

8 Anwendungsmodellierung mit Petrinetzen

In diesem Kapitel werden Platz/Transitions-Netze zu solchen Netzen erweitert, die statt der einfachen, „schwarzen“ Marken, allgemeinere Datentypen wie **integer** o.ä. erlauben. Dies erlaubt große Netze in sehr viel kleinerer und kompakterer Form darzustellen, wobei viele Eigenschaften der Platz/Transitions-Netze übertragbar sind.¹ Bücher zu gefärbten Netzen sind insbesondere [GV03] (Kapitel 1 bis 4), [JK09], [Rei10] und der Artikel [Kum01] zu Referenznetzen.

Als erster Schritt beim Übergang von Platz/Transitions- zu gefärbten Netzen werden *kantenkonstante Netze* betrachtet, die zwar schon die genannten Farbmengen haben, aber noch keine Verschmelzung von ähnlichen Transitionen erlauben. Dies wird erst im nächsten Schritt durch die Einführung von Variablen als Kantengewicht möglich, wodurch gefärbte Petrinetze eingeführt werden (Abschnitt 8.2). Anschließend wird in Abschnitt 8.5 das Werkzeug RENEW kurz beschrieben, das den graphischen Entwurf unterstützt und eine Ausführung des Schaltens erlaubt.

8.1 Kantenkonstante Netze

In dem Beispielnetz von Abbildung 6.5 (auf Seite 89) wurden die Objekte *Wagen a* und *Wagen b* durch einfache Marken repräsentiert. Diese Marken sind an sich nicht zu unterscheiden. In dem Netz 6.5 werden sie nur durch ihre Lage in den unterschiedlichen Plätzen p_1 und p_{10} gekennzeichnet. In Hinblick auf eine direktere Modellierung der realen Situation und um ein kompakteres Modell zu erhalten, wäre es vorteilhaft die Rennwagen direkt durch unterscheidbare Marken (Bezeichner) a und b auf *einem* Platz darzustellen, wie dies in Abb. 8.1 durch den Platz $p_{1\&10}$ gezeigt wird. Dieser Platz repräsentiert gleichsam die Aufstellungszone für den Rennbeginn.

8.1.1 Grundlagen

Unterscheidbare Marken werden als Sorten oder Typen gruppiert, die Farben heißen. Jedem Platz p wird durch $cd(p)$ (engl. colour domain) eine Farbe zugeordnet, hier also

¹Dies gilt jedoch nicht für die Eigenschaft der Beschränktheit, die (wie in Abschnitt 7.4 gezeigt) hier nicht entscheidbar ist. Enthält das Netz jedoch nur endliche Datentypen (die hier *Farben* genannt werden), so sind gefärbte Netze gleichmächtig zu Platz/Transitions-Netzen (die Beschränktheit ist dann z.B. entscheidbar) und der genannte Kompaktifizierungs-Vorteil bleibt bestehen.

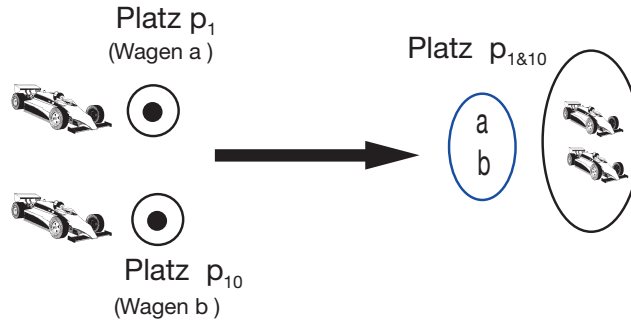
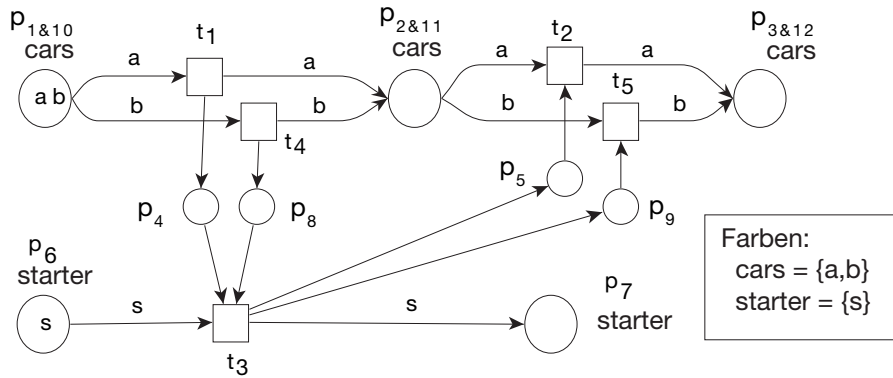


Abbildung 8.1: Übergang zu individuellen Marken

$cd(p_{1\&10}) = cars = \{a, b\}$ (in der grafischen Darstellung kursiv beim Bezeichner des Platzes). Andere Beispiele von Farben sind *integer* oder *boolean*. Wie im Netz 8.2 zu sehen, spezifizieren Ausdrücke an Kanten wie “a”, welche Marke entfernt bzw. hinzugefügt wird.

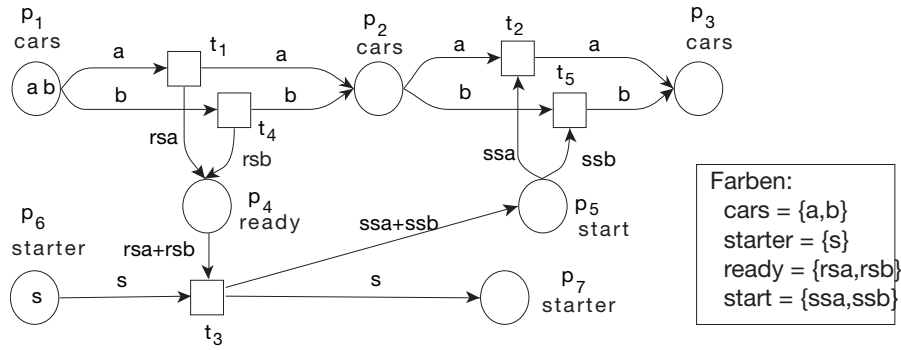

Abbildung 8.2: Das kantenkonstante gefärbte Netz \mathcal{N}_1

Plätzen ohne Farbspezifikation wird per default die Farbe $token = \{\bullet\}$, also die bekannte Marke zugeordnet. Solche Netze heißen *kantenkonstante gefärbte Netze*, da die Kanten mit Ausdrücken über Konstanten (und nicht wie später mit Ausdrücken über Konstanten und Variablen) gewichtet sind.

Das Netz 8.3 ist ein weiteres Beispiel, in dem auch die Nachrichten als Farben *ready* und *start* modelliert sind. Die Kante (p_4, t_3) ist mit dem Ausdruck $rsa + rsb$ gewichtet. Dies bedeutet, dass die Menge $\{rsa, rsb\}$ von p_4 abzuziehen ist. In Netzen werden neben Mengen auch Multimengen benutzt, in denen Elemente mehrfach auftreten können. Multimengen werden als Abbildungen oder als formale Summen dargestellt.

Definition 8.1 Eine Multimenge bg (engl. *bag*, *multi-set*), über einer nicht leeren Menge A , ist eine Abbildung $bg : A \rightarrow \mathbb{N}$, die auch als formale Summe $\sum_{a \in A} bg(a) \cdot a$ dargestellt wird.

\emptyset bezeichnet die leere Multimenge mit $\emptyset(a) = 0$ für alle $a \in A$.

Abbildung 8.3: Das kantenkonstante gefärbte Netz \mathcal{N}_2

$Bag(A)$ bezeichnet die Menge aller Multimengen über A .

$|bg| := \sum_{a \in A} bg(a)$ ist die Mächtigkeit (oder Kardinalität) von bg .

Die Mengenoperationen Vereinigung, Inklusion und Differenz werden punktweise auf $Bag(A)$ als Summe (+), Inklusion (\leq) und Differenz ($-$) übertragen:

- $bg_1 + bg_2 := \sum_{a \in A} (bg_1(a) + bg_2(a))'a$
- $bg_1 \leq bg_2 : \iff \forall a \in A : bg_1(a) \leq bg_2(a)$
- $bg_1 - bg_2 := \sum_{a \in A} (Max(bg_1(a) - bg_2(a), 0))'a$

Beispiel 8.2 $bg_1 = \{a, a, b, b, b, d\}_b$ ist Multimenge über $A = \{a, b, c, d\}$: (der Index b unterscheidet die schließende Klammer von der Mengenklammer) oder als formale Summe: $bg_1 = 2'a + 3'b + d$.

Mit $bg_2 = a + 2'b + c$ erhält man $bg_1 + bg_2 = 3'a + 5'b + c + d$ und $bg_1 - bg_2 = 1'a + 1'b + 0'c + 1'd = a + b + d$.

Multimengen wie $\{a, b, d\}_b$, die Mengen sind, werden auch als Mengen dargestellt: $\{a, b, d\}$

8.1.2 Definition

Die Struktur eines kantenkonstanten gefärbten Petrinetzes oder kurz eines kantenkonstanten Netzes liefert den Rahmen für die jeweiligen Systemmodelle:

Definition 8.3 Ein kantenkonstantes gefärbtes Petrinetz (KKN) wird als Tupel $\mathcal{N} = (P, T, F, \mathcal{C}, cd, W, \mathbf{m}_0)$ definiert, wobei

- (P, T, F) ein endliches Netz (Def. 6.1) ist,
- \mathcal{C} ist eine Menge von Farbenmengen,
- $cd: P \rightarrow \mathcal{C}$ ist die Farbzuzuweisungsabbildung (colour domain mapping). Sie wird durch $cd: F \rightarrow \mathcal{C}$, $cd(x, y) := \text{if } x \in P \text{ then } cd(x) \text{ else } cd(y) \text{ fi}$ auf F erweitert.
- $W: F \rightarrow Bag(\bigcup \mathcal{C})$ mit $W(x, y) \in Bag(cd(x, y))$ heißt Kantengewichtung.

- $\mathbf{m}_0 : P \rightarrow \text{Bag}(\bigcup \mathcal{C})$ mit $\mathbf{m}_0(p) \in \text{Bag}(cd(p))$ für alle $p \in P$ ist die Anfangsmarkierung.

Für das kantenkonstante Netz 8.3 ist beispielsweise $cd(p_4) = \text{ready} = \{rsa, rsb\}$, $cd((p_4, t_3)) = cd(p_4) = \{rsa, rsb\}$, $W(p_4, t_3) = 1'rsa + 1'rsb \in \text{Bag}(cd(p_4)) = \text{Bag}(\{rsa, rsb\})$ und $\mathbf{m}_0(p_1) = 1'a + 1'b \in cd(p_1)$.

8.1.3 Verhalten

Wie für P/T-Netze wird nun das Verhalten der oben definierten kantenkonstanten Netze definiert:

Definition 8.4 Sei $\mathcal{N} = (P, T, F, \mathcal{C}, cd, W, \mathbf{m}_0)$ ein KKN.

- Die Markierung des KKN \mathcal{N} ist ein Vektor \mathbf{m} mit $\mathbf{m}(p) \in \text{Bag}(cd(p))$ für jedes $p \in P$.
(Dies kann auch als Abbildung $\mathbf{m} : P \rightarrow \text{Bag}(\bigcup \mathcal{C})$ mit $\mathbf{m}(p) \in \text{Bag}(cd(p))$ für jedes $p \in P$ aufgefasst werden.)
- Eine Transition $t \in T$ heißt aktiviert in einer Markierung \mathbf{m} falls $\forall p \in \bullet t. \mathbf{m}(p) \geq W(p, t)$ (als Relation: $\mathbf{m} \xrightarrow{t}$).
- Die Erweiterung von W auf $(P \times T) \cup (T \times P)$ ist definiert als:

$$\widetilde{W}(x, y) := \begin{cases} W(x, y) & \text{falls } (x, y) \in F \\ \emptyset & \text{sonst} \end{cases}$$

Ist t in \mathbf{m} aktiviert, dann ist die Nachfolgemarkierung definiert durch $\mathbf{m}'(p) = (\mathbf{m}(p) - \widetilde{W}(p, t)) + \widetilde{W}(t, p)$. (Beachte, dass es sich um Multimengenoperationen handelt!)

- Wir definieren $W(\bullet, t) := (\widetilde{W}(p_1, t), \dots, \widetilde{W}(p_{|P|}, t))$ als Vektor der Länge $|P|$ – und sinngemäß ebenso $W(t, \bullet)$.

Aktivierung ergibt sich dann als: $\mathbf{m} \xrightarrow{t} \iff \mathbf{m} \geq W(\bullet, t)$.

Die Nachfolgemarkierung ist $\mathbf{m}' = \mathbf{m} - W(\bullet, t) + W(t, \bullet)$.

Dabei sind die Multimengenoperatoren komponentenweise auf Vektoren zu erweitern.

Beispiel 8.5 Für das Netz 8.3 ist die Anfangsmarkierung (als Vektor dargestellt):

$$\mathbf{m}_0 = (a + b, \emptyset, \emptyset, \emptyset, \emptyset, s, \emptyset).$$

Da

$$W(\bullet, t_1) = (a, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$$

ist die Transition t_1 in \mathbf{m}_0 aktiviert und die Nachfolgemarkierung ist

$$\begin{aligned} \mathbf{m}' &= \mathbf{m}_0 + W(t_1, \bullet) - W(\bullet, t_1) \\ &= (a + b, \emptyset, \emptyset, \emptyset, \emptyset, s, \emptyset) + (\emptyset, a, \emptyset, rsa, \emptyset, \emptyset, \emptyset) - (a, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \\ &= (b, a, \emptyset, rsa, \emptyset, s, \emptyset) \end{aligned}$$

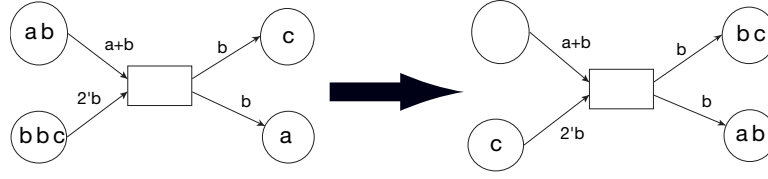


Abbildung 8.4: Schaltregel für kantenkonstante Netze

$a + b$ bzw. rsa sind hier beispielsweise die Multimengen $\{a, b\}_b$ bzw. $\{rsa\}_b$, dargestellt als formale Summen.

Die Schaltregel wird in Abb. 8.4 an einem abstrakten Beispiel erläutert, wobei Multimengen vorkommen, die keine Mengen sind!

Definition 8.6 Die Nachfolgemarkierungsrelation von Definition 8.4 wird wie üblich auf Wörter über T erweitert:

- $\mathbf{m} \xrightarrow{w} \mathbf{m}'$ falls w das leere Wort λ ist und $\mathbf{m} = \mathbf{m}'$,
- $\mathbf{m} \xrightarrow{wt} \mathbf{m}'$ falls $\exists \mathbf{m}'' : \mathbf{m} \xrightarrow{w} \mathbf{m}'' \wedge \mathbf{m}'' \xrightarrow{t} \mathbf{m}'$ für $w \in T^*$ und $t \in T$.

Für eine Menge S^0 von Markierungen sei dann $\mathbf{R}(\mathcal{N}, S^0) := \{\mathbf{m}_2 | \exists w \in T^*, \mathbf{m}_1 \in S^0 : \mathbf{m}_1 \xrightarrow{w} \mathbf{m}_2\}$ die Menge der von S^0 aus erreichbaren Markierungen und $\mathbf{R}(\mathcal{N}) := \mathbf{R}(\mathcal{N}, \{\mathbf{m}_0\})$ die Erreichbarkeitsmenge von $\mathbf{R}(\mathcal{N})$.

Falls \mathcal{N} implizit gegeben ist, schreiben wir auch $\mathbf{R}(\mathbf{m})$ für $\mathbf{R}(\mathcal{N}, \{\mathbf{m}\})$. Eine Transitionsfolge $w \in T^*$ heißt aktiviert in \mathbf{m} (in Zeichen: $\mathbf{m} \xrightarrow{w}$), falls $\exists \mathbf{m}_1 : \mathbf{m} \xrightarrow{w} \mathbf{m}_1$.

$FS(\mathcal{N}) := \{w \in T^* \mid \mathbf{m}_0 \xrightarrow{w}\}$ ist die Menge der Schaltfolgen (firing sequence set) von \mathcal{N} .

Der Erreichbarkeitsgraph eines KKN \mathcal{N} ist ein Tupel $RG(\mathcal{N}) := (Kn, Ka)$ mit Knotenmenge $Kn := \mathbf{R}(\mathcal{N})$ und Kantenmenge $Ka := \{(\mathbf{m}_1, t, \mathbf{m}_2) \mid \mathbf{m}_1 \xrightarrow{t} \mathbf{m}_2\}$

Damit ist die Semantik eines kantenkonstanten gefärbten Petrinetzes durch ein Transitionssystem definiert: Der Erreichbarkeitsgraph eines kantenkonstanten gefärbten Netzes \mathcal{N} kann als Transitionssystem $TS = (\mathbf{R}(\mathcal{N}), T, Ka, \{\mathbf{m}_0\})$ aufgefasst werden. Dann ist $R(TS) = \mathbf{R}(\mathcal{N})$ und $FS(TS) = FS(\mathcal{N})$.

Beispiel 8.7 Es folgt die Markierungs-/Transitionsfolge für die Schaltfolge $t_4, t_1, t_3, t_2, t_5 \in FS(\mathcal{N}_2)$, wobei zur Abkürzung die Multimengen als Mengen geschrieben sind.

$$\begin{pmatrix} \{a, b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_4} \begin{pmatrix} \{a\} \\ \{b\} \\ \emptyset \\ \{rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_1} \begin{pmatrix} \emptyset \\ \{a, b\} \\ \emptyset \\ \{rsa, rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_3} \begin{pmatrix} \emptyset \\ \{a, b\} \\ \emptyset \\ \{ssa, ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{t_2} \begin{pmatrix} \emptyset \\ \{b\} \\ \{a\} \\ \emptyset \\ \{ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{t_5} \begin{pmatrix} \emptyset \\ \emptyset \\ \{a, b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \end{pmatrix}$$

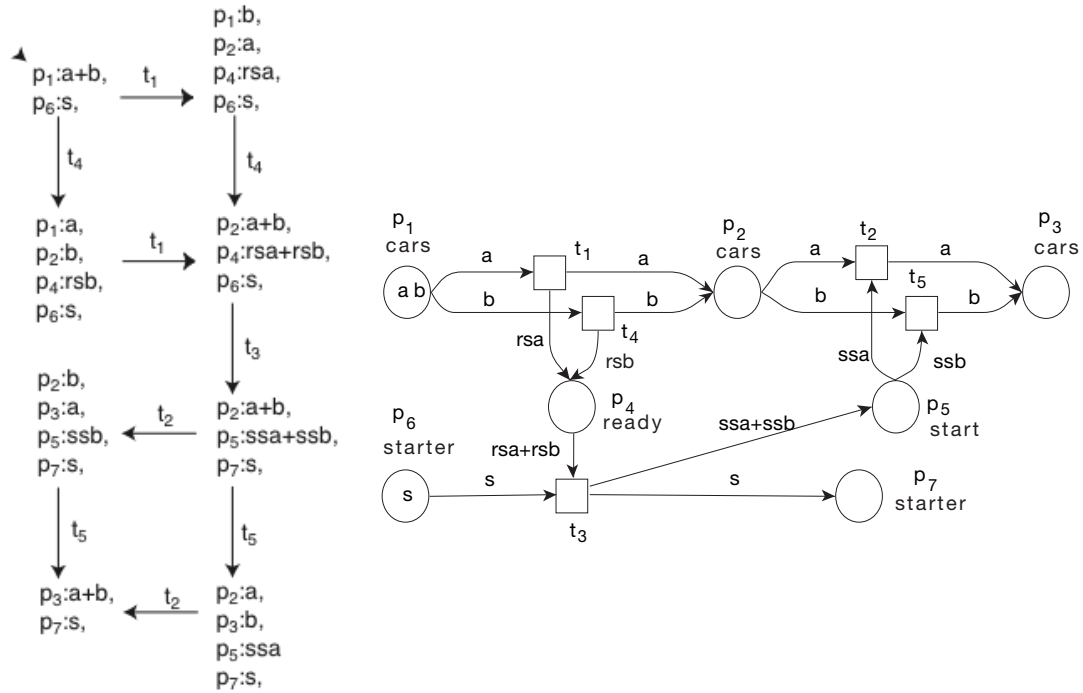


Abbildung 8.5: Erreichbarkeitsgraph des kantenkonstanten Netzes (links) von Abb. 8.3 (rechts)

Abbildung 8.5 zeigt beispielhaft den Erreichbarkeitsgraphen des kantenkonstanten Netzes von Abb. 8.3.

8.2 Gefärbte Netze

Bei der Einführung von kantenkonstanten Netzen wurden Plätze verschmolzen. Um ein Netz noch kompakter zu machen, liegt es nahe, auch Transitionen zusammenzulegen, insbesondere wenn sie verhaltensähnliche Aktionen modellieren. Wenn dabei das gleiche Verhalten dargestellt werden soll, dann ist die zusammengelegte Transition zu parametrisieren.

8.2.1 Grundlagen

Die Zusammenfassung von Plätzen beim Übergang von einfachen Netzen über kantenkonstante Netze zu gefärbten Netzen ist im oberen Teil von Abbildung 8.6 als Vergrößerung von Plätzen („Platz-Faltung“) dargestellt.

Die Zusammenfassung kann durch Variablen geschehen. Beispielsweise modelliert die Transition t_1 im Netz in Abb. 8.8 durch die Variablenbelegung $\beta_1 := [x = a, y = rsa]$ die Transition t_1 im Netz in Abb. 8.3, während mit $\beta_2 := [x = b, y = rsb]$ die Transition t_4 in Abb. 8.3 simuliert wird.

Definition 8.8 Sei $Var := \{x_1, x_2, x_3, \dots\}$ eine Menge von Variablen mit Wertebereichen $dom(x)$ für jedes $x \in Var$. Eine Belegung ist eine Zuordnung (Abbildung) $\beta = [x_1 = u_1, x_2 = u_2, x_3 = u_3, \dots]$ von Werten $u_i \in dom(x_i)$ zu den Variablen.

Natürlich wäre $\beta_3 := [x = a, y = rsb]$ keine zulässige Belegung, denn dies wird durch Schutzbedingungen (guards) ausgeschlossen. Guards sind über den Variablen an einer Transition definierte Prädikate. Sie können auch als Testbedingung an einer Verzweigung eingesetzt werden.

Allgemein stehen an den Kanten eines solchen gefärbten Netzes Ausdrücke über den Variablen, die mit einer zu wählenden Belegung Multimengen definieren, die dann die gleiche Rolle wie bei kantenkonstanten Netzen spielen. Die Form der Ausdrücke lassen wir hier offen, um flexibel zu bleiben. Wichtig ist nur, dass sie zu einer Belegung eine passende Multimenge liefern, d.h. eine Multimenge über der Farbe des angrenzenden Platzes. Entsprechendes gilt für Guards. Wichtig ist hier, dass sie zu einer Belegung einen Wahrheitswert liefern, der über die Zulässigkeit der Belegung entscheidet. Im übrigen ist die Definition eines gefärbten Netzes derjenigen eines kantenkonstanten Netzes ähnlich.

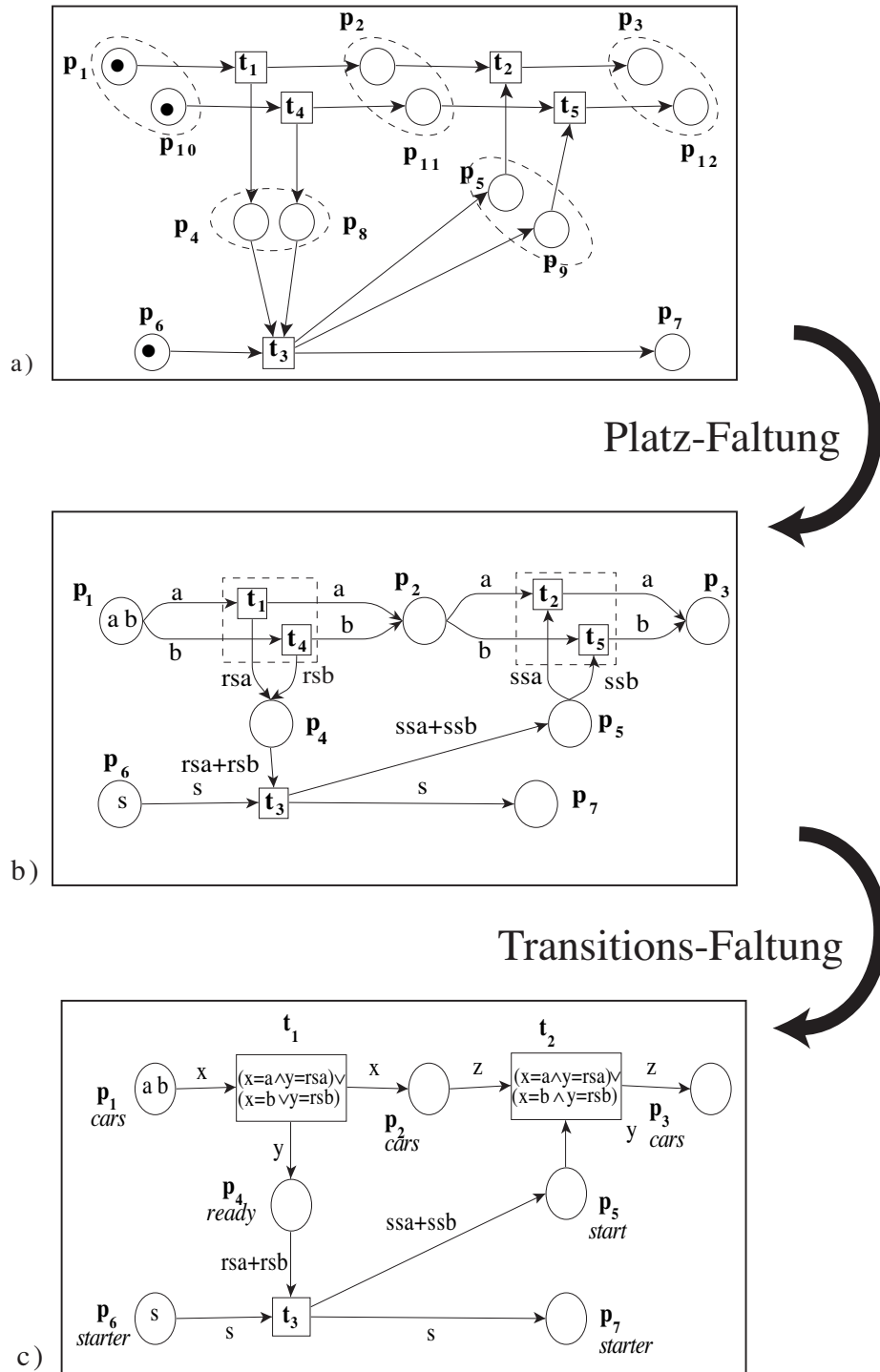


Abbildung 8.6: Netzfaltung zum kantenkonstanten und gefärbten Netz

8.2.2 Definition

Die Struktur eines gefärbten Petrinetzes oder kurz eines gefärbten Netzes liefert den Rahmen für die jeweiligen Systemmodelle:

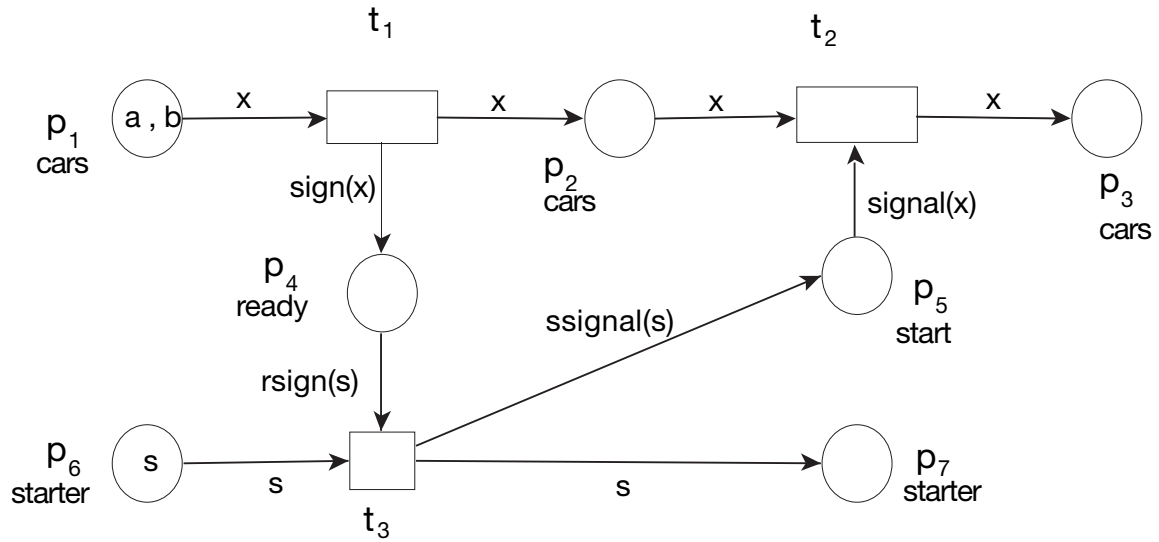
Definition 8.9 Ein gefärbtes Petrinetz (engl. *coloured Petri net*, *CPN*) wird als Tupel

$$\mathcal{N} = (P, T, F, \mathcal{C}, cd, Var, Guards, \widehat{W}, \mathbf{m}_0)$$

definiert, wobei gilt:

- (P, T, F) ist ein endliches Netz (Def. 6.1),
- \mathcal{C} ist eine Menge von Farbenmengen,
- $cd: P \rightarrow \mathcal{C}$ ist die Farbzuzuweisungsabbildung (*colour domain mapping*). Sie wird durch $cd: F \rightarrow \mathcal{C}$, $cd(x, y) := \text{if } x \in P \text{ then } cd(x) \text{ else } cd(y) \text{ fi}$ auf F erweitert.
- Var ist eine Menge von Variablen mit Wertebereichen $dom(x)$ für jedes $x \in Var$.
- $Guards = \{guard_t \mid t \in T\}$ ordnet jeder Transition $t \in T$ ein Prädikat $guard_t$ mit Variablen aus Var zu.
- $\widehat{W} = \{W_\beta \mid \beta \text{ ist Belegung von } Var\}$ ist eine Menge von Kantengewichtungen der Form $W_\beta: F \rightarrow Bag(\bigcup \mathcal{C})$, wobei $W_\beta(x, y) \in Bag(cd(x, y))$ für alle $(x, y) \in F$ gilt.
- $\mathbf{m}_0: P \rightarrow Bag(\bigcup \mathcal{C})$ mit $\mathbf{m}_0(p) \in Bag(cd(p))$ für alle $p \in P$ ist die Anfangsmarkierung.

Kantenausdrücke von gefärbten Netzen können auch speziell definierte Funktionen enthalten. Diese werden wie in Abb. 8.7 zweckmäßigerweise in den Deklarationsteil aufgenommen. Der Vorteil dieser Darstellung liegt in der Reduzierung der Anzahl der Variablen. Dadurch werden in diesem Beispiel Guards in Transitionen nicht benötigt.



Farben:

$\text{cars} = \{a, b\}$

$\text{starter} = \{s\}$

$\text{ready} = \{rsa, rsb\}$ "ready signs"

$\text{start} = \{ssa, ssb\}$ "start signs"

Variablen: x, s

$\text{dom}(x) = \text{cars}, \text{dom}(s) = \text{starter}$

Funktionen:

$\text{id}_1 : a \rightarrow a; b \rightarrow b$ $\text{id}_2 : s \rightarrow s$

$\text{sign} : a \rightarrow rsa; b \rightarrow rsb$ $\text{signal} : a \rightarrow ssa; b \rightarrow ssb;$

$\text{rsign} : s \rightarrow rsa + rsb;$ $\text{ssignal} : s \rightarrow ssa + ssb$

Abbildung 8.7: Gefärbtes Netz mit definierten Funktionen

8.2.3 Verhalten

Wie für P/T-Netze und die KKN wird nun das Verhalten der oben definierten gefärbten Netze definiert:

Definition 8.10 Sei $\mathcal{N} = (P, T, F, \mathcal{C}, cd, Var, Guards, \widehat{W}, \mathbf{m}_0)$ ein gefärbtes Netz.

- a) Die Markierung des gefärbten Netzes \mathcal{N} ist ein Vektor \mathbf{m} mit $\mathbf{m}(p) \in Bag(cd(p))$ für jedes $p \in P$
(auch als Abbildung $\mathbf{m} : P \rightarrow Bag(\bigcup \mathcal{C})$ mit $\mathbf{m}(p) \in Bag(cd(p))$ für jedes $p \in P$ aufzufassen).
- b) Sei β eine Belegung für Var . Die Transition $t \in T$ heißt β -aktiviert in einer Markierung \mathbf{m} , falls $guard_t(\beta) = true$ und $\forall p \in \bullet t. \mathbf{m}(p) \geq W_\beta(p, t)$
(als Relation: $\mathbf{m} \xrightarrow{t, \beta}$).
- c) Die Erweiterung von W_β auf $(P \times T) \cup (T \times P)$:

$$\widetilde{W}_\beta(x, y) := \begin{cases} W_\beta(x, y) & \text{falls } (x, y) \in F \\ \emptyset & \text{sonst} \end{cases}$$

Ist t in \mathbf{m} β -aktiviert, dann ist die Nachfolgemarkierung definiert durch $\mathbf{m}'(p) = \mathbf{m}(p) - \widetilde{W}_\beta(p, t) + \widetilde{W}_\beta(t, p)$. (Man beachte, dass es sich um Multimengenoperationen handelt.)

- d) Definiert man $W_\beta(\bullet, t) := (\widetilde{W}_\beta(p_1, t), \dots, \widetilde{W}_\beta(p_{|P|}, t))$ als Vektor der Länge $|P|$ und entsprechend $W_\beta(t, \bullet) := (\widetilde{W}_\beta(t, p_1), \dots, \widetilde{W}_\beta(t, p_{|P|}))$, dann kann die Nachfolgemarkierung einfacher durch Vektoren definiert werden: $\mathbf{m}' = \mathbf{m} - W_\beta(\bullet, t) + W_\beta(t, \bullet)$. Dabei sind die Multimengenoperatoren komponentenweise auf Vektoren zu erweitern.
- e) Schalten wird auch ohne Belegung notiert: $\mathbf{m} \xrightarrow{t} \mathbf{m}' : \iff \exists \beta. \mathbf{m} \xrightarrow{t, \beta} \mathbf{m}'$

Die Schaltregel für CPN entspricht der für KKN, wenn man sich (für den Moment des Schaltens) alle Ausdrücke durch die Belegung β als ausgewertet vorstellt und dann mit der Schaltregel für KKN schaltet. Diese Idee illustriert Abb. 8.9.

1. Für die Transition t in Abb. 8.9 a) wählen wir eine Belegung β für $Var(t)$ mit $guard_t(\beta) = true$ (wie in Abb. 8.9 b)).
2. Wir werten mit dieser Belegung die Ausdrücke an den Kanten zu Multimengen aus (wie in Abb. 8.9 c).
3. Wir wenden die Schaltregel für kantenkonstante Netze (Abb. 8.4) an (wie in Abb. 8.9d).

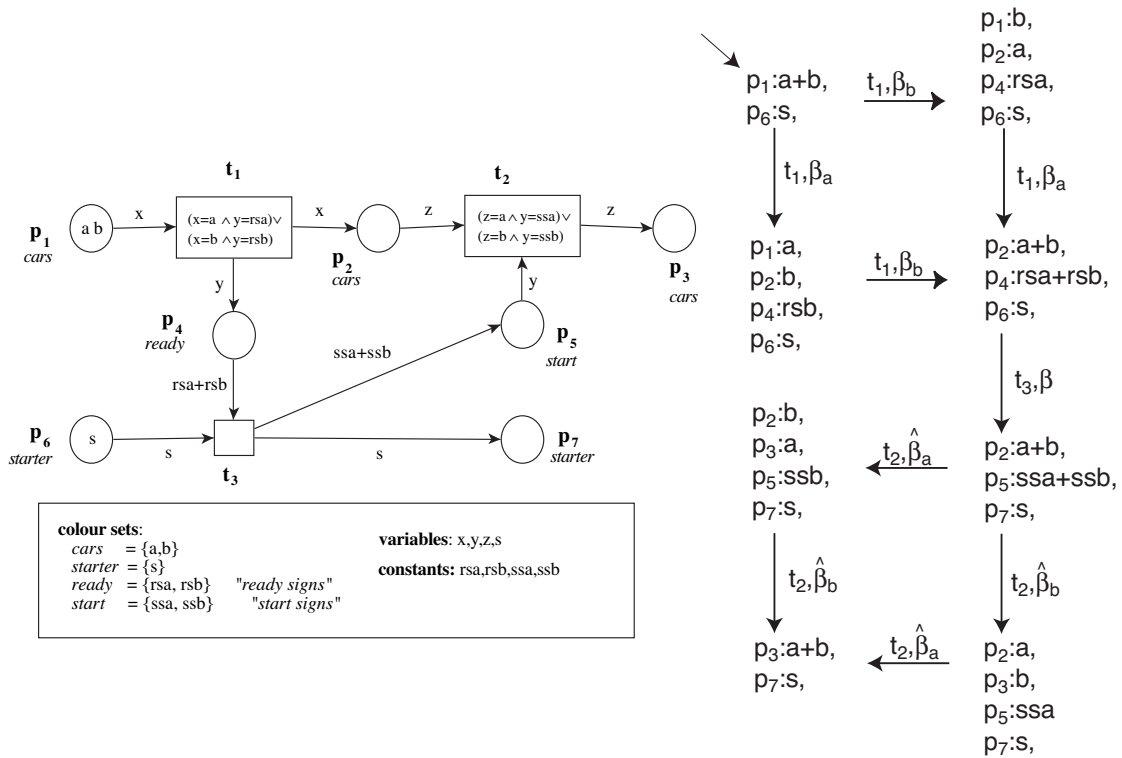


Abbildung 8.8: Gefärbtes Netz und sein Erreichbarkeitsgraph

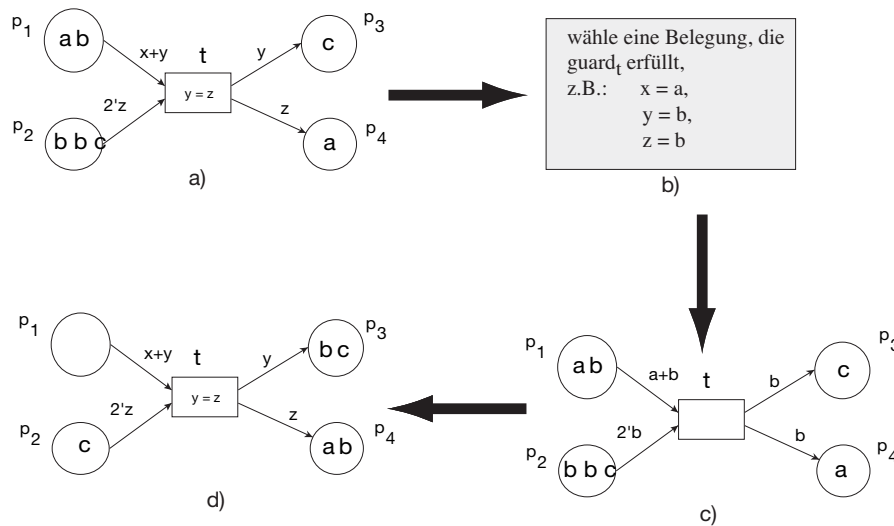


Abbildung 8.9: Schaltregel für gefärbte Netze

Die Definition von Schaltfolgen, erreichbare Markierungen usw. ergeben sich dann ganz analog zu KKN:

Definition 8.11 Die Nachfolgemarkierungsrelation von Definition 8.10 wird wie üblich auf Wörter über T erweitert:

- $\mathbf{m} \xrightarrow{w} \mathbf{m}'$, falls w das leere Wort λ ist und $\mathbf{m} = \mathbf{m}'$,
- $\mathbf{m} \xrightarrow{wt} \mathbf{m}'$, falls $\exists \mathbf{m}'' : \mathbf{m} \xrightarrow{w} \mathbf{m}'' \wedge \mathbf{m}'' \xrightarrow{t} \mathbf{m}'$ für $w \in T^*$ und $t \in T$.

Die Menge $\mathbf{R}(\mathcal{N}) := \{\mathbf{m} \mid \exists w \in T^* : \mathbf{m}_0 \xrightarrow{w} \mathbf{m}\}$ ist die Menge der erreichbaren Markierungen (auch: Erreichbarkeitsmenge).

Eine Transitionsfolge $w \in T^*$ heißt aktiviert in \mathbf{m} (in Zeichen: $\mathbf{m} \xrightarrow{w}$), falls gilt: $\exists \mathbf{m}_1 : \mathbf{m} \xrightarrow{w} \mathbf{m}_1$

$FS(\mathcal{N}) := \{w \in T^* \mid \mathbf{m}_0 \xrightarrow{w}\}$ ist die Menge der Schaltfolgen von \mathcal{N} .

Der Erreichbarkeitsgraph eines gefärbten Netzes \mathcal{N} ist der Graph $RG(\mathcal{N}) := (Kn, Ka)$ mit der Knotenmenge $Kn := \mathbf{R}(\mathcal{N})$ (siehe Def. 8.11) und der Kantenmenge $Ka := \{(\mathbf{m}_1, (t, \beta), \mathbf{m}_2) \mid \mathbf{m}_1 \xrightarrow{t, \beta} \mathbf{m}_2\}$.

Der Erreichbarkeitsgraph ist ein Transitionssystem (ohne Endzustände).

Beispiel 8.12 Für das gefärbte Netz 8.8 ist beispielsweise $cd(p_4) = ready = \{rsa, rsb\}$, $cd((p_4, t_3)) = cd(p_4) = \{rsa, rsb\}$, $W_\beta(p_1, t_1) = 1'a \in Bag(cd(p_1)) = Bag(\{a, b\})$ für $\beta = [x = a, y = rsa]$ und $\mathbf{m}_0(p_1) = 1'a + 1'b \in cd(p_1)$, $guard_{t_1} = (x = a \wedge y = rsa) \vee (x = b \wedge y = rsb)$.

Für das Netz 8.8 ist die Anfangsmarkierung: $\mathbf{m}_0 = (a + b, \emptyset, \emptyset, \emptyset, \emptyset, s, \emptyset)$.

Für $\beta = [x = a, y = rsa]$ und $W_\beta(\bullet, t_1) = (a, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ ist die Transition t_1 in \mathbf{m}_0 aktiviert und die Nachfolgemarkierung ist

$$\begin{aligned} \mathbf{m}' &= \mathbf{m}_0 + W_\beta[t_1, \bullet] - W_\beta(\bullet, t_1) \\ &= (a + b, \emptyset, \emptyset, \emptyset, \emptyset, s, \emptyset) + (\emptyset, a, \emptyset, rsa, \emptyset, \emptyset, \emptyset) - (a, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \\ &= (b, a, \emptyset, rsa, \emptyset, s, \emptyset) \end{aligned}$$

Mit $a + b$ bzw. rsa sind hier beispielsweise die Multimengen $\{a, b\}_b$ bzw. $\{rsa\}_b$ als formale Summen dargestellt.

Eine Markierungs-Schaltfolge für das gefärbte Netz von Abb. 8.8 sieht folgendermaßen aus, wobei die Belegungen $\beta_a = [x = a, y = rsa]$, $\beta_b = [x = b, y = rsb]$ und $\hat{\beta}_a = [x = a, y = ssa]$, $\hat{\beta}_b = [x = b, y = ssb]$ gewählt wurden. Für die Transition t_3 kann eine beliebige Belegung eingesetzt werden, da alle Kantenausdrücke nur Konstante enthalten.

$$\begin{pmatrix} \{a, b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{(t_1, \beta_b)} \begin{pmatrix} \{a\} \\ \{b\} \\ \emptyset \\ \{rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{(t_1, \beta_a)} \begin{pmatrix} \emptyset \\ \{a, b\} \\ \emptyset \\ \{rsa, rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{(t_3, \beta)} \begin{pmatrix} \emptyset \\ \{a, b\} \\ \emptyset \\ \{ssa, ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{(t_2, \hat{\beta}_a)} \begin{pmatrix} \emptyset \\ \{b\} \\ \{a\} \\ \emptyset \\ \{ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{(t_2, \hat{\beta}_b)} \begin{pmatrix} \emptyset \\ \emptyset \\ \{a, b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \end{pmatrix}$$

Der Erreichbarkeitsgraph des gefärbten Netzes ist in Abbildung 8.8, rechts dargestellt.

8.2.4 Ausdrucksmächtigkeit

Gefärbte Netze mit der Farbe *integer* besitzen einen engen Bezug zu P/T-Netzen. Vergleiche das gefärbte Netz mit dem P/T-Netz in Abb. 8.10: Beide stellen einen Zähler dar.

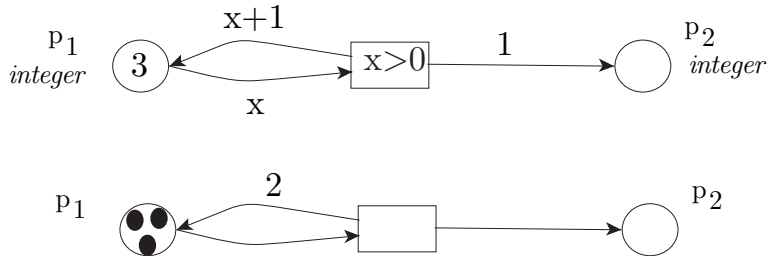


Abbildung 8.10: Gefärbtes Netz und P/T-Netz mit ähnlichem Verhalten

Der entscheidende Unterschied ist aber, dass im CPN nur eine Marke mit dem Wert $n = 3$ auf der Zählerstelle liegt – nicht aber $n = 3$ Marken. Daher kann der Wert des Zählers als Ganzes erfasst werden, insbesondere auch auf den Wert $n = 0$ getestet werden. Also kann man mit CPN auch Zählerautomaten modellieren (vgl. Abschnitt 7.4), woraus eine Vielzahl von Unentscheidbarkeitsresultaten folgt. Das zentrale Resultat ist das folgende:

Satz 8.13 *Das Erreichbarkeitsproblem ist für CPN unentscheidbar.*

Gefärbte Netze mit ausschließlich endlichen Farbmengen sind dagegen nicht mächtiger als P/T-Netze, da sie immer zu solchen aufgefaltet werden können (vgl. Abbildung 8.6). Eine Übersicht über die Turing-Mächtigkeit bei Petrinetzen gibt die Tabelle 7.1.

8.2.5 Beispiel: Der Datenbank-Manager

Das Beispiel *Datenbank-Manager* [Jen87] ist ein schönes Beispiel eines gefärbten Netzes mit einer größeren Anzahl von gefärbten Marken.

Eine Menge von $n > 0$ Datenbanken soll von n Prozessen, genannt *Datenbank-Manager*, $DBM = \{d_1, d_2, \dots, d_n\}$ so verwaltet werden, dass sie immer den gleichen Inhalt haben. Um dies zu erreichen, sollen die Manager miteinander kommunizieren. Jeder Manager kann seine eigene Datenbank aktualisieren. Dabei muss er an jeden anderen Manager eine Nachricht senden, die diesen über die Aktualisierung informiert. Der Manager muss warten, bis alle anderen diese Nachricht erhalten, die Aktualisierung durchgeführt und eine entsprechende Rückmeldung zurückgesandt haben. Erst dann kehrt der Manager in den Zustand *inactive* zurück.

Dabei sollen weder die Datenbanken noch ihre Aktualisierung dargestellt werden, sondern nur der Nachrichtenaustausch.

Zustände der Manager : *inactive, waiting, performing*

Nachrichten : $MS = \{(s, r) | s, r \in DBM \wedge s \neq r\}$

Zustände der Nachrichtenkanäle : *unused, sent, received, acknowledged*

wechselseitiger Ausschluss : *exclusion*

Spezifikation für das gefärbte Netz von Abb. 8.11:

Farben : $DBM = \{d_1, d_2, \dots, d_n\}$
 $MS = \{(s, r) | s, r \in DBM \wedge s \neq r\}$
 $E = \{e\}$

Variablen : $Var = \{e, r, s\}$
 $dom(e) = E, \quad dom(r) = dom(s) = DBM$

Funktionen :

$MINE : DBM \rightarrow Bag(MS) \quad MINE(s) := \sum_{r \neq s} (s, r)$
 $REC : MS \rightarrow DBM \quad REC((s, r)) := r$
 $ABS : DBM \rightarrow E \quad ABS(s) := e$

Anfangsmarkierung: $\mathbf{m}_0(p) := \begin{cases} DBM & \text{falls } p = inactive \\ MS & \text{falls } p = unused \\ \{e\} & \text{falls } p = exclusion \\ \emptyset & \text{sonst} \end{cases}$

Nicht alle Funktionen dieser Spezifikation werden in der graphischen Darstellung von Abb. 8.11 benutzt. Sie sind jedoch nützlich um die Funktion $W_\beta(x, y) \in Bag(cd(x, y))$ (wobei $(x, y) \in F$) aus der Definition 8.9 von gefärbten Netzen zu definieren. Beispielsweise ergibt sich für $(x, y) = (T2, unused)$ und die Belegung $\beta = [s = d_1]$ der Wert $W_\beta(T2, unused) = MINE(d_1)$.

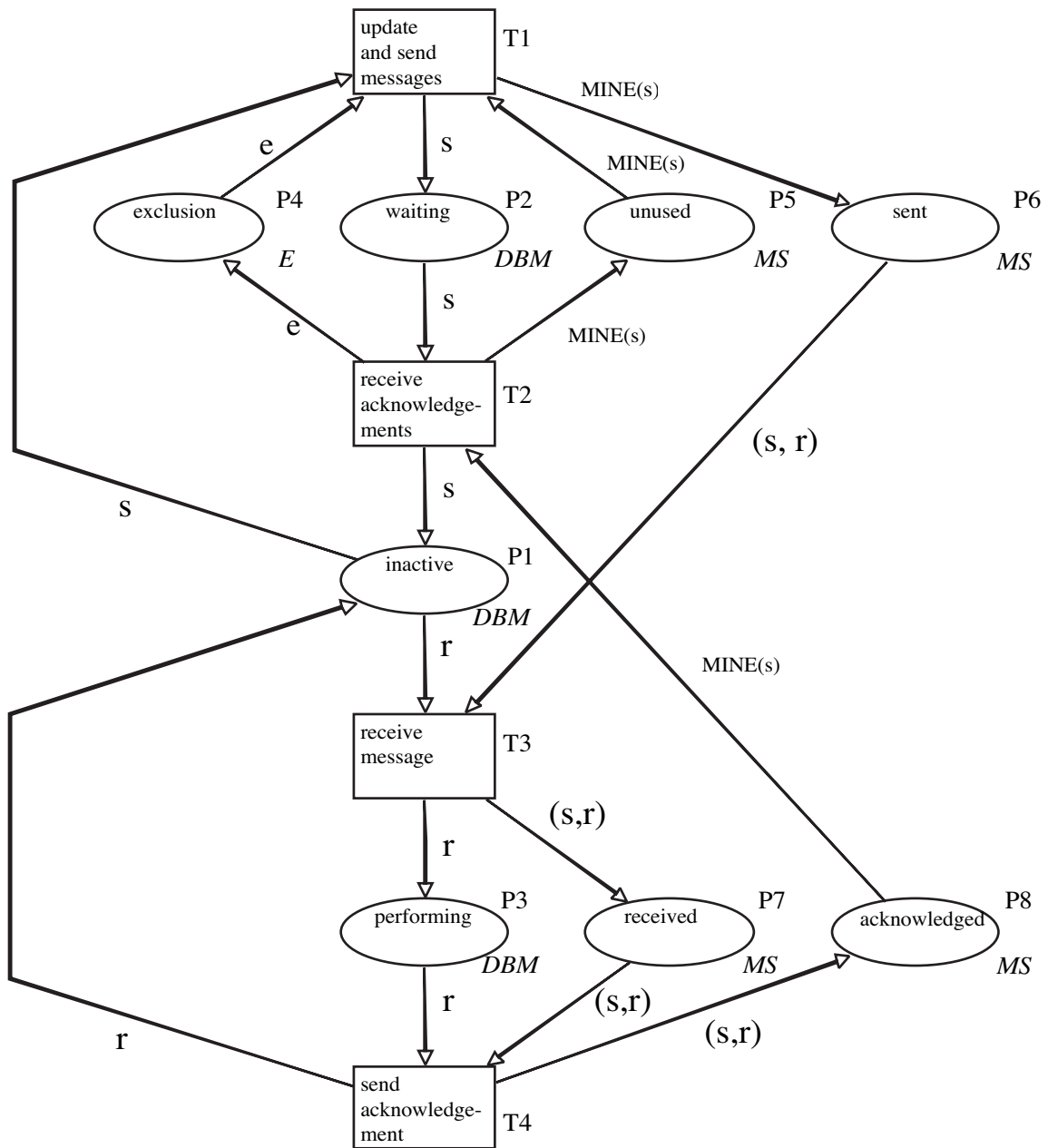


Abbildung 8.11: Datenbank-Manger

8.3 Workflow-Netze

Workflow-Netze oder Workflow-Petrinetze umfassen eine der aktuell wichtigsten Klassen der Petrinetze, insbesondere im Bereich der Wirtschaftsinformatik. Im Folgenden werden die Hintergründe, Definitionen und wichtigsten Konzepte eingeführt.

8.3.1 Historie

Workflow-Systeme steuern und verwalten den Ablauf von Workflow-Prozessen (zu deutsch etwa „Geschäftsprozesse“). In den 60-er Jahren bestanden Informationssysteme aus einer Anzahl von Anwenderprogrammen (engl. application systems, APPL), die direkt auf das Betriebssystem (engl. operating system, OS) aufsetzten (vergl. Abb. 8.12). Für jedes dieser Programme wurde eine eigene Benutzungsschnittstelle und ein eigenes Datenbanksystem entwickelt. In den 70-er Jahren wurde die langfristige Datenhaltung aus den Anwendungsprogrammen (engl. Applications) ausgelagert. Zu diesem Zweck wurden Datenbanksysteme (engl. database management systems, DBMSs) entwickelt. In den 80-er Jahren fand mit den Benutzeroberflächen eine ähnliche Entwicklung statt. Benutzungsschnittstellensysteme (engl. user interface management systems, UIMSs) erlaubten es, die Kommunikation mit dem Benutzer aus dem Anwendungsprogramm auszulagern. Eine ähnliche Entwicklung setzte in den 90-er Jahren mit der Entwicklung von Workflow-Programmen (engl. workflow management systems, WFMSs) ein. Sie erlauben es, die Verwaltung von Geschäftsprozessen aus spezifischen Anwendungsprogrammen herauszuhalten (siehe z.B. [Aal00], [Aal98], [AH02]).

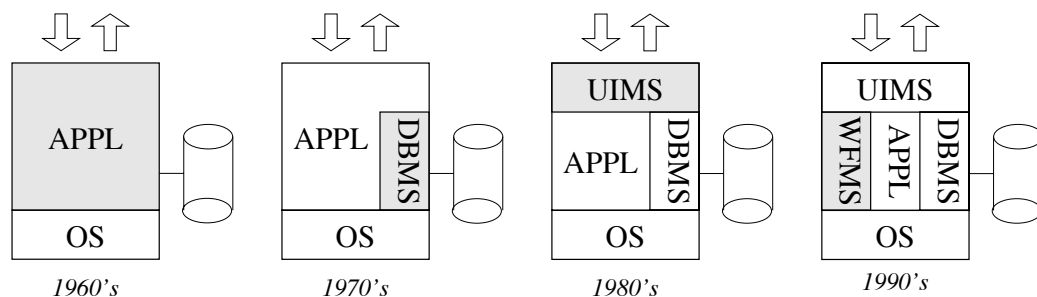


Abbildung 8.12: Workflow-Management-Systeme in historischer Perspektive

Die Aufgabe von Workflow-Systemen (manchmal auch *office logistics* genannt) ist es, sicherzustellen, dass die richtigen Aktionen eines Geschäftsablaufes durch die richtigen Personen zur richtigen Zeit ausgeführt werden. Abstrakter kann man ein Workflow-System als eine Software definieren, die Geschäftsabläufe vollständig modelliert, verwaltet und steuert.

Der Begriff Workflow-System wird häufig nicht so spezifisch gebraucht. Beispielsweise wird so auch Software zur Gruppenkommunikation genannt, die lediglich die Versendung

von Nachrichten und den Austausch von Information unterstützt (wie z.B. Lotus Notes, Microsoft Exchange). Die Abbildung 8.13 setzt allgemeine kooperative Prozesse und Workflow-Systeme in eine Skalierung zwischen informationszentriert und prozesszentriert bzw. weniger und mehr strukturiert.

8.3.2 Modellierung durch Workflow-Netze

Für Workflow-Systeme gibt es hunderte von Modellierungsansätze und Rechnerwerkzeuge, die jedoch häufig keine wohldefinierte und eindeutige Semantik besitzen. Wegen ihrer Orientierung auf Aktionen und deren Koordination bieten sich natürlich besonders Petrinetze an, deren Semantik wohldefiniert und bekannt ist. Ein solcher Ansatz nach [Aal00], [Aal98] und [AH02] wird hier vorgestellt.

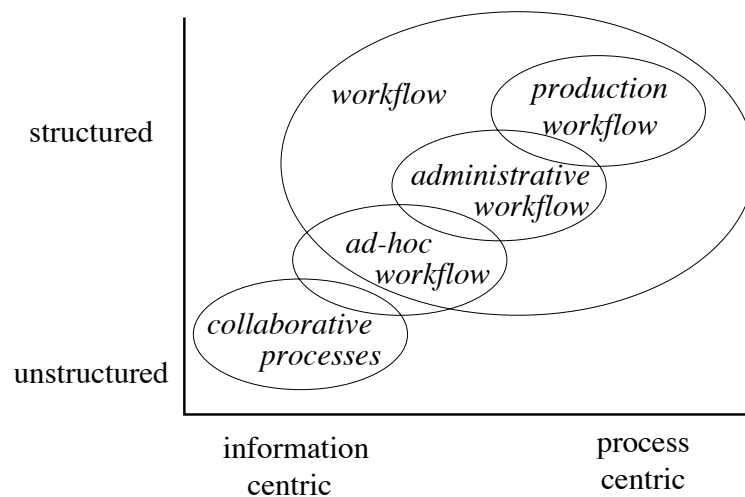


Abbildung 8.13: Workflow Prozesse und kooperative Prozesse im Vergleich

8.3.3 Intuitives Beispiel

Im Folgenden betrachten wir ein Workflow-System zur Beschwerdebearbeitung (siehe Abbildung. 8.14). Es enthält folgende Aktionen:

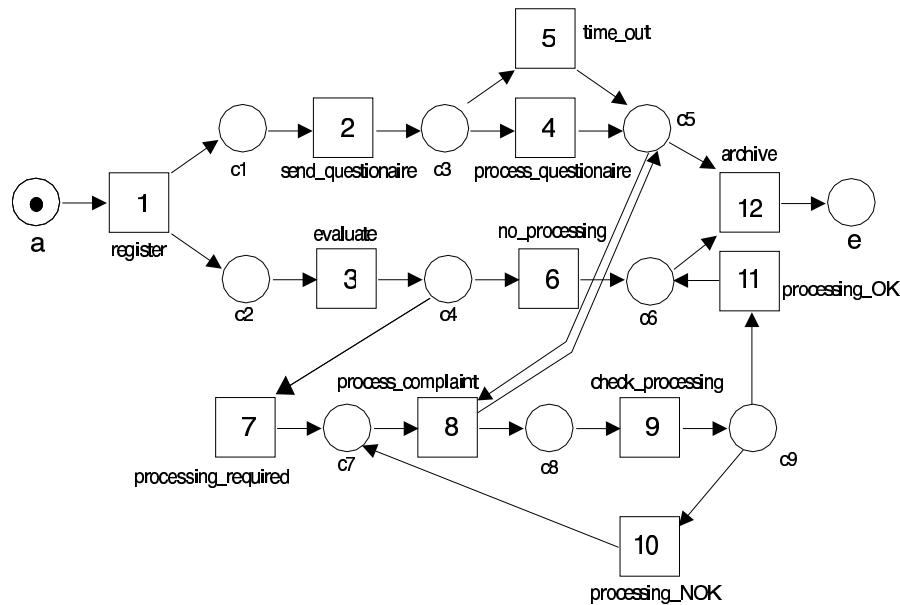


Abbildung 8.14: Ein Petrinetz für einen Vorgang zur Beschwerdebearbeitung

1. Aufnahme einer Beschwerde (register)
2. Fragebogen an Beschwerdeführer (send_questionnaire)
3. Bewertung (evaluate) (nebenläufig zu 2.)
4. Fragebogenauswertung (process_questionnaire), falls Rücklauf innerhalb von 2 Wochen, sonst:
5. Nichtberücksichtigung des Fragebogens (time_out).
6. Je nach Ergebnis der Bewertung (3.): Aussetzung der Bearbeitung (no_processing) oder
7. Beginn der eigentlichen Prüfung (processing_required)
8. Bearbeitung der Beschwerde (process_complaint) unter Berücksichtigung des Fragebogens²
9. Bewertung der Bearbeitung (check_processing) mit dem Ergebnis
10. erneute Prüfung (processing_nok) oder
11. Abschluss (processing_ok)
12. Ablage (archive)

Auch Plätze haben Bedeutung: z. B. hat c_2 die Bedeutung: „Bewertung kann beginnen“.

²Nebenbedingung c_5

8.3.4 Definition

Im Folgenden werden nur einfache Netze betrachtet, d.h. die Kantenbewertung ist 1 auf F , also: $W(x, y) = 1 \iff (x, y) \in F$.

Es ist sinnvoll nur einen Anfangsplatz a und einen Endplatz e vorzusehen, da Anfang und Ende meist hervorgehobene Ereignisse sind. Dann sollten alle Transitionen auf Pfaden zwischen diesen liegen. Wir erhalten so folgende Normalform:

Definition 8.14 Ein P/T -Netz $\mathcal{N} = (P, T, F, \mathbf{m}_a)$ heißt Workflow-Netz (WF-Netz), falls

- a) es zwei besondere Plätze $\{a, e\} \subseteq P$ enthält mit $\bullet a = \emptyset$ („Start, Quelle, Anfangsplatz“) und $e^\bullet = \emptyset$ („Ende, Senke, Endplatz“),
- b) alle Plätze und Transitionen auf Pfaden zwischen a und e liegen und
- c) die Anfangsmarkierung \mathbf{m}_a durch $[\mathbf{m}_a(p) := \text{if } p = a \text{ then } 1 \text{ else } 0 \text{ fi}]$ sowie eine Endmarkierung \mathbf{m}_e durch: $[\mathbf{m}_e(p) := \text{if } p = e \text{ then } 1 \text{ else } 0 \text{ fi}]$ festgelegt sind.

Lemma 8.15 In jedem Workflow-Netz sind Quelle a und Senke e eindeutig festgelegt.

Beweis: Als Übung. □

Das Verhalten der Workflow-Netze ist eindeutig über das Verhalten der P/T -Netze festgelegt. Dies ist für die Fragen nach der jeweiligen Semantik von Bedeutung.

8.3.5 Grundlegende Strukturen in Workflow-Modellen

Der Aufbau von Workflow-Modellen erfolgt meist auf der Grundlage von einfachen und komplexeren Strukturfragmenten. Diese werden im Folgenden *Muster* (engl. „patterns“) genannt.³

Einfache Muster sind dabei die folgenden, typische Strukturen sind sequentielle, nebenläufige, wiederholende und alternative Bearbeitung (letztere mit und ohne expliziten Testbedingungen).

- Sequentielle und nebenläufige Bearbeitung (siehe Abb. 8.15)
- Iteration (siehe Abb. 8.16)
- Alternative Bearbeitung mit und ohne expliziten Testbedingungen (siehe Abb. 8.17)

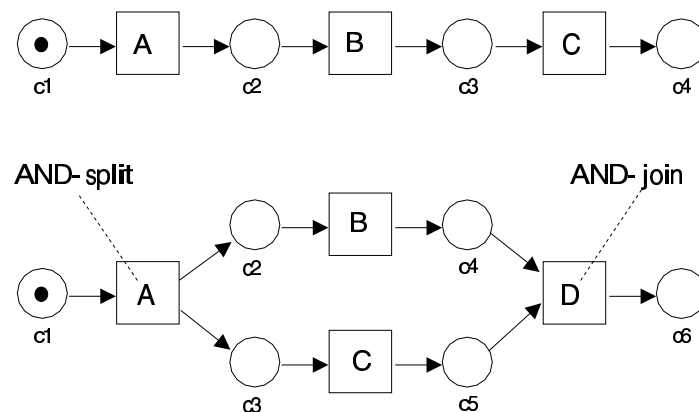


Abbildung 8.15: Sequentielle und nebenläufige Bearbeitung

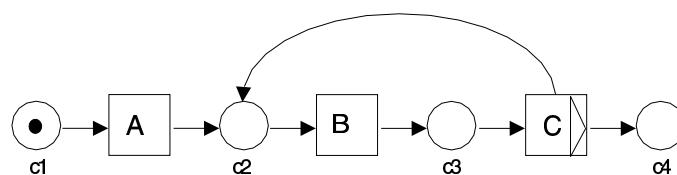


Abbildung 8.16: Iteration

³Unter <http://www.workflowpatterns.com/patterns> ist eine Sammlung solcher Strukturmuster für Workflows zu finden.

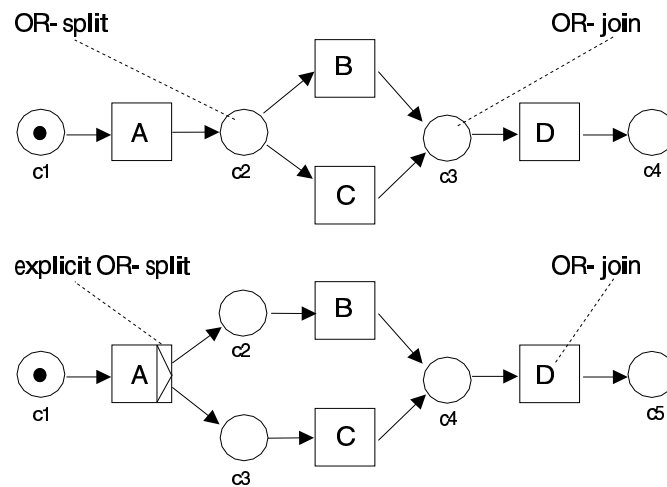


Abbildung 8.17: Alternative Bearbeitung mit und ohne expliziten Testbedingungen

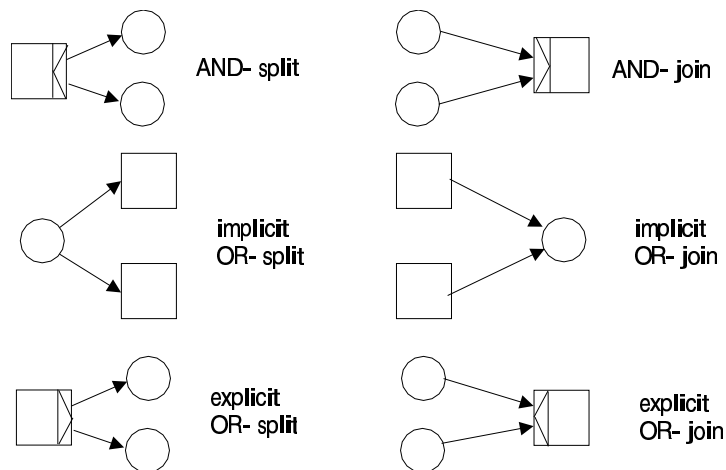


Abbildung 8.18: Graphische Symbole für Verzweigung

Dabei werden zum Teil vereinfachende graphische Symbole benutzt (siehe Abb. 8.18). Hier ist zu beachten, dass diese teilweise fundamental gegen das übliche Schaltverhalten der Transitionen verstoßen. Zur besseren Lesbarkeit in anwendungsnahen Kontexten ist dies aber eine sinnvolle Modellierung. Von entscheidender Bedeutung ist jedoch, dass die Semantik der jeweiligen Symbole den jeweiligen Verwendern / Lesern der Modelle bekannt ist. Die Diskussionen zu diesen Modellierungssprachen oder Modellierungstechniken finden sich in der hier insgesamt angegebenen Literatur.

Ein Workflow kann auch von der Interaktion mit der Umgebung abhängen (restriktives System, Trigger, Ressourcen), wie zum Beispiel bei den Ereignissen: „Bearbeiter ist in Urlaub oder krank“ oder „Antwort auf eine Anfrage trifft nicht ein“.

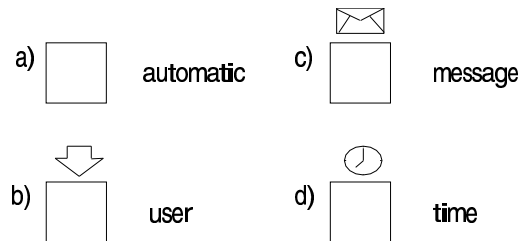


Abbildung 8.19: Trigger eines Workflow-Systems

Es gibt folgende Arten von Triggern (Abbildung 8.19):

- a) Automatisch (keine externe Eingabe notwendig).
- b) Benutzer (user): ein Bearbeiter oder Benutzer oder eine Funktionseinheit nimmt einen Auftrag an und bearbeitet ihn.
- c) Nachricht (message): eine Nachricht von außen wird benötigt (Brief, Anruf, e-mail, Fax).
- d) Zeit (time): es besteht eine Zeitbedingung für die Bearbeitung.

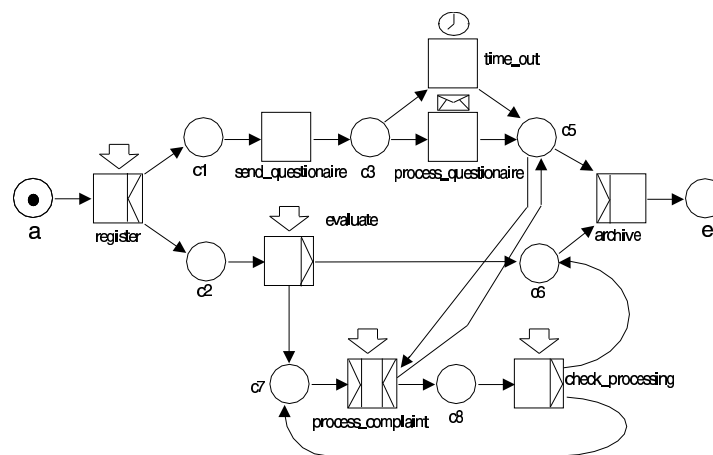


Abbildung 8.20: Workflow aus Abb. 8.14 mit Triggern

Es kann auch Wechselwirkungen zwischen Triggern und Verzweigungen geben (ersetze beispielsweise die implizite Verzweigung in $c3$ der Abbildung 8.20 durch explizite). Ein anderes Problem besteht darin, dass manchmal Zustandsinformation („milestones“) erforderlich ist (Beispiel: $c5$ in Abb. 8.20).

8.3.6 Korrektheit

Workflow-Systeme werden u.U. fehlerhaft entworfen, weil die abzubildenden Prozesse nicht richtig verstanden wurden. Es stellt sich die Frage, ob in diesem Sinne semantisch inkorrekte Entwürfe durch formale Methoden erkannt werden können.

Die Erfahrung aus der Praxis zeigt, dass ...

- Workflow-Prozesse oft nicht richtig verstanden werden (Mehrdeutigkeit, Widersprüche, Verklemmungen),
- allein schon die (versuchsweise) Modellierung durch Petrinetze Mängel aufdeckt und
- bei fertiggestellten Petrinetz-Modellen von Workflow-Systemen Mängel durch strukturelle Untersuchungen aufgedeckt oder durch Werkzeuge (automatisch) gefunden werden.

Beispiel 8.16 Wir betrachten dazu das nicht korrekte Workflow-System von Abbildung 8.21. Es zeigt zum Beispiel die folgenden Fälle von Fehlverhalten:

- Wenn die Transitionen 2 und 5 schalten, dann verbleibt bei Termination (d.h. wenn eine Marke in e ankommt) immer noch eine Marke in c_5 . Das deutet darauf hin, dass ein Teilprozess nicht vollendet wurde. (Die Marke kann auch beim nächsten Durchlauf zu Fehlverhalten führen.)
- Wenn die Transitionen 3 und 4 schalten, dann verbleibt bei Termination eine Marke in c_4 .
- Wenn die Transitionen 2 und 4 schalten, dann erreichen bei Termination 2 Marken den Endplatz e .

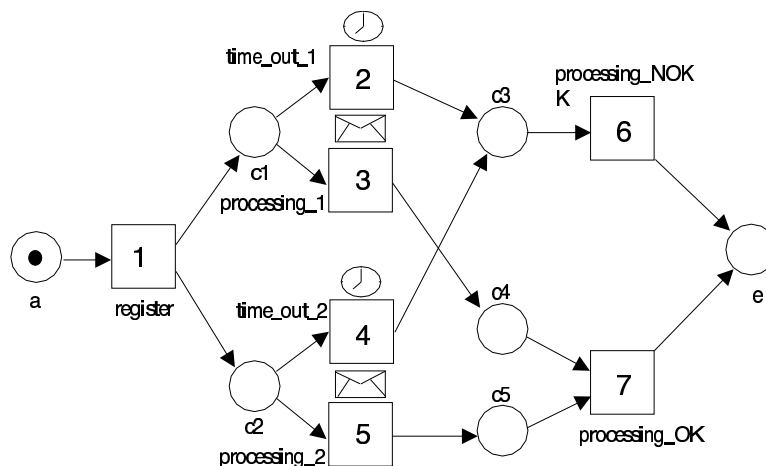


Abbildung 8.21: Problematisches Workflownetz für Beschwerdebearbeitung

Da fehlerhafte Workflow-Systeme oft aus inhaltlichen Missverständnissen entstehen, stellt sich die Frage, ob solche im Grunde semantische Fehler durch formale Kriterien vermieden werden können.

Die Definition 8.17 formalisiert dazu folgende drei Kriterien:

- a) Aus jeder erreichbaren Markierung ist eine ordnungsgemäße Termination möglich.
- b) Genau eine Marke in dem Endplatz e ist die einzige Möglichkeit zu terminieren.
- c) Jede Transition kann in einer möglichen Schaltfolge schalten, denn sonst wäre sie nutzlos.

Definition 8.17 Ein WF-Netz $\mathcal{N} = (P, T, F, \mathbf{m}_a)$ heißt korrekt, falls gilt:

- a) $\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}) \exists w \in T^* : \mathbf{m} \xrightarrow{w} \mathbf{m}_e$ (Termination möglich)
- b) $\forall \mathbf{m} \in \mathbf{R}(\mathcal{N}) : \mathbf{m}(e) \geq 1 \Rightarrow \mathbf{m} = \mathbf{m}_e$ (korrekte Termination)
- c) $\forall t \in T \exists \mathbf{m} \in \mathbf{R}(\mathcal{N}) : \mathbf{m} \xrightarrow{t}$ (Nützlichkeit)

Das problematische WF-Netz aus Abbildung 8.21 erfüllt Eigenschaft c) – aber weder a) noch b).

8.3.7 Abschluss eines WF-Netzes

Wir transformieren ein WF-Netz, indem wir eine neue Transition t^* zwischen e und a hinzufügen (vgl. Abbildung 8.22) und so das Netz „kurzschließen“:

Definition 8.18 Für ein WF-Netz $\mathcal{N} = (P, T, F, \mathbf{m}_a)$ mit Anfangsplatz a und Endplatz e heißt $\overline{\mathcal{N}} = (P, T', F', \mathbf{m}_a)$ der Abschluss von \mathcal{N} , falls gilt:

- a) $T' := T \cup \{t^*\}$ für eine neue Transition $t^* \notin T$
- b) $F' := F \cup \{(e, t^*), (t^*, a)\}$

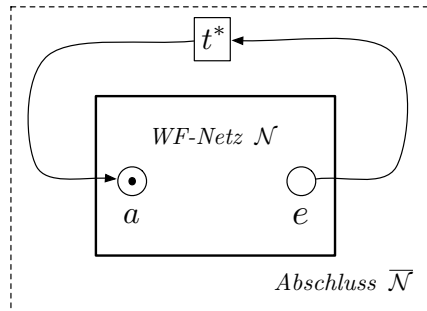


Abbildung 8.22: Transformation eines WF-Netzes: Hinzufügung einer neuen Transition t^*

Aufgabe 8.19 Zeigen Sie, dass für das WF-Netz \mathcal{N} aus Abbildung 8.21 der Abschluss $\overline{\mathcal{N}}$ nicht beschränkt ist (z.B. dadurch dass $\mathbf{R}(\overline{\mathcal{N}})$ nicht endlich ist). Ebenso zeige man, dass $\overline{\mathcal{N}}$ nicht lebendig ist.

Satz 8.20 ([Aal97]) Ein WF-Netz \mathcal{N} ist genau dann korrekt, wenn sein Abschluss $\overline{\mathcal{N}}$ lebendig und beschränkt ist.

Beweis: Als Übung. □

Aufgabe 8.21 Analysieren Sie das WF-Netz von Abb. 8.23 mit Hilfe des Verfahrens nach Satz 8.20.

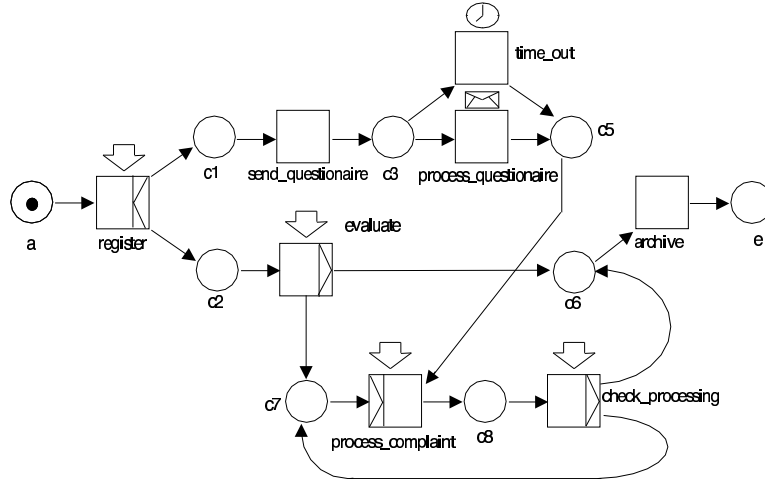


Abbildung 8.23: Ein WF-Netz zur Behandlung von Beschwerden

Die Eigenschaft „korrektes WF-Netz“ ist ein gutes Beispiel dafür, wie eine domänenspezifische Eigenschaft durch eine einfache Transformation auf eine basale Petrinetzeigenschaft zurückgeführt werden kann, die intensiv untersucht wurde und für die Software-Werkzeuge bereitstehen. Es handelt sich um die Eigenschaften „beschränkt“ und „lebendig“, die in der Tabelle 6.1 auf Seiten 105 definiert werden und zu denen im selben Kapitel Entscheidungsalgorithmen entwickelt werden. Beschränktheit bedeutet, dass es eine obere Schranke für die Anzahl der Marken auf den Plätzen gibt. Diese Eigenschaft ist damit äquivalent, dass die Erreichbarkeitsmenge $\mathbf{R}(\mathcal{N})$ endlich ist. Lebendigkeit bedeutet dagegen, dass von jeder erreichbaren Markierung $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ jede Transition über eine geeignete Schaltfolge aktiviert werden kann. Intuitiv bedeutet dies, dass – egal wie sich das System verhält – nie Teile des Systems permanent ausfallen oder terminieren.

Umfangreiche weitere Literatur zu Workflows und angrenzenden Gebieten wie z.B. Process Mining finden sich auf der persönlichen Internetseite von Wil van der Aalst (siehe <http://www.wis.win.tue.nl/~wvdaalst/>).