

Prof. Dr. Jürgen Dassow
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik

P E T R I — N E T Z E

Vorlesungsskript

Magdeburg, Oktober 2008 – Januar 2009

Vorwort

Petri-Netze gehören zu den meist benutzten Werkzeugen in der Informatik. Sie werden zur Modellierung in einer Vielzahl von Bereichen mit zum Teil sehr unterschiedlicher Ausprägung angewendet. Sie werden in der Informatik z.B. zur Simulation von „klassischen“ Berechnungen, als formales Modell zur Beschreibung von Kommunikationsprotokollen, fehlertoleranten Systemen und Betriebssystemen eingesetzt, finden aber auch in der Wirtschaftsinformatik Verwendung bei der Beschreibung von Geschäftsprozessen und anderen organisatorischen Abläufen. In einigen Fällen werden Petri-Netze schon zu den Beschreibungssprachen gerechnet. Durch die Vielfalt der Anwendungen und deren Bedeutung sollte daher jeder Informatiker, also auch jeder Studierende der Informatik oder informatiknaher Fächer gewisse Grundkenntnisse über Petri-Netze besitzen. Daher scheint es mir sehr wichtig, dass Vorlesungen über Petri-Netze angeboten werden.

Es gibt zwei Aspekte bei Petri-Netzen. Der eine betrifft den Entwurf von Petri-Netzen, der andere die Analyse von Petri-Netzen. Beim zweitgenannten Aspekt geht es darum, festzustellen, ob ein gegebenes Petri-Netz gewisse Eigenschaften hat oder nicht. Dieses ist wichtig, da man stets zu prüfen hat, ob ein entworfenes Netz die Anforderungen erfüllt, die an den Entwurf gestellt wurden. Zwei Methoden stehen zum Testen zur Verfügung. Man kann Experimente durchführen und überprüfen, ob bei diesen das Resultat mit dem der Praxis übereinstimmt, oder man verwendet Algorithmen, mittels deren das Vorhandensein oder Fehlen gewisser Eigenschaften festgestellt werden kann. Der zweite Ansatz hat den Vorteil, dass er ein gesichertes endgültiges Resultat liefert, während man bei Experimenten auch bei der korrekten Widerspiegelung in allen Test nicht abschließend weiß, ob das Netz die Eigenschaft hat.

In dieser Vorlesung, die der Vertiefungsrichtung Theoretische Informatik zugerechnet wird, werden wir im Wesentlichen – neben den erforderlichen Grundbegriffen der Theorie der Petri-Netze – die algorithmische Analyse von Petri-Netzen behandeln. Abschließend werden einige Resultate über Petri-Netz-Sprachen behandelt.

Die Vorlesung ist für Studierende der Informatik und aller informatiknahen Fachrichtungen geeignet. Vorausgesetzt werden – bis auf den letzten Teil zu Sprachen – nur Grundkenntnisse der Mathematik (insbesondere Graphentheorie und lineare Algebra) und einige Algorithmen, wie sie in jeder Vorlesung zu Algorithmen und Datenstrukturen vermittelt werden. Im letzten Teil sind Basiskonzepte der Theoretischen Informatik (formale Sprachen und Automaten) erforderlich, die aber auch in anderen Teilen an einigen Stellen hilfreich sind.

Definitionen, Sätze und andere Aussagen, Beispiele usw. werden einheitlich kapitelweise durchnummeriert; daher folgt auf Definition x.y das Beispiel x.y+1 und dann der

Satz $x.y+2$ und darauf Folgerung $x.y+3$ usw.. Dadurch soll die Anzahl der Suchräume klein gehalten werden. Nur Abbildungen und Gleichungen werden gesondert nummeriert. Durch \square wird das Ende eines Beweises angezeigt. Wird auf einen Beweis verzichtet, so steht \square bereits am Ende der Formulierung der Aussage.

Magdeburg/Lostau, im Herbst 2008/Winter 2009

Jürgen Dassow

Inhaltsverzeichnis

Vorwort	1
1 Einführende Beispiele und Bemerkungen	5
2 Netzgraphen	11
Literaturverzeichnis	21

Kapitel 1

Einführende Beispiele und Bemerkungen

Bevor wir eine Theorie der Petri-Netze in systematischer Weise entwickeln, wollen wir einige Beispiele geben und intuitiv erklären. Dabei sollen zum einen einige Anwendungsbereiche von Petri-Netzen vorgestellt werden, zum anderen wollen wir einige Erläuterungen geben, die in den folgenden Kapiteln als formale und präzise Definitionen erneut auftauchen, dann aber durch diese Betrachtungen motiviert sind.

Wir beginnen mit dem Netz aus Abbildung 1.1, durch das die Werte der Booleschen Funktionen $(x \wedge y)$ und $(x \vee y)$ bei Eingabe von x und y berechnet werden. Dies geschieht auf folgende Weise: Die zu Beginn auf den Stellen x und y (oben) liegenden Marken \bullet werden entsprechend den Pfeilen weitergeschoben. Ein Schieben „durch“ ein Quadrat ist nur dann möglich, wenn alle Kreise, von denen ein Pfeil zum Quadrat geht, mit einer Marke \bullet versehen ist. Außerdem wird bei Abarbeitung eines Quadrats jeder Kreis, zu dem ein Pfeil vom Quadrat ausgeht, mit einer Marke belegt. Können zwei Quadrate aktiv werden (die Kreise, von denen Pfeile zu den Quadraten führen, sind alle markiert) und gibt es einen Kreis, von dem zu beide Quadraten eine Kante führt, so wird nichtdeterministisch eines der Quadrate gewählt und zum Verschieben der Marken genutzt.

Zu Beginn liegen die beiden Marken in den mit x und y bezeichneten Kreisen in der ersten Reihe. Wir wählen jetzt sowohl für x als auch y eine Belegung mit 0 oder 1, d.h. wir schieben die Marken durch das mit $x = 0$ bzw. mit $x = 1$ bezeichnete Quadrat und entsprechend für y . Man rechnet nun leicht nach, dass

- im ersten Kreis der vierten Reihe genau dann eine Marke liegt, wenn $(x \wedge y)$ den Wert 1 annimmt,
- im zweiten Kreis der vierten Reihe genau dann eine Marke liegt, wenn $(x \wedge y)$ den Wert 0 annimmt,
- im dritten Kreis der vierten Reihe genau dann eine Marke liegt, wenn $(x \vee y)$ den Wert 1 annimmt,
- im vierten Kreis der vierten Reihe genau dann eine Marke liegt, wenn $(x \vee y)$ den Wert 0 annimmt.

Damit sind die Werte von $(x \wedge y)$ und $(x \vee y)$ berechnet. Die restlichen Knoten erlauben, dass die Marken wieder nach oben geschoben werden und damit für weitere Belegungen die Werte ausgerechnet werden können.

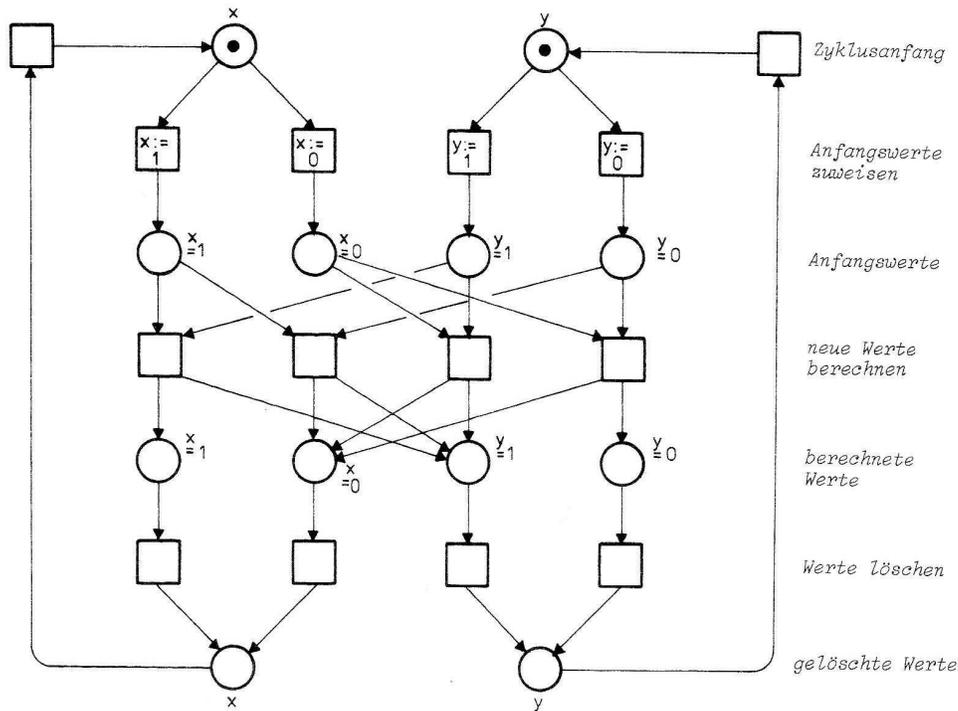


Abbildung 1.1: Petri-Netz zur Berechnung von $x \wedge y$ und $x \vee y$

Als nächstes geben wir in Abbildung 1.2 zwei Netze an, mittels derer die Summe zweier natürlicher Zahlen x und y berechnet wird, indem schrittweise x um 1 verkleinert und y um 1 vergrößert wird.

Die beiden Netze entsprechen einer Umsetzung eines Programms und unterscheiden sich nur in der Reihenfolge der Vergrößerung und Verkleinerung. Wegen der Umsetzung des Programms sind diese beiden Operationen sequentiell auszuführen. Dies ist aber nicht zwingend im Netz erforderlich. Hier könnte eine Parallelität gestattet werden, wie dies in Abbildung 1.3 geschehen ist. Es ist aber anzumerken, dass es sich nicht wirklich um Parallelität handelt, denn die Addition von 1 bzw. Subtraktion von 1 können in einer beliebigen Reihenfolge hintereinander oder gleichzeitig oder zeitlich leicht versetzt (d.h. nicht hintereinander, aber auch nicht gleichzeitig) erfolgen. Es ist nur notwendig beide Operationen auszuführen, bevor der nächste Zyklus eingeleitet wird. Eine derartige Situation wird nebenläufig genannt und ist ein typisches Merkmal von Petri-Netzen, dass durch viele andere Modellierungen nicht erfasst werden kann.

Ein anderes Phänomen, dass durch Petri-Netze modelliert werden kann, sind Konflikte. Diesen Aspekt haben wir schon bei der Berechnung der Booleschen Funktionen gesehen. Dort konnten von den Quadraten in der zweiten Zeile nur das erste oder zweite bzw. nur das dritte oder vierte benutzt werden. Bei Konflikten kann also von mehreren möglichen Quadraten nur eines benutzt werden.

Ein Konflikt wird bei der Modellierung eines Ausschnitts eines Betriebssystems in Abbildung 1.4 genutzt, um zu verhindern, dass gewisse Prozesse nicht gleichzeitig ablaufen können. Auf der linken Seite befinden sich zwei Kreise aus jeweils sechs Knoten. Jeder

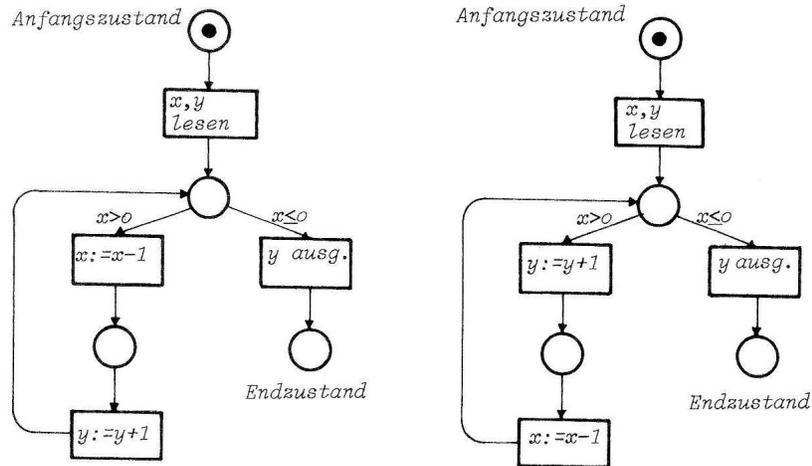


Abbildung 1.2: Petri-Netz zur sequentiellen Berechnung der Addition zweier Zahlen

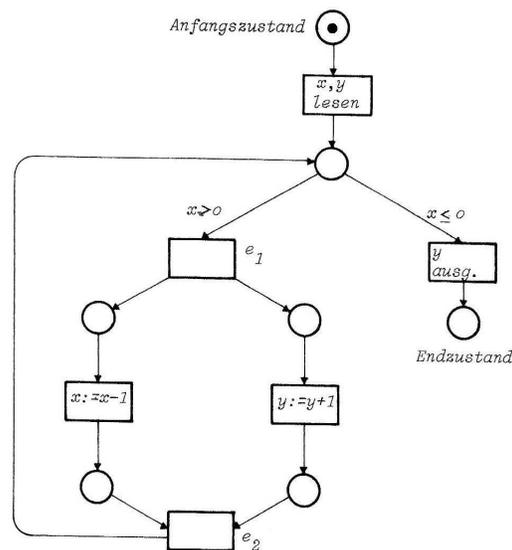


Abbildung 1.3: Petri-Netz zur nebenläufigen Berechnung der Addition zweier Zahlen

dieser Kreise beschreibt einen Schreibprozess. Die drei Kreise aus sechs Knoten auf der rechten Seite beschreiben drei Leseprozesse. Offensichtlich kann nur erlaubt werden, dass höchstens ein Prozess schreibt; auch die Anzahl gleichzeitiger Lesezugriffe ist in der Regel beschränkt; in unserem Beispiel sollen höchst zwei Prozesse gleichzeitig lesen. Der in der Mitte befindliche Knoten dient als Steuerelement. Er trägt zu Beginn zwei Marken. Da entsprechend der Gewichtung 2 für die Kanten vom Steuerelement zu den Schreibprozessen jeder Schreibprozess nur aktiv werden kann, wenn er beide Marken für sich nimmt, kann stets nur ein Prozess schreiben und beim Schreiben eines Prozesse kann nicht gelesen

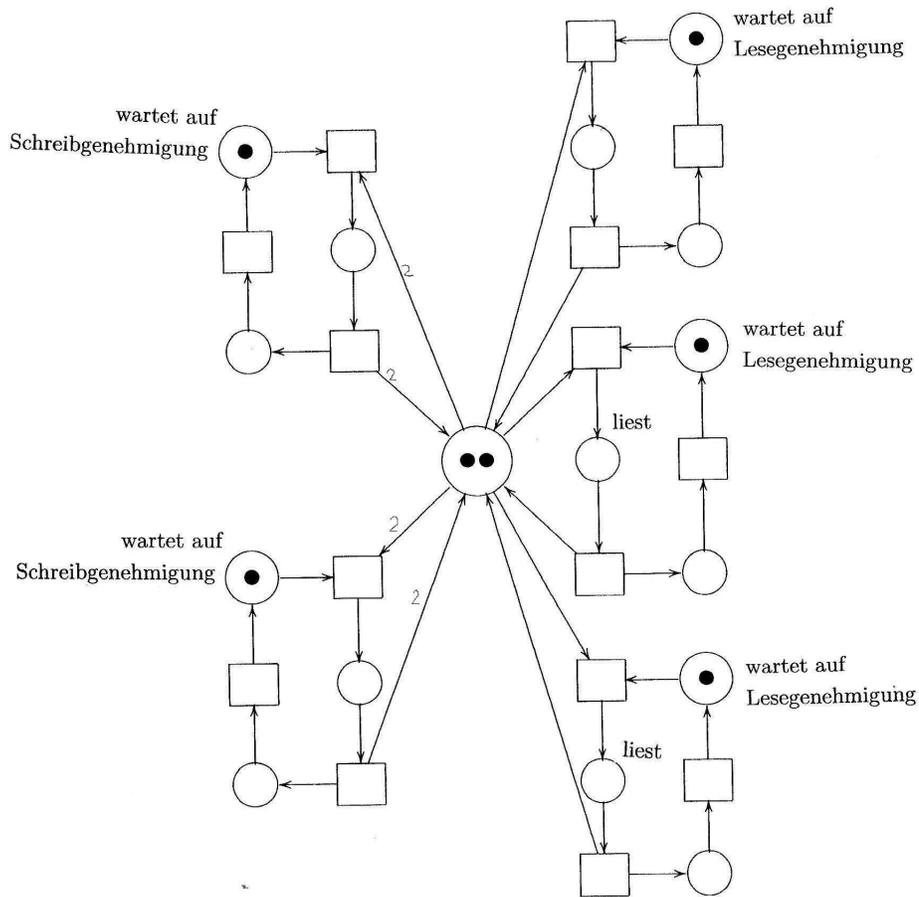


Abbildung 1.4: Petri-Netz eine Teils des Betriebssystems

werden. Nach dem Schreiben werden die beiden Marken wieder ans Steuerelement zurückgegeben. Falls ein Leseprozess aktiv wird, so entnimmt er dem Steuerelement genau eine Marke (die nach dem Lesen zurückgelegt wird). Damit kann kein Schreibprozess aktiv werden und höchstens ein zweiter Leseprozess, der die noch verbliebene Marke des Steuerelements nutzt. Damit sind unsere Forderungen für die Gleichzeitigkeit von Prozessen alle erfüllt.

Als nächstes Beispiel geben wir die „fünf dinierenden Philosophen“. Das Netz ist in Abbildung 1.5 angegeben. Die Philosophen sitzen an einem runden Tisch und können essen, wofür sie für jede Hand eine Gabel benötigen, oder denken. Der Tisch ist nun so gedeckt, dass jeder Philosoph zwar seinen eigenen Teller hat, aber zwischen zwei Tellern jeweils nur eine Gabel liegt. Daher können die Nachbarn eines essenden Philosophen stets nur denken, da ihnen mindestens eine Gabel zum Essen fehlt. Falls alle Philosophen denken, so steht jeder mit seinen Nachbarn hinsichtlich des Essens in einem Konflikt.

Abschließend wollen wir das Petri-Netz aus Abbildung 1.6 betrachten, das die Prozesse in einer Bibliothek beschreibt. Hierbei verwenden wir im Gegensatz zu den bisherigen Netzen verschiedenen Marken benutzen, die Bestellzetteln, Karteikarten und Büchern entsprechen. Falls jemand eine Bestellung an der Bestelltheke abgibt, gibt es zwei Möglichkei-

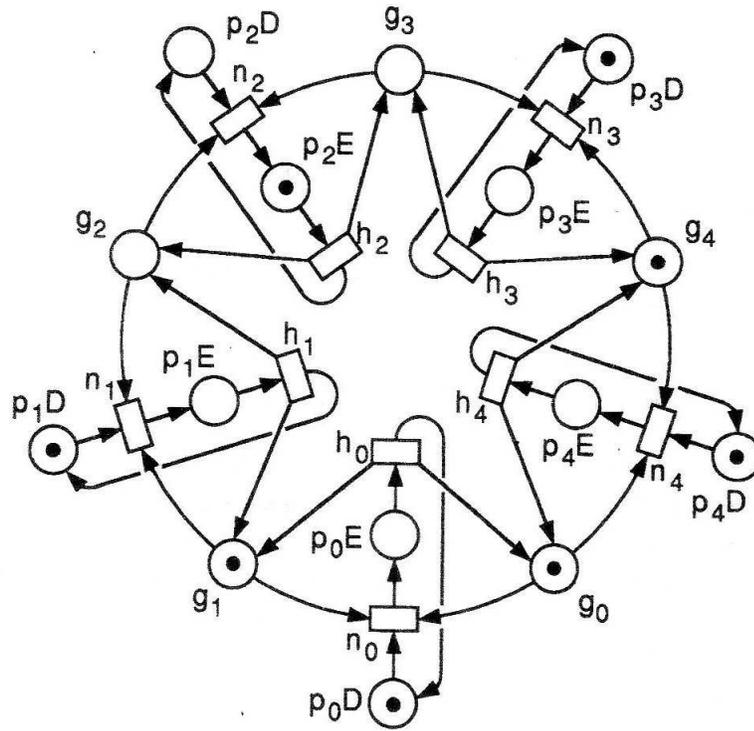


Abbildung 1.5: Petri-Netz der dinierenden Philosophen

ten. Zum einen kann das Buch ausgeliehen sein, d.h. die Karteikarte des Buches befindet sich in der Kartei der ausgeliehenen Bücher; damit kann nur das obere Rechteck aktiviert werden, wodurch das Buch ausgeliehen bleibt (Karteikarte wird wieder in die Kartei der ausgeliehenen Bücher gelegt) und an der Abholtheke eine Nachricht „Buch ist entliehen“ hinterlegt wird. Zum anderen kann das Buch im Lager sein (und die Karteikarte ist im Buch), womit nur das zweite Rechteck von oben aktiviert ist; nun wird das Buch zur Abholtheke gebracht und seine Karteikarte geht in die Kartei der ausgeliehenen Bücher. Man sieht leicht, dass auch der Rückgabeprozess im Netz modelliert ist.

Das Netz aus Abbildung 1.7 gibt eine weitere Abstraktion der Prozesse in der Bibliothek.

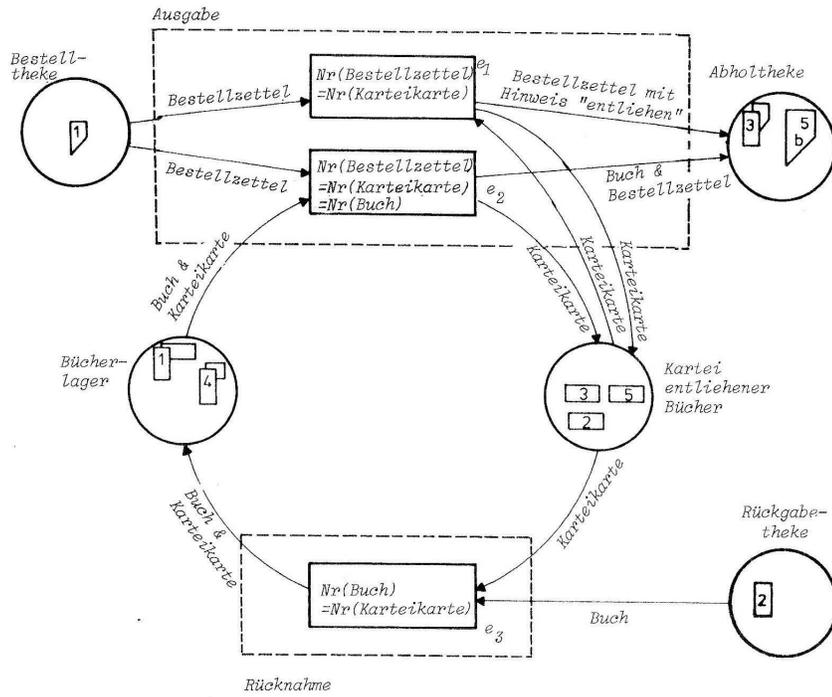


Abbildung 1.6: Modell einer Bibliothek

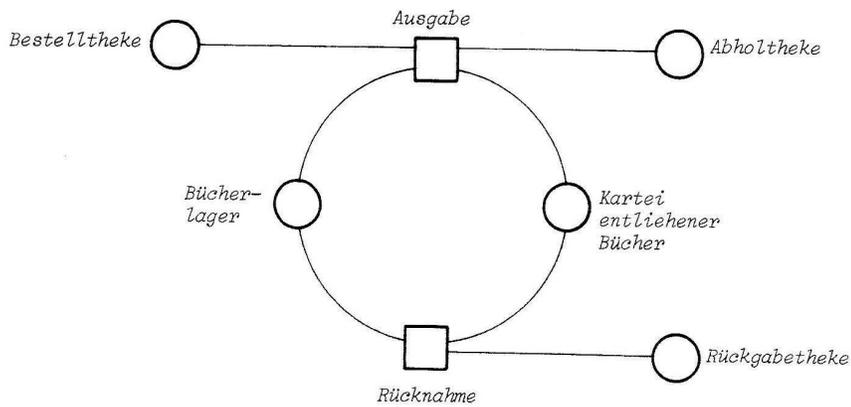


Abbildung 1.7: Vereinfachtes Petri-Netz für die Prozesse in einer Bibliothek

Kapitel 2

Netzgraphen

Wie wir in den einführenden Bemerkungen gesehen haben, sind Petri-Netze ein Art von Graphen, die durch die Veränderung der Position der Marken Prozesse beschreiben können. In diesem Abschnitt wollen wir einige Begriffe und spezielle Klassen von Graphen einführen, die zwar für die Theorie der Petri-Netze bedeutsam sind, aber eigentlich nur die graphentheoretische Seite betreffen.

Definition 2.1 Ein Netzgraph ist ein Tripel $N = (S, T, F)$, wobei

- S und T endliche nichtleere Mengen mit $S \cap T = \emptyset$ sind, und
- F eine Teilmenge von $(S \times T) \cup (T \times S)$ ist.

Die Elemente aus S nennen wir *Stellen* oder *Plätze*. Die Elemente aus T werden *Transitionen* genannt. Die Elemente aus F sind Kanten.

Wir werden manchmal anstelle von Netzgraphen auch einfach nur von Netzen sprechen.

Zur Darstellung eines Netzgraphen verwenden wir die üblichen Darstellung von Graphen, wobei wir zwei unterschiedliche Arten von Knoten verwenden; die Stellen werden als Kreise und die Transitionen als Vierecke angegeben.

Beispiel 2.2 Wir betrachten die Netze

$$N_1 = (S_1, T_1, F_1) \quad \text{und} \quad N_2 = (S_2, T_2, F_2)$$

mit

$$\begin{aligned} S_1 &= \{s_1, s_2, s_3\}, & T_1 &= \{t_1, t_2\}, \\ F_1 &= \{(s_1, t_1), (s_1, t_2), (s_2, t_1), (s_2, t_2), (t_1, s_3), (t_2, s_3)\}, \\ S_2 &= \{s'_1, s'_2, s'_3, s'_4\}, & T_2 &= \{t'_0, t'_1, t'_2\}, \\ F_2 &= \{(s'_1, t'_1), (s'_1, t'_2), (s'_2, t'_2), (s'_3, t'_0), (s'_4, t'_2), (t'_0, s'_1), (t'_1, s'_3), (t'_2, s'_4)\}. \end{aligned}$$

In Abbildung 2.1 sind die Netzgraphen N_1 und N_2 graphisch dargestellt.

Man sieht, dass die graphische Darstellung eines Netzgraphen $N = (S, T, F)$ erheblich übersichtlicher ist als die Darstellung durch Angabe der Mengen S , T und F . Wir werden daher zukünftig meist nur die graphische Darstellung angeben.

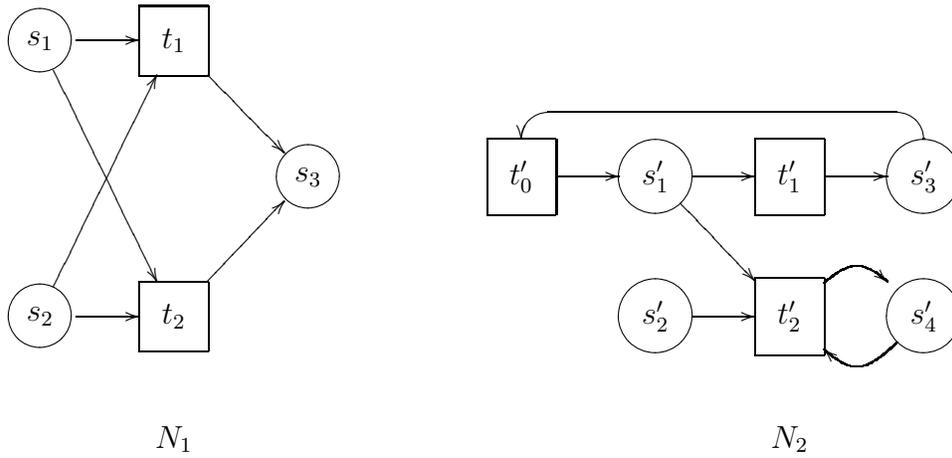


Abbildung 2.1: Zwei Beispiele für Netzgraphen

Aus graphentheoretischer Sicht ist ein Netzgraph ein paarer (oder bipartiter) Graph, denn diese Graphen sind dadurch definiert, dass ihre Knotenmenge in zwei disjunkte Teilmengen V_1 und V_2 zerlegt werden kann und keine Kanten zwischen zwei Knoten aus V_1 und keine Kanten zwischen zwei Knoten aus V_2 existieren. Durch die Mengen S und T eines Netzgraphen $N = (S, T, F)$ ist diese Zerlegung gerade gegeben.

Definition 2.3 Für ein Element $x \in S \cup T$ sind der Vorbereich von x bzw. der Nachbereich von x durch

$$\bullet x = \{y \mid (y, x) \in F\} \quad \text{und} \quad x \bullet = \{y \mid (x, y) \in F\}$$

definiert.

Offensichtlich gilt für $x \in S$, dass der Vorbereich und Nachbereich von x in T enthalten ist, während Vorbereich und Nachbereich von Elementen $z \in T$ Teilmengen von S sind.

Für die Netzgraphen N_1 und N_2 aus Beispiel 2.2 ergibt sich unter anderem

$$\begin{aligned} \bullet s_1 &= \emptyset, \quad s_1 \bullet = \{t_1, t_2\}, \quad \bullet t_1 = \{s_1, s_2\}, \quad t_2 \bullet = \{s_3\}, \\ \bullet s'_1 &= \{t'_0\}, \quad s'_2 \bullet = \{t'_2\}, \quad \bullet t'_2 = \{s'_2, s'_4\}, \quad t'_2 \bullet = \{s'_4\} \end{aligned}$$

Definition 2.4 Ein Netzgraph $N = (S, T, F)$ heißt schlicht, wenn keine zwei verschiedenen Knoten den gleichen Vor- und den gleichen Nachbereich haben.

In einem schlichten Netzgraphen gilt also für alle x und y aus $S \cup T$, dass aus $\bullet x = \bullet y$ und $x \bullet = y \bullet$ die Beziehung $x = y$ folgt.

Der Netzgraph N_1 aus Beispiel 2.2 ist nicht schlicht, denn es gelten $\bullet s_1 = \bullet s_2$ und $s_1 \bullet = s_2 \bullet$ für die zwei verschiedenen Stellen s_1 und s_2 . Der Netzgraph N_2 aus Beispiel 2.2 ist schlicht.

Definition 2.5 i) Ein Netzgraph $N' = (S', T', F')$ ist ein Teilnetzgraph des Netzgraphen $N = (S, T, F)$ falls $S' \subseteq S$, $T' \subseteq T$ und $F' \subseteq F$ gelten.

ii) Der Teilnetzgraph $N' = (S', T', F')$ des Netzgraphen $N = (S, T, F)$ heißt induzierter Teilnetzgraph von N , falls $F' = F \cap ((S' \times T') \cup (T' \times S'))$ gilt.

Bei einem Teilnetzgraphen sind sowohl die Knoten- als auch Kantenmenge eine Teilmenge der entsprechenden Mengen des Netzgraphen N . Beim induzierten Teilnetzgraph sind alle Kanten von N , die Knoten aus S' und T' miteinander verbinden auch in N' vorhanden.

Definition 2.6 Eine Schlinge in einem Netzgraphen $N = (S, T, F)$ ist eine Teilnetzgraph $N' = (\{s\}, \{t\}, \{(s, t), (t, s)\})$ von N .

Der Netzgraph N_1 aus Beispiel 2.2 enthält keine Schlinge, während N_2 aus Beispiel 2.2 eine Schlinge enthält, die aus der Stelle s'_4 und der Transition t'_2 gebildet wird.

Wir betrachten nun die Netzgraphen N_3 und N_4 aus Abbildung 2.2. Auf den ersten

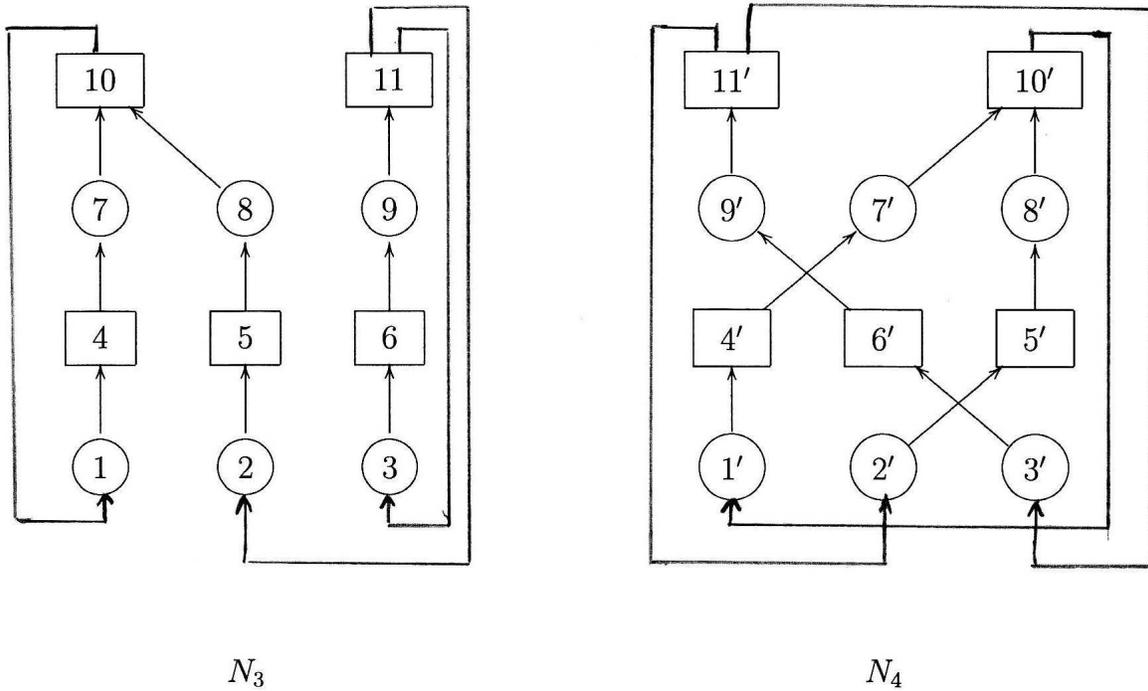


Abbildung 2.2: Zwei isomorphe Netzgraphen

Blick sehen die Netzgraphen sehr verschieden aus, denn bei einem kreuzen sich die Kanten und bei dem anderen nicht. Nimmt man dagegen die Beschreibung beider Graphen durch Mengen, so stellt man fest, dass sich beide Beschreibungen nur dadurch unterscheiden, dass die Knotennamen bei N_3 die Form x haben, während bei N_4 die gestrichelte Variante x' verwendet wird. Wir wollen dies hier nur dadurch dokumentieren, dass wir die Vor- und Nachbereiche von 1, 5, 11, 1', 5' und 11' angeben:

- 1 = {10}, •1' = {10'},
- 1• = {4}, 1'• = {4'},
- 5 = {2}, •5' = {2'},
- 5• = {8}, 5'• = {8'},
- 11 = {9}, •11' = {9'},
- 11• = {2, 3}, 11'• = {2', 3'}.

Somit sind die beiden Netzgraphen bis auf die Bezeichnungen identisch. Daher ist eigentlich nicht nötig, diese beiden Graphen zu unterscheiden. Wir präzisieren diesen Ansatz durch folgende Definition.

Definition 2.7 Eine Abbildung $\varphi : (S \cup T) \rightarrow (S' \cup T')$ heißt Isomorphismus des Netzes $N = (S, T, F)$ auf das Netz $N' = (S', T', F')$, falls folgende Bedingungen erfüllt sind:

- φ ist eine bijektive Abbildung von $S \cup T$ auf $S' \cup T'$ (φ ist eineindeutig und surjektiv),
- $\varphi(S) \subseteq S'$ und $\varphi(T) \subseteq T'$ (φ ist stellen- und transitionserhaltend),
- $(x, y) \in F$ gilt genau dann, wenn $(\varphi(x), \varphi(y)) \in F'$ gültig ist.

Wir sagen dann auch, dass N isomorph zu N' ist.

Es ist leicht zu sehen, dass die zweite Bedingung wegen der Bijektivität und Surjektivität von φ zu $\varphi(S) = S'$ und $\varphi(T) = T'$ verschärft werden kann.

Offensichtlich ist die Abbildung φ , die durch $\varphi(x) = x'$ für $x \in \{1, 2, \dots, 10, 11\}$ gegeben ist, ein Isomorphismus von N_3 auf N_4 .

In der Algebra spricht man von einem Isomorphismus zwischen zwei Strukturen, wenn die Abbildung bijektiv, surjektiv und operationserhaltend ist. Die dritte Bedingung in Definition 2.7 besagt, dass im Fall der Netzgraphen φ relationserhaltend sein soll.

Die folgende Aussagen sind einfach zu beweisen (die Beweise können analog zu denen für entsprechende Aussagen in der Algebra geführt werden).

Lemma 2.8 *i) Die identische Abbildung $\varphi(x) = x$ der Knotenmenge $S \cup T$ eines Netzgraphen $N = (S, T, F)$ auf sich selbst ist ein Isomorphismus von N auf sich selbst.*

ii) Die Umkehrabbildung eines Isomorphismus φ von N auf N' ist ein Isomorphismus von N' auf N .

iii) Die Komposition (Hintereinanderausführung) $\varphi_1 \circ \varphi_2$ der Isomorphismen φ_1 von N auf N' und φ_2 von N' auf N'' ist ein Isomorphismus von N auf N'' .

iv) Die Isomorphie zwischen Netzgraphen ist eine Äquivalenzrelation.

Wegen Lemma 2.8 iv) können wir die Menge der Netzgraphen in Äquivalenzklassen isomorpher Netzgraphen einteilen. Im Folgenden werden wir Netzgraphen der gleichen Äquivalenzklasse nicht unterscheiden und stets eine für unsere Belange passende Repräsentation angeben.

Wir werden jetzt einige spezielle Klassen von Netzgraphen einführen und die Beziehung zwischen diesen Graphklassen untersuchen.

Definition 2.9 Ein Netzgraph $N = (S, T, F)$ heißt Zyklus, wenn es eine natürliche Zahl $n \in \mathbf{N}$ und Elemente $s_1, s_2, \dots, s_n, t_1, t_2, \dots, t_n$ derart gibt, dass

$$\begin{aligned} S &= \{s_1, s_2, \dots, s_n\}, & T &= \{t_1, t_2, \dots, t_n\} \\ F &= \{(s_i, t_i) \mid 1 \leq i \leq n\} \cup \{(t_i, s_{i+1}) \mid 1 \leq i \leq n-1\} \cup \{(t_n, s_1)\} \end{aligned}$$

gelten.

Genauer hätte es heißen müssen, dass es einen zu N isomorphen Netzgraphen mit den angegebenen Mengen gibt; wegen der Identifizierung isomorpher Netzgraphen kann die Definition wie oben gegeben werden.

Die graphische Repräsentation eines Zyklus ist durch einen Netzgraphen der Form aus Abbildung 2.3 gegeben.

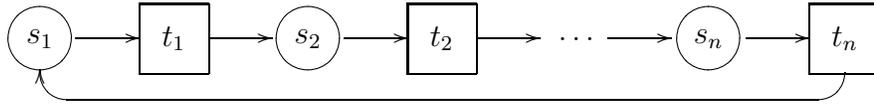


Abbildung 2.3: Form eines Zyklus

Definition 2.10 i) Ein Netzgraph heißt Zustandsmaschine, wenn

$$|\bullet t| = |t \bullet| = 1 \text{ für alle } t \in T$$

gilt.

ii) Ein Netzgraph heißt Synchronisationsgraph, wenn

$$|\bullet s| = |s \bullet| = 1 \text{ für alle } s \in S$$

gilt.

iii) Ein Netzgraph heißt verallgemeinerte Zustandsmaschine, wenn

$$|\bullet t| \leq 1 \text{ und } |t \bullet| \leq 1 \text{ für alle } t \in T$$

gilt.

iv) Ein Netzgraph heißt verallgemeinerter Synchronisationsgraph, wenn

$$|\bullet s| \leq \text{ und } |s \bullet| \leq 1 \text{ für alle } s \in S$$

gilt.

v) Ein verallgemeinerter Synchronisationsgraph, der keinen Teilgraph enthält, der ein Zyklus ist, heißt Kausalnetz.

v) Ein Netzgraph heißt Free-Choice-Netz, wenn für alle $s \in S$ und $t \in T$ gilt, dass

$$(s, t) \in F \cap (S \times T) \text{ impliziert } (s \bullet = \{t\} \vee \bullet t = \{s\}).$$

vi) Ein Netzgraph heißt erweitertes Free-Choice-Netz, wenn für alle $t_1 \in T$ und $t_2 \in T$ gilt, dass

$$\bullet t_1 \cap \bullet t_2 \neq \emptyset \text{ impliziert } \bullet t_1 = \bullet t_2.$$

Wir bezeichnen mit $\mathcal{G}(Z)$, $\mathcal{G}(ZM)$, $\mathcal{G}(VZM)$, $\mathcal{G}(SG)$, $\mathcal{G}(VSG)$, $\mathcal{G}(Kausal)$, $\mathcal{G}(FNC)$, und $\mathcal{G}(EFCN)$ die Mengen der Zyklen, bzw. Zustandsmaschinen, bzw. verallgemeinerten Zustandsmaschinen, bzw. Synchronisationsgraphen, bzw. verallgemeinerten Synchronisationsgraphen, bzw. Free-Choice-Netze, bzw. erweiterten Free-Choice-Netze.

Für $X \in \{ZM, VZM, SG, VSG, Kausal, FNC, EFCN\}$ sind in der Abbildung 2.4 Netzgraphen N_X mit $N_X \in \mathcal{G}(X)$ angegeben.

Wir geben nun äquivalente Charakterisierungen für einige der in Definition 2.10 eingeführten Klassen an.

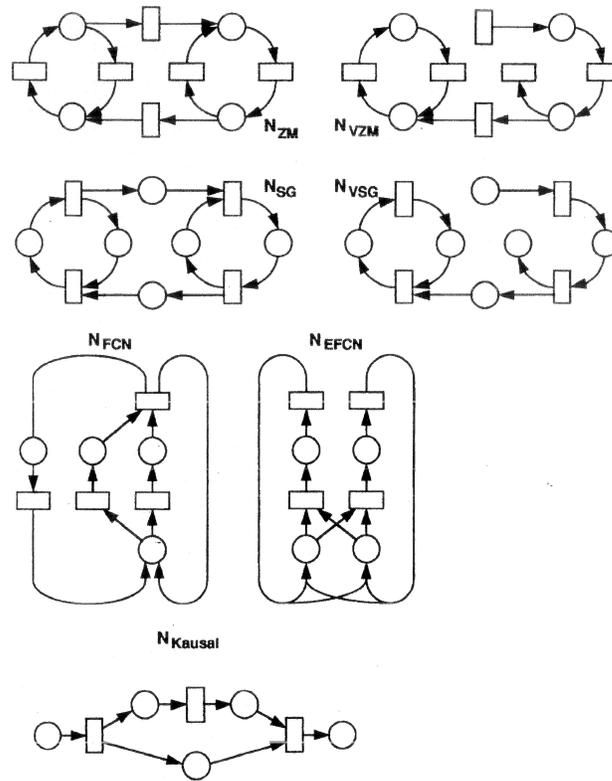
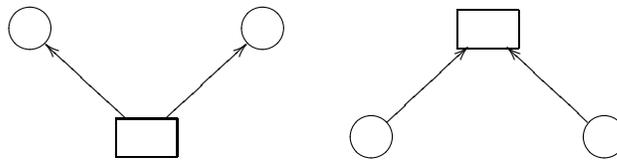


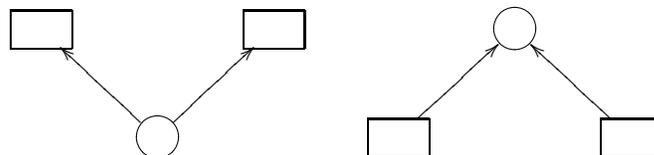
Abbildung 2.4: Vertreter für die Klassen spezieller Netzgraphen

Lemma 2.11 *i) Ein Netzgraph ist genau dann eine Zustandsmaschine, wenn er keinen Netzgraphen der Form*



als Teilnetzgraphen enthält.

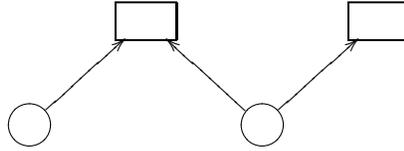
ii) Ein Netzgraph ist genau dann ein Synchronisationsgraph, wenn er keinen Netzgraphen der Form



als Teilnetzgraphen enthält.

iii) Ein Netzgraph ist genau dann ein Free-Choice-Netz, wenn er keinen Netzgraphen

der Form



als Teilnetzgraphen enthält.

iv) Ein Netzgraph ist genau dann ein erweitertes Free-Choice-Netz, wenn er bei Beschränkung auf Kanten aus $S \times T$ keinen induzierten Netzgraphen der Form aus iii) enthält.

Beweis. Die Aussagen i), ii) und iii) enthalten im Wesentlichen nur eine graphische Umsetzung der Definition der Klasse und werden daher ohne Beweis angegeben.

iv) Es sei N kein erweitertes Free-Choice-Netz. Nach Definition ist dies damit gleichwertig, dass es Stellen s und s' und Transitionen t_1 und t_2 derart gibt, dass $s \in \bullet t_1$, $s \in \bullet t_2$ (d.h. $\bullet t_1 \cap \bullet t_2 \neq \emptyset$), $s' \in \bullet t_1$ und $s' \notin \bullet t_2$ (d.h. $\bullet t_1 \neq \bullet t_2$) gelten. Der induzierte Teilnetzgraph mit der Stellenmenge $\{s, s'\}$ und der Transitionsmenge $\{t_1, t_2\}$ – bei Beschränkung auf Kanten aus $S \times T$ – liefert gerade den in iii) gezeigten Graphen.¹

Es sei N ein erweitertes Free-Choice-Netz. Wenn N einen induzierten Teilnetzgraphen der Form aus iii) enthält, so gibt es offenbar zwei Transitionen t_1 und t_2 mit $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ und $\bullet t_1 \neq \bullet t_2$, womit wir einen Widerspruch zur Definition zur Voraussetzung, dass N ein erweitertes Free-Choice-Netz ist, erhalten. \square

Die zweite Gruppe von Aussagen charakterisiert die Free-Choice-Netze und erweiterten Free-Choice-Netze in Analogie zu der Definition der beiden anderen Typen von Choice-Netzen. In allen Fällen ist dann die Voraussetzung der Implikation gleich; es wird ein nichtleerer Durchschnitt der Nachbereiche zweier Stellen gefordert.

Lemma 2.12 i) Ein Netzgraph $N = (S, T, F)$ ist genau dann ein Free-Choice-Netz, wenn für alle $s_1, s_2 \in S$

$$s_1 \bullet \cap s_2 \bullet \neq \emptyset \text{ impliziert } (s_1 \bullet = s_2 \bullet = \{t\} \text{ für ein } t \in T). \quad (2.1)$$

ii) Ein Netzgraph $N = (S, T, F)$ ist genau dann ein erweitertes Free-Choice-Netz, wenn für alle $s_1, s_2 \in S$

$$s_1 \bullet \cap s_2 \bullet \neq \emptyset \text{ impliziert } s_1 \bullet = s_2 \bullet. \quad (2.2)$$

Beweis. i) Es sei zuerst N ein Free-Choice-Netz. Wegen $s_1 \bullet \cap s_2 \bullet \neq \emptyset$ gibt es ein $t \in T$ mit $t \in s_1 \bullet$ und $t \in s_2 \bullet$. Damit gilt auch $\{s_1, s_2\} \subseteq \bullet t$. Somit ist $\bullet t$ für $(s, t) \in F$ nicht einelementig. Es muss also der Nachbereich von s_1 einelementig sein. Dies gibt $s_1 \bullet = \{t\}$. Analog zeigt man $s_2 \bullet = \{t\}$, woraus sofort $s_1 \bullet = s_2 \bullet = \{t\}$ folgt. Damit ist (2.1) für Free-Choice-Netze nachgewiesen.

Es gelte nun (2.1). Es sei (s, t) mit $s \in S$ und $t \in T$ eine Kante von N . Falls der Vorbereich von t einelementig ist, so gilt $\bullet t = \{s\}$ und damit auch die Alternative $(s \bullet = \{t\} \vee \bullet t = \{s\})$, womit die Bedingung für ein Free-Choice-Netz erfüllt ist. Falls

¹Man bemerke, dass wegen der Induzierttheit des Teilgraphen keine Kante von s' nach t_2 existieren kann. Bei Aussage iii) dieses Lemmas wäre das erlaubt.

der Vorbereich von t neben s noch ein Element s' enthält, so ist $t \in s \bullet \cap s' \bullet \neq \emptyset$. Nun erhalten wir aus der Gültigkeit von (2.1) sofort, dass es ein $t' \in T$ mit $s \bullet = s' \bullet = \{t'\}$ gibt. Da aber $t \in s \bullet$ liegt, ergibt sich $t = t'$ und somit $s \bullet = \{t\}$. Damit gilt erneut ($s \bullet = \{t\} \vee \bullet t = \{s\}$).

ii) Es sei zuerst wieder N ein erweitertes Free-Choice-Netz. Angenommen, die Implikation (2.2) gilt nicht. Dann gelten $s_1 \bullet \cap s_2 \bullet \neq \emptyset$ und $s_1 \bullet \neq s_2 \bullet$. Damit gibt es dann ohne Beschränkung der Allgemeinheit Transitionen t_1 und t_2 mit

$$t_1 \in s_1 \bullet \cap s_2 \bullet \quad \text{und} \quad t_2 \in s_2 \bullet \setminus s_1 \bullet. \quad (2.3)$$

Daher haben wir $s_2 \in \bullet t_1 \cap \bullet t_2$. Da N ein erweitertes Free-Choice-Netz ist, ergibt sich $\bullet t_1 = \bullet t_2$. Wegen $s_1 \in \bullet t_1$, erhalten wir $s_1 \in \bullet t_2$ und damit $t_2 \in s_1 \bullet$. Andererseits folgt aus der zweiten Beziehung aus (2.3) $t_2 \notin s_1 \bullet$. Dieser Widerspruch zeigt, dass unsere Annahme falsch ist, womit (2.2) bewiesen ist.

Es gelte nun (2.2). Völlig analog zeigt man, dass dann auch gilt, dass aus $\bullet t_1 \cap \bullet t_2$ die Gleichheit $\bullet t_1 = \bullet t_2$ folgt. (Man vertausche nur Stellen und Transitionen und gleichzeitig Vor- und Nachbereich im vorstehenden Beweis der umgekehrten Richtung.) \square

Zwischen den Klassen spezieller Netzgraphen gelten die folgenden Inklusionsbeziehungen.

Satz 2.13 *Es gilt das Diagramm aus Abbildung 2.5. Dabei bedeutet ein Pfeil von X nach Y , dass X eine echte Teilmenge von Y ist, und zwei Mengen X und Y sind genau dann unvergleichbar, wenn es keinen Weg von X nach Y bzw. umgekehrt gibt.*

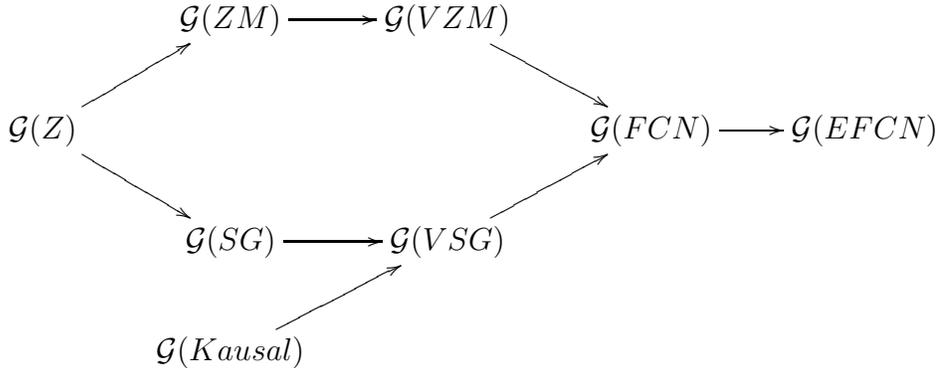


Abbildung 2.5: Hierarchie spezieller Netzgraphklassen

Beweis. Bei einem Zyklus (mit den Bezeichnungen aus Definition 2.9) gelten offenbar $s_i \bullet = \{t_i\}$ für $1 \leq i \leq n$, $\bullet s_i = \{t_{i-1}\}$ für $2 \leq i \leq n$ und $\bullet s_1 = \{t_n\}$. Damit ist jeder Vor- und Nachbereich einer jeden Stelle einelementig. Somit ist jeder Zyklus ein Synchronisationsgraph.

Analog beweist man, dass jeder Zyklus eine Zustandsmaschine ist.

Es sei $N = (S, T, F)$ eine verallgemeinerte Zustandsmaschine und $(s, t) \in F$ mit $s \in S$ und $t \in T$ eine Kante von N . Wegen $|\bullet t| \leq 1$ und $s \in \bullet t$, ist $|\bullet t| = 1$ und damit $\bullet t = \{s\}$. Damit gilt auch ($\bullet t = \{s\} \vee s \bullet = \{t\}$). Somit ist N ein Free-Choice-Netz.

Analog zeigt man, dass jeder verallgemeinerte Synchronisationsgraph ein Free-Choice-Netz ist.

Die Inklusionen zwischen den Klassen von Free-Choice-Netzen und erweiterten Free-Choice-Netzen folgt aus Lemma 2.12 sofort. Die restlichen Inklusionen ergeben sich schon aus den Definitionen.

Die Echtheit aller Inklusionen und die Unvergleichbarkeiten folgen aus den Beispielnetzgraphen aus Abbildung 2.4. Wie man jeweils leicht nachprüft, gilt für alle X und Y mit $X \subseteq Y$ die Beziehung $N_Y \not\subseteq X$, womit $X \subset Y$ gezeigt ist. \square

Definition 2.14 *Es sei $N = (S, T, F)$ ein Netzgraph.*

i) Wir sagen, dass das Netz N zusammenhängend ist, wenn es für zwei beliebige Knoten x und y aus $S \cup T$ einen ungerichteten Weg von x nach y gibt.

ii) Wir sagen, dass das Netz N stark zusammenhängend ist, wenn es für zwei beliebige Knoten x und y aus $S \cup T$ einen gerichteten Weg von x nach y gibt.

Entsprechend der Definition liegt Zusammenhang dann vor, wenn es eine Folge von Knoten x_0, x_1, \dots, x_n , $n \geq 0$, derart gibt, dass folgende Bedingungen erfüllt sind:

$$x = x_0, y = x_n, (x_i, x_{i+1}) \in F \text{ oder } (x_{i+1}, x_i) \in F \text{ für } 0 \leq i \leq n - 1.$$

Beim starken Zusammenhang wird nur geändert, dass $(x_i, x_{i+1}) \in F$ für $0 \leq i \leq n - 1$ gelten muss, d.h. es muss stets der Richtung der Kanten gefolgt werden.

Will man (starken) Zusammenhang testen, so hat man nach der Definition alle Paare aus $(S \cup T) \times (S \cup T)$ zu betrachten. Dies kann für den strengen Zusammenhang etwas vereinfacht werden, da man sich auf $S \times S$ bzw. $T \times T$ beschränken kann. Dies ist der Inhalt des folgenden Satzes.

Satz 2.15 *Die folgenden Aussagen sind für einen Netzgraphen $N = (S, T, F)$, der $\bullet x \neq \emptyset$ und $x \bullet \neq \emptyset$ für alle $x \in S \cup T$ erfüllt, gleichwertig.*

i) Der Netzgraph N ist stark zusammenhängend.

ii) Für zwei beliebige Stellen s und s' aus S gibt es einen gerichteten Weg von s nach s' .

iii) Für zwei beliebige Transitionen t und t' aus T gibt es einen gerichteten Weg von t nach t' .

Beweis. i) \implies ii) und i) \implies iii) sind trivial, da in ii) bzw. iii) nur die Existenz von Wegen zwischen Stellen bzw. Transitionen gefordert wird.

ii) \implies i). Wir haben zu zeigen, dass für beliebige Transitionen t und t' und eine beliebige Stelle s Wege von t nach t' , von s nach t und von t nach s existieren.

Es seien t und t' zwei Transitionen. Wegen $t \bullet \neq \emptyset$, gibt es eine Stelle s in $t \bullet$. Somit existiert eine Kante $(t, s) \in F$. Analog zeigt man, dass eine Stelle s' mit $(s', t') \in F$ existiert. Nach Voraussetzung gibt es einen gerichteten Weg von s nach s' . Wenn wir diesen Weg am Beginn durch (t, s) und am Ende durch (s', t') verlängern, erhalten wir einen Weg von t nach t' .

Analog beweist man die Existenz von Wegen von Transitionen zu Stellen und von Stellen zu Transitionen.

Der Beweis für $\text{iii}) \implies \text{i})$ kann mit gleichen Argumenten wie $\text{ii}) \implies \text{i})$ erbracht werden; man hat nur Stellen und Transitionen zu vertauschen. \square

Im Rahmen dieser Vorlesung wollen wir stets annehmen, dass die Netzgraphen, die den Petri-Netzen zugrundeliegen, zusammenhängend sind. Dies ist dadurch begründet, dass man sich in den meisten Fällen auf die zusammenhängenden Komponenten zurück ziehen kann.

Literaturverzeichnis

- [1] B. BAUMGARTEN, *Petri-Netze. Grundlagen und Anwendungen.* BI-Wissenschaftsverlag, Mannheim, 1990.
- [2] M. JANTZEN, The large marking problem. *Petri Net Newsletter* **14** (1983) 24–25.
- [3] S. R. KOSARAJU, Decidability of reachability in vector addition systems. *Proc. 14th Ann. ACM STOC* (1982) 267-281.
- [4] E. W. MAYR, An algorithm for the general petri net reachability problem. *SIAM J. Comput.* **13** (1984) 441-460.
- [5] H. MÜLLER, The reachability problem for VAS. In: *Lecture Notes in Computer Science* **188** (1984) 376–391.
- [6] W. REISIG, *Petrinetze. Eine Einführung.* Springer-Verlag, Berlin, 1991.
- [7] A. SCHRIJVER, *Theory of Linear and Integer Programming.* Wiley, 1986.
- [8] P. H. STARKE, *Petri-Netze.* VEB Deutscher Verlag der Wissenschaften, Berlin, 1980.
- [9] P. H. STARKE, *Analyse von Petri-Netz-Modellen.* B. G. Teubner, Stuttgart, 1990.
- [10] J. WEHLER, *Petri Nets.* Vorlesungsskript, Ludwig-Maximilians-Universität München, 1999.