

5 Halbordnungssemantik (Partial Order Semantics)

Während Prozesse von sequentiellen Prozessen durch eine totale (auch *linear* genannte) Ordnung gekennzeichnet sind, tritt für nebenläufige Prozesse (wie sie beispielsweise durch eine Synchronisation von Transitionssystemen entstehen) an deren Stelle die partielle oder strikte Ordnung.

Allgemein werden Prozesse als Anordnungen von Handlungen angesehen, wobei hier zunächst Handlungen (Aktionen) gemeint sind, die von einer Maschine, einem Prozessor, allgemein von einer Funktionseinheit ausgeführt werden.

Eigenschaften von Handlungen:

- *Extensionalität*: Handlungen sind durch ihre Wirkung beschreibbar.
- *Unteilbarkeit* (auch: Atomizität): Handlungen werden vom Prozessor ununterbrochen ausgeführt.
- *Anordnung*:
 - a) Ordnung durch eine totale Ordnung: sequentieller Prozess
 - b) Ordnung durch eine partielle Ordnung: nicht-sequentieller Prozess, d.h. zeitlich/kausal unabhängige Handlungen sind möglich

Mehrere sequentielle Prozesse wirken durch *Synchronisation* zusammen und bilden so einen Gesamtprozess, der eine Menge partiell geordneter Handlungen darstellt. Kausal unabhängige Handlungen heißen *nebenläufig*.

Nebenläufige Prozesse können auf (mindestens) zwei Weisen dargestellt werden:

- a) als partielle Ordnung, wie z.B. in Abb. 5.1 als Relation $R = \{(a, c), (b, c)\}$ oder
- b) als lineare Ordnung in Form von Folgen: $u := a; b; c$ und $v := b; a; c$.

Die Darstellungsform a) heißt *partial order semantics* oder *PO-Semantik*, während b) *interleaving semantics* oder *Folgen-Semantik* heißt.

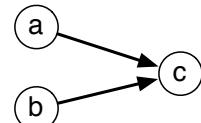
Beispiel 5.1 Um die Nebenläufigkeit von Zuweisungen a und b , wie im Programm

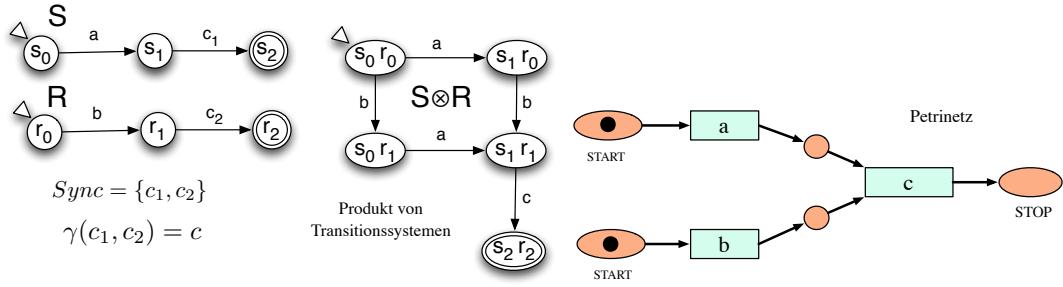
START $a : x := 3; b : y := 4; c : z := x + y$ **STOP**

zu modellieren, kann – wie in Abbildung 5.1 gezeichnet – ein Produkt von Transitionssystemen oder ein Petrinetz verwendet werden.

Die Semantik wird durch Folgenmenge $\{abc, bac\}$

oder durch die nebenstehende Striktordnung beschrieben:



Abbildung 5.1: Nebenläufige Handlungen a und b , gefolgt von c

5.1 Partielle und strikte Halbordnung

Dieser Abschnitt gibt eine formale Definition partieller Ordnungen.

Definition 5.2 Sei A eine Menge und $R \subseteq A \times A$ eine (binäre) Relation.

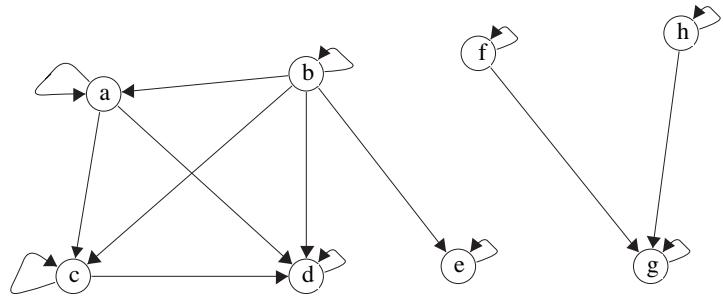
- a) (A, R) heißt partielle Ordnung (partially ordered set, poset), falls gilt:
 - 1. $\forall a \in A. (a, a) \in R$ “Reflexivität”
 - 2. $\forall a, b \in A. (a, b) \in R \wedge (b, a) \in R \Rightarrow a = b$ “Antisymmetrie”
 - 3. $\forall a, b, c \in A. (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$ “Transitivität”
 Schreibweise: $a \leq b$ für $(a, b) \in R$
- b) (A, R) heißt strikte Ordnung oder Striktordnung (partially ordered set, poset, Halbordnung), falls gilt:
 - 1. $\forall a \in A. (a, a) \notin R$ “Irreflexivität”
 - 2. $\forall a, b, c \in A. (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$. “Transitivität”
 Schreibweise: $a < b$ für $(a, b) \in R$
- c) (A, R) heißt totale oder lineare Ordnung (totally ordered set), falls gilt:
 - 1. (A, R) ist partielle Ordnung.
 - 2. $\forall a, b \in A. a \neq b$ impliziert $(a, b) \in R \vee (b, a) \in R$. “Vollständigkeit”
 Eine Striktordnung mit 2. heißt totale oder lineare Striktordnung.

In Abb. 5.2 ist oben eine partielle Ordnung R und darunter die Striktordnung $S = R - id$ dargestellt. Striktordnungen lassen sich oft übersichtlicher als Präzedenzgraph (“Hasse-Diagramm”) darstellen, der nur die direkten Nachfolger enthält (Abb. 5.3).

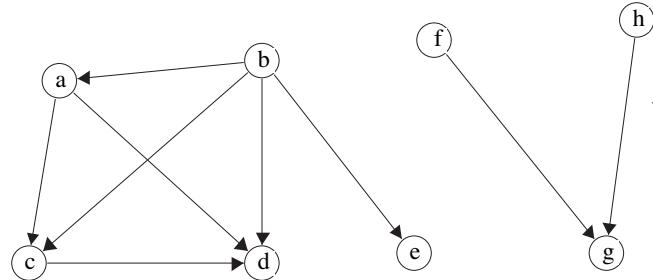
Definition 5.3 Sei $(A, <)$ eine strikte Ordnung und $a, b \in A$.

- 1. b heißt direkter Nachfolger von a (in Zeichen: $a \lessdot b$), falls:

$$a \lessdot b : \Leftrightarrow a < b \wedge \neg \exists c \in A. a < c \wedge c < b$$
- 2. (A, \lessdot) heißt Präzedenzrelation zu $(A, <)$.
- 3. Die Menge ${}^\circ A = \{a \in A \mid \neg \exists b \in A : b < a\}$ ist die Menge der minimalen Elemente der Ordnung $<$.

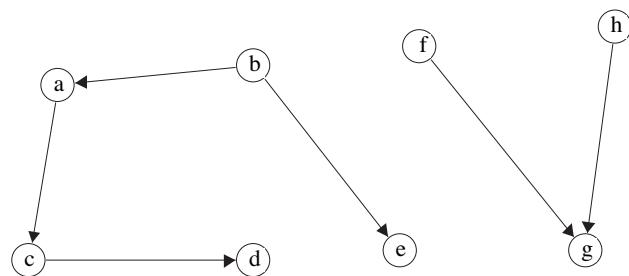


$$R = \{(a,a), (a,c), (b,a), \dots\}$$



$$S = R - id = \{(b,a), (a,c), (b,c), \dots\}$$

Abbildung 5.2: Partielle Ordnung R und Striktordnung S



$$Q = S - S^2 = \{(b,a), (a,c), (c,d), \dots\}$$

Abbildung 5.3: Präzedenzgraph Q

Entsprechend ist $A^\circ = \{a \in A \mid \neg \exists b \in A : a < b\}$ die Menge der maximalen Elemente.

Anmerkung:

- a) Es gilt: $\lessdot = (\langle - \rangle^2)$ (d.h. die Relation \langle ohne transitiv gebildete Paare)
 (“ $-$ ” Mengendifferenz, $\langle^2 := (\langle \circ \langle) = \{(a, b) \mid \exists c. a < c \wedge c < b\}$ Relationenprodukt)
- b) Gilt $\lessdot \subseteq \lessdot^+$ (transitive Hülle), dann heißt (A, \lessdot) kombinatorisch. In diesem Fall ist \langle durch \lessdot festgelegt. Für endliche Mengen A ist (A, \lessdot) immer kombinatorisch.
- c) Falls A unendlich ist, muss dies nicht gelten:
 Für die rationale Zahlen (\mathbb{Q}, \lessdot) gilt zum Beispiel $\lessdot = \emptyset$.
- d) Ist eine Striktordnung (A, \lessdot) isomorph zu einer Teilmenge von \mathbb{N} (mit der von \mathbb{N} geerbten Striktordnung und damit total), dann wird sie gerne mit einer Folge beschrieben:

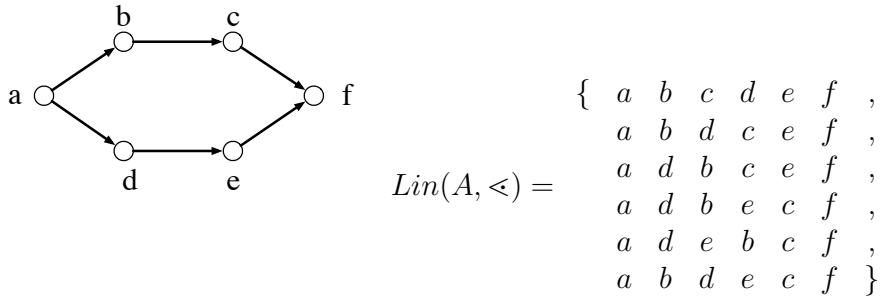
$$\begin{array}{lll} a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_n & \text{mit} & a_1 a_2 \dots a_n \quad (\text{card}(A) = n) \\ & \text{oder} & a_1; a_2; \dots; a_n \\ a_1 \rightarrow a_2 \rightarrow \dots & \text{mit} & a_1 a_2 \dots \quad (\text{card}(A) = \text{card}(\mathbb{N})) \\ & \text{oder} & a_1; a_2; \dots \end{array}$$

- e) Für eine Striktordnung (A, \lessdot) ist die Menge der linearen Vervollständigungen (auch: Linearisierung) von (A, \lessdot) :

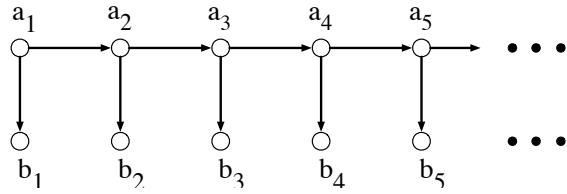
$$\text{Lin}(A, \lessdot) := \{(A, \lessdot_1) \mid \lessdot_1 \text{ ist lineare Striktordnung mit } \lessdot \subseteq \lessdot_1\} \quad (5.1)$$

Ist (A, \lessdot) kombinatorisch mit $\lessdot \subseteq \lessdot^+$ dann wird: $\text{Lin}(A, \lessdot) := \text{Lin}(A, \lessdot^+)$ definiert.

Beispiel 5.4 Die Ordnung (A, \lessdot) sei durch folgenden Graph definiert (Konstruktionsprinzip?):



Beispiel: Die Ordnung (A, \lessdot) sei durch folgenden Graph definiert:



$$\begin{aligned} \text{Lin}(A, \lessdot) = \{ & \quad a_1 \quad b_1 \quad a_2 \quad b_2 \quad a_3 \quad b_3 \quad a_4 \quad b_4 \quad a_5 \quad b_5 \quad \dots \\ & \quad a_1 \quad a_2 \quad b_1 \quad b_2 \quad a_3 \quad b_3 \quad a_4 \quad b_4 \quad a_5 \quad b_5 \quad \dots \\ & \quad a_1 \quad a_2 \quad b_1 \quad a_3 \quad b_2 \quad b_3 \quad a_4 \quad b_4 \quad a_5 \quad b_5 \quad \dots \\ & \quad \vdots \end{aligned}$$

Konstruktionsprinzip?
(so nicht möglich, da $\text{Lin}(A, \lessdot)$ überabzählbar)

Aufgabe 5.5 (Linearisierung)

- Ist a, b, c, d, e, f, g, h eine Linearisierung der Striktordnung von Abb. 5.2 ?
- Berechnen Sie für die Striktordnung S von Abb. 5.2 die Relation S^2 .

5.2 Logische und vektorielle Zeitstempel

In diesem Abschnitt wird die Rolle partieller Ordnungen in verteilten Systemen betrachtet. Unter anderem wird gezeigt, wie durch Zeitstempel eine strikte Ordnung zu einer totalen Ordnung (*Lamport-Ordnung* genannt) verschärft werden kann, um dadurch eine globale Sicht auf das verteilte System zu ermöglichen.

Kausalität und Zeit spielen eine wichtige Rolle beim Entwurf verteilter Systeme. Oft ist es wichtig, die relative Ordnung von Ereignissen und Aktionen zu kennen. In verteilten Systemen ist meist keine zentrale einheitliche Zeitmessung möglich. Trotzdem kann aber die relative Ordnung von Ereignissen durch sogenannte *logische Uhren* festgehalten werden, indem *logische Zeitstempel* gesetzt und den zu versendenden Nachrichten beigefügt werden. Logische Uhren wurden in [Lam78] eingeführt und werden in den Lehrbüchern [Lyn96], [AW98] und [Mat89] behandelt.

Definition 5.6 Ein Nachrichten-Modell ist ein System von n Funktionseinheiten bzw. Prozessoren p_0, \dots, p_{n-1} , die

- lokale Rechenschritte ausführen und
- Nachrichten an andere versenden.

Ein *Ereignis* ist entweder das Absenden oder das Empfangen von Nachrichten. Zur Vereinfachung wird außerdem angenommen, dass in einem Ereignis höchstens eine einzige Nachricht gesendet oder empfangen wird. Diese Ereignisse sind in der Ereignismenge $\Phi = \{\phi_1, \phi_2, \dots\}$ enthalten.

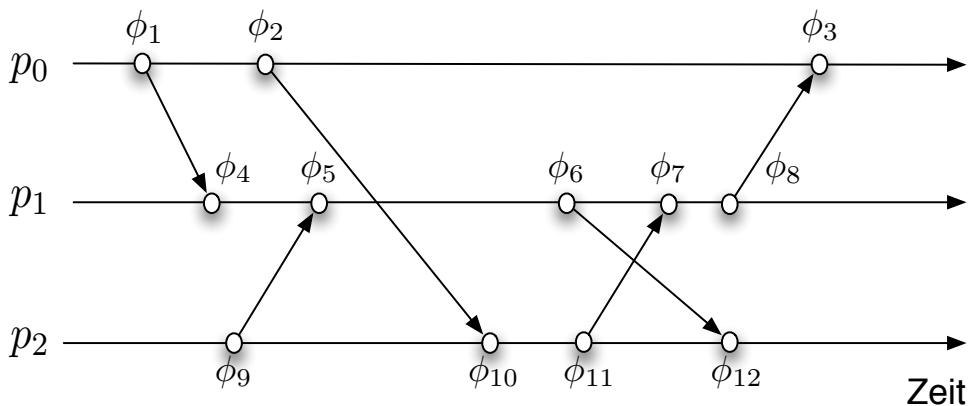


Abbildung 5.4: Nachrichten-Modell mit $n=3$ Prozessoren durch lokale Zeitskalen

Definition 5.7 Es wird eine Relation $\text{vor} \subseteq \Phi \times \Phi$ definiert. Für $\phi_1, \phi_2 \in \Phi$ gelte (ϕ_1 vor ϕ_2), falls folgendes gilt:

- a) Gehören ϕ_1 und ϕ_2 zu dem selben Prozessor (d.h. sie liegen auf der selben (linear geordneten) Zeitachse), dann gilt (ϕ_1 vor ϕ_2) genau dann, wenn ϕ_1 vor ϕ_2 auf der Zeitachse liegt.

- b) Gehören ϕ_1 und ϕ_2 zu verschiedenen Prozessoren (d.h. sie liegen auf verschiedenen Zeitachsen) und ist ϕ_1 das Sendeereignis einer Nachricht, die in ϕ_2 empfangen wird, dann gilt ($\phi_1 \text{ vor } \phi_2$).
- c) Gibt es ein Ereignis ϕ mit ($\phi_1 \text{ vor } \phi$) und ($\phi \text{ vor } \phi_2$), dann gilt auch ($\phi_1 \text{ vor } \phi_2$) (transitiver Abschluss).

(Φ, vor) ist eine strikte Ordnung, denn nach c) ist sie transitiv. Wäre sie nicht irreflexiv, dann müsste für ein Ereignis ϕ_1 die Beziehung $(\phi_1 \text{ vor } \phi_1)$ gelten. Dies kann nur daher kommen, dass $(\phi_1 \text{ vor } \phi_2)$ und $(\phi_2 \text{ vor } \phi_1)$ für ein Ereignis ϕ_2 auf einer anderen Zeitachse gilt. Sei oBdA. ϕ_1 ein Sendeereignis. Dann kann es kein Empfangsereignis sein, d.h. es muss ein Ereignis ϕ_3 auf der Zeitachse von ϕ_1 geben mit $(\phi_1 \text{ vor } \phi_2 \text{ vor } \phi_3 \text{ vor } \phi_1)$. Dann gilt aber gleichzeitig $(\phi_3 \text{ vor } \phi_1)$ und $(\phi_1 \text{ vor } \phi_3)$, was für Ereignisse auf der selben Zeitachse nach Bedingung a) ausgeschlossen ist.

Für Anwendungen in realen Systemen kann es wichtig sein, (Φ, vor) in einer linearen Ordnung zu erweitern, d. h. eine lineare Vervollständigung zu finden, die die **vor**-Relation nicht verletzt. Dazu müssen die lokale Zeitskalen in konsistente globale Zeitskalen transformiert werden. Dies wollen wir anhand des Anwendungsbeispiels eines Bankssystems darstellen.

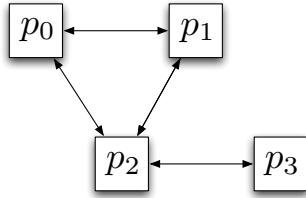


Abbildung 5.5: Banksystem mit 4 Filialen p_0, \dots, p_3

Beispiel 5.8 Funktionseinheiten (Filialen) p_i enthalten lokale Variablen x_i mit aktuellem Kontostand. Über die Kanäle wird als Nachricht eine Menge m_{ij} von p_i nach p_j transferiert (siehe Abb. 5.6).

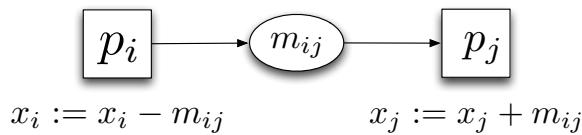


Abbildung 5.6: Übertragung von Nachrichten über Kanal

Es liege eine fortwährende Aktivität vor, d.h. jede Funktionseinheit versendet unendlich oft eine Nachricht an jede andere Funktionseinheit. Falls eine Funktionseinheit terminiert, so kann sie ständig Nachrichten mit $m = 0$ absenden. Der Anfangszustand des

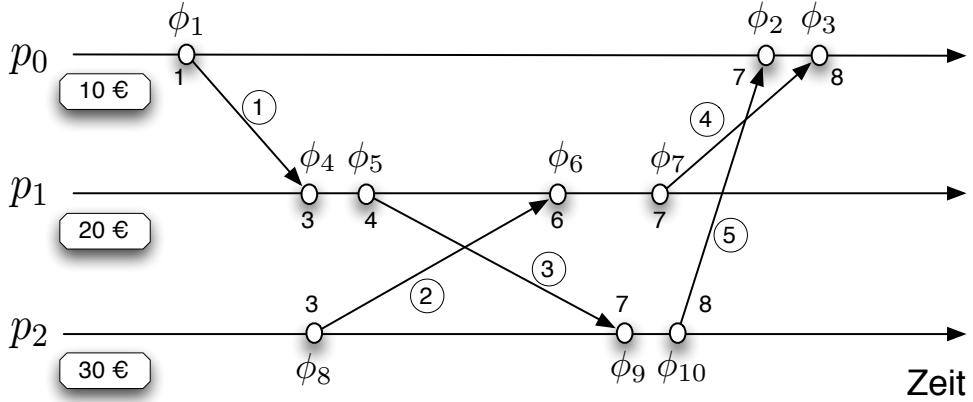


Abbildung 5.7: Zeitskala zu Beispiel 5.8

Modells ist $(x_0, \dots, x_{n-1}) = (a_0, \dots, a_{n-1})$ und keine Nachricht ist unterwegs. $c := \sum_{i=0}^{n-1} a_i$ ist die Gesamtgeldmenge.

$$\text{Gesamtgeldmenge : } 10 + 20 + 30 = 60$$

$$m(\phi_8, \phi_6) = 2$$

1, 3, 4, ..., 8 sind die Zeitzustände der lokalen Uhren

Problem: Die Bankleitung möchte immer wieder in Intervallen die insgesamt umlaufende Geldmenge ermitteln. Diese sollte immer gleich 60 sein.

Verfahren 1:

Die Bankleitung fordert alle Funktionseinheiten auf, die Kontostände zu einem bestimmten Zeitpunkt t mitzuteilen.

Erwartung: $\sum = 60$.

Beispiel 5.9 Zeitpunkt $t = 5$

$$\begin{array}{rcl} p_0 & : & x_0 = 10 - 1 = 9 \\ p_1 & : & x_1 = 20 + 1 - 3 = 18 \\ p_2 & : & x_2 = 30 - 2 = 28 \\ \hline & & \sum 55 ! \end{array}$$

Verfahren 2:

Die Bankleitung bittet die Summe der abgesandten minus der Summe der eingegangenen Beträge zu x_i jeweils hinzuzuzählen.

Erwartung: $\sum = 60$.

Beispiel 5.10 Zeitpunkt $t = 5$

$$\begin{array}{rcl} p_0 & : & x_0 = 9 + 1 = 10 \\ p_1 & : & x_1 = 18 - 1 + 3 = 20 \\ p_2 & : & x_2 = 28 + 2 = 30 \\ \hline & & \sum 60 \end{array}$$

Aber es kann auch der Fall von Abb. 5.8 eintreten, wo zum Zeitpunkt t ein unerwarteter Überschuss von 63 Geldeinheiten beobachtet wird. Solche Effekte sind durch einfache Zählverfahren nicht zu beheben, jedoch durch die im Folgenden eingeführten *Zeitstempelverfahren*.

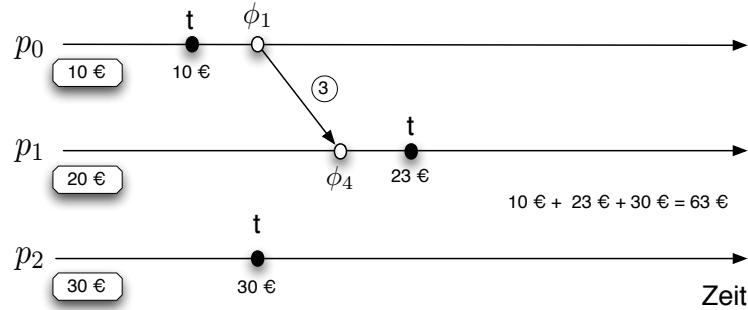


Abbildung 5.8: Im Zeitpunkt t wird virtueller Überschuss durch Nachrichten „aus der Zukunft“ beobachtet.

Es stellt sich die Frage, wie die Relation **vor** im System beobachtet werden kann. Dazu bestimmen wir eine *logische Uhr* $LT(\phi)$ mit

$$\phi_1 \text{ vor } \phi_2 \Rightarrow LT(\phi_1) < LT(\phi_2).$$

Zur Realisierung führt jede Funktionseinheit p_i eine Variable LT_i mit Anfangswert $LT_i = 0$ mit. Den Nachrichten wird der neue Wert des Sendeereignisses beigefügt (*logische Zeitstempel*). Ein Ereignis ϕ von p_i setzt LT_i auf einen um 1 größeren Wert als das Maximum des alten Wertes und eines ggf. in ϕ empfangenen Zeitstempels.

Definition 5.11 Sei ϕ ein Ereignis von p_i . Dann bezeichnet $LT(\phi)$ den von ϕ berechneten Wert von LT_i .

Satz 5.12 Für die Ereignisse $\phi_1, \phi_2 \in \Phi$ gilt:

$$\phi_1 \text{ vor } \phi_2 \Rightarrow LT(\phi_1) < LT(\phi_2)$$

Man beachte, dass das Gegenbeispiel von Abb. 5.8 hier nicht mehr auftreten kann!

Beispiel 5.13 Zeitpunkt $t = 1,5$ in Abb. 5.9:

$$\begin{array}{l} p_0 : \phi = \phi_1, \phi' = \phi_2 \quad c_0 = 9 \\ p_1 : t \text{ liegt vor } \phi_4 \quad c_1 = 20 \\ p_2 : \phi = \phi_8, \phi' = \phi_9 \quad c_2 = 28 \\ \hline \sum \quad \quad \quad 57 \end{array}$$

Es kommen an: 1 in ϕ_4 und 2 in ϕ_6 . Die Summe ist $57 + 3 = 60$.

Jedoch: woher weiß die Leitung, ob alle Nachrichten angekommen sind? Die Funktionseinheiten werden aufgefordert mitzuteilen, wieviele Nachrichten an welche andere Funktionseinheit abgesandt bzw. von solchen empfangen wurden.

Algorithmus 5.1 Verfahren der Bankleitung

1. Man führe logische Uhren ein.
 2. Man lege ein $t \in \mathbb{Q}$ mit $t \geq 0$ fest.
 3. Für jede Funktionseinheit p_i :
 - Bestimme in Bezug auf die lokale Zeit aufeinander folgende Ereignisse ϕ und ϕ' von p_i mit $LT(\phi) \leq t < LT(\phi')$, falls t nicht vor dem ersten Ereignis von p_i liegt.
 - Setze c_i auf den Wert von x_i zwischen ϕ und ϕ' oder auf den Anfangswert, falls t vor dem ersten Ereignis von p_i liegt. Sende c_i an Leitung.
 - Sende den Wert jeder Geldsendung an Leitung, die ab ϕ' ankommt, aber einen Zeitstempel $\leq LT(\phi)$ hat.
-

Im Beispiel 5.13 sind dies zum Zeitpunkt $t=1,5$:

p_0 : eine an p_1

p_1 : keine

p_2 : eine an p_1

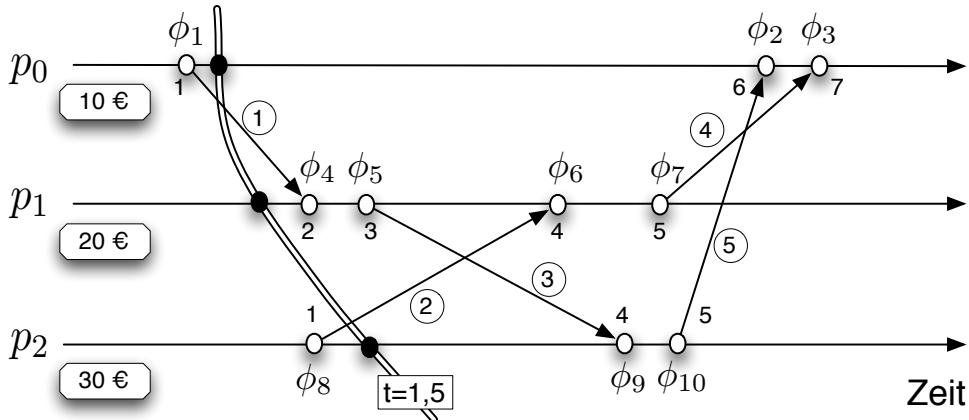


Abbildung 5.9: Zeitskala zu Beispiel 5.13

Die Relation $<$ ist i.A. keine totale Ordnung. Oft wird in verteilten Algorithmen jedoch eine totale Ordnung auf den Ereignissen benötigt, z.B. um eindeutige Prioritäten zu regeln. Zu diesem Zweck kann die Relation $<$ zu einer totalen Relation erweitert werden, indem in den Fällen, in denen zwei Ereignisse gleiche Zeitstempel haben, als „tie-break“ Regel die hier ja gegebene totale Ordnung auf den Prozessorbezeichnern hinzugenommen wird. Diese Ordnung heißt nach ihrem Finder *Lamport-Ordnung*.

Definition 5.14 Werden die Ereignisse ϕ durch ihren Zeitstempel $LT(\phi)$ und ihren Prozess(or) p_i charakterisiert, dann ist die Lamport-Ordnung $<_L$ auf den Ereignissen definiert als:

$(LT(\phi), p_i) <_L (LT(\phi'), p_j)$ genau dann, wenn

- a) $LT(\phi) < LT(\phi')$ oder
- b) $LT(\phi) = LT(\phi')$ und $i < j$.

Beispiel 5.15 $(2, p_5) <_L (3, p_5)$, $(2, p_5) <_L (3, p_4)$, $(2, p_5) >_L (2, p_4)$

Wir haben gesehen, dass gilt:

$$\phi_1 \text{ vor } \phi_2 \Rightarrow LT(\phi_1) < LT(\phi_2)$$

Stellt die Relation $<$ die vor-Relation exakt dar, d.h. gilt auch die folgende Umkehrung?

$$LT(\phi_1) < LT(\phi_2) \Rightarrow \phi_1 \text{ vor } \phi_2$$

Diese Umkehrung gilt nicht, da $<$ - wie gesagt - keine totale Ordnung ist. Dazu noch ein Gegenbeispiel: In der Zeitskala von Abb. 5.10 ist Folgendes zu beobachten: Es gilt $LT(\phi_9) = 1 < LT(\phi_2) = 2$ aber nicht $\phi_9 \text{ vor } \phi_2$. Der Grund ist, dass $LT(\phi) \in \mathbb{N}$ und \mathbb{N} linear geordnet ist.

Abhilfe: Statt \mathbb{N} wählen wir die nicht linear, aber partiell geordnete Menge \mathbb{N}^n für $n > 1$. Zur Erinnerung: Die *partielle Ordnung* auf \mathbb{N}^n ist komponentenweise definiert:

$\vec{v}_1 \leq \vec{v}_2 \Leftrightarrow \forall i \in \{1, \dots, n\} : \vec{v}_1[i] \leq \vec{v}_2[i]$. Die *strikte Ordnung* auf \mathbb{N}^n schließt Gleichheit aus: $\vec{v}_1 < \vec{v}_2 : \Leftrightarrow \vec{v}_1 \leq \vec{v}_2 \wedge \vec{v}_1 \neq \vec{v}_2$. Zwei Vektoren \vec{v}, \vec{v}' heißen *unvergleichbar*, falls $\neg(\vec{v} \leq \vec{v}') \wedge \neg(\vec{v}' \leq \vec{v})$ gilt.

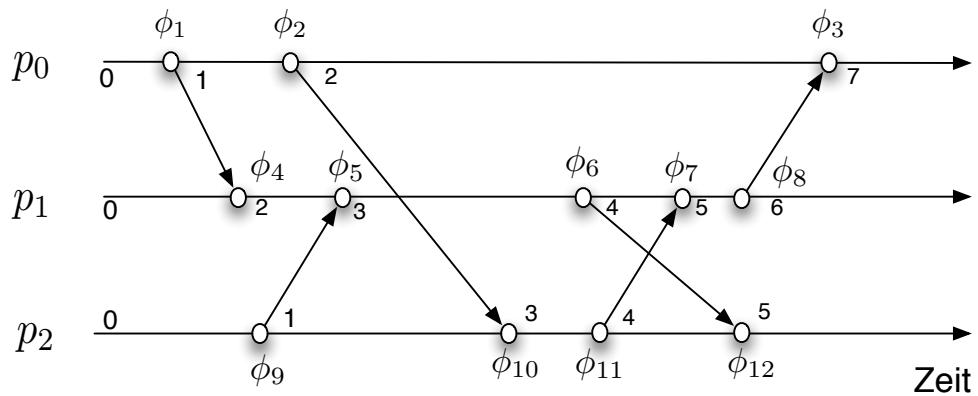


Abbildung 5.10: Senden mit logischen Zeitstempeln

Definition 5.16 ϕ_1 heißt unabhängig von ϕ_2 bzw. ϕ_1 heißt nebenläufig zu ϕ_2 , geschrieben als $\phi_1 \parallel \phi_2$, falls gilt:

$$\phi_1 \parallel \phi_2 \Leftrightarrow \neg(\phi_1 \text{ vor } \phi_2) \wedge \neg(\phi_2 \text{ vor } \phi_1)$$

Gesucht ist eine strikte Ordnung auf Φ , die \parallel darstellt.

Definition 5.17 (Vektorzeit, vektorieller Zeitstempel) Jede Funktionseinheit p_i führt eine Variable \vec{v}_i mit Werten in \mathbb{N}^n (lokale Vektorzeit) und dem Nullvektor $\vec{0}$ als Anfangswert.

Falls p_i ein Ereignis ϕ bearbeitet, wird der Zeitstempel \vec{v}_i wie folgt aktualisiert:

- Für die eigene Komponente i von p_i gelte:
 $\vec{v}_i[i] \mapsto \vec{v}_i[i] + 1$ (Inkrementieren des eigenen Stempels)
- Für die anderen Komponenten $j \neq i$ von p_i gelte:
 $\vec{v}_i[j] \mapsto \max(\vec{v}_i[j], \vec{V}T_m[j])$ falls eine Nachricht m mit dem Vektorzeitstempel $\vec{V}T_m$ empfangen wird. Wird keine Nachricht empfangen, bleibt $\vec{v}_i[j]$ unverändert ($j \neq i$).
(Aktualisieren der anderen Komponenten)

Die Abbildung $VC : \Phi \rightarrow \mathbb{N}^n$ wird definiert durch $VC(\phi) = \vec{v}_i$, wobei \vec{v}_i der in ϕ durch p_i berechnete Wert ist.

Beispiel: Die Anschrift an ϕ in Abb. 5.11 ist $VC(\phi)$.

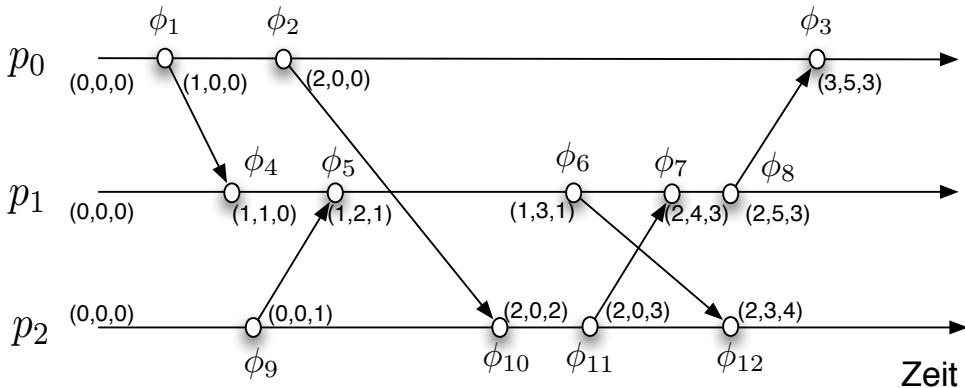


Abbildung 5.11: Zeitskala zur Vektorzeit

Satz 5.18 Die Vektorzeit charakterisiert die **vor**-Relation, denn es gilt:

$$\begin{aligned} VC(\phi_1) < VC(\phi_2) &\iff \phi_1 \text{ vor } \phi_2 \\ VC(\phi_1), VC(\phi_2) \text{ unvergleichbar} &\iff \phi_1 \parallel \phi_2 \end{aligned}$$

Aufgabe 5.19 Der Bankleitung werden ständig alle Vektor-Zeiten $VC(\phi)$ gesandt. Kann Sie darauf ein Verfahren aufbauen, um das Bilanz-Problem zu lösen?

6 Petrinetze und Nebenläufigkeit

Dieses Kapitel führt in den Formalismus der Petrinetze ein. Wichtige ihrer Eigenschaften lassen sich folgendermaßen zusammenfassen:

- 1.) graphische und äquivalente algebraische/textuelle Darstellung
- 2.) (formal abgesicherte) Algorithmen für die Analyse
- 3.) Abstraktion und hierarchische Strukturen
- 4.) hoch entwickelte Theorie der Nebenläufigkeit (concurrency)
- 5.) Rechnerwerkzeuge für Modellierung, Simulation und Analyse
- 6.) Universalität in Anwendbarkeit (Anwendungen in fast allen Gebieten)
- 7.) Varianten des gleichen Modellierungskonzeptes (Zeit-Netze, stochastische Netze, high-level, objektorientiert, ...)

Dies bewirkt unter anderem, dass sie „einfach“ zu vermitteln sind, dass Werkzeuge „einfach“ miteinander verknüpft werden können sowie die Verträglichkeit von verschiedenen Abstraktionsebenen. Die meist benutzten Vertreter der im letzten Punkt der vorstehenden Aufzählung genannten „high-level Netze“ sind die *gefärbten Netze*, bei denen es keine Einschränkung für den Datentyp der Plätze gibt. Diese werden gesondert behandelt. Wie gesagt, besteht ein großer Vorteil der Petrinetze darin, dass viele Eigenschaften dieses komplexeren Modells sehr ähnlich zu den Eigenschaften der in diesem Kapitel eingeführten elementaren Platz/Transitions-Netze sind. Empfehlenswerte Monographien zu Petrinetzen sind [GV03], [Rei82] und [Rei10].

6.1 Petrinetze

Petrinetze werden durch ein einfaches Beispiel eingeführt, um wesentliche Prinzipien wie *Lokalität*, *Nebenläufigkeit*, *grafische* und *formaltextuelle Darstellung* daran zu erläutern. Das Beispiel stellt die Synchronisation von Objekten dar, wie sie in vielen Anwendungen in anderer, aber prinzipiell ähnlicher Form vorkommt.

6.1.1 Grundlegende Prinzipien

Das Beispiel stellt den Startvorgang eines Autorennens dar (siehe [GV03]). Zur Vereinfachung werden nur drei Objekte modelliert: zwei Autos und ein Starter (Abb. 6.1). Wenn die Fahrer der Wagen ihre Vorbereitungen abgeschlossen haben, geben Sie ein Fertigzeichen („ready sign“). Sobald der Starter die Fertigzeichen von allen Wagen erhalten hat, gibt er das Startsignal und die Wagen fahren los.

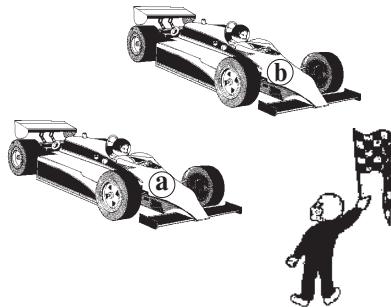


Abbildung 6.1: Start zweier Rennwagen

Man stelle sich vor, der Vorgang soll (z.B. für eine Simulation) modelliert werden. Dabei könnten die folgenden Bedingungen und Aktionen als relevant betrachtet werden:

Liste der Bedingungen	Liste der Aktionen
p_1 : car a; preparing for start	
p_2 : car a; waiting for start	
p_3 : car a; running	
p_4 : ready sign of car a	
p_5 : start sign for car a	t_1 : car a; send ready sign
p_6 : starter; waiting for ready signs	t_2 : car a; start race
p_7 : starter; start sign given	t_3 : starter; give start sign
p_8 : ready sign of car b	t_4 : car b; send ready sign
p_9 : start sign for car b	t_5 : car b; start race
p_{10} : car b; preparing for start	
p_{11} : car b; waiting for start	
p_{12} : car b; running	

Die Unterscheidung von „aktiven“ und „passiven“ Systemkomponenten ist eine wichtige (aber nicht immer eindeutige) Abstraktion. Diese wird durch Netze unterstützt:

I	Das Prinzip der Dualität in Petrinetzen
Es gibt zwei disjunkte Mengen von Grundelementen: <i>P-Elemente</i> (state elements, Plätze, Stellen) und <i>T-Elemente</i> (Transition-Element, Transitionen). Dinge der realen Welt werden entweder als passive Elemente aufgefasst und als P-Elemente dargestellt (z.B. Bedingungen, Plätze, Betriebsmittel, Wartepools, Kanäle usw.) oder als aktive Elemente, die durch T-Elemente repräsentiert werden (z.B. Ereignisse, Aktionen, Ausführungen von Anweisungen, Übermitteln von Nachrichten usw.).	

Anmerkung:

- Statt von Plätzen spricht man auch von „Stellen“ und „S-Elementen“.
- Beispielsweise kann ein Programm ein aktives oder passives Element sein, je nach Kontext.
- Beachte: In der Prozessalgebra wird diese Dualität nicht berücksichtigt, zumindest nicht explizit.

Um zu einem ausführbaren Modell zu kommen, ordnen wir den Bedingungen für den Anfangszustand Wahrheitswerte TRUE und FALSE zu. Im Anfangszustand m_1 bereiten sich die Wagen a und b auf den Start vor (d.h. $p_1 = p_{10} = T$ (TRUE)) und der Starter wartet auf die Fertigzeichen (d.h. $p_6 = T$). Der Anfangszustand ist also als Vektor m_1 darstellbar:

$$\mathbf{m}_1 = [p_1 = T, p_2 = F, p_3 = F, p_4 = F, p_5 = F, p_6 = T, \\ p_7 = F, p_8 = F, p_9 = F, p_{10} = T, p_{11} = F, p_{12} = F]$$

Zwei Aktionen, nämlich t_1 und t_4 , können hier stattfinden. Betrachten wir die Aktion t_1 . Mit ihr gibt der Wagen a das Startzeichen und beendet die Vorbereitungsphase ($p_1 = F$). Dann wartet er auf den Start ($p_2 = T$), nachdem er das Fertigzeichen abgegeben hat ($p_4 = T$). Daraus ergibt sich der neue Zustandsvektor \mathbf{m}_2 als:

$$\mathbf{m}_2 = [p_1 = F, p_2 = T, p_3 = F, p_4 = T, p_5 = F, p_6 = T, \\ p_7 = F, p_8 = F, p_9 = F, p_{10} = T, p_{11} = F, p_{12} = F]$$

Die graphische Darstellung dieses Zustandsüberganges in Abb. 6.2 zeigt, dass nur einige Bedingungen beteiligt sind. Sie sind in runde Grafikelemente gefasst und werden Vor- und Nachbedingung der Aktion genannt. Zusammen mit der Aktion (Rechteck) stellen sie den Übergang wesentlich einfacher und adäquater dar. Vor- und Nachbedingung nennen wir die Lokalität der Aktion. Sie allein bestimmt kausale (und zeitliche) Abhängigkeiten mit anderen Aktionen.

Die Aktion t_1 kann stattfinden, wenn p_1 gilt (TRUE) und p_2, p_4 nicht gelten (FALSE). p_1, p_2 und p_4 heißen Bedingungen der Aktion t_1 , wobei p_1 Vorbedingung und p_2, p_4 Nachbedingungen heißen. Zusammen mit dem Aktionsbezeichner nennen wir die Menge $\{t_1, p_1, p_2, p_4\}$ die Lokalität von t_1 .

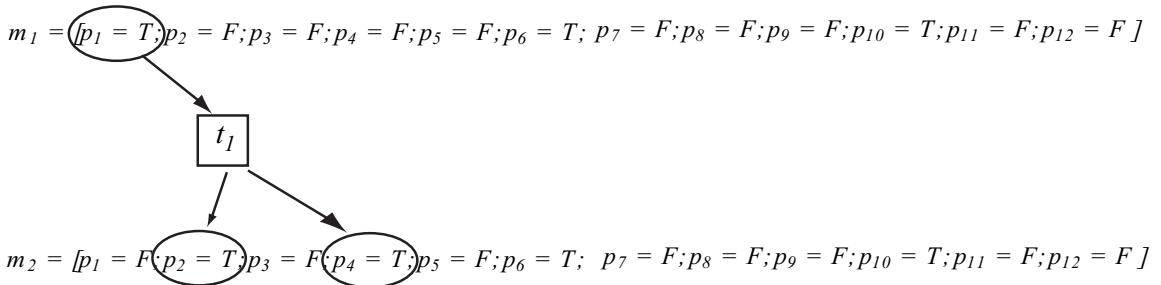


Abbildung 6.2: Lokalität der Aktion t_1

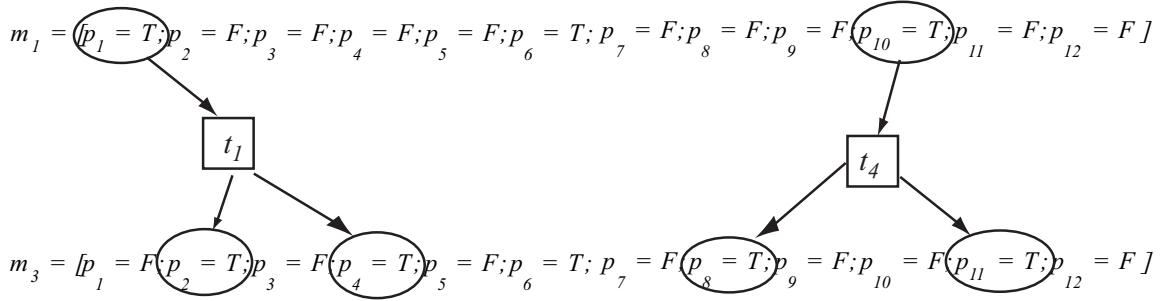


Abbildung 6.3: Nebenläufige Aktionen t_1 und t_4

II	Das Prinzip der Lokalität für Petrinetze
<p>Das Verhalten einer Transition wird ausschließlich durch ihre <i>Lokalität</i> bestimmt, welche sich aus ihr und der Gesamtheit ihrer Eingangs- und Ausgangselemente zusammensetzt.</p>	

Die Bedeutung dieser Begriffsbildung wird deutlicher, wenn wir einbeziehen, dass die zweite Aktion in \mathbf{m}_2 stattfinden kann, die \mathbf{m}_2 nach \mathbf{m}_3 überführt:, wobei:

$$\mathbf{m}_3 = [p_1 = F, p_2 = T, p_3 = F, p_4 = T, p_5 = F, p_6 = T, \\ p_7 = F, p_8 = T, p_9 = F, p_{10} = F, p_{11} = T, p_{12} = F].$$

Die Lokalität von t_4 ist $\{t_4, p_{10}, p_8, p_{11}\}$. Also teilen t_1 und t_4 keine Bedingung und sind damit völlig unabhängig. Die Abb. 6.3 drückt dies auch grafisch aus. Dies ist die Basis zur Modellierung von Nebenläufigkeit in Petrinetzen.

III	Das Prinzip der Nebenläufigkeit für Petrinetze
<p>Transitionen mit disjunkter Lokalität finden unabhängig (nebenläufig, concurrently) statt.</p>	

Die Abb. 6.4 zeigt *alle* Aktionen mit ihren Vor- und Nachbedingungen. In dieser Form heißen sie Transitionen und die Bedingungen Plätze oder Stellen. Zusammen bilden sie ein *Netz*. Identifiziert man Stellen mit gleichem Bezeichner, so erhält man Abb. 6.5. (Einige Plätze enthalten *Marken*, die den Anfangszustand markieren, worauf wir später zurückkommen.)

IV	Das Prinzip der grafischen Darstellung von Petrinetzen
<p>P-Elemente (auch S-Elemente) werden durch runde grafische Elemente (Kreise, Ellipsen,...) dargestellt (rund wie im Buchstaben P oder S).</p> <p>T-Elemente werden durch eckige grafische Elemente (Rechtecke, Balken,...) dargestellt (eckig wie im Buchstaben T).</p> <p>Kanten (auch „Pfeile“) verbinden T-Elemente mit den P-Elementen ihrer Lokalität. Also gibt es nur Kanten zwischen T-Elementen und P-Elementen.</p>	

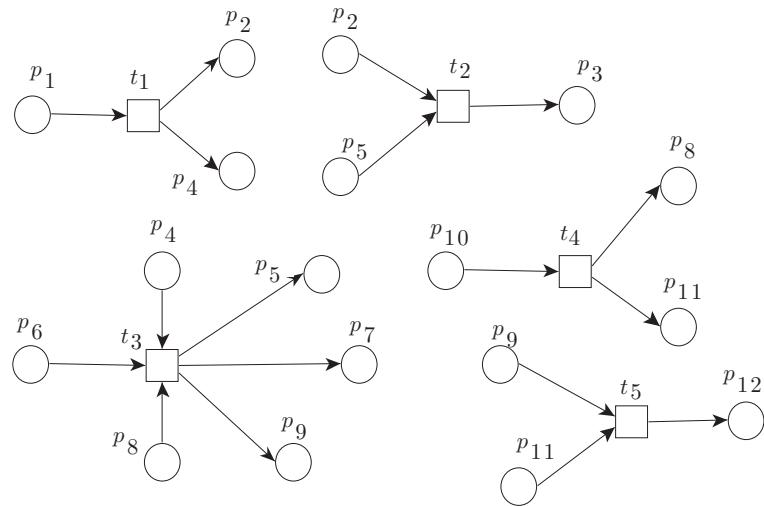


Abbildung 6.4: Einzelaktionen als Transitionen

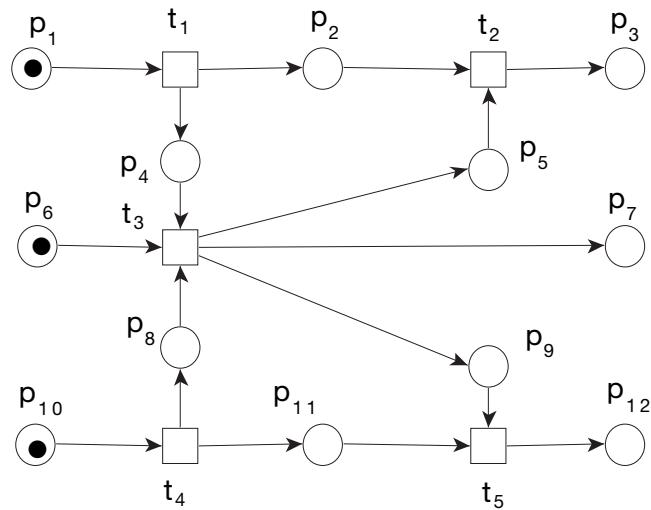


Abbildung 6.5: Netz \mathcal{N} zum Beispiel

In vielen Fällen (z.B. in Texten, zur formalen Beschreibung oder als Datenstruktur) sind formaltextuelle Darstellungen nützlich. Eine solche folgt als mathematische Definition.

Definition 6.1 Ein Netz ist ein Tripel $\mathcal{N} = (P, T, F)$, wobei gilt:

- P ist eine Menge von Plätzen (auch: Stellen).
- T eine Menge von Transitionen (disjunkt zu P).
- $F \subseteq (P \times T) \cup (T \times P)$ ist die Flussrelation.

Falls P und T endlich sind, dann heißtt auch das Netz \mathcal{N} endlich.

Für das Beispielnetz erhält man $P = \{p_1, \dots, p_{12}\}$, $T = \{t_1, \dots, t_5\}$ und $F = \{(p_1, t_1), (t_1, p_2), (t_1, p_4), \dots\}$.

Wichtig ist, dass die graphische und mathematische Darstellung äquivalent sind.

V	Das Prinzip der formaltextuellen Darstellung von Petrinetzen
Zu jeder graphischen Darstellung eines Petrinetzes gibt es eine äquivalente formaltextuelle Darstellung und umgekehrt.	

Die folgende Notation für Eingangs- und Ausgangs-Elemente ist üblich:

Definition 6.2 Für ein Netz $\mathcal{N} = (P, T, F)$ und ein Element $x \in P \cup T$ bezeichnet
 $\bullet x := \{y \in P \cup T \mid (y, x) \in F\}$ die Menge der Eingangselemente und $x^\bullet := \{y \in P \cup T \mid (x, y) \in F\}$ die Menge der Ausgangsselemente von x .

Für eine Menge von Elementen $X \subseteq P \cup T$ seien entsprechend

$$\bullet X := \bigcup_{x \in X} \bullet x \text{ und } X^\bullet := \bigcup_{x \in X} x^\bullet$$

$loc(y) := \{y\} \cup \bullet y \cup y^\bullet$ heißt Lokalität des Platzes oder der Transition y .

Ist x ein Platz, dann heißen $\bullet x$ bzw. x^\bullet Eingangs- bzw. Ausgangs-Transitionen. Analog für Transitionen.

Beispiel 6.3 Für $X = \{t_1, p_5, p_{11}\}$ im Netz 6.5 erhält man $\bullet X = \{p_1, t_3, t_4\}$ und $X^\bullet = \{p_2, p_4, t_2, t_5\}$.

Die Gültigkeit einer Bedingung wird durch eine Marke in dem entsprechenden Platz dargestellt (siehe Abb. 6.5). Die Schaltregel wird informell in Abb. 6.6 gezeigt: Eine Transition kann schalten, wenn alle Eingangsstellen eine Marke enthalten und die Ausgangsstellen unbelegt sind. Das Ergebnis des Schaltens ist das Entfernen der Marken in den Eingangsstellen und das Hinzufügen der Marken zu den Ausgangsstellen.

Definition 6.4 Sei \mathcal{N} ein Petrinetz und $M \subseteq P$ eine Markierung.

Die Transition $t \in T$ ist in M aktiviert, wenn $\bullet t \subseteq M$ und $t^\bullet \cap M = \emptyset$ gilt.

Die Nachfolgemarkierung M' ergibt sich dann als $M' = (M \setminus \bullet t) \cup t^\bullet$.

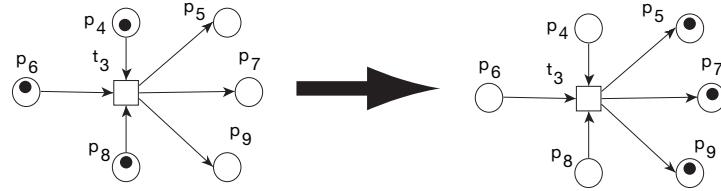


Abbildung 6.6: Schaltregel

Um das simultane Schalten von Transitionen zu beschreiben, wollen wir definieren, wann eine Transitionsmenge $U \subseteq T$ aktiviert ist. Dies ist der Fall, wenn der Vorbereich von U markiert, der Nachbereich von U unmarkiert ist und die Umgebungen aller Transitionen in U disjunkt zueinander sind, d.h. wenn alle Transitionen nebenläufig zueinander sind.

Definition 6.5 Sei \mathcal{N} ein Petrinetz und $M \subseteq P$ eine Markierung.

Eine Transitionsmenge $U \subseteq T$ ist in M aktiviert, wenn gilt:

1. Vorbereich markiert: $\bullet U \subseteq M$.
2. Nachbereich leer: $U^\bullet \cap C = \emptyset$.
3. Transitionen nebenläufig: $\forall t, t' \in U : t \neq t' \Rightarrow loc(t) \cap loc(t') = \emptyset$.

Die Nachfolgemarkierung M' ist definiert als $M' = (M \setminus \bullet U) \cup U^\bullet$.

Setzen wir $U = \{t\}$, so erhalten wir die bekannte sequentielle Schaltregel.

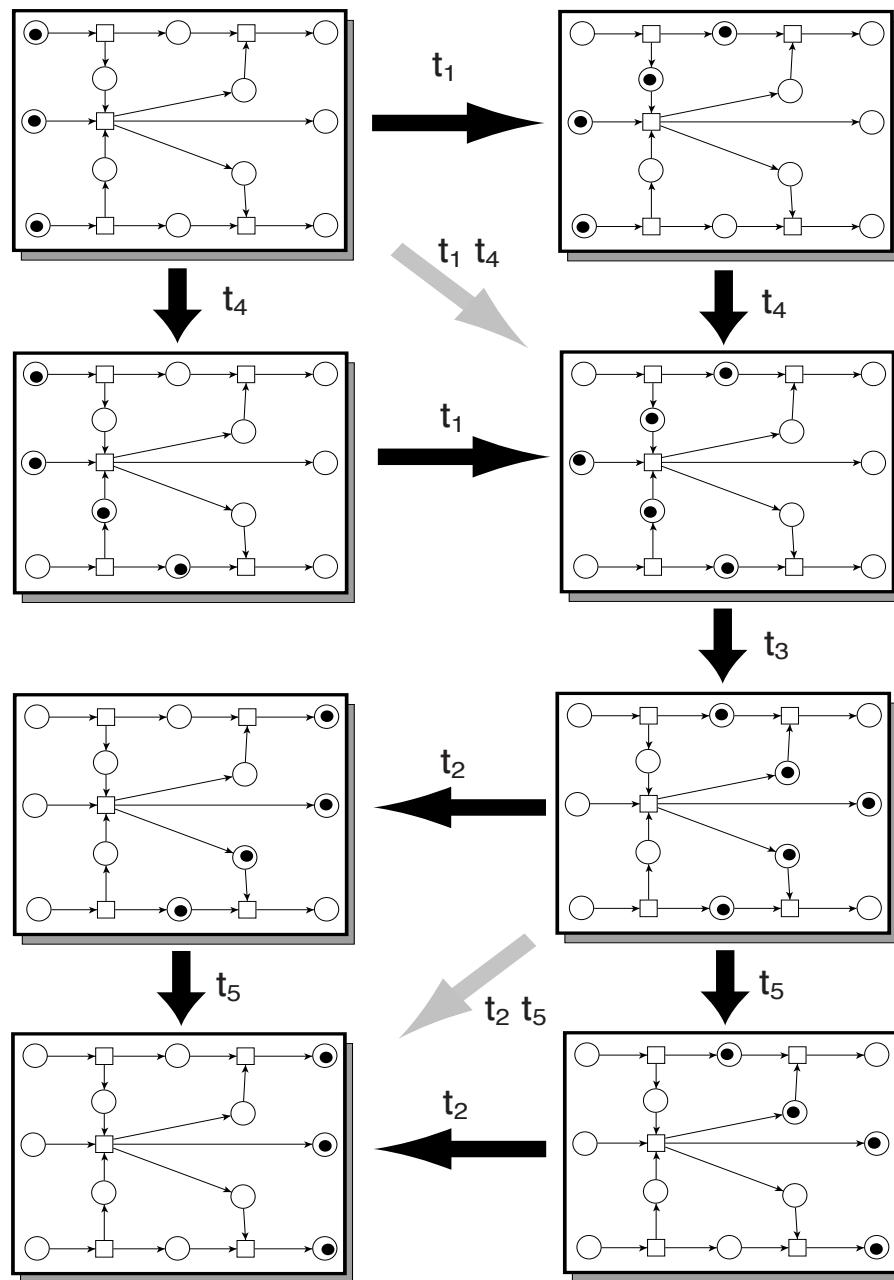
Beispiel 6.6 Die Abbildung 6.7 zeigt alle möglichen Folgen von Transitionssereignissen. Nebenläufige (z.B. t_1 und t_4) Transitionen sind sowohl als simultaner Schritt wie auch in Folgensemantik dargestellt.

6.1.2 Verfeinerung und Vergrößerung von Netzen

Hierarchiebildung ist ein wesentliches Konzept zur Darstellung und Strukturierung von Systemen. Die entsprechenden Begriffe der Vergrößerung und Verfeinerung sind für Netze ohne Anschriften (Kantenbewertung, Markierung) definiert.

Die Konstruktion von Systemhierarchien durch Vergrößerung (Abstraktion) und Verfeinerung ist eine wichtige Methode des Systementwurfs. Petrinetze unterstützen dies durch besondere mit ihrer Struktur kompatible Konzepte. Diese werden unabhängig von Markierungen und speziellen Netzmodellen gebildet und daher für einfache Netze definiert. Wir beginnen mit dem Begriff des *Randes* einer Menge von Plätzen und Transitionen, der die Schnittstelle des zu vergrößernden Teiles bilden wird.

Definition 6.7 Sei $\mathcal{N} = (P, T, F)$ ein Netz, $X := P \cup T$ und $Y \subseteq X$ eine Menge von Elementen. Dann heißt $\partial(Y) := \{y \in Y \mid \exists x \notin Y . x \in loc(y)\}$ der Rand (engl. border) der Menge Y .



Y heißt Platz-berandet (*place-bordered*) oder offen, wenn $\partial(Y) \subseteq P$, und Transitions-berandet (*transition-bordered*) oder abgeschlossen, falls $\partial(Y) \subseteq T$.

Anmerkung: Eine Menge Y kann gleichzeitig offen und abgeschlossen sein, wie z.B.: $Y := P \cup T$. In diesem Fall hängt es von der Interpretation bzw. Anwendung ab, ob Y durch einen Platz oder eine Transition ersetzt wird.

Platz-berandete Mengen heißen auch *offen*. Transitions-berandete Mengen heißen *abgeschlossen*. Die Bezeichnung ist in Anlehnung an die Topologie gewählt, denn offene und abgeschlossene Mengen definieren eine Topologie, die eine Formalisierung von Nachbarschaft auf der graphischen Struktur von Netzen darstellt.

Um eine Vergrößerung mit der Netzstruktur verträglich zu gestalten, sollten im Normalfall Platz- bzw. Transitions-berandete Mengen durch einen Platz bzw. eine Transition ersetzt werden.

Die Menge $Y = \{p_3, p_4, t_2, t_3, t_4\}$ des Netzes in Abb. 6.8 ist Transitions-berandet und wird daher zu einer Transition t_Y vergröbert. Auf diese Weise erhält man wieder ein Netz $\mathcal{N}[Y] = (P[Y], T[Y], F[Y])$, das in Abb. 6.9 dargestellt ist. $P[Y]$ enthält alle Plätze mit Ausnahme derjenigen aus Y . $T[Y]$ enthält alle Transitionen mit Ausnahme derjenigen aus Y und das neue Element t_Y . $F[Y]$ ist die Vereinigung von 3 Kantenmengen, nämlich (1) derjenigen, die kein Ende in Y haben, (2) derjenigen die von außerhalb von Y zu t_Y führen und (3) derjenigen, die von t_Y nach außerhalb führen. Diese Operation wird nun formalisiert.

Definition 6.8 Sei $\mathcal{N} = (P, T, F)$ ein Netz und Y eine nicht leere Transitions-berandete Menge von Elementen.

Dann heißt $\mathcal{N}[Y] = (P[Y], T[Y], F[Y])$ elementare Vergrößerung von \mathcal{N} in Bezug auf Y , falls gilt:

1. $P[Y] = P \setminus Y$.
2. $T[Y] = (T \setminus Y) \cup \{t_Y\}$, wobei t_Y ein neues Element ist.
3. $F[Y] = \{(x, y) \mid x \notin Y \wedge y \notin Y \wedge (x, y) \in F\} \cup \{(x, t_Y) \mid x \notin Y \wedge \exists y \in Y . (x, y) \in F\} \cup \{(t_Y, x) \mid x \notin Y \wedge \exists y \in Y . (y, x) \in F\}$.

Wenn Y eine Platz-berandete Menge ist, dann ist $\mathcal{N}[Y] = (P[Y], T[Y], F[Y])$ analog definiert:

1. $P[Y] = (P \setminus Y) \cup \{p_Y\}$, wobei p_Y ein neues Element ist,
2. $T[Y] = T \setminus Y$,
3. $F[Y] = \{(x, y) \mid x \notin Y \wedge y \notin Y \wedge (x, y) \in F\} \cup \{(x, p_Y) \mid x \notin Y \wedge \exists y \in Y . (x, y) \in F\} \cup \{(p_Y, x) \mid x \notin Y \wedge \exists y \in Y . (y, x) \in F\}$.

Anmerkung: Die Definition von $\mathcal{N}[Y]$ ist mehrdeutig, falls Y gleichzeitig Platz- und Transitions-berandet ist. Dann schreiben wir $\mathcal{N}[Y^{(p)}]$ falls Y als Platz-berandete Menge aufgefasst wird und $\mathcal{N}[Y^{(t)}]$ im anderen Fall.

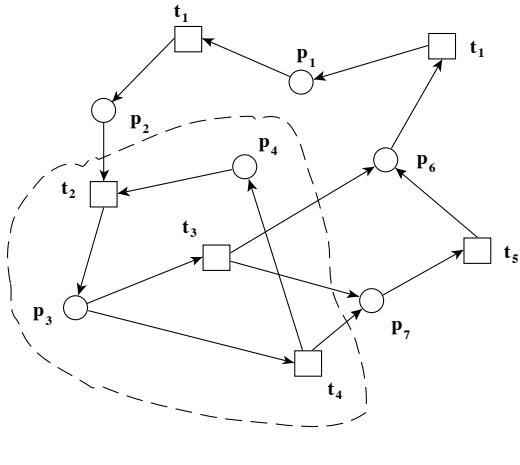


Abbildung 6.8: Eine transitionsberandete Menge

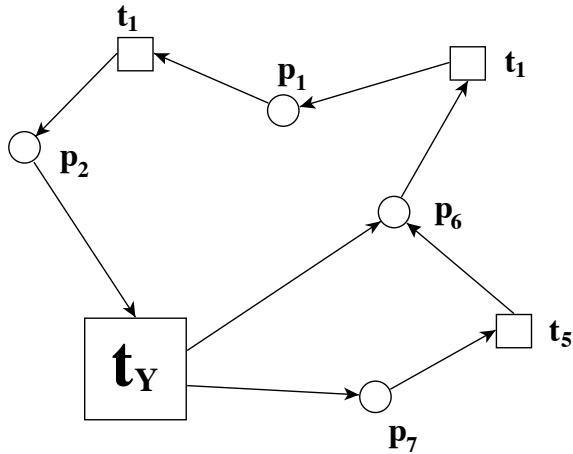


Abbildung 6.9: Vergrößerung des Netzes

Definition 6.9 a) Wenn $\mathcal{N}_2 = \mathcal{N}_1[Y]$ eine einfache Vergrößerung von \mathcal{N}_1 für eine Platz- oder Transitions-berandete Menge Y ist, dann heißt \mathcal{N}_1 einfache Verfeinerung (simple refinement) von \mathcal{N}_2 .

Für eine Menge $\{Y_1, Y_2, \dots, Y_n\}$ von paarweise disjunkten, Platz- oder Transitions-berandeten Teilmengen von $P_1 \cup T_1$ wird $\mathcal{N}_2 = (\dots((\mathcal{N}_1[Y_1])[Y_2])\dots[Y_n])$ Vergrößerung (abstraction) von \mathcal{N}_1 genannt und \mathcal{N}_1 ist eine Verfeinerung (refinement) von \mathcal{N}_2 . \mathcal{N}_2 wird durch $\mathcal{N}_2 = \mathcal{N}_1[Y_1, Y_2, \dots, Y_n]$ bezeichnet.

b) Eine Vergrößerung $\mathcal{N}_2 = \mathcal{N}_1[Y_1, Y_2, \dots, Y_n]$ von \mathcal{N}_1 wird als Faltung (folding) bezeichnet, wenn jedes Y_i entweder eine Menge von Plätzen, d.h. $Y_i \subseteq P_1$, oder eine Menge von Transitionen, d.h. $Y_i \subseteq T_1$, ist.

In der Definition einer strikten Abstraktion wird Y_i im ersten Fall durch einen Platz p_{Y_i} und im zweiten Fall durch eine Transition t_{Y_i} ersetzt. \mathcal{N}_1 wird strikte Verfeinerung von \mathcal{N}_2 genannt.

Durch folgende Konvention können Mehrdeutigkeiten vermieden werden: falls in a) oder b) eine Menge Y_i ($1 \leq i \leq n$) sowohl Platz- als auch Transitions-berandet ist, kann die Vergrößerung durch $\mathcal{N}_2 = \mathcal{N}_1[Y_1, \dots, Y_i^{(d)}, \dots, Y_n]$ bezeichnet werden, wobei $d = p$ bzw. $d = t$ ist und Y_i als a Platz- bzw. Transitions-berandete Menge betrachtet wird.

Abb. 6.10 zeigt eine nicht einfache Vergrößerung. Das obere Netz stellt das aus der Vorlesung F4 bekannte Beispielnetz zum Start eines Autorennens dar. Dabei sind die vergrößerten Mengen $Y = \{t_1, t_2, t_3, t_4, t_5, p_2, p_4, p_5, p_8, p_9, p_{11}\}$, $Y_1 = \{t_6, p_{13}, t_7\}$ und $Y_2 = \{t_8, p_{15}, t_9\}$, in der oberen Abbildung durch eine gestrichelte Linien dargestellt. Es handelt sich um drei Transitions-berandete Mengen, die zu den Transitionen t_Y , t_{Y_1} und t_{Y_2} im Netz $\mathcal{N}[Y, Y_1, Y_2]$ der unteren Abbildung verwandelt werden.

Es ist einfach zu zeigen, dass die Vergrößerung eines Netzes wieder ein Netz ist, d.h. der Definition 6.1 genügt. Die Vergrößerung im unteren Teil von Abb. 6.10 hat sinngemäß das entsprechende Verhalten des Netzes darüber. Eine Vergrößerung muss jedoch

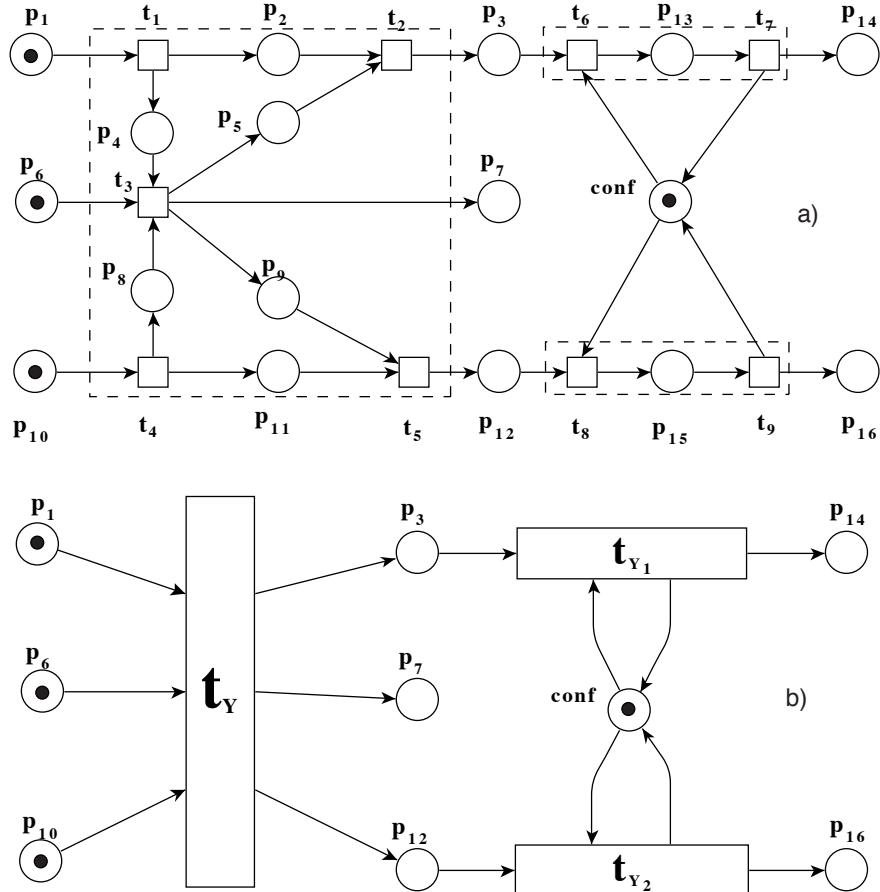


Abbildung 6.10: Eine (nicht einfache) Vergrößerung

nicht eine sinnvolle Interpretation haben. Insbesondere überträgt sich der Begriff nicht zwangsläufig auf seine Semantik (d.h. die Menge seiner Prozesse). Als Beispiel betrachte man das Netzfragment in Abb. 6.11 a). Intuitiv hat die Vergrößerung in Abb. 6.11 d) das entsprechende Verhalten: Marken können von links nach rechts „durchlaufen“. Allerdings ist dieses Netz auch Vergrößerung von Abb. 6.11 b). Das Verhalten ist hier jedoch völlig verschieden.

Die Vergrößerung in Abb. 6.11d) lässt sich in diesem Fall sinnvoll interpretieren, und zwar als Verschmelzung zweier Plätze als Schnittstelle zweier Komponenten. Die Operation wird als *Verschmelzung* oder *Fusion* bezeichnet und zuweilen wie in Abbildung 6.11 c) dargestellt. Die Abbildungen 6.11 e)-h) zeigen die entsprechenden Fälle für Transitionenberandete Mengen. Es handelt sich um die *Verschmelzung* oder *Fusion* von Transitionen.

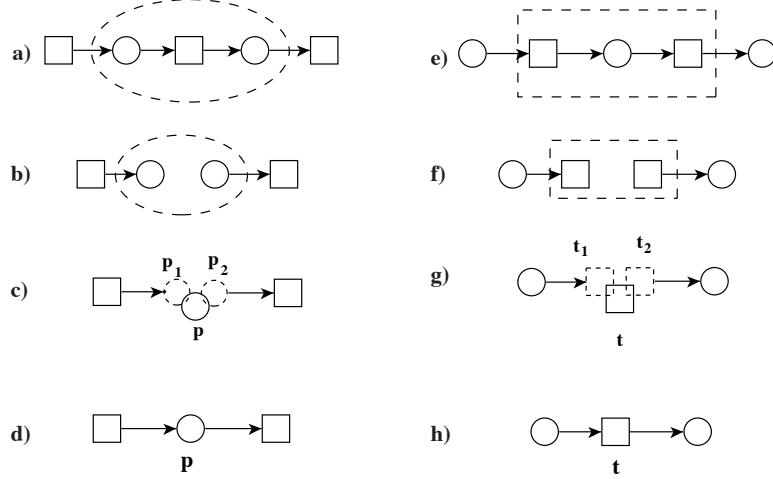


Abbildung 6.11: Vergrößerung und Verschmelzung (Fusion)

Mit Abb. 6.12 sind zwei Netze gegeben, deren Komposition durch Platzverschmelzung das Netz von Abb. 6.10 a) ergibt.

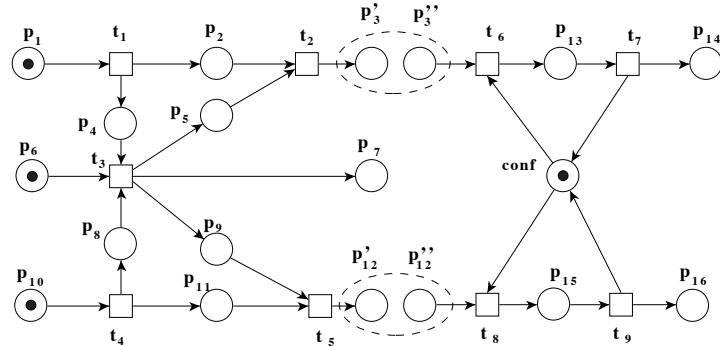


Abbildung 6.12: Komposition durch Verschmelzung (Fusion)

6.1.3 Netzmorphismen

Wie in algebraischen Theorien allgemein üblich, werden strukturerhaltende Abbildungen *Homomorphismen* oder kürzer *Morphismen* genannt. Im folgenden wird Strukturverträglichkeit für Netze eingeführt.

Unter Strukturverträglichkeit ist zu verstehen, dass bei einer Abbildung ϕ eines Netzes $\mathcal{N}_1 = (P_1, T_1, F_1)$ auf $\mathcal{N}_2 = (P_2, T_2, F_2)$ die Kanten F „zusammen“ mit den Plätzen