 beide tot und es liegt ein **Deadlock** (bzw. eine **Verklemmung**) vor. Das Deadlock-Problem wird im Rahmen von Vorlesungen über Betriebssysteme detaillierter behandelt. In unserem Kontext bleibt festzustellen, daß Petri-Netze Modelle sind, mit denen sich derartige unerwünschte Phänomene exakt beschreiben lassen.

Bei Stellen-Transitions-Netzen ist weiter interessant, wie hoch die Zahl der Marken an einem Platz werden kann. Ein Stellen-Transitions-Netz heißt **k-beschränkt**, wenn bei jeder erreichbaren Markierung M für alle p aus P: $M(p) \leq k$ gilt.

In vielen Anwendungsfällen muß sichergestellt werden, daß bestimmte Totzustände, Verklemmungen oder Kapazitätsüberschreitungen nicht auftreten. Ideal wäre es daher, wenn man ein Petri-Netz mit einem Prüfprogramm automatisch auf erwünschte bzw. unerwünschte Eigenschaften hin untersuchen könnte. Leider geht dies für die meisten interessierenden Eigenschaften nicht generell (vgl. [Re85]). Nichtsdestotrotz kann man oft im Einzelfall von Hand einen "Beweis" führen, daß eine gewünschte Eigenschaft vorhanden ist.

Eine Alternative zu analytischen (automatisierten oder manuellen) Untersuchungen sind Simulationen. Hierbei gibt ein Simulator eine zufällige Folge von externen Ereignissen vor (anders gesehen wählt er unter den schaltbereiten Transitionen eine aus) und simuliert das Verhalten des Netzes, wobei dann laufend die interessierenden Eigenschaften überprüft und z.B. Durchschnittwerte über die Zahl der Marken auf bestimmten Plätzen gebildet werden. Simulatoren können, ähnlich wie beim Testen von Programmen, indessen nur die Anwesenheit von Fehlern zeigen, nicht hingegen deren Abwesenheit.

6 Weitere Spezialformen

Die bisher vorgestellten Petri-Netze haben sehr einfache und klare Strukturen; dies ist vorteilhaft für die Verstehbarkeit und Erlernbarkeit, nachteilig aber für komplexere praktische Probleme, weil entweder die resultierenden Netze zu groß werden oder bestimmte Phänomene überhaupt nicht modelliert werden können. In diesem Abschnitt skizzieren wir daher einige Erweiterungen der Grundformen.

6.1 Prädikat-Transitions-Netze (PrT-Netze)

Transitionen in Stellen-Transitions-Netzen kann man auffassen als Funktionen, die Eingabedaten in Ausgabedaten transformieren, wobei die Typen der Datenobjekte und die Transformationsfunktionen sehr einfach sind. Die Grundidee von **Prädikat-Transitions-Netzen** besteht darin, diese Restriktion aufzuheben und Datenobjekte beliebigen Typs zuzulassen. Das Schaltverhalten einer Transition muß dementsprechend abhängig vom Wert der eingegebenen Marken definiert werden; für jeden Ausgangsplatz berechnet die Transition den Wert neuer dort abgelegter Marken. Prädikat-Transitions-Netze haben daher folgende Merkmale:

- Als Marken sind Werte mit beliebigen benutzerdefinierten Typen zulässig.
- Auf einem Platz können beliebig viele Marken liegen.
- Wenn eine Transition k Eingangsplätze hat und auf diesen Plätzen jeweils eine oder mehrere Marken liegen, dann bildet jedes k-Tupel, das je eine Marke von jedem Eingangsplatz enthält, ein **Eingabetupel** $(m_1, ..., m_k)$
- Wann bzw. mit welchen Eingabetupeln eine Transition t schaltbereit ist, wird durch ein Prädikat P_t bestimmt: sofern $P_t(m_1, ..., m_k) = true$ ist, nennt man $(m_1, ..., m_k)$ ein **schaltbares Tupel**. Eine Transition heißt schaltbereit, wenn wenigstens ein schaltbares Tupel vorliegt.

  Es kann mehrere schaltbare Tupel für eine Transition geben, ferner können mehrere Transition schaltbereit sein. Unter allen schalt-

baren Tupeln aller schaltbereiten Transitionen wird eines nichtdeterministisch ausgewählt, das schaltet.
- Wenn Transition t mit Eingabetupel $(m_1, ..., m_k)$ schaltet, wird
  - für alle i, $1 \leq i \leq k$, wird m_i vom jeweiligen Platz entfernt
  - für jeden Ausgangsplatz p ein Wert $A_{t,p}(m_1, ..., m_k)$ berechnet und dort abgelegt.

Prädikat-Transitions-Netze erlauben es, z.B. Vorgangsketten in Betrieben oder sogar bei der Softwareentwicklung, bei denen Dokumente von verschiedenen Instanzen bearbeitet werden, relativ leicht zu modellieren.

6.2 Hierarchische Petri-Netze

Analog zu hierarchischen ZÜD liegt es auch bei Petri-Netzen nahe, komplexe Systeme auf verschiedenen Abstraktionsebenen zu modellieren und das gesamte Modell als eine Hierarchie von Diagrammen zu strukturieren, in der einzelne Diagrammkomponenten durch ein eigenes Netz verfeinert werden. Prinzipiell können sowohl Plätze als auch Transitionen verfeinert werden. Da Transitionen die eigentlichen dynamischen Elemente sind und Petri-Netze gerade Ablaufstrukturen modellieren, ist eine Verfeinerung der Transitionen vorzuziehen.

Die Netze auf oberen Ebenen einer Verfeinerungshierarchie bezeichnet man auch als **Kanal-Instanzen-Netze**: Die Transitionen sind die aktiven Instanzen, die Datenobjekte verarbeiten; die Plätze bzw. Kanäle sind passive Systemteile, die Datenobjekte speichern bzw. letztlich zwischen den aktiven Instanzen transportieren.

Literatur

[Re85] Reisig, W.: Systementwurf mit Petri-Netzen; Springer-Verlag; 1985

Glossar

Bedingungs-Ereignis-Netz: Synonym zu Boolesches Petri-Netz

Kanal-Instanzen-Netz: Bezeichnung für Petri-Netze auf oberen Ebenen einer Verfeinerungshierarchie von Petri-Netzen

Konflikt: Ein Konflikt zwischen mehreren schaltbereiten Transitionen liegt vor, wenn diese einen gemeinsamen Eingangsplatz (oder Ausgangsplatz) haben und dann, wenn eine der Transitionen schaltet, die anderen ihre Schaltbereitschaft verlieren

Marke (*token*): liegt auf einem Platz in einem Petri-Netz und stellt einen lokalen Zustand dar

Markierung: Konfiguration von Marken in einem Petri-Netz

Nat-Netz: Synonym zu Stellen-Transitions-Netz

Petri-Netz: formales mathematisches Modell für ein verteiltes, paralleles System mit lokalen Zustandsübergängen; besteht aus einem Graphen, dessen Knoten Plätze oder Transitionen sind; es gibt mehrere, unterschiedlich komplexe Varianten

Petri-Netz, Boolesches: Petri-Netz, bei dem maximal eine Marke auf einem Platz stehen kann

Petri-Netz, hierarchisches: Petri-Netz, bei dem Plätze oder Transitionen durch ein Petri-Netz verfeinert werden

Petri-Netz, terminierendes: Petri-Netz, in dem alle Abläufe endlich sind, also immer nach endlich vielen Schaltungen alle Transitionen tot sind

Platz oder **Stelle:** Art von Knoten in einem Petri-Netz; nimmt Marken auf, die den aktuellen Zustand darstellen

Prädikat-Transitions-Netz: verallgemeinerte Variante von Petri-Netzen, die mit beliebigen Datenobjekten als Marken arbeitet

Stellen-Transitions-Netz: Petri-Netz, bei dem mehrere Marken auf einem Platz stehen können

Transition: Art von Knoten in einem Petri-Netz, der mit Eingangs- und Ausgangsplätzen verbunden ist; Transitionen können schalten; beim Schalten verbrauchen sie Marken auf den Eingangsplätzen und erzeugen neue Marken auf den Ausgangsplätzen

Transition, lebendige: Transition, die bei einer gegebenen Markierung immer wieder schaltbereit wird

Verklemmung (*deadlock*): eine Verklemmung ist eine zyklische Wartesituation; in einem Petri-Netz mit einer bestimmten Markierung bedeutet dies, daß zwei oder mehr Transitionen nicht schaltbereit werden, weil sie zum Schalten Marken benötigen, die nur von anderen wartenden Transitionen erzeugt werden können.

Index