



Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik

Petri-Netz-Formalismen und Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen

**Bernhard Bachmann, Timo Kleine-Döpke, Hermann-Josef Kruse,
Lennart Ochel, Sabrina Proß**

Band 2



FH Bielefeld
University of
Applied Sciences

Petri-Netz-Formalismen und Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen

Bernhard Bachmann, Timo Kleine-Döpke,
Hermann-Josef Kruse, Lennart Ochel,
Sabrina Proß

Forschungs- und Entwicklungsschwerpunkt
Angewandte Mathematische Modellierung und Optimierung (AMMO)
Fachbereich Ingenieurwissenschaften und Mathematik
Fachhochschule Bielefeld

Impressum

**Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik**

ISSN

2196-6192

Erscheinungsort

Bielefeld

Verantwortliche und ausführende Stelle (Herausgeber)

im Sinne des § 55 I RStV und des Presserechts NRW:
Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Wilhelm-Bertelsmann-Str. 10
33602 Bielefeld

Der Fachbereich wird innerhalb der Fachhochschule
Bielefeld vom Dekan Prof. Dr.-Ing. Prof. h.c. Lothar Budde
geleitet und vertreten.

Verantwortliche Chefredaktion

Prof. Dr.-Ing. Rolf Naumann
Fachhochschule Bielefeld
Fachbereich Ingenieurwissenschaften und Mathematik
Am Stadtholz 24
33609 Bielefeld
Telefon +49.521.106-7483
Telefax +49.521.106-7191
rolf.naumann@fh-bielefeld.de

Vorwort

Die Motivation zum Verfassen dieses Buches ergab sich durch die Dissertation von Sabrina Proß (2013). In ihrem Buch behandelt sie das Modellieren und Optimieren von hybriden Prozessen in biologischen Systemen und entwickelt auf der Basis von Petri-Netzen einen Formalismus mit der Bezeichnung „Extended Hybrid Petri Nets“ (kurz: xHPN), um damit insbesondere biologische Prozesse modellieren, analysieren und optimieren zu können. Um Petri-Netze als graphische Modellierungsmethode einsetzen zu können, wurde mit Hilfe der objektorientierten Modellierungssprache *Modelica* eine Petri-Netz-Bibliothek namens PNlib erstellt.¹

Im Rahmen eines internen AMMO²-Projektes sollte diese Modelica-Bibliothek um wesentliche Elemente aus dem Bereich der *gefärbten* Petri-Netze (Coloured Petri Nets) erweitert werden. Zum Zwecke der Selbstschulung und als Arbeitsgrundlage für alle Projektmitglieder entstand nebenbei ein sehr ausführliches „Glossar“, in welchem die wichtigsten Begriffe und Zusammenhänge des umfangreichen Gebiets der Petri-Netze zusammengetragen wurden. Durch Einfügen von veranschaulichenden Beispielen zu den einzelnen Begriffen und ausführlichen Beschreibungen der Zusammenhänge entstand aus dem Glossar ein Buch, das sich auch für interessierte Personen eignet, die noch kein oder nur ein sehr eingeschränktes Vorwissen über Petri-Netze besitzen. Allerdings wird das facettenreiche Gebiet der Petri-Netze und insbesondere auch der gefärbten Petri-Netze in diesem Buch nur eingeschränkt betrachtet, wobei der Fokus auf die Umsetzung der Theorie in praxisrelevante Anwendungsmöglichkeiten für Modelica-Modelle gerichtet ist, indem verschiedene Auflösungsmöglichkeiten von allgemeinen Konflikten beim Feuerprozess von (gefärbten) Petri-Netzen entwickelt werden.

Petri-Netze dienen der Modellierung von realen Prozessabläufen. Für den Fall, dass Petri-Netz-Modelle zur Simulation eingesetzt werden und die Simulationen automatisch und ohne (oder nur teilweise mit) Interaktionen eines Anwenders ablaufen sollen, muss sichergestellt sein, dass im Modell auftretende Entscheidungskonflikte durch zuvor implementiertes Entscheidungsverhalten automatisch gelöst werden. Dabei ist unterschiedliches Entscheidungsverhalten in unterschiedlichen Realsituationen denkbar, weil sie zufallsgesteuert, prioritäts- oder nutzenorientiert sein können.

¹Die Bibliothek PNlib wurde auf der 9. Internationalen Modelica-Konferenz 2012 preisgekrönt (<http://www.fh-bielefeld.de/presse/archiv/library-award-fuer-mathematiker-der-fh-bielefeld>).

²AMMO steht für „Angewandte Mathematische Modellierung & Optimierung“ und ist ein Forschungsschwerpunkt des Fachbereiches *Ingenieurwissenschaften & Mathematik* der Fachhochschule Bielefeld (<http://www.fh-bielefeld.de/ammo>).

Um einem Entwickler von Petri-Netz-Modellen („Modellierer“) die Möglichkeit an die Hand zu geben, verschiedene, den Realsituationen angepasste Entscheidungsmuster zur automatisierten Konfliktlösung implementieren zu können, bedarf es einer „Werkzeugkiste“, aus der er sich bedienen kann.

In diesem Buch werden einige solcher Werkzeuge zur automatisierten Konfliktlösung vorgestellt. Dabei wird die Entwicklung dieser Konfliktlösungstools in drei Stufen vorgenommen. Zunächst werden „normale“ Petri-Netze behandelt und an ihnen die Prinzipien der Konfliktlösungsansätze erklärt. Danach werden diese Lösungsansätze auf die allgemeineren Fälle von gefärbten Petri-Netzen übertragen, und zwar zunächst mit konstanten und schließlich mit variablen Pfeilbewertungen.

Inhaltsverzeichnis

1	Einführung in Petri-Netze	1
1.1	Anschauliche Darstellung von Petri-Netzen	1
1.2	Formale Darstellung von Petri-Netzen	5
1.2.1	Seriellles Feuern	8
1.2.2	Nebenläufiges Feuern	10
2	Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen	13
2.1	Formale Definition eines allgemeinen Konflikts	16
2.2	Lokale Lösungsansätze für allgemeine Konflikte	18
2.2.1	Petri-Netze mit Bewertungen	19
2.2.2	Algorithmische Beschreibungen von lokalen Freigabeprozessen	23
2.2.2.1	Lokale Strategien mit impliziter Zielfunktion	23
2.2.2.2	Lokale Strategien mit expliziter Zielfunktion	30
2.2.3	Petri-Netze mit Konfliktlösung	39
2.3	Kollektive Lösungsansätze	43
2.3.1	Kollektive	46
2.3.2	Kollektive Bewertungen	49
2.3.3	Algorithmen zur kollektiven Konfliktlösung	51
2.3.4	Probleme bei verallgemeinerter Kollektiventscheidung	54
3	Feuerprozess-Szenarien	58
4	Einführung in gefärbte Petri-Netze	63
5	Allgemeine Konfliktlösungen bei Feuerprozessen von gefärbten Petri-Netzen	80
5.1	Verallgemeinerung eines allgemeinen Konflikts für MCPN	81
5.2	Lokale Lösungsansätze für allgemeine Konflikte bei MCPN	84
5.2.1	Gefärbte Petri-Netze mit Bewertungen	85

5.2.2	Beispiele für algorithmische Beschreibungen von lokalen Freigabeprozessen	88
5.3	MCPN mit Konfliktlösung	103
6	Gefärbte Petri-Netze mit Variablen	107
6.1	CPN mit einfachen Variablen	110
6.2	Problematik der Modus-Wahl	127
6.2.1	Wahlstrategien ohne Berücksichtigung der Häufigkeit	129
6.2.2	Wahlstrategien mit Berücksichtigung der Häufigkeit	133
7	Konfliktlösung bei CPN mit Variablen	137
8	Anwendungsbeispiel: Biergarten	145
8.1	Problemstellung	145
8.2	Modell eines gefärbten Petri-Netzes mit Variablen	147
9	Ausblick	158
9.1	„Allgemeine“ gefärbte Petri-Netze	158
9.2	Kapazitierte (gefärbte) Petri-Netze	160
9.3	Erweiterte (gefärbte) Petri-Netze	162
9.4	(Gefärbte) Petri-Netze mit Zeit	163
	Symbolverzeichnis	165
	Literaturverzeichnis	168
	Anhang: Multimengen	170

1 Einführung in Petri-Netze

In diesem Kapitel soll möglichst anschaulich in die Grundidee von Petri-Netzen eingeführt und deren Formalisierung vermittelt werden.¹

1.1 Anschauliche Darstellung von Petri-Netzen

Petri-Netze lassen sich als schwach zusammenhängende, 2-gefärbte Digraphen auffassen.² Die beiden „Farben“ bzw. die entsprechenden Knoten heißen *Plätze* (engl. *places*) bzw. *Transitionen* (engl. *transitions*). Grafisch werden Plätze als kreisrunde und Transitionen als rechteckige Knoten dargestellt. Jeder Platz darf eine Anzahl $k \geq 0$ von sogenannten *Marken* (oder *Token*) speichern. Eine konkrete Festlegung, wie viele Marken die verschiedenen Plätze beinhalten, nennt man *Markierung* oder *Zustand* des Petri-Netzes.³

Das folgende Beispiel eines Petri-Netzes besteht aus fünf Plätzen (p_i , $i = 1, \dots, 5$) und vier Transitionen (t_j , $j = 1, \dots, 4$). Der aktuelle Zustand dieses Petri-Netzes ist durch die Anzahl z_i von Marken auf den verschiedenen Plätzen p_i festgelegt, also $z_1 = 2$, $z_2 = 3$, $z_3 = z_4 = z_5 = 1$.

¹Eine ausführlichere und weiterführende Einführung in die Begriffswelt von Petri-Netzen findet man u.a. in [Bau97], [Rei82], [Rei10], [Sta90], [PW03].

²Zu den graphentheoretischen Begriffen siehe [BM08], [Bra94], [Die10], [KV12], [NS96], [Nol76], [Sch03].

³Man mache sich klar, dass die Marken in einem Petri-Netz als (temporäre) Knotenbewertungen aufgefasst werden können.

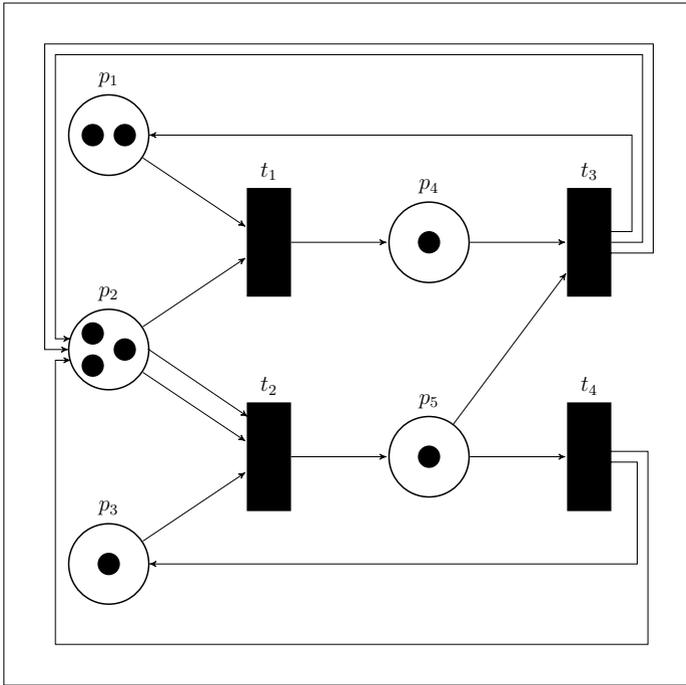


Abbildung 1.1: Beispiel eines Petri-Netzes in einem initialen Zustand

Alle Plätze, von denen aus Pfeile zu einer Transition t führen, werden als *Vorbedingungen* (*input places*) von t bezeichnet. Alle Plätze, zu denen Pfeile von t aus hinführen, heißen *Nachbedingungen* (*output places*) von t . Im obigen Beispiel stellen die Plätze p_4 und p_5 die Vorbedingungen und die Plätze p_1 und p_2 die Nachbedingungen der Transition t_3 dar.

Es sind auch parallele Pfeile (Mehrfachpfeile) zwischen Plätzen und Transitionen und umgekehrt erlaubt (wie im Beispiel zwischen p_2 und t_2 bzw. t_3 und p_2). Dabei spricht man von *k-fachen Vor-* bzw. *Nachbedingungen*. Diese Form der Darstellung lässt sich aber auch durch entsprechende Pfeilbewertung ersetzen:

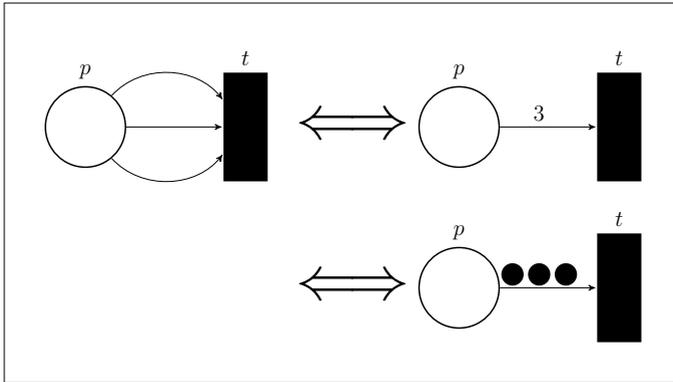


Abbildung 1.2: Darstellungsformen von Pfeilgewichtungen

Darüber hinaus ist grundsätzlich zugelassen, dass ein Platz p gleichzeitig Vor- und Nachbedingung einer Transition t ist (siehe folgende Skizze). Hierbei spricht man von sog. *Petri-Schlingen*.

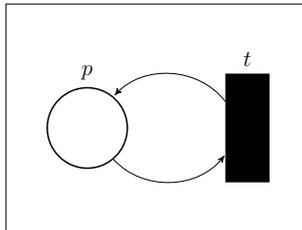


Abbildung 1.3: Beispiel einer Petri-Schlinge

Alle Transitionen „spielen“ nun weitgehend unabhängig voneinander das folgende *Tokenspiel* auf dem Petri-Netz: Falls auf *allen* Vorbedingungen von Transition t passend viele Marken liegen, d.h. mindestens k Marken auf jeder k -fachen Vorbedingung, so heißt die Transition t *aktiviert* oder *feuerbar*. Eine feuerbare Transition „feuert“, indem sie in einem Schritt k Marken von jeder k -fachen Vorbedingung entfernt und m Marken auf jede m -fache Nachbedingung legt.

Im obigen Beispiel (Abb. 1.1) sind aktuell alle Transitionen feuerbar. Dabei liefert beispielsweise das Feuern von Transition t_2 den in Abb. 1.4 dargestellten Zustand des initialen Petri-Netzes.

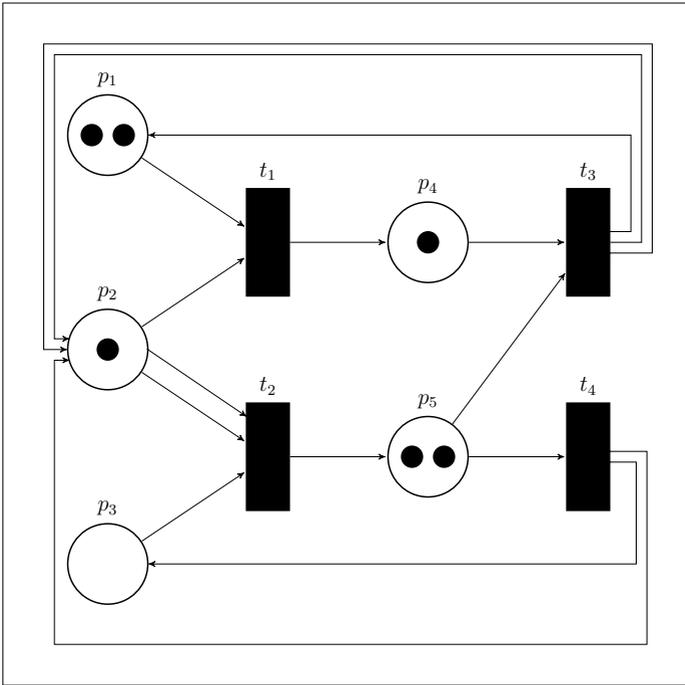


Abbildung 1.4: Zustand des Petri-Netzes aus Abb. 1.1 nach Feuern der Transition t_2

Im initialen Zustand des Petri-Netzes (Abb. 1.1) sind einige Transitionen auch *nebenläufig* *feuerbar*, d.h. sie lassen sich gleichzeitig (*parallel*) genauso wie hintereinander (*seriell*) ausführen, ohne dass sich die Ergebnisse nach parallelem bzw. seriellem Feuern unterscheiden. Beispielsweise sind die Transitionen t_1 und t_2 nebenläufig feuerbar, ja sogar t_1 , t_2 und t_3 oder t_1 , t_2 und t_4 . Demgegenüber sind die Transitionen t_3 und t_4 nicht nebenläufig feuerbar, da ein Feuern von t_3 (oder t_4) die einzige Marke von p_5 entfernt und damit das Feuern der anderen Transition t_4 (bzw. t_3) unmöglich macht.

Durch ein nebenläufiges Feuern der Transitionen t_1 , t_2 und t_3 ergibt sich aus dem initialen Zustand des Petri-Netzes (Abb. 1.1) der Folgezustand gemäß Abb. 1.5.

Wenn für eine Menge $\{t_{i_1}, \dots, t_{i_k}\}$ von feuerbaren Transitionen nebenläufiges Feuern nicht möglich ist, spricht man von einem „Feuerkonflikt“ dieser Transitionenmenge bzw. der zugehörigen Plätze. Maßnahmen zur Auflösung von Feuerkonflikten stehen im Mittelpunkt der Betrachtungen von Kapitel 2.

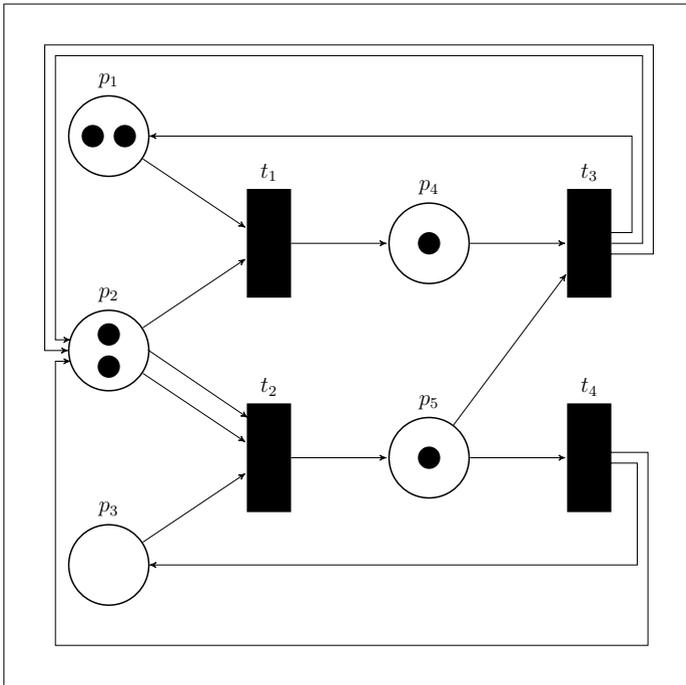


Abbildung 1.5: Zustand des Petri-Netzes aus Abb. 1.1 nach nebenläufigem Feuern von t_1 , t_2 und t_3

1.2 Formale Darstellung von Petri-Netzen

Um mögliche Zustandsänderungen eines Petri-Netzes formalisieren zu können, wird zunächst eine formale Definition für Petri-Netze gebracht:

Definition 1.1:

Ein **Netz** N ist ein Tupel $N = (P, T, F, B)$ mit einer endlichen Menge $P = \{p_1, \dots, p_m\}$ und einer endlichen Menge $T = \{t_1, \dots, t_n\}$, wobei $P \cap T = \emptyset$; zudem gelte $F \subseteq P \times T$, $B \subseteq T \times P$. Die Elemente aus P werden **Plätze** (places) und die Elemente aus T werden **Transitionen** (transitions) genannt. Die Elemente aus $F \cup B$ heißen **Pfeile** oder **Bögen** (arcs) oder **gerichtete Kanten** von N .

Offensichtlich stellt $G = (V, E)$ mit $V = P \cup T$ und $E = F \cup B$ einen 2-gefärbten Digraph dar.

Definition 1.2:

Ein **Petri-Netz** N ist ein Tupel $N = (P, T, F, B, f)$, wobei $N = (P, T, F, B)$ ein Netz gemäß Def. 1.1 ist. Die Abbildung $f : F \cup B \rightarrow \mathbb{N}$ heißt **Gewichtungsfunktion** von N , die jedem Pfeil $(p, t) \in P \times T$ bzw. $(t, p) \in T \times P$ eine natürliche Zahl als das zugehörige **Gewicht** $f(p, t)$ bzw. $f(t, p)$ zuweist.¹

Der entsprechende gewichtete Digraph $G = (V, E, f)$ lässt sich als ein (ungewichteter) Multidigraph² G^* interpretieren, indem jeder Pfeil $(i, j) \in E$ durch $f(i, j)$ parallele (ungewichtete) Pfeile ersetzt wird. Entsprechend wird auch von der gewichteten bzw. ungewichteten Darstellungsform eines Petri-Netzes gesprochen.³

Definition 1.3:

Es sei ein Petri-Netz $N = (P, T, F, B, f)$ gegeben.

- a) Jedem Platz $p \in P$ wird durch $\bullet p = \{t \in T \mid (t, p) \in B\}$ ein sog. **Vorbereich** (*set of inputs*) und durch $p\bullet = \{t \in T \mid (p, t) \in F\}$ ein sog. **Nachbereich** (*set of outputs*) zugewiesen.
- b) Jeder Transition $t \in T$ wird durch $\bullet t = \{p \in P \mid (p, t) \in F\}$ ein sog. **Vorbereich** (*set of inputs*) und durch $t\bullet = \{p \in P \mid (t, p) \in B\}$ ein sog. **Nachbereich** (*set of outputs*) zugewiesen.
- c) Die Elemente aus einem Vor- bzw. Nachbereich werden auch **Vor-** bzw. **Nachbedingungen** genannt.

Definition 1.4:

Es sei ein Petri-Netz $N = (P, T, F, B, f)$ gegeben. Eine Abbildung $\mathbf{z} : P \rightarrow \mathbb{N}_0$ heißt **Zustand** oder **Markierung** von N , wobei \mathbf{z} eindeutig als Vektor $\mathbf{z} = (z_1, \dots, z_m)$ mit $z_i = \mathbf{z}(p_i)$ darstellbar ist. Die Menge aller Abbildungen von P in \mathbb{N}_0 wird der **Zustandsraum** von N genannt.⁴

Eine Markierung von N lässt sich als eine Knotenbewertung der Knoten(teilm)enge P auffassen. Die Knotenwerte $\mathbf{z}(p)$ spiegeln sich in der graphischen Darstellung des Petri-Netzes N als die Anzahl der Marken an den Plätzen $p \in P$ wieder.

¹Hierbei wird die vereinfachte Schreibweise $f(p, t)$ statt $f((p, t))$ vereinbart.

²Zum Begriff Multidigraph siehe einschlägige Literatur.

³Im Folgenden wird insbesondere die gewichtete Form von Petri-Netzen verwendet werden.

⁴Mit \mathbb{N}_0 ist die Menge $\{0, 1, 2, 3, \dots\}$ aller nichtnegativen ganzen Zahlen bezeichnet, d.h. $\mathbb{N} \cup \{0\}$.

Beispiel:

In Abb. 1.6 ist das Petri-Netz aus Abb. 1.1 in gewichteter Form dargestellt: $N = (P, T, F, B, f)$ mit $P = \{p_1, p_2, p_3, p_4, p_5\}$, $T = \{t_1, t_2, t_3, t_4\}$, $F = \{(p_1, t_1), (p_2, t_1), (p_2, t_2), (p_3, t_2), (p_4, t_3), (p_5, t_3), (p_5, t_4)\}$, $B = \{(t_1, p_4), (t_2, p_5), (t_3, p_1), (t_3, p_2), (t_4, p_3), (t_4, p_2)\}$.

Die Gewichte sind $f(t_3, p_2) = f(p_2, t_2) = 2$, sonst 1.

Die Markierung (bzw. der Zustand) von N ist $\mathbf{z} = (2, 3, 1, 1, 1)$.

Die Vor- bzw. Nachbereiche der Plätze sind $\bullet p_1 = \{t_3\}$, $\bullet p_2 = \{t_3, t_4\}$, $\bullet p_3 = \{t_4\}$,

$\bullet p_4 = \{t_1\}$, $\bullet p_5 = \{t_2\}$ bzw. $p_1 \bullet = \{t_1\}$, $p_2 \bullet = \{t_1, t_2\}$, $p_3 \bullet = \{t_2\}$, $p_4 \bullet = \{t_3\}$, $p_5 \bullet = \{t_3, t_4\}$.

Die Vor- bzw. Nachbereiche der Transitionen sind $\bullet t_1 = \{p_1, p_2\}$, $\bullet t_2 = \{p_2, p_3\}$,

$\bullet t_3 = \{p_4, p_5\}$,

$\bullet t_4 = \{p_5\}$ bzw. $t_1 \bullet = \{p_4\}$, $t_2 \bullet = \{p_5\}$, $t_3 \bullet = \{p_1, p_2\}$, $t_4 \bullet = \{p_2, p_3\}$.

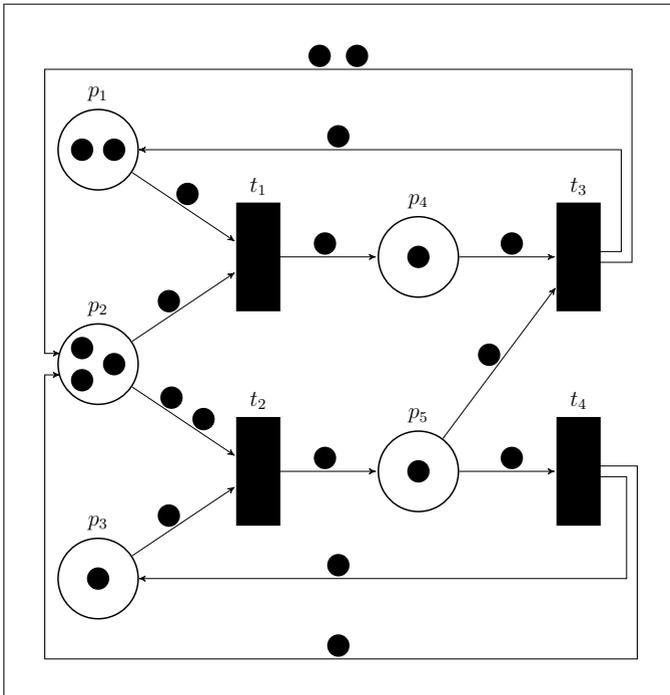


Abbildung 1.6: Beispiel eines gewichteten Petri-Netzes (vgl. Abb. 1.1)

1.2.1 Serielles Feuern

Definition 1.5:

Es sei $N = (P, T, F, B, f)$ ein Petri-Netz und \mathbf{z} ein Zustand von N .

- a) Eine Transition $t \in T$ von N heißt **aktiviert** oder **seriell feuerbar** im Zustand \mathbf{z} , wenn für alle $p \in \bullet t$ gilt: $\mathbf{z}(p) \geq f(p, t)$.
- b) Eine im Zustand \mathbf{z} aktivierte Transition t wird kurz auch **z-feuerbar** genannt.
- c) Das **Feuern** einer z-feuerbaren Transition $t \in T$ von N ist der Übergang vom Zustand $\mathbf{z} = (z_1, \dots, z_m)$ in den Zustand $\mathbf{z}' = (z'_1, \dots, z'_m)$, wobei für $i = 1, \dots, m$ gilt:

$$z'_i = \begin{cases} z_i - f(p_i, t) + f(t, p_i) & \text{falls } p_i \in \bullet t \wedge p_i \in t \bullet \\ z_i - f(p_i, t) & \text{falls } p_i \in \bullet t \wedge p_i \notin t \bullet \\ z_i + f(t, p_i) & \text{falls } p_i \notin \bullet t \wedge p_i \in t \bullet \\ z_i & \text{falls } p_i \notin \bullet t \wedge p_i \notin t \bullet \end{cases}$$

Das Feuern einer Transition t von N kann allgemein als ein Vektorfeld $\mathbf{f}_t : \mathbb{N}_0^m \rightarrow \mathbb{N}_0^m$ mit

$$\mathbf{f}_t(\mathbf{z}) = \begin{cases} \mathbf{z}' & \text{falls } t \text{ z-feuerbar ist} \\ \mathbf{z} & \text{sonst} \end{cases}$$

aufgefasst werden.

Die Zustandsübergänge lassen sich wie folgt beschreiben: Die Symbolik $\mathbf{z}\langle t \rangle \mathbf{z}'$ bedeutet, dass die Transition t eines im Zustand \mathbf{z} vorliegendes Petri-Netzes N aktiviert ist und N nach dem Feuern der Transition t in den Zustand \mathbf{z}' übergeht. Man sagt auch kurz: t feuert von \mathbf{z} nach \mathbf{z}' .

Mittels dieser Schreibweise lassen sich nun auch allgemeinere Zustandsübergangsketten beschreiben, die man auch als (**serielle**) **Rechnungen** bezeichnet:

$$\mathbf{z}^1 \langle t_{j_1} \rangle \mathbf{z}^2 \langle t_{j_2} \rangle \mathbf{z}^3 \dots \mathbf{z}^r \langle t_{j_r} \rangle \mathbf{z}^{r+1},$$

wobei $j_k \in \{1, \dots, m\}$ für $k = 1, \dots, r$ gilt. Man beachte, dass sichergestellt ist, dass jede Transition t_{j_k} auch \mathbf{z}^k -feuerbar ist, und eine Transition mehrfach in einer Rechnung vorkommen kann. Man mache sich ebenfalls klar, dass jede in einer Rechnung vorkommende

Transition erst dann aktiviert sein muss, wenn ihr Feuern ansteht.

Beispiel:

Im obigen Beispiel (Abb. 1.6) sind aktuell alle Transitionen feuerbar. Nun wird die Transition t_2 seriell gefeuert:

Vom Zustand $\mathbf{z} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ wird in den Zustand $\mathbf{z}' = \begin{pmatrix} 2 \\ 3 - 2 \\ 1 - 1 \\ 1 \\ 1 + 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 2 \end{pmatrix}$ übergegangen.

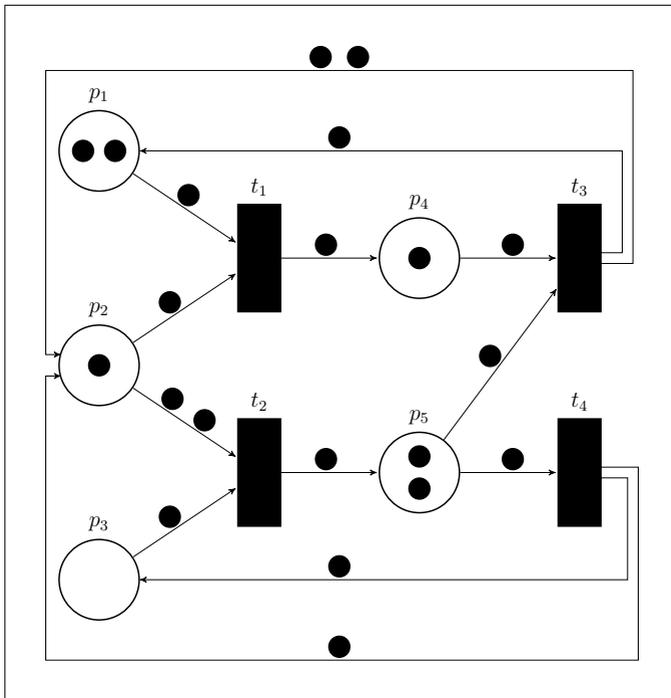


Abbildung 1.7: Zustand des Petri-Netzes aus Abb. 1.6 nach Feuern der Transition t_2

1.2.2 Nebenläufiges Feuern

Neben dem seriellen Feuern, bei dem selbst bei seriellen Rechnungen immer nur genau eine aktivierte Transition bzgl. eines Zustandes \mathbf{z} für einen Zustandsübergang herangezogen wird, soll nun auch *nebenläufiges* Feuern von mehreren Transitionen formalisiert werden. Hierzu muss der Begriff der Feuerbarkeit verallgemeinert werden.

Definition 1.6:

Es sei $N = (P, T, F, B, f)$ ein Petri-Netz im Zustand \mathbf{z} und $\{t_1, \dots, t_r\}$ eine Menge von Transitionen.¹ Mit

$$\bullet\{t_1, \dots, t_r\} = \bigcup_{j=1}^r \bullet t_j = \{p \in P \mid \exists j \in \{1, \dots, r\} : (p, t_j) \in F\}$$

werde der *gemeinsame Vorbereich* der Transitionen t_1, \dots, t_r bezeichnet. Die Transitionenmenge $\{t_1, \dots, t_r\}$ heißt (*nebenläufig*) *feuerbar* im Zustand \mathbf{z} , wenn alle Transitionen aus $\{t_1, \dots, t_r\}$ aktiviert sind (gemäß Def. 1.5) und für alle $p \in \bullet\{t_1, \dots, t_r\}$ gilt:

$$\sum_{t \in \{t_1, \dots, t_r\} \cap p \bullet} f(p, t) \leq \mathbf{z}(p).$$

Beispiel:

Im obigen Beispiel (Abb. 1.7) sind aktuell die Transitionen t_1, t_3 und t_4 feuerbar. Werden

nun die Transitionen t_1, t_3 und t_4 nebenläufig gefeuert, so wird vom Zustand $\mathbf{z}' = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \\ 2 \end{pmatrix}$ in

den Zustand $\mathbf{z}'' = \begin{pmatrix} 2 - 1 + 1 \\ 1 - 1 + 2 + 1 \\ 0 + 1 \\ 1 + 1 - 1 \\ 2 - 1 - 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 1 \\ 0 \end{pmatrix}$ übergegangen.

¹Um eine Doppelindizierung zu vermeiden, wird o.B.d.A. angenommen, dass es sich um die r ersten Transitionen aus T handelt.

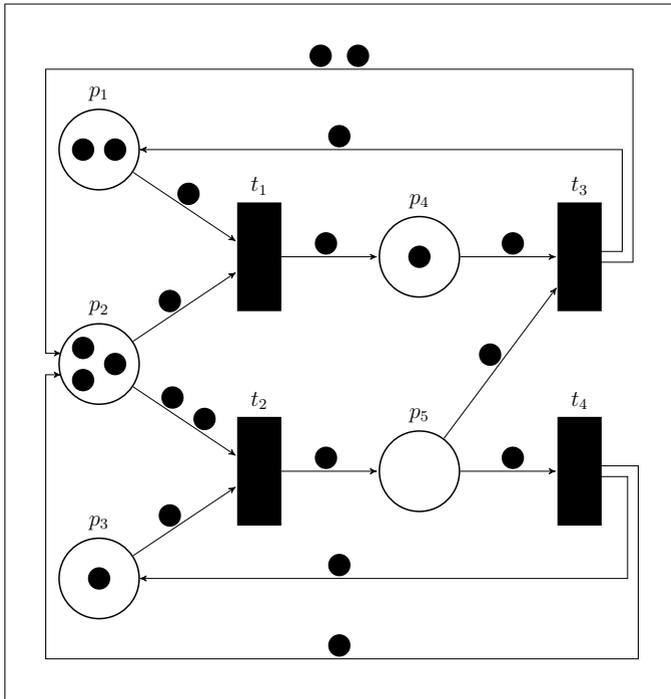


Abbildung 1.8: Zustand des Petri-Netzes aus Abb. 1.7 nach nebenläufigem Feuern der Transitionen t_1, t_3 und t_4

Bemerkung:

Man mache sich klar, dass nebenläufige Feuerbarkeit von k Transitionen $\{t_1, \dots, t_k\}$ im Zustand \mathbf{z} gemäß Def. 1.6 den Sachverhalt impliziert, dass diese Transitionen in *jeder* beliebigen Reihenfolge seriell feuierbar sind (*serielle Vertauschbarkeit*), wobei das jeweilige serielle Feuern in ein und denselben Endzustand überführt. Umgekehrt gilt aber nicht, dass sich aus serieller Vertauschbarkeit stets auch die nebenläufige Feuerbarkeit ergibt, wie das folgende Beispiel zeigt.

Beispiel:

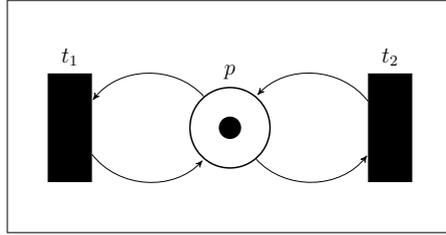


Abbildung 1.9: Beispiel für ein nicht nebenläufig feuerbares Petri Netz mit serieller Vertauschbarkeit

Offensichtlich sind die beiden seriellen Rechnungen $\mathbf{z}\langle t_1 \rangle \mathbf{z}\langle t_2 \rangle \mathbf{z}$ und $\mathbf{z}\langle t_2 \rangle \mathbf{z}\langle t_1 \rangle \mathbf{z}$ möglich (und beide liefern den Endzustand \mathbf{z}), allerdings liegt keine nebenläufige Feuerbarkeit gemäß obiger Definition vor:

$$\sum_{t \in \{t_1, t_2\}} f(p, t) = 1 + 1 = 2 > 1 = \mathbf{z}(p).$$

Die Symbolik $\mathbf{z}\langle t_1, t_2 \rangle \mathbf{z}'$ bedeutet, dass ein nebenläufiges Feuern der Transitionen t_1 und t_2 eines im Zustand \mathbf{z} vorliegendes Petri-Netzes N möglich ist und N dabei in den Zustand \mathbf{z}' übergeht. Die allgemeine Darstellung für einen Zustandsübergang durch nebenläufiges Feuern von k Transitionen ist $\mathbf{z}\langle t_{j_1}, t_{j_2}, \dots, t_{j_k} \rangle \mathbf{z}'$. Mittels dieser Schreibweise lassen sich nun auch allgemeinere Zustandsübergangsketten beschreiben, die man auch als **Rechnungen** bezeichnet:

$$\mathbf{z}^1 \langle t_{j_1}, t_{j_2}, \dots, t_{j_{r_1}} \rangle \mathbf{z}^2 \langle t_{k_1}, t_{k_2}, \dots, t_{k_{r_2}} \rangle \mathbf{z}^3 \dots \mathbf{z}^q \langle t_{l_1}, t_{l_2}, \dots, t_{l_{r_q}} \rangle \mathbf{z}^{q+1}.$$

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

Beim nebenläufigen Feuern kann es in Abhängigkeit von der Modellierung eines konkreten Prozesses durch ein Petri-Netz und dem Zustand des Petri-Netzes im Prozessverlauf zu sogenannten *allgemeinen Konfliktsituationen* kommen¹, wie das folgende Beispiel zeigt:

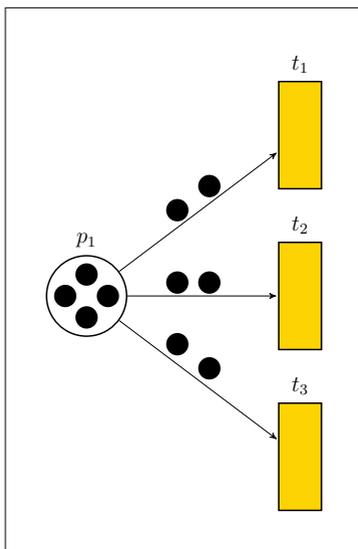


Abbildung 2.1: Allgemeine Konfliktsituation

Wie man gut erkennen kann, ist jede Transition aktiviert und seriell feuierbar, die Menge $\{t_1, t_2, t_3\}$ ist allerdings nicht nebenläufig feuierbar. Nun kann es aber von Interesse sein,

¹Neben den hier beschriebenen *allgemeinen* Konflikten können auch noch andere spezielle Konfliktsituationen auftreten, zum Beispiel kapazitive Input-Konflikte und kapazitive Output-Konflikte oder aktuelle Zeitkonflikte, welche hier nicht näher behandelt werden; vgl. [Pro13].

einen Teil der Transitionenmenge oder sogar maximal viele Transitionen trotzdem zu feuern. Im obigen Beispiel wären jeweils zwei Transitionen nebenläufig feuern, und zwar in den Konstellationen $\{t_1, t_2\}$ oder $\{t_1, t_3\}$ oder $\{t_2, t_3\}$. Es stellt sich also der Entscheidungskonflikt, welche dieser Konstellationen zum Feuern freigegeben werden soll.

Die oben beschriebene „lokale“ Konfliktsituation bezieht sich nur auf einen einzigen Platz. Im Allgemeinen aber kann sich auch ein „globaler“ Entscheidungskonflikt einstellen, wie in der folgenden Abbildung exemplarisch dargestellt wird.

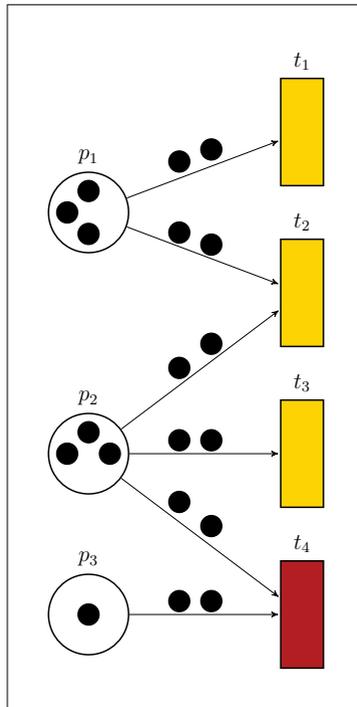


Abbildung 2.2: Mehrere allgemeine Konflikte mit gegenseitiger Beeinflussung

Die Plätze p_1 und p_2 des obigen Petri-Netzes liefern jeweils einen allgemeinen Konflikt, aber weil die Transition t_2 zum jeweiligen Nachbereich von p_1 und p_2 gehört, beeinflussen sich diese beiden Konflikte. Der Platz p_1 kann entweder t_1 oder t_2 zum Feuern freigegeben; der Platz p_2 kann es entweder t_2 oder t_3 ermöglichen zu feuern. Die Transition t_4 ist keine Option, weil diese aufgrund fehlender Token in p_3 nicht aktiviert ist.

Oft ist es bereits die Aufgabe der Netzmodellierung, derartige Konflikte auszuschließen, weil sie beispielsweise nicht Untersuchungsgegenstand eines Prozesses, welcher durch ein Petri-Netz abgebildet werden soll, sind bzw. sein sollen. Umgekehrt kann es sich bei solchen Konflikten durchaus um typische Problemstellungen im Ablauf von zu modellierenden Prozessen handeln, etwa Entscheidungssituationen bei begrenzten Ressourcen, für die verschiedene Lösungsmöglichkeiten mit Hilfe von Petri-Netzen analysiert werden sollen.

Im Folgenden soll es sich unabhängig von der originären Bedeutung um abstrakte Konfliktsituationen handeln, die sich immer dadurch ergeben, dass es nicht möglich ist, sämtliche aktivierten Transitionen nebenläufig zu feuern. Es stellt sich dann die Aufgabe, aus der Menge $TA(\mathbf{z})$ aller im Zustand \mathbf{z} aktivierten Transitionen eine derartige Teilmenge $TE(\mathbf{z})$ zu entnehmen, deren Elemente im Zustand \mathbf{z} nebenläufig feuern sind.¹

Die zu bestimmende Menge von nebenläufig feuerbaren Transitionen kann je nach Problemstellung an unterschiedliche Bedingungen gekoppelt werden. So kann es von Interesse sein, überhaupt eine solche feuerbare Transitionenmenge ohne weitere Bedingungen zu finden. Sollte es allerdings mehr als eine feuerbare Transitionenmenge geben, so könnten diese verschiedenen Optionen als unterschiedlich „gut“ oder „nützlich“ zu bewerten sein, falls ein entsprechendes Gütekriterium zugrunde liegt.

Es stellt sich also die grundsätzliche Aufgabe, für verschiedene Interessenslagen passende Auswahlverfahren zu entwickeln und zu implementieren, die im Konfliktfall zur Anwendung gelangen und zur „automatischen Auflösung“ dieser Konfliktsituationen führen.

Um Konflikte in Petri-Netzen „automatisiert“ mittels implementierter Strategien zur Freigabe für nebenläufiges Feuern von Transitionen lösen zu lassen, werden zunächst Bewertungsfunktionen eingeführt. Mit Hilfe dieser Bewertungsfunktionen werden verschiedene Freigabestrategien vorgestellt, wobei im Wesentlichen zwischen lokalen und kollektiven Freigabeprozessen zu unterscheiden sein wird.

Nach der formalen Einführung des Begriffes „Allgemeiner Konflikt“ in Abschnitt 2.1 werden in Abschnitt 2.2 „Lokale Lösungsansätze“ und später in Abschnitt 2.3 „Kollektive Lösungsansätze“ vorgestellt. Bei den lokalen Lösungsansätzen wird jeder Platz p einzeln betrachtet und jeweils eine *lokale Freigabemenge* $TE(p)$ bestimmt. Diese lokalen Freigabemengen un-

¹An dieser Stelle sei angemerkt, dass nicht unbedingt *alle* aktivierten Transitionen eines Petri-Netzes, sondern auch *ausgewählte* Transitionen, also eine Teilmenge von $TA(\mathbf{z})$, einer solchen Konfliktlösung unterzogen werden können.

terliegen der Vorgabe der *maximalen Feuerbarkeit*, indem sie gewährleisten, dass alle Transitionen aus $TE(p)$ nebenläufig feuerbar sind und die Hinzunahme einer weiteren aktivierten Transition aus dem Nachbereich von p die nebenläufige Feuerbarkeit verletzen würde („maximal feuerbar“). Die lokalen Freigabemengen induzieren auf kanonische Weise eine Menge $TE(\mathbf{z})$ von nebenläufig feuerbaren Transitionen des Petri-Netzes in einem Zustand. Schließlich werden in Abschnitt 2.3 auch noch kollektive Lösungsansätze behandelt.

2.1 Formale Definition eines allgemeinen Konflikts

Zunächst werden folgende Bezeichnungen eingeführt:

Definition 2.1:

Das Tupel $(P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz (PN) mit der Markierung \mathbf{z} .

- a) $TA(\mathbf{z})$ bezeichnet die Menge aller aktivierten Transitionen des Petri-Netzes:

$$TA(\mathbf{z}) := \{t \in T \mid \forall p \in \bullet t : f(p, t) \leq \mathbf{z}(p)\}.$$

- b) $TA_{in}(p, \mathbf{z})$ bezeichnet die Menge der aktivierten Transitionen des Vorbereichs eines Platzes $p \in P$:

$$TA_{in}(p, \mathbf{z}) := \{t \in \bullet p \mid t \in TA(\mathbf{z})\} = TA(\mathbf{z}) \cap \bullet p.$$

- c) $TA_{out}(p, \mathbf{z})$ bezeichnet die Menge der aktivierten Transitionen des Nachbereichs eines Platzes $p \in P$:

$$TA_{out}(p, \mathbf{z}) := \{t \in p\bullet \mid t \in TA(\mathbf{z})\} = TA(\mathbf{z}) \cap p\bullet.$$

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch TA statt $TA(\mathbf{z})$ bzw. $TA_{in}(p)$ statt $TA_{in}(p, \mathbf{z})$ bzw. $TA_{out}(p)$ statt $TA_{out}(p, \mathbf{z})$ geschrieben.

Definition 2.2:

Das Tupel $N = (P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz mit der Markierung \mathbf{z} .

Ein Platz $p \in P$ induziert eine **allgemeine Konfliktsituation** bezüglich \mathbf{z} , wenn gilt:

$$\sum_{t \in TA_{out}(p)} f(p, t) > \mathbf{z}(p).$$

Man mache sich klar, dass ein Platz p nur dann einen allgemeinen Konflikt auslösen kann, wenn die Menge $TA_{out}(p)$ mehr als eine (aktive) Transition enthält.

Definition 2.3:

Das Tupel $N = (P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz mit der Markierung \mathbf{z} .

Eine Menge von Transitionen $\{t_1, \dots, t_r\}$ aus der Menge $TA_{out}(p)$ eines Platzes $p \in P$ ist **maximal feuierbar**, wenn folgende Bedingung für alle $t_0 \in (TA_{out}(p) \setminus \{t_1, \dots, t_r\})$ gilt:

$$\left(\sum_{t \in \{t_1, \dots, t_r\}} f(p, t) \leq \mathbf{z}(p) \right) \wedge \left(f(p, t_0) + \sum_{t \in \{t_1, \dots, t_r\}} f(p, t) > \mathbf{z}(p) \right)$$

Beispiel:

In Abb. 2.3 sind die einelementigen Transitionenmengen $\{t_1\}$, $\{t_2\}$ und $\{t_3\}$ jeweils zwar feuierbar, aber nicht maximal feuierbar. Die zweielementigen Transitionenmengen $\{t_1, t_2\}$ und $\{t_1, t_3\}$ sind maximal feuierbar, während die Transitionenmenge $\{t_2, t_3\}$ nicht feuierbar ist.

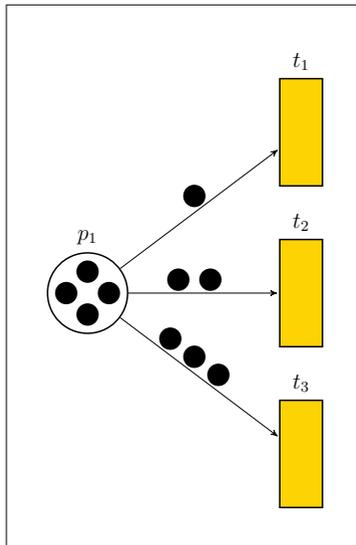


Abbildung 2.3: Maximal feuerebare Transitionenmengen

2.2 Lokale Lösungsansätze für allgemeine Konflikte

Zunächst sollen die Konfliktsituationen lokal behandelt werden, d.h. es wird ein einzelner Platz p betrachtet. Bei dieser Betrachtungsweise bleibt eine gegenseitige Beeinflussung von Konflikten im Hinblick auf verschiedene Plätze noch unberücksichtigt.

Definition 2.4:

Das Tupel $N = (P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz mit der Markierung \mathbf{z} .

Mit $TE_{out}(p, \mathbf{z})$ werde die **Menge der lokal freigegebenen Transitionen** des Nachbereichs eines Platzes $p \in P$ bezeichnet. Diese Transitionenmenge muss selbstverständlich feuierbar sein, d.h. es muss gelten:

$$\sum_{t \in TE_{out}(p, \mathbf{z})} f(p, t) \leq \mathbf{z}(p)$$

und somit

$$TE_{out}(p, \mathbf{z}) \subseteq TA_{out}(p, \mathbf{z}).$$

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch $TE_{out}(p)$ statt $TE_{out}(p, \mathbf{z})$ geschrieben.

Prinzipiell ist jede maximal zulässige Teilmenge $TE_{out}(p)$ von $TA_{out}(p)$ eine mögliche Lösung des allgemeinen Konfliktes bzgl. p . Um eine solche Menge zu erhalten, könnte man die folgende triviale Vorgehensweise verwenden:

Setze $TE_{out}(p) := \emptyset, S := TA_{out}(p)$.

Solange $S \neq \emptyset$ führe aus:

Wähle beliebige Transition $t_0 \in S$ aus.

Falls $f(p, t_0) + \sum_{t \in TE_{out}(p)} f(p, t) \leq \mathbf{z}(p)$,

setze: $TE_{out}(p) := TE_{out}(p) \cup \{t_0\}$.

Setze: $S := S \setminus \{t_0\}$.

Nun könnte es je nach realer Problematik durchaus sein, dass nicht alle möglichen Lösungen gleich „gut“ oder „nützlich“ sind. Um nun bestimmte Arten von Lösungen zu bevorzugen, werden Bewertungen und dazugehörige Freigabe-Algorithmen vorgestellt, die – passend zum realen Problem – dem Petri-Netz-Modell einverleibt werden können.

2.2.1 Petri-Netze mit Bewertungen

Es werden drei verschiedene Bewertungen eines Petri-Netzes eingeführt:

1. Allgemeine Bewertungen
2. Prioritäten (insbesondere normierte Prioritäten)
3. Wahrscheinlichkeiten (insbesondere aktualisierte Wahrscheinlichkeiten)

Definition 2.5:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\sigma : F \rightarrow \mathbb{R}_{\geq}$ heißt **allgemeine Bewertungsfunktion** von N , wobei jedem Pfeil $(p, t) \in F$ eine *allgemeine Bewertung* $\sigma(p, t)$ zugewiesen wird.

Beispiel:

Die Pfeile aus Abb. 2.4 haben folgende allgemeine Bewertung:

$$\begin{aligned}\sigma(p_1, t_1) &= 6 \\ \sigma(p_1, t_2) &= 5 \\ \sigma(p_1, t_3) &= 4711 \\ \sigma(p_1, t_4) &= 4,5\end{aligned}$$

Allgemeine Bewertungen können als Kosten, Zeitaufwände oder Nutzen betrachtet werden, die es in bestimmten Problemfällen zu minimieren bzw. zu maximieren gilt. Insbesondere werden in Anlehnung an [Pro13] zwei Spezialfälle betrachtet, wobei die Pfeilbewertungen als *Prioritätszahlen* ($\sigma : F \rightarrow \mathbb{N}$) oder als *Wahrscheinlichkeiten* ($\sigma : F \rightarrow [0; 1]$) aufgefasst werden.

Definition 2.6:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\varrho : F \rightarrow \mathbb{N}$ heißt (**allgemeine**) **Prioritätsfunktion** von N und $\varrho(p, t)$ die *Priorität* des Pfeils $(p, t) \in F$, wenn für jeden Platz $p \in P$ gilt:

$$\varrho(p, t_i) \neq \varrho(p, t_k) \text{ für alle } t_i, t_k \in p \bullet \text{ mit } t_i \neq t_k,$$

und **normierte Prioritätsfunktion** von N , wenn für jeden Platz $p \in P$ zusätzlich $\varrho(p, t) \in \{1, \dots, |p \bullet|\}$ gilt, wobei $|p \bullet|$ die Ausgangsvalenz von p , also die Mächtigkeit von $p \bullet$ ist.

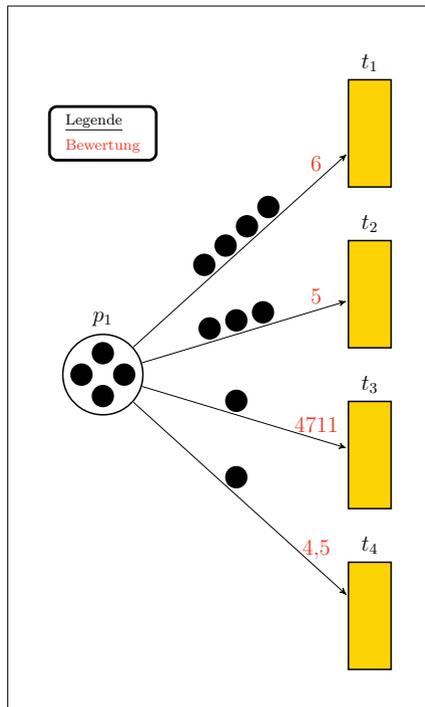


Abbildung 2.4: Petri-Netz mit allgemeinen Bewertungen

Beispiel:

Die Pfeile aus Abb. 2.5 haben folgende (normierte) Prioritäten:

$$\varrho(p_1, t_1) = 1$$

$$\varrho(p_1, t_2) = 3$$

$$\varrho(p_1, t_3) = 2$$

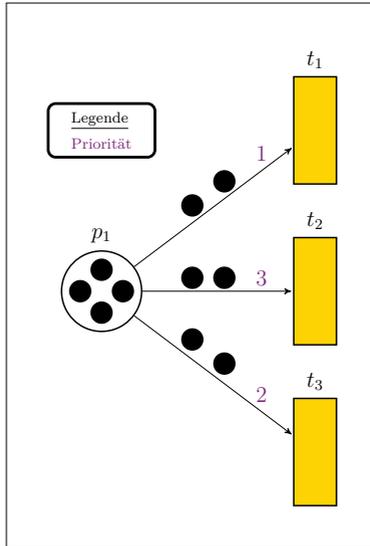


Abbildung 2.5: Petri-Netz mit normierten Prioritäten

Bemerkung:

Man beachte, dass die oben eingeführten Prioritätsfunktionen umkehrbare Abbildungen darstellen, zumal die einzelnen Prioritäten $\varrho(p, t)$ paarweise verschiedene Werte sind, wodurch die Pfeile $p \in P$ einer strikten Ordnung unterliegen. Davon abweichend ließen sich durchaus auch (nicht strikte) Prioritätsfunktionen einführen, bei denen gleiche Prioritäten zugelassen sind, d.h.

$$\varrho(p, t_i) = \varrho(p, t_k) \text{ für } t_i, t_k \in p \bullet \text{ mit } t_i \neq t_k.$$

Auf diese explizite Verallgemeinerung wird hier allerdings verzichtet, zumal diese Möglichkeit implizit durch Def. 2.5 gegeben wird.

Man beachte auch, dass die in [Pro13, S. 79] eingeführten Prioritätsfunktionen *normierte* Prioritätsfunktionen gemäß Def. 2.6 sind. Diese sind in Abweichung zu den allgemeinen Prioritätsfunktionen äquidistant angeordnet.

Bei Bewertungen als Wahrscheinlichkeiten ist zu unterscheiden zwischen einer *allgemeinen Wahrscheinlichkeitsfunktion*, die unabhängig vom Zustand des Petri-Netzes definiert ist, und von einer bzgl. Zustand \mathbf{z} *aktualisierten Wahrscheinlichkeitsfunktion*.

Definition 2.7:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\omega : F \rightarrow [0, 1]$ heißt **allgemeine Wahrscheinlichkeitsfunktion** von N und $\omega(p, t)$ die *allgemeine Wahrscheinlichkeit* des Pfeils $(p, t) \in F$, wenn für jeden Platz $p \in P$ gilt:

$$\sum_{t \in p^\bullet} \omega(p, t) = 1.$$

Sei zudem ein Zustand \mathbf{z} gegeben und $F_z := \{(p, t) \in F \mid t \in TA(\mathbf{z})\}$. Die Abbildung $\omega_z : F_z \rightarrow [0; 1]$ mit

$$\omega_z(p, t) = \frac{\omega(p, t)}{\sum_{t_j \in TA_{out}(p)} \omega(p, t_j)}$$

heißt die **bzgl. \mathbf{z} aktualisierte Wahrscheinlichkeitsfunktion** von N und $\omega_z(p, t)$ die *aktualisierte Wahrscheinlichkeit* des Pfeils $(p, t) \in F_z$.

Hierbei muss darauf geachtet werden, dass aktualisierte Wahrscheinlichkeiten nur für Plätze $p \in P$ in Betracht kommen können, deren Nachbereiche im Zustand \mathbf{z} aktivierte Transitionen besitzen, deren Wahrscheinlichkeiten ungleich null sind, damit gilt:

$$\sum_{t_j \in TA_{out}(p)} \omega(p, t_j) > 0.$$

Beispiel:

Die Pfeile aus Abb. 2.6 haben folgende Wahrscheinlichkeiten:

$$\begin{aligned} \omega(p_1, t_1) &= 0,2 & \omega_z(p_1, t_1) &= 0,4 \\ \omega(p_1, t_2) &= 0,1 & \omega_z(p_1, t_2) &= 0,2 \\ \omega(p_1, t_3) &= 0,2 & \omega_z(p_1, t_3) &= 0,4 \\ \omega(p_1, t_4) &= 0,5 \end{aligned}$$

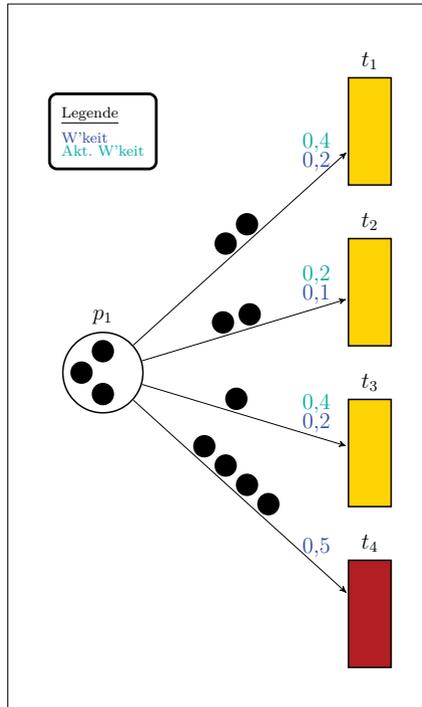


Abbildung 2.6: Petri-Netz mit Wahrscheinlichkeiten

2.2.2 Algorithmische Beschreibungen von lokalen Freigabeprozessen

Die im Folgenden beschriebenen Algorithmen stellen eine lokale Freigabe von Transitionen dar, und zwar immer in Bezug auf einen bestimmten Platz p von N . Des Weiteren lassen sich die Algorithmen unterteilen in solche, mit einer impliziten Zielfunktion, und solche, die eine explizite haben.

2.2.2.1 Lokale Strategien mit impliziter Zielfunktion

Strategien mit impliziter Zielfunktion zielen in erster Linie darauf ab, eine maximal feuerbare Lösung $TE_{out}(p)$ zu ermitteln, die nach einem bestimmten Bildungsprinzip entwickelt wird, ohne dabei eine explizite Lösungsgüte in Betracht zu ziehen.

Für Prioritäten

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \varrho)$ mit Prioritätsfunktion ϱ . Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Prioritäten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben ist und zudem die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ in aufsteigender Reihenfolge¹ ihrer Prioritäten geordnet ist, d.h.

$$\varrho(p, t_i) < \varrho(p, t_j) \text{ für alle } t_i, t_j \in TA_{out}(p) \text{ mit } i < j.$$

Algorithmus 2.8: Lokale Freigabe mittels Prioritätenverfahren

Setze $TE_{out}(p) := \emptyset$.

Für $i = 1, \dots, \kappa(p)$ führe aus:

Falls gilt: $f(p, t_i) + \sum_{t \in TE_{out}(p)} f(p, t) \leq \mathbf{z}(p)$,

setze: $TE_{out}(p) := TE_{out}(p) \cup \{t_i\}$.

Ergebnis: Die Menge $TE_{out}(p)$ beinhaltet alle aktivierten Transitionen, die vom Platz p zum Feuern freigegeben werden. Bei der Bildung der Transitionenmenge $TE_{out}(p)$ gehen die Prioritäten der Transitionen bzgl. p maßgeblich ein.

Beispiel:

In Abb. 2.7 hat p_1 einen allgemeinen Konflikt; es erfolgt die Freigabe nach Prioritäten. Dabei wird in der folgenden Reihenfolge geprüft:

Setzung: $TE_{out}(p_1) := \emptyset$.

Prüfungen:

$$t_1 \text{ wg. } \varrho(p_1, t_1) = 1:$$

$$f(p_1, t_1) = 2 \leq 4 = \mathbf{z}(p_1)$$

$$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_1\} = \{t_1\}$$

¹An dieser Stelle sei angemerkt, dass man Prioritäten auch in absteigender Reihenfolge ordnen könnte. Dies hängt davon ab, wie man Prioritäten dargestellt haben möchte. Im ersten Fall sagt man, dass die Prioritätszahl 1 die höchste Priorität darstellt; im zweiten Fall spiegelt die Prioritätszahl 1 die niedrigste Priorität wider. Die Bedeutungszuweisung der Prioritätszahlen wird grundsätzlich dem Modellierer überlassen.

t_3 wg. $\varrho(p_1, t_3) = 2$:

$$f(p_1, t_1) + f(p_1, t_3) = 4 \leq 4 = \mathbf{z}(p_1)$$

$$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_3\} = \{t_1, t_3\}$$

t_2 wg. $\varrho(p_1, t_2) = 3$:

$$f(p_1, t_1) + f(p_1, t_3) + f(p_1, t_2) = 6 \not\leq 4 = \mathbf{z}(p_1)$$

$$\Rightarrow t_2 \text{ kann nicht (zusätzlich) freigegeben werden}$$

Ergebnis: $TE_{out}(p_1) = \{t_1, t_3\}$, d.h. es werden die Transitionen t_1 und t_3 bzgl. Platz p_1 freigegeben.

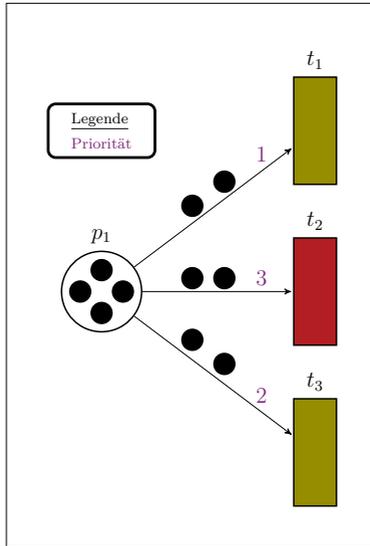


Abbildung 2.7: Lokale Freigabe bei Petri-Netzen mit Prioritätenverfahren

Für Wahrscheinlichkeiten

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \omega)$ mit Wahrscheinlichkeitsfunktion ω . Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Wahrscheinlichkeiten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben ist. Durch eine Indizierung der Elemente $t_i \in TA_{out}(p)$, $i = 1, \dots, \kappa(p)$, ist eine Anordnung der Transitionen vorgegeben, so dass jeder Transition $t_i \in TA_{out}(p)$,

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

$i = 1, \dots, \kappa(p)$, auf kanonische Weise die kumulierte Wahrscheinlichkeit

$$\omega_{\Sigma}(p, t_i) = \sum_{j=1}^i \omega_z(p, t_j)$$

zugewiesen wird.

Algorithmus 2.9: Lokale Freigabe mittels Wahrscheinlichkeiten-Verfahren

Setze:

$$TE_{out}(p) := \emptyset$$

$$L := TA_{out}(p) \text{ (Kandidatenliste)}$$

$$\omega_L := 1 \text{ (Summierte Restkandidaten-W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(0) := 0 \text{ (Hilfs-W'keit für Monte-Carlo-Methode)}$$

Für $i = 1, \dots, \kappa(p)$ führe aus:

Setze:

$$\pi(i) := i \text{ (Reihenfolgezähler)}$$

$$\omega^{aktuell}(\pi(i)) := \omega_z(p, t_i) \text{ (aktualisierte W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(\pi(i)) := \omega_{\Sigma}(p, t_i) \text{ (aktualisierte kumulierte W'keit)}$$

Solange $L \neq \emptyset$ führe aus:

Ermittle Zufallszahl $\rho \in (0; 1]$.

Bestimme $t_i \in L$ mit $\rho \in (\omega_{\Sigma}^{aktuell}(\pi(i) - 1); \omega_{\Sigma}^{aktuell}(\pi(i))]$.

Falls gilt: $f(p, t_i) + \sum_{t \in TE_{out}(p)} f(p, t) \leq \mathbf{z}(p)$,

$$\text{setze: } TE_{out}(p) := TE_{out}(p) \cup \{t_i\}.$$

Setze: $L := L \setminus \{t_i\}$; $\omega_L := \omega_L - \omega_z(p, t_i)$.

Für $j = i + 1, \dots, \kappa(p)$ setze: $\pi(j) := \pi(j) - 1$.

Für alle $t_j \in L$ führe aus:

Setze:

$$\omega^{aktuell}(\pi(j)) := \frac{\omega_z(p, t_j)}{\omega_L};$$

$$\omega_{\Sigma}^{aktuell}(\pi(j)) := \omega_{\Sigma}^{aktuell}(\pi(j) - 1) + \omega^{aktuell}(\pi(j)).$$

Ergebnis: Die Menge $TE_{out}(p)$ beinhaltet alle aktivierten Transitionen, die vom Platz p zum Feuern freigegeben werden.

Bemerkung:

Die Menge L stellt eine „Kandidatenliste“ für eine potentielle Übernahme in die Freigabemenge $TE_{out}(p)$ dar. Sobald eine per Monte-Carlo-Methode ausgewählte Transition t aus L in die Menge $TE_{out}(p)$ übergeht oder aber festgestellt wird, dass diese anstehende Übernahme aus Kapazitätsgründen nicht (mehr) möglich ist, wird diese Transition aus L entfernt, wodurch sich die Kandidatenliste verringert. Die Positionen der verbliebenen Transitionen in der verkürzten Liste L werden mittels „Reihenfolgezähler“ π verwaltet und entsprechend aktualisiert.

Beispiel:

In Abb. 2.8 hat Platz p_1 einen allgemeinen Konflikt; die Freigabe erfolgt diesmal nach Wahrscheinlichkeiten. Es wird folgendermaßen vorgegangen:

Berechnung der aktualisierten und kumulierten Wahrscheinlichkeiten:

$$\begin{aligned} \omega_z(p_1, t_1) &= \frac{0,2}{0,5} = 0,4; \quad \omega_\Sigma(p_1, t_1) = 0,4; \\ \omega_z(p_1, t_2) &= \frac{0,1}{0,5} = 0,2; \quad \omega_\Sigma(p_1, t_2) = 0,4 + 0,2 = 0,6; \\ \omega_z(p_1, t_3) &= \frac{0,2}{0,5} = 0,4; \quad \omega_\Sigma(p_1, t_3) = 0,4 + 0,2 + 0,4 = 1. \end{aligned}$$

Setzungen: $TE_{out}(p_1) := \emptyset; L := TA_{out}(p_1) = \{t_1, t_2, t_3\}; \omega_L := 1; \omega_\Sigma^{aktuell}(0) := 0$

i	1	2	3
$\pi(i)$	1	2	3
$\omega^{aktuell}(\pi(i))$	0,4	0,2	0,4
$\omega_\Sigma^{aktuell}(\pi(i))$	0,4	0,6	1

Schritt 1:

$$L = \{t_1, t_2, t_3\} \neq \emptyset$$

$$\begin{aligned} \rho &= 0,3 \text{ (Zufallszahl)}^1 \\ \rho \in (0; 0,4] &\Rightarrow \text{wähle } t_1 \Rightarrow i = 1 \\ f(p_1, t_1) &= 2 \leq 3 = z(p_1) \\ &\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_1\} = \{t_1\} \\ L &:= \{t_1, t_2, t_3\} \setminus \{t_1\} = \{t_2, t_3\} \\ \omega_L &:= 1 - 0,4 = 0,6 \end{aligned}$$

¹Der Hinweis „Zufallszahl“ soll besagen, dass hier mit Hilfe eines Zufallszahlengenerators eine reelle Zahl als Realisation einer über dem Intervall $[0; 1]$ gleichverteilten Zufallsvariablen X erzeugt wird.

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

i	2	3
$\pi(i)$	1	2
$\omega^{\text{aktuell}}(\pi(i))$	$\frac{1}{3}$	$\frac{2}{3}$
$\omega_{\Sigma}^{\text{aktuell}}(\pi(i))$	$\frac{1}{3}$	1

Schritt 2:

$$L = \{t_2, t_3\} \neq \emptyset$$

$$\rho = 0,1 \text{ (Zufallszahl)}$$

$$\rho \in (0; \frac{1}{3}] \Rightarrow \text{wähle } t_2 \Rightarrow i = 2$$

$$f(p_1, t_1) + f(p_1, t_2) = 4 \not\leq 3 = \mathbf{z}(p_1)$$

$$L := \{t_2, t_3\} \setminus \{t_2\} = \{t_3\}$$

$$\omega_L := 0,6 - 0,2 = 0,4$$

i	3
$\pi(i)$	1
$\omega^{\text{aktuell}}(\pi(i))$	1
$\omega_{\Sigma}^{\text{aktuell}}(\pi(i))$	1

Schritt 3:

$$L = \{t_3\} \neq \emptyset$$

$$\rho = 0,9 \text{ (Zufallszahl)}$$

$$\rho \in (0; 1] \Rightarrow \text{wähle } t_3 \Rightarrow i = 3$$

$$f(p_1, t_1) + f(p_1, t_2) + f(p_1, t_3) = 3 \leq 3 = \mathbf{z}(p_1)$$

$$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_3\} = \{t_1, t_3\}$$

$$L := \{t_3\} \setminus \{t_3\} = \emptyset$$

$$\omega_L := 0,4 - 0,4 = 0$$

Schritt 4:

$$L = \emptyset \Rightarrow \text{Terminiere}$$

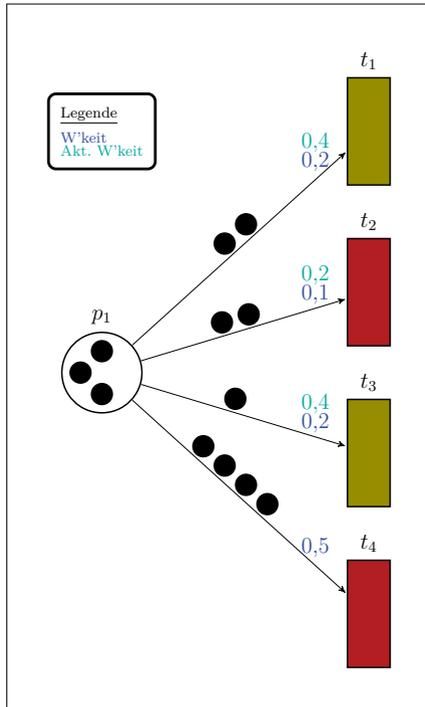


Abbildung 2.8: Lokale Freigabe bei Petri-Netzen mit Wahrscheinlichkeitsverfahren

Ergebnis: $TE_{out}(p_1) = \{t_1, t_3\}$; d.h. p_1 gibt t_1 und t_3 frei.

Eine alternative Lösung ist in Abb. 2.9 dargestellt. Diese ergibt sich, wenn beispielsweise folgende Zufallszahlen in der angegebenen Reihenfolge erzeugt werden: 0,8, 0,9, 0,3.

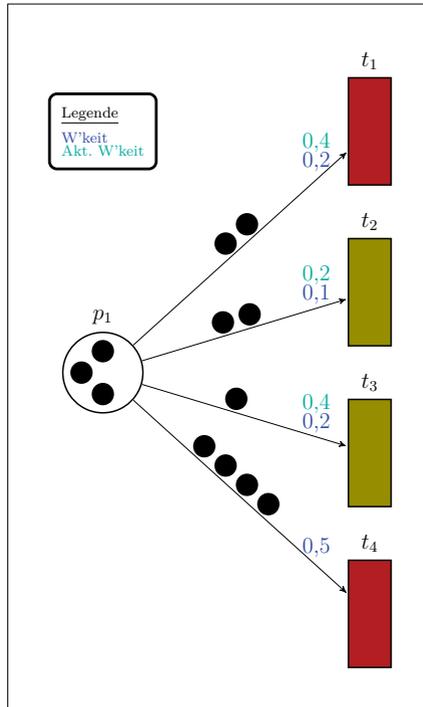


Abbildung 2.9: Lokale Freigabe bei Petri-Netzen mit Wahrscheinlichkeitsverfahren

Bemerkung:

Man beachte, dass mit diesem Algorithmus auch eine Lösung ohne Bewertung modelliert werden kann, indem allen Pfeilen implizit dieselbe Wahrscheinlichkeit zugeordnet wird.

2.2.2.2 Lokale Strategien mit expliziter Zielfunktion

Freigabe-Strategien mit expliziter Zielfunktion zielen nicht nur darauf ab, eine maximal feuerbare Lösung zu finden, sondern darüber hinaus die Optimierung einer bestimmten Zielfunktion zu erreichen. Ein solches Optimum lässt sich bei geringer Anzahl von Transitionen mit der *vollständigen Enumeration* exakt bestimmen. Bei größerer Anzahl von Transitionen würde das allerdings einen enormen Rechenaufwand nach sich ziehen. Daher wird im Folgenden eine Auswahl von möglichen Lösungsheuristiken vorgestellt.

Für Nutzen

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \psi)$ im Zustand \mathbf{z} mit allgemeiner Bewertungsfunktion $\psi : F \rightarrow \mathbb{R}_{\geq}$, die als Nutzen aufgefasst wird. Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Nutzen**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben ist.

Hier werden nun drei Algorithmen vorgestellt:

Der erste ist ein einfacher Greedy-Algorithmus, der in Anlehnung an Algorithmus 2.8 immer versucht, die Transition mit dem jeweils noch größten Nutzen freizugeben. Dafür seien die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ o.B.d.A. in absteigender Reihenfolge ihres Nutzens $\psi(p, t_j)$ geordnet, d.h.

$$\psi(p, t_1) \geq \psi(p, t_2) \geq \dots \geq \psi(p, t_{\kappa(p)}) \text{ für alle } t_j \in TA_{out}(p).$$

Der zweite Algorithmus ist der obige Greedy-Algorithmus, angewandt auf die **Nutzenquotienten** η_j , wobei die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ nun o.B.d.A. in absteigender Reihenfolge ihrer Nutzenquotienten η_j geordnet sind, d.h.

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_{\kappa(p)} \text{ für alle } t_j \in TA_{out}(p) \text{ mit } \eta_j = \frac{\psi(p, t_j)}{f(p, t_j)}.$$

Beide Algorithmen sind im Prinzip das oben beschriebene Prioritätenverfahren, allerdings mit einer anderen Anordnung der Transitionen, wobei die „verallgemeinerten Prioritäten“ $\psi(p, t_j)$ oder η_j wären (wohlgemerkt: i.d.R. $\psi(p, t_j) \notin \mathbb{N}$ und $\eta_j \notin \mathbb{N}$).

Zudem sei angemerkt, dass im Falle von zu minimierenden Bewertungsfunktionen (Kosten, Aufwände, Bearbeitungszeiten) die obigen Freigabestrategien lediglich dahingehend zu modifizieren sind, dass an Stelle einer absteigenden eine aufsteigende Reihenfolge für die Transitionen zu wählen ist.

Der folgende Algorithmus arbeitet nach dem Branch- & Bound-Prinzip, wobei die Transitionen wieder nach dem Nutzenquotienten geordnet sind.

Algorithmus 2.10: Lokale Freigabe mittels Branch & Bound

Setze:

$$X_0 := \{(x_1, x_2, \dots, x_{\kappa(p)}) \mid x_j \in \{0, 1\}, j = 1, \dots, \kappa(p)\} \text{ (Ursprungsknoten)}$$

$$g_0 := 0 \text{ (Gewicht der Teillösung)}$$

$$n_0 := \sum_{j=1}^{\kappa(p)} \psi(p, t_j) \text{ (theoretischer Nutzen des Problems)}$$

$$N := 0 \text{ (Effizienzschranke (Bound))}$$

$$TP := \{X_0\} \text{ (Menge aller Teilknoten)}$$

$$s := 0 \text{ (Zähler der Teilknoten)}$$

$$m_0 := 0 \text{ (Stufe des Teilknotens)}$$

$$L := \emptyset \text{ (Lösung)}$$

1. Schritt: Falls $TP = \emptyset$ terminiere.

Wähle X_r aus TP mit $n_r = \max(n_i \mid X_i \in TP)$ (bei Mehrdeutigkeit wähle willkürlich)

2. Schritt: Falls $m_r < \kappa(p)$, führe aus:

Setze:

$$X_{s+1} := X_r \text{ mit } x_{m_r+1} = 1; X_{s+2} := X_r \text{ mit } x_{m_r+1} = 0;$$

$$n_{s+1} := n_r; n_{s+2} := n_r - \psi(p, t_{m_r+1});$$

$$g_{s+1} := g_r + f(p, t_{m_r+1}); g_{s+2} := g_r;$$

$$m_{s+1} := m_{s+2} := m_r + 1;$$

$$TP_{\text{gültig}} := \{X_{s+i} \mid g_{s+i} \leq \mathbf{z}(p) \wedge n_{s+i} > N, i = 1, 2\}$$

$$TP := (TP \setminus \{X_r\}) \cup TP_{\text{gültig}};$$

Falls $(m_r = \kappa(p) - 1) \wedge (TP_{\text{gültig}} \neq \emptyset)$, führe aus:

Wähle X_l aus $TP_{\text{gültig}}$ mit $n_l = \max(n_i \mid X_i \in TP_{\text{gültig}})$

Setze:

$$N := n_l$$

$$L := X_l$$

Für alle $X_i \in TP$ führe aus:

Falls $n_i \leq N$, setze:

$$TP := TP \setminus \{X_i\};$$

Gehe zum Schritt 1.

Ergebnis: Das binäre Tupel $(x_1, x_2, \dots, x_{\kappa(p)}) \in L$ ist folgendermaßen zu interpretieren:

$x_i = 1 \equiv$ Transition t_i wird freigegeben,

$x_i = 0 \equiv$ Transition t_i wird nicht freigegeben,

so dass gilt:

$$TE_{out}(p) := \{t_i \in TA_{out}(p) \mid x_i = 1\}.$$

Die Menge $TE_{out}(p)$ beinhaltet nun alle aktivierten Transitionen, die vom Platz p zum Feu-ern freigegeben werden.

Als besonders effektiv hat sich die B&B-Variante erwiesen, die eine erste Startlösung schon als Effizienzschranke N nutzt (vgl. Diving B&B in [Kä06, S. 25f.]) Man beachte, dass dabei passieren kann, dass die Lösung des B&B die leere Menge ist, was so zu interpretieren ist, dass es keine Verbesserungsmöglichkeit durch das B&B-Verfahren gibt und die Startlösung bereits das Optimum ist.

Beispiel:

In Abb. 2.10 ist ein Petri-Netz mit Nutzen abgebildet. In diesem Petri-Netz hat Platz p_1 einen allgemeinen Konflikt; die Freigabe erfolgt nach Nutzenwerten. Je nach Freigabe-Prozedur ergeben sich drei Lösungsmöglichkeiten:

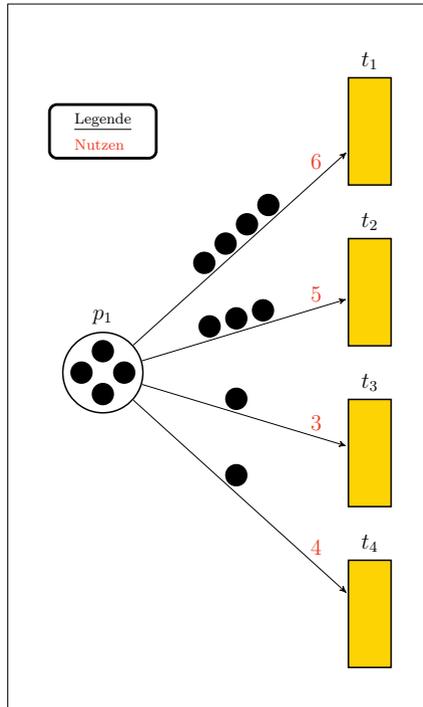


Abbildung 2.10: Petri-Netz mit Nutzen

1. Lösung mit Greedy-Heuristik für Nutzen:

Hier werden die Transitionen nach ihrem Nutzen in absteigender Reihenfolge geordnet, so dass sich die Bearbeitungsreihenfolge t_1, t_2, t_4, t_3 ergibt. Nun wird folgendermaßen verfahren:

Setzung:

$$TE_{out}(p_1) := \emptyset.$$

Prüfungen:

$$t_1 \text{ wg. } \psi(p_1, t_1) = 6:$$

$$f(p_1, t_1) = 4 \leq 4 = \mathbf{z}(p_1) \Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_1\} = \{t_1\}$$

$$t_2 \text{ wg. } \psi(p_1, t_2) = 5:$$

$$f(p_1, t_1) + f(p_1, t_2) = 7 \not\leq 4 = \mathbf{z}(p_1) \Rightarrow t_2 \text{ wird nicht freigegeben.}$$

t_4 wg. $\psi(p_1, t_4) = 4$:

$$f(p_1, t_1) + f(p_1, t_4) = 5 \not\leq 4 = \mathbf{z}(p_1) \Rightarrow t_4 \text{ wird nicht freigegeben.}$$

t_3 wg. $\psi(p_1, t_3) = 3$:

$$f(p_1, t_1) + f(p_1, t_3) = 5 \not\leq 4 = \mathbf{z}(p_1) \Rightarrow t_3 \text{ wird nicht freigegeben.}$$

Ergebnis: $TE_{out}(p_1) = \{t_1\}$, d.h. Platz p_1 gibt allein die Transition t_1 frei. Der gesamte Nutzen dieser Freigabe beträgt $N = 6$.

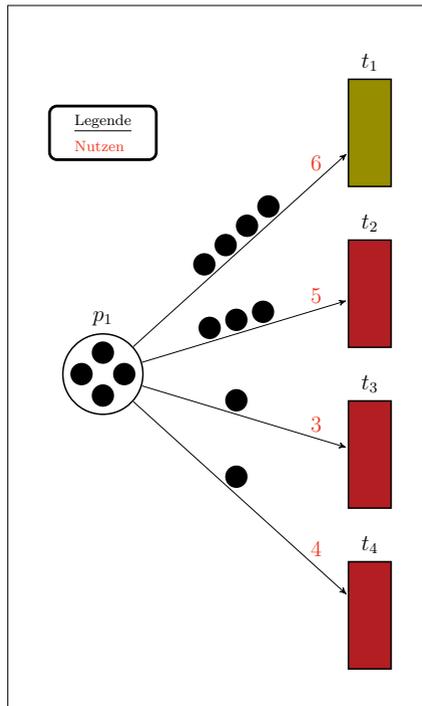


Abbildung 2.11: Lokale Freigabe bei Petri-Netz mit Greedy-Heuristik für Nutzen

2. Lösung mit Greedy-Heuristik für Nutzenquotient:

Hier werden die Transitionen nun nach ihren Nutzenquotienten in absteigender Reihenfolge geordnet, so dass sich die Reihenfolge t_4, t_3, t_2, t_1 auf Grund der zugehörigen Nutzenquotienten $\frac{4}{1} = 4, \frac{3}{1} = 3, \frac{5}{3} = 1,\bar{6}$ und $\frac{6}{4} = 1,5$ ergibt. Nun wird folgendermaßen verfahren:

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

Setzung:

$$TE_{out}(p_1) := \emptyset.$$

Prüfungen:

$$t_4 \text{ wg. } \frac{\psi(p_1, t_4)}{f(p_1, t_4)} = \frac{4}{1} = 4:$$

$$f(p_1, t_4) = 1 \leq 4 = \mathbf{z}(p_1) \Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_4\} = \{t_4\}$$

$$t_3 \text{ wg. } \frac{\psi(p_1, t_3)}{f(p_1, t_3)} = \frac{3}{1} = 3:$$

$$f(p_1, t_4) + f(p_1, t_3) = 2 \leq 4 = \mathbf{z}(p_1) \Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_3\} = \{t_4, t_3\}$$

$$t_2 \text{ wg. } \frac{\psi(p_1, t_2)}{f(p_1, t_2)} = \frac{5}{3} = 1, \bar{6}:$$

$$f(p_1, t_4) + f(p_1, t_3) + f(p_1, t_2) = 5 \not\leq 4 = \mathbf{z}(p_1) \Rightarrow t_2 \text{ wird nicht freigegeben.}$$

$$t_1 \text{ wg. } \frac{\psi(p_1, t_1)}{f(p_1, t_1)} = \frac{6}{4} = 1,5:$$

$$f(p_1, t_4) + f(p_1, t_3) + f(p_1, t_1) = 6 \not\leq 4 = \mathbf{z}(p_1) \Rightarrow t_1 \text{ wird nicht freigegeben.}$$

Ergebnis: $TE_{out}(p_1) = \{t_3, t_4\}$, d.h. Platz p_1 gibt t_3 und t_4 frei. Der gesamte Nutzen dieser Freigabe wäre $N = 7$.

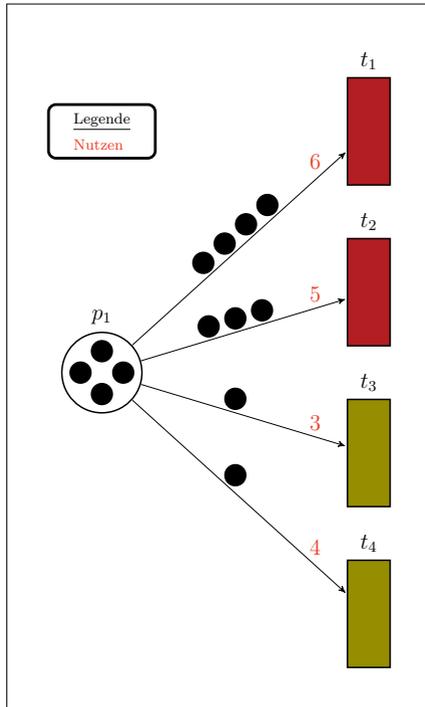


Abbildung 2.12: Lokale Freigabe bei Petri-Netz mit Greedy-Heuristik für Nutzenquotient

3. Lösung mit Branch & Bound:

Hier werden die Transitionen zunächst nach ihren Nutzenquotienten in absteigender Reihenfolge geordnet, so dass sich wieder die Reihenfolge t_4, t_3, t_2, t_1 ergibt. Des Weiteren wird hier die Methode mit einer schon vorgegebenen Startlösung von $N = 7$ aus dem oberen Greedy-Verfahren verwendet. Die einzelnen Schritte des B&B-Verfahrens lassen sich in Abb. 2.13 verfolgen.

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

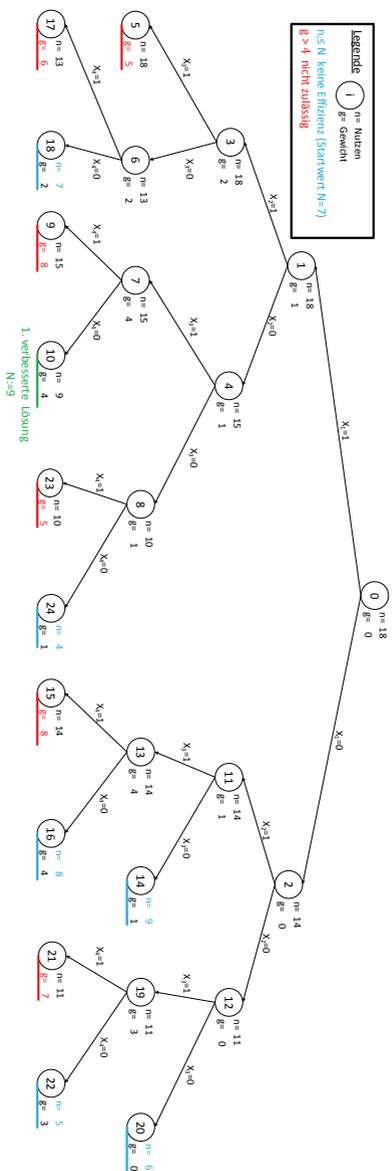


Abbildung 2.13: Lösung nach Branch & Bound

Am Anfangsknoten 0 ist der theoretisch noch zu erzielende Gesamtnutzen $n = 18$ vermerkt. Dieser ergibt sich aber nur, wenn das Tupel $(1, 1, 1, 1)$ realisierbar ist, d.h. $x_1 = x_2 = x_3 = x_4 = 1$ und $TE_{out}(p_1) = \{t_1, t_2, t_3, t_4\}$. Zudem wird die Anzahl g der Tokens, die in Platz p_1 zur Verfügung stehen müssten, vermerkt. Da zu Beginn des Verfahrens noch keine Transitionen freigegeben sind (d.h. $x_1 = x_2 = x_3 = x_4 = 0$ bzw. $TE_{out}(p_1) = \emptyset$), gilt $g = 0$.

Es wird die Transition mit dem größten Nutzenquotienten gewählt, also t_4 , und es werden die beiden Fälle $x_1 = 1$ und $x_1 = 0$ analysiert. Durch entsprechende Verzweigung (branching) ergeben sich die beiden Knoten 1 und 2. Am Knoten 1 sind die Konsequenzen aus der Entscheidung $x_1 = 1$ bzw. $TE_{out}(p_1) := \{t_4\}$ vermerkt, d.h. der theoretisch noch maximal mögliche Gesamtnutzen ist unverändert geblieben, während sich wegen $f(p_1, t_4) = 1$ ein „Verbrauch“ von 1 Token eingestellt hat. Solange $g \leq \mathbf{z}(p_1)$ gilt (*Zulässigkeitsgrenze*), ist der Knoten für weitere Verzweigungen zulässig. Am Knoten 2 sind die Konsequenzen aus der Entscheidung $x_1 = 0$ bzw. weiterhin $TE_{out}(p_1) = \emptyset$ vermerkt, d.h. der theoretisch noch mögliche Gesamtnutzen hat sich um $\psi(p_1, t_4) = 4$ verringert, also $n = 14$, während die Tokenanzahl g unverändert bleibt. In der Folge wird immer an demjenigen Knoten weiter verzweigt, der den maximalen Wert n aufweist.

Eine erste zulässige Lösung ist am Knoten 10 erzielt: Das zugehörige binäre Tupel $(1, 0, 1, 0)$ entspricht der Freigabemenge $TE_{out}(p_1) = \{t_2, t_4\}$ mit dem Gesamtnutzen $N = 9$. Im weiteren Verlauf des Verfahrens wird diese Zwischenlösung $N = 9$ als untere Grenze (*Effizienzgrenze*) für das weitere Verzweigen eingesetzt, indem von zulässigen Knoten, deren Nutzenwert n nicht größer als die untere Grenze ist, auch nicht weiter verzweigt wird. Die Abbrüche auf Grund von Zulässigkeits- oder Effizienzgrenzen sind in Abb. 2.13 durch rote Striche für *nicht zulässig* bzw. blaue Striche für *nicht effizient* gekennzeichnet. Das Erreichen von zulässigen Lösungen wird durch grüne Striche verdeutlicht.

Insgesamt ergibt sich ein Baum mit 25 Knoten, wobei sich keine weitere Verbesserung ergibt, d.h. Platz p_1 gibt die Transitionen t_2 und t_4 frei. Der gesamte Nutzen dieser Freigabe beträgt $N = 9$. Es ist leicht nachzuvollziehen, dass hiermit der maximal mögliche Gesamtnutzen für eine Freigabe bzgl. p_1 erzielt ist.

2.2.3 Petri-Netze mit Konfliktlösung

Die oben beschriebenen FreigabeprozEDUREN basieren auf einer lokalen Betrachtung von allgemeinen Konfliktsituationen von einzelnen Plätzen eines Petri-Netzes N . Ausgangspunkt ist jeweils ein Platz p von N , der gemäß Def. 2.2 einer allgemeinen Konfliktsituation unterliegt.

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

In Abb. 2.2 wird bereits gezeigt, wie sich lokale Konflikte durch die globale Betrachtungsweise auf gleichzeitig mehrere allgemeine Konfliktsituationen gegenseitig beeinflussen können. Bevor allerdings ein Konzept zur kollektiven Lösung von mehrfachen allgemeinen Konfliktsituationen behandelt wird (hierzu siehe Abschn. 2.3), sollen hier zunächst die verschiedenen lokalen FreigabeprozEDUREN für einen globalisierten Freigabeprozess hinzugezogen werden.

Zusätzlich zur gleichzeitigen Betrachtung von mehreren Plätzen in jeweils allgemeinen Konfliktsituationen muss davon ausgegangen werden, dass die Plätze eines Petri-Netzes einerseits verschiedene Arten von Bewertungen aufweisen und andererseits ihre lokalen Konflikte mit unterschiedlichen FreigabeprozEDUREN gelöst werden können. Solche Petri-Netze, in denen die allgemeinen Konfliktsituationen sämtlicher Plätze durch lokale FreigabeprozEDUREN gelöst werden, wodurch auf kanonische Weise eine *globale Freigabemenge* $TE(\mathbf{z})$ entsteht (vgl. Def. 2.12), werden auch *Petri-Netze mit Konfliktlösung* genannt. Hierbei ist jedem Platz p eindeutig eine Bewertungsart (Prio, W'keit, Nutzen) und eine dazu passende FreigabeprozEDURE zugewiesen.

Definition 2.11:

Das Tupel $(P, T, F, B, f, \mathbf{z}, \alpha, \delta, \zeta)$ ist ein **Petri-Netz mit Konfliktlösung**, wenn gilt:

- $(P, T, F, B, f, \mathbf{z})$ ist ein Petri-Netz im Zustand \mathbf{z} .
- Die Abbildung $\alpha : P \rightarrow \{\text{Prio}, \text{W'keit}, \text{Nutzen}\}$ ist eine **Bewertungsartfunktion**, welche jedem Platz $p \in P$ eine bestimmte Art von Bewertung zuordnet.
- Die Abbildung $\delta : F \rightarrow \left\{ \begin{array}{ll} \mathbb{N} & \text{für } \alpha(p) = \text{Prio} \\ [0, 1] & \text{für } \alpha(p) = \text{W'keit} \\ \mathbb{R}_{\geq} & \text{für } \alpha(p) = \text{Nutzen} \end{array} \right\}$ ist eine **Bewertungsfunktion**, welche jedem Pfeil von einem Platz $p \in P$ zu einer Transition $t \in p\bullet$ eine Bewertung $\delta(p, t)$ entsprechend der Bewertungsart des Platzes und unter den folgenden Bedingung zuordnet:

- dass jede Priorität nur einmal von jedem Platz p benutzt wird

$$\delta(p, t_j) \neq \delta(p, t_k) \quad \forall t_j, t_k \in p\bullet \quad \text{mit } t_j \neq t_k \quad \text{falls } \alpha(p) = \text{Prio},$$

- dass die Summe der Wahrscheinlichkeiten gleich 1 ist

$$\sum_{t \in p\bullet} \delta(p, t) = 1 \quad \forall p \quad \text{falls } \alpha(p) = \text{W'keit}.$$

• Die Abbildung

$$\zeta : P \rightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} \{\text{lokales Prioritäten-Verfahren}\} \\ \{\text{lokales W'keiten-Verfahren}\} \end{array} \right\} \quad \left. \begin{array}{l} \text{für } \alpha(p) = \text{Prio} \\ \text{für } \alpha(p) = \text{W'keit} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{lokale vollständige Enumeration} \\ \text{lokales Greedy-V. für Nutzen} \\ \text{lokales Greedy-V. für Nutzenquotient} \\ \text{lokales B\&B-Verfahren} \end{array} \right\} \quad \left. \begin{array}{l} \\ \\ \text{für } \alpha(p) = \text{Nutzen} \end{array} \right\} \end{array} \right\}$$

ist eine **Lösungsstrategiefunktion**, welche jedem Platz $p \in P$ eine bestimmte Konfliktlösungsstrategie zuordnet.

Bemerkung:

Hier wird nur eine Auswahl von Bewertungsarten und Lösungsstrategien vorgestellt, was bedeutet, dass man die obige Definition um zusätzliche Bewertungsarten oder Lösungsstrategien erweitern könnte. Des Weiteren sei angemerkt, dass bei Plätzen, bei denen nie eine Konfliktsituation eintreten kann (z.B. bei nur einer Transition im Nachbereich), auch keine Konfliktlösung explizit angegeben werden muss, um so das Modell übersichtlicher zu halten. Vollständigkeitshalber könnte man bei solchen Plätzen mit Wahrscheinlichkeiten arbeiten, wobei die einzige Transition im Nachbereich die Wahrscheinlichkeit 1 hat.

Definition 2.12:

Das Tupel $(P, T, F, B, f, \mathbf{z}, \alpha, \delta, \zeta)$ sei ein Petri-Netz mit Konfliktlösung. Die Transitionenmenge

$$TE(\mathbf{z}) := \{t \in TA(\mathbf{z}) \mid \forall p \in \bullet t : t \in TE_{out}(p, \mathbf{z})\}$$

wird die **Menge der global freigegebenen Transitionen** des Petri-Netzes im Zustand \mathbf{z} genannt.

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch TE statt $TE(\mathbf{z})$ geschrieben.

Beispiel:

Die Plätze aus Abb. 2.14 haben folgende Bewertungsarten:

$$\begin{aligned} \alpha(p_1) &= \text{Prio} \\ \alpha(p_2) &= \text{W'keit} \end{aligned}$$

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

Die Pfeile aus Abb. 2.14 haben folgende Bewertungen:

$$\begin{aligned}\delta(p_1, t_1) &= 1 \\ \delta(p_1, t_2) &= 2 \\ \delta(p_2, t_1) &= 0,5 \\ \delta(p_2, t_2) &= 0,5\end{aligned}$$

Die Plätze aus Abb. 2.14 haben folgende Lösungsstrategien:

$$\begin{aligned}\zeta(p_1) &= \text{lokales Prioritäten-Verfahren} \\ \zeta(p_2) &= \text{lokales W'keiten-Verfahren}\end{aligned}$$

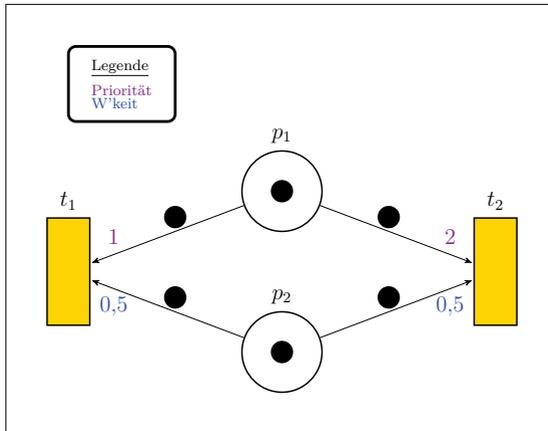


Abbildung 2.14: Beispiel für ein Petri-Netze mit Konfliktlösung

In Abb. 2.14 hat sowohl p_1 als auch p_2 einen allgemeinen Konflikt. Die Freigabe erfolgt bei p_1 nach Prioritäten und bei p_2 nach Wahrscheinlichkeiten.

Es können nun zwei Fälle eintreten.

1. Fall:

p_1 wird t_1 freigeben: $TE_{out}(p_1) = \{t_1\}$.

p_2 wird t_1 freigeben: $TE_{out}(p_2) = \{t_1\}$.

Daraus ergibt sich die Freigabemenge $TE = \{t_1\}$, weil alle Plätze im Vorbereich der Transition t_1 die Freigabe erteilt haben.

2. Fall:

p_1 wird t_1 freigeben: $TE_{out}(p_1) = \{t_1\}$.

p_2 wird t_2 freigeben: $TE_{out}(p_2) = \{t_2\}$.

$\Rightarrow TE = \emptyset$, weil keine Transition von allen Plätzen in ihrem Vorbereich freigegeben wird.

Dieses Beispiel zeigt, dass es durchaus wünschenswert ist, über die individuelle Betrachtungsweise der einzelnen Plätze hinweg auch kollektiv nach einer Lösung von globalen Konfliktsituationen zu suchen, indem zumindest eine nichtleere oder sogar eine möglichst mächtige Freigabemenge $TE(\mathbf{z})$ ermittelt wird.

2.3 Kollektive Lösungsansätze

Die lokale Konfliktlösung mit Nutzenmaximierung betrachtet immer nur einen Platz, um den Nutzen für diesen Platz zu maximieren. Hierbei weisen die Plätze ein individuelles Verhalten auf, wie es sich möglicherweise auch in vielen realen Systemen oder Prozessen darstellt und somit eine passende Modellbeschreibung ist. Nun könnte es aber auch der Fall sein, dass mehrere (oder gar alle) Plätze ein „altruistisches“ Verhalten aufweisen sollen, um entsprechende Sachverhalte von realen Systemen und Prozessen damit besser modellieren zu können. Dies bedeutet, dass nicht jeder Platz versucht, seinen individuellen Nutzen für sich zu maximieren, sondern den globalen Gesamtnutzen für alle bzw. mehrere Plätze zu maximieren.

Betrachtet man das Petri-Netz in Abb. 2.15, so ist ersichtlich, dass unabhängig davon, welches lokale Verfahren zur individuellen Nutzenmaximierung verwendet wird, der Platz p_1 die Transition t_1 und der Platz p_2 die Transition t_2 freigeben wird. Im Endeffekt ist allerdings keine Transition global freigegeben (also gemeinsam von beiden Plätzen). Danach ergäbe sich der globale Gesamtnutzen $N = 0$.

Nun ist das Ergebnis $N = 0$ das insgesamt schlechteste Ergebnis für den globalen Gesamtnutzen. Daher ist im Hinblick auf einen zu maximierenden Gesamtnutzen eine lokale Betrachtungsweise problematisch. Eine andere Lösungsmöglichkeit wäre, dass die Plätze sich gemeinsam „absprechen“ sollten, welche Transition(en) sie gemeinsam freigeben. Mögliche Anwendungen zur Modellierung von Kollektiventscheidungen liegen zum Beispiel vor, wenn ein Team sich in Konfliktsituationen abspricht und gemeinsam entscheidet, um im Nachhinein den Gesamtnutzen untereinander aufzuteilen, ähnlich wie bei kooperativen Spielen mit Seitenzahlungsmöglichkeiten.

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

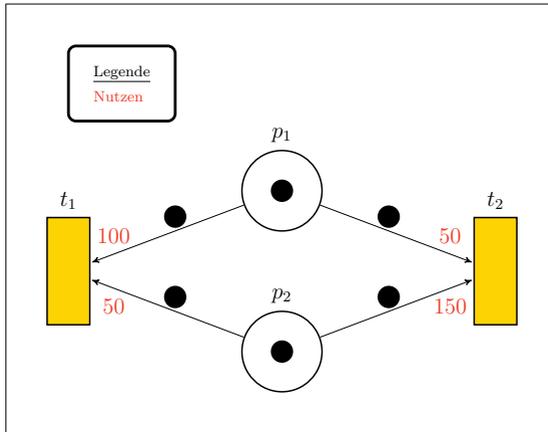


Abbildung 2.15: Petri-Netz zur lokalen Nutzenmaximierung

Im obigen Beispiel ist ersichtlich, dass nur eine von beiden Transitionen global freigebbar ist, entweder Transition t_1 oder t_2 . Bei Freigabe von t_1 wäre der Nutzen $N = 100 + 50 = 150$, bei Freigabe von t_2 ergäbe sich $N = 150 + 50 = 200$. Es wäre also nun möglich, diesen Nutzen an die Transitionen zu schreiben und dann diejenige Transition zu wählen, deren Freigabe den größten Nutzen induziert. Dies führt zu einem Petri-Netz mit Transitionenbewertungen (siehe Abb. 2.16).

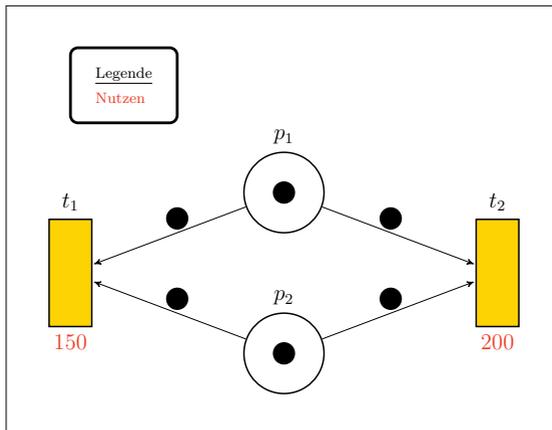


Abbildung 2.16: Petri-Netz zur kollektiven Nutzenmaximierung

Die Plätze, die nun zusammen entscheiden, nennt man eine *kollektive Gruppe* oder ein *Kollektiv*. Auf ein Kollektiv lassen sich ähnliche Konfliktlösungen anwenden wie bei der lokalen

Entscheidung, nur wird hier keine lokale Freigabemenge erzeugt, sondern eine kollektive Freigabemenge, die im Extremfall, wenn alle Plätze des Petri-Netzes zum Kollektiv zählen, gleich der globalen Freigabemenge ist.

Ein weiteres Beispiel für den Unterschied zwischen lokaler und kollektiver Entscheidung wäre das folgende: Es gibt zwei Mitarbeiter und es sind zwei Aufträge abzuarbeiten; für jeden der beiden Aufträge werden die beiden Mitarbeiter gebraucht. Die Auswahl des Auftrages, den jeder Mitarbeiter erledigen will, hängt – bildlich gesprochen – von einem Münzwurf ab, z.B. bei Kopf für Auftrag 1 und Zahl für Auftrag 2. Dann könnte es sein, dass der eine Mitarbeiter den Auftrag 1 und der andere den Auftrag 2 erledigen will, wodurch keiner der beiden Aufträge letztendlich erledigt werden kann. Eine sinnvollere Strategie wäre es, dass beide Mitarbeiter zusammen nur eine einzige Münze werfen und beide somit automatisch ein und denselben Auftrag erledigen, was einer Kollektivlösung entspricht (vgl. Abb. 2.17).

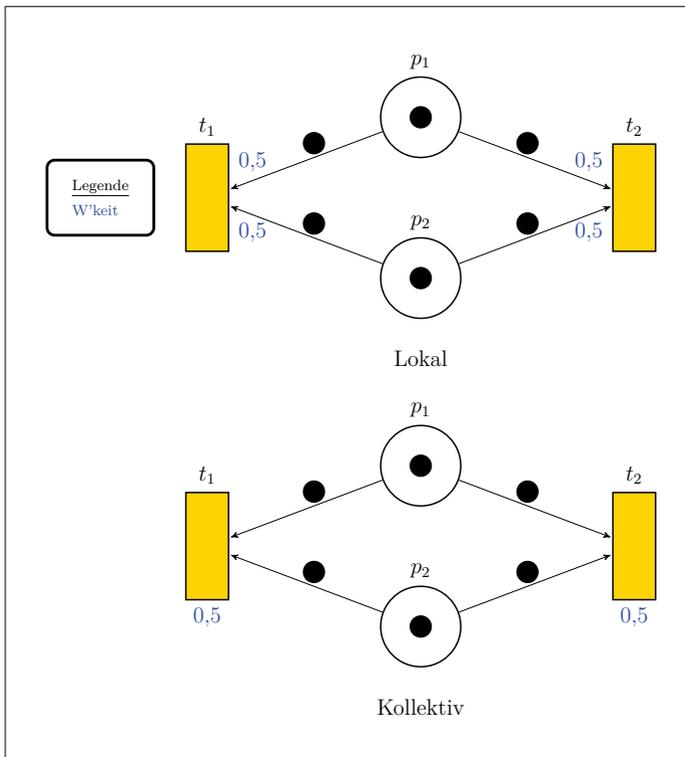


Abbildung 2.17: Unterschied zwischen lokaler und kollektiver Modellierung

2.3.1 Kollektive

Die bis jetzt beschriebenen lokalen Konfliktlösungen (Abschn. 2.2) erzeugen lediglich eine lokale Freigabe von Transitionen, und zwar immer nur in Bezug auf einen bestimmten Platz p von N . Wenn diese Prozedur für alle Plätze von N durchgeführt worden ist, so ist nicht gewährleistet, dass alle lokal freigegebenen Transitionen

$$t \in \bigcup_{p \in P} TE_{out}(p)$$

nebenläufig feuierbar sind. Dieses liegt darin begründet, dass sich mehrere allgemeine Konflikte gegenseitig beeinflussen (siehe Abb. 2.14). Für die Berücksichtigung der gegenseitigen Beeinflussung erfolgen die Einführung von Transitionenbewertungen statt Pfeilbewertungen und die Bildung von *Kollektiven* und von dazugehörigen *kollektiven Freigabeprozeduren*. Dabei ist unter einem Kollektiv eine Menge von Plätzen zu verstehen, für die nicht mehr nur jeweils eine lokale Freigabemenge gebildet wird, sondern eine Freigabemenge über das gesamte Kollektiv, wodurch insbesondere auch die gegenseitige Beeinflussung von mehreren allgemeinen Konflikten Berücksichtigung findet.

Definition 2.13:

Das Tupel $N = (P, T, F, B, f)$ sei ein Petri-Netz. Eine nichtleere Teilmenge \mathcal{K} von P ist ein **Kollektiv**, wenn gilt:

$$t \in \mathcal{K} \bullet \Rightarrow \bullet t \subseteq \mathcal{K},$$

wobei

$$\mathcal{K} \bullet = \bigcup_{p \in \mathcal{K}} p \bullet$$

als der **gemeinsame Nachbereich des Kollektivs** bezeichnet wird.

Man mache sich klar, dass in jedem Petri-Netz (mindestens) ein Kollektiv existiert, zumal P selbst ein Kollektiv ist.

Definition 2.14:

Das Tupel $N = (P, T, F, B, f)$ sei ein Petri-Netz und es sei ein Kollektiv \mathcal{K} gegeben. Falls $\mathcal{K} \subset P$, so ist \mathcal{K} ein **echtes Kollektiv**.

Es gibt Petri-Netze, die keine bzw. mehrere echte Kollektive besitzen. Des Weiteren ist die Vereinigung von Kollektiven wieder ein Kollektiv. Für zwei Kollektive $\mathcal{K}_1, \mathcal{K}_2$ gilt entweder $\mathcal{K}_1 \subseteq \mathcal{K}_2$ oder $\mathcal{K}_2 \subseteq \mathcal{K}_1$ oder $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$.

Definition 2.15:

Das Tupel $N = (P, T, F, B, f)$ sei ein Petri-Netz. Eine Menge $\mathcal{KM} = \{\mathcal{K}_1, \dots, \mathcal{K}_v\}$ von disjunkten Kollektiven $\mathcal{K}_1, \dots, \mathcal{K}_v$ wird eine **Kollektivmenge** genannt.

Hiernach kann kein Platz in zwei Kollektiven einer Kollektivmenge enthalten sein.

Beispiel:

Die in Abb. 2.18 gezeigten Plätze p_1, \dots, p_5 bilden ein Kollektiv, wohingegen jede echte Teilmenge von diesen fünf Plätzen kein Kollektiv bilden kann.

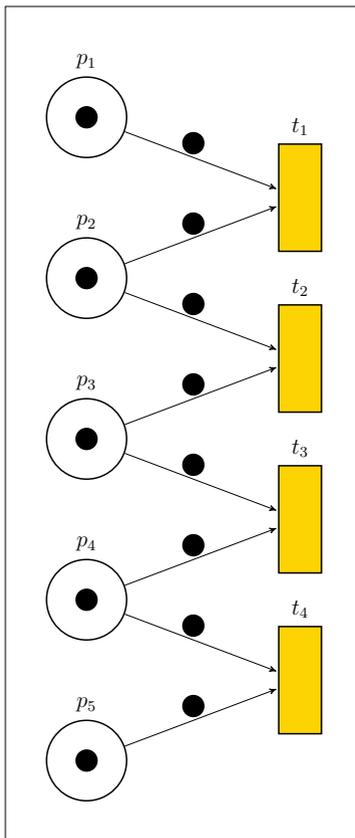


Abbildung 2.18: Beispiel eines Kollektivs

Definition 2.16:

Das Tupel $N = (P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz mit der Markierung \mathbf{z} . Zudem sei eine Kollektivmenge $\mathcal{KM} = \{\mathcal{K}_1, \dots, \mathcal{K}_v\}$ gegeben, dann bezeichnet man mit $\mathbf{z}(\mathcal{K})$ den **Zustand eines Kollektivs** $\mathcal{K} \in \mathcal{KM}$, wobei $\mathbf{z}(\mathcal{K})$ alle $\mathbf{z}(p)$ mit $p \in \mathcal{K}$ umfasst, die man o.B.d.A. auch als Vektor:

$$\mathbf{z}(\mathcal{K}) = (\mathbf{z}(p_1), \dots, \mathbf{z}(p_{|\mathcal{K}|})) \text{ mit } p_i \in \mathcal{K}, i = 1, \dots, |\mathcal{K}|,$$

darstellen kann.

Beispiel:

Die in Abb. 2.19 gezeigten Plätze p_1, \dots, p_3 bilden ein Kollektiv \mathcal{K} , wohingegen p_4 nicht zu diesem Kollektiv gehört. Aus dem Zustand $\mathbf{z} = (2, 1, 2, 3)$ des Petri-Netzes ergibt sich somit der Zustand des Kollektivs \mathcal{K} :

$$\mathbf{z}(\mathcal{K}) = (2, 1, 2).$$

Definition 2.17:

Das Tupel $N = (P, T, F, B, f, \mathbf{z})$ sei ein Petri-Netz mit der Markierung \mathbf{z} . Zudem sei eine Kollektivmenge $\mathcal{KM} = \{\mathcal{K}_1, \dots, \mathcal{K}_v\}$ gegeben.

- $TA_{in}(\mathcal{K}, \mathbf{z})$ bezeichnet die **Menge der aktivierten Transitionen des Vorbereichs** eines Kollektivs $\mathcal{K} \in \mathcal{KM}$:

$$TA_{in}(\mathcal{K}, \mathbf{z}) := TA(\mathbf{z}) \cap \bullet\mathcal{K}, \text{ wobei } \bullet\mathcal{K} = \bigcup_{p \in \mathcal{K}} \bullet p.$$

- $TA_{out}(\mathcal{K}, \mathbf{z})$ bezeichnet die **Menge der aktivierten Transitionen des Nachbereichs** eines Kollektivs $\mathcal{K} \in \mathcal{KM}$:

$$TA_{out}(\mathcal{K}, \mathbf{z}) := TA(\mathbf{z}) \cap \mathcal{K} \bullet.$$

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch $TA_{in}(\mathcal{K})$ statt $TA_{in}(\mathcal{K}, \mathbf{z})$ bzw. $TA_{out}(\mathcal{K})$ statt $TA_{out}(\mathcal{K}, \mathbf{z})$ geschrieben.

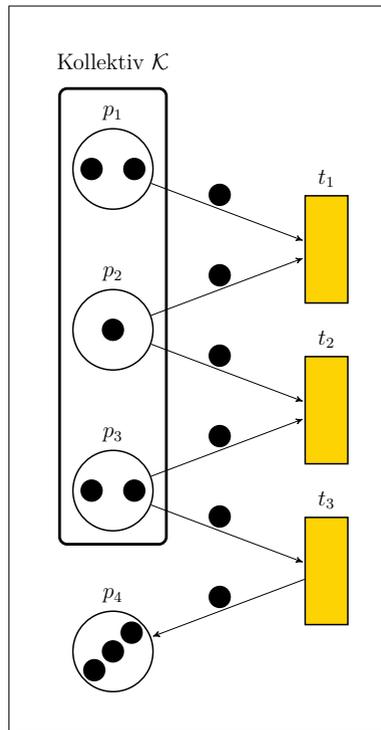


Abbildung 2.19: Zustand eines Kollektivs

2.3.2 Kollektive Bewertungen

Die Plätze eines Kollektivs können nun je nach mitgegebener Bewertung und Freigabeprozedur eine Menge von Transitionen gemeinsam freigeben.

Definition 2.18:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\sigma : \mathcal{KM}\bullet \rightarrow \mathbb{R}_{\geq}$ heißt **allgemeine kollektive Bewertungsfunktion** von N , wobei jeder Transition $t \in \mathcal{KM}\bullet$ eine *allgemeine kollektive Bewertung* $\sigma(t)$ zugewiesen wird (vgl. Def. 2.5) und $\mathcal{KM}\bullet = \bigcup_{\mathcal{K} \in \mathcal{KM}} \mathcal{K}\bullet$.

Definition 2.19:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\varrho : \mathcal{KM}\bullet \rightarrow \mathbb{N}$ heißt **kollektive Prioritätsfunktion** von N , wobei jeder Transition $t \in \mathcal{KM}\bullet$ eine *kollektive Priorität* $\varrho(t)$ zugewiesen wird und für jedes Kollektiv $\mathcal{K} \in \mathcal{KM}$ gilt:

$$\varrho(t_i) \neq \varrho(t_k) \text{ für alle } t_i, t_k \in \mathcal{K}\bullet$$

und **normierte kollektive Prioritätsfunktion** von N , wenn für jede Transition $t \in \mathcal{K}\bullet$ zusätzlich $\varrho(t) \in \{1, \dots, |\mathcal{K}\bullet|\}$ gilt (vgl. Def. 2.6).

Bei den Wahrscheinlichkeiten ist zu unterscheiden zwischen einer *allgemeinen kollektiven Wahrscheinlichkeitsfunktion*, die unabhängig vom Zustand des Petri-Netzes definiert ist, und einer bzgl. Zustand \mathbf{z} *aktualisierten kollektiven Wahrscheinlichkeitsfunktion*.

Definition 2.20:

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f)$. Eine Abbildung $\omega : \mathcal{KM}\bullet \rightarrow [0, 1]$ heißt **allgemeine kollektive Wahrscheinlichkeitsfunktion** von N , wobei jeder Transition $t \in \mathcal{KM}\bullet$ eine *allgemeine kollektive Wahrscheinlichkeit* $\omega(t)$ zugewiesen wird und für jedes Kollektiv $\mathcal{K} \in \mathcal{KM}$ gilt:

$$\sum_{t \in \mathcal{K}\bullet} \omega(t) = 1.$$

Sei zudem ein Zustand \mathbf{z} gegeben. Die Menge $TA(\mathbf{z})$ bezeichne die Menge aller aktivierten Transitionen von N bzgl. Zustand \mathbf{z} . Die Abbildung $\omega_{\mathbf{z}} : (\mathcal{KM}\bullet \cap TA(\mathbf{z})) \rightarrow [0, 1]$ mit

$$\omega_{\mathbf{z}}(t) = \frac{\omega(t)}{\sum_{t_j \in TA_{out}(\mathcal{K})} \omega(t_j)}$$

heißt die bzgl. \mathbf{z} **aktualisierte kollektive Wahrscheinlichkeitsfunktion** von N und $\omega_{\mathbf{z}}(t)$ die *aktualisierte Wahrscheinlichkeit* der Transition $t \in \mathcal{K}\bullet$ (vgl. Def. 2.7).

Hierbei muss darauf geachtet werden, dass aktualisierte Wahrscheinlichkeiten nur für Transitionen $t \in \mathcal{K}\bullet$ in Betracht kommen können, die im Zustand \mathbf{z} aktiviert und deren Wahrscheinlichkeiten ungleich null sind, damit gilt:

$$\sum_{t_j \in TA_{out}(\mathcal{K})} \omega(t_j) > 0.$$

2.3.3 Algorithmen zur kollektiven Konfliktlösung

Die im Folgenden beschriebenen Algorithmen stellen eine kollektive Freigabe von Transitionen dar, und zwar in Bezug auf ein Kollektiv von Plätzen von N .

Für Prioritäten

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \varrho)$ mit Prioritätsfunktion ϱ im Zustand \mathbf{z} . Ein Kollektiv $\mathcal{K} \in \mathcal{KM}$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(\mathcal{K})$ von aktivierten Transitionen aus $TA_{out}(\mathcal{K})$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **kollektiven Freigabemenge** $TE_{out}(\mathcal{K})$ **mittels Prioritäten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Transitionen $t_1, \dots, t_{|TA_{out}(\mathcal{K})|} \in TA_{out}(\mathcal{K})$ in aufsteigender Reihenfolge ihrer Prioritäten geordnet ist, d.h.

$$\varrho(t_i) < \varrho(t_j) \text{ für alle } t_i, t_j \in TA_{out}(\mathcal{K}) \text{ mit } i < j.$$

Algorithmus 2.21: Kollektive Freigabe mittels Prioritäten

Setze $TE_{out}(\mathcal{K}) := \emptyset$.

Für $i = 1, \dots, |TA_{out}(\mathcal{K})|$ führe aus:

$$\text{Falls } \forall p \in \bullet t_i \text{ gilt: } f(p, t_i) + \sum_{t \in (TE_{out}(\mathcal{K}) \cap p^\bullet)} f(p, t) \leq \mathbf{z}(p),$$

$$\text{setze: } TE_{out}(\mathcal{K}) := TE_{out}(\mathcal{K}) \cup \{t_i\}.$$

Ergebnis: Die Menge $TE_{out}(\mathcal{K})$ beinhaltet alle aktivierten Transitionen, die vom Kollektiv \mathcal{K} zum Feuern freigegeben werden.

Man beachte, dass dieser Algorithmus nach wie vor auch auf Nutzenbewertungen anwendbar ist, nur die Reihenfolge der Transitionen anders bestimmt wird.

Für Wahrscheinlichkeiten

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \omega)$ mit Wahrscheinlichkeitsfunktion ω im Zustand \mathbf{z} . Ein Kollektiv $\mathcal{K} \in \mathcal{KM}$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(\mathcal{K})$ von aktivierten Transitionen aus $TA_{out}(\mathcal{K})$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **kollektiven Freigabemenge** $TE_{out}(\mathcal{K})$ **mittels Wahrscheinlichkeiten**. Durch eine Indizierung der Elemente $t_i \in TA_{out}(\mathcal{K})$, $i = 1, \dots, |TA_{out}(\mathcal{K})|$, ist eine Anordnung der Transitionen vorgegeben, so dass jeder Transition $t_i \in TA_{out}(\mathcal{K})$, $i = 1, \dots, |TA_{out}(\mathcal{K})|$, auf kanonische Weise die kumulierte Wahrschein-

lichkeit

$$\omega_{\Sigma}(t_i) = \sum_{j=1}^i \omega_z(t_j)$$

zugewiesen wird.

Algorithmus 2.22: Kollektive Freigabe mittels Wahrscheinlichkeiten

Setze:

$$TE_{out}(\mathcal{K}) := \emptyset.$$

$$L := TA_{out}(\mathcal{K}) \text{ (Kandidatenliste)}$$

$$\omega_L := 1 \text{ (Summierte Restkandidaten-W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(0) := 0 \text{ (Hilfs-W'keit für Monte-Carlo-Methode)}$$

Für $i = 1, \dots, |TA_{out}(\mathcal{K})|$ führe aus:

Setze:

$$\pi(i) := i \text{ (Reihenfolgezähler)}$$

$$\omega^{aktuell}(\pi(i)) := \omega_z(t_i) \text{ (aktualisierte W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(\pi(i)) := \omega_{\Sigma}(t_i) \text{ (aktualisierte kumulierte W'keit)}$$

Solange $L \neq \emptyset$ führe aus:

Ermittle Zufallszahl $\rho \in (0; 1]$.

Bestimme $t_i \in L$ mit $\rho \in (\omega_{\Sigma}^{aktuell}(\pi(i) - 1); \omega_{\Sigma}^{aktuell}(\pi(i))]$.

Falls $\forall p \in \bullet t_i$ gilt: $f(p, t_i) + \sum_{t \in (TE_{out}(\mathcal{K}) \cap p\bullet)} f(p, t) \leq \mathbf{z}(p)$,

setze: $TE_{out}(\mathcal{K}) := TE_{out}(\mathcal{K}) \cup \{t_i\}$.

Setze: $L := L \setminus \{t_i\}$; $\omega_L := \omega_L - \omega_z(t_i)$.

Für $j = i + 1, \dots, |TA_{out}(\mathcal{K})|$ setze: $\pi(j) := \pi(j) - 1$.

Für alle $t_j \in L$ führe aus:

Setze:

$$\omega^{aktuell}(\pi(j)) := \frac{\omega_z(t_j)}{\omega_L};$$

$$\omega_{\Sigma}^{aktuell}(\pi(j)) := \omega_{\Sigma}^{aktuell}(\pi(j) - 1) + \omega^{aktuell}(\pi(j)).$$

Ergebnis: Die Menge $TE_{out}(\mathcal{K})$ beinhaltet alle aktivierten Transitionen, die vom Kollektiv \mathcal{K} zum Feuern freigegeben werden.

Für Nutzen

Gegeben sei ein Petri-Netz $N = (P, T, F, B, f, \mathbf{z}, \psi)$ im Zustand \mathbf{z} mit *allgemeiner kollektiver Bewertungsfunktion* $\psi : \mathcal{KM}\bullet \rightarrow \mathbb{R}_{\geq}$, die als Nutzen aufgefasst wird. Ein Kollektiv

$\mathcal{K} \in \mathcal{KM}$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(\mathcal{K})$ von aktivierten Transitionen aus $TA_{out}(\mathcal{K})$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **kollektiven Freigabemenge** $TE_{out}(\mathcal{K})$ **mittels Nutzen**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(\mathcal{K})$ mit $\kappa(\mathcal{K}) \in \mathbb{N}$ gegeben ist.

Der folgende Algorithmus arbeitet nach dem Branch- & Bound-Prinzip, wobei die Transitionen $t_1, \dots, t_{\kappa(\mathcal{K})} \in TA_{out}(\mathcal{K})$ nun o.B.d.A. in absteigender Reihenfolge ihrer **Nutzenquotienten** η geordnet sind, d.h.

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_{\kappa(\mathcal{K})} \text{ für alle } t_j \in TA_{out}(\mathcal{K}) \text{ mit } \eta_j = \frac{\psi(p, t_j)}{\sum_{p \in \bullet t_j} f(p, t_j)}.$$

Zudem beachte man, dass das Gewicht \mathbf{g} hier ein Vektor ist, in dem die „Gewichte“ gleich geordnet sind wie beim Zustand des Kollektives $\mathbf{z}(\mathcal{K})$.

Algorithmus 2.23: Kollektive Freigabe mittels Branch & Bound

Setze:

$X_0 := \{(x_1, x_2, \dots, x_{\kappa(\mathcal{K})}) \mid x_j \in \{0, 1\}, j = 1, \dots, \kappa(\mathcal{K})\}$ (Ursprungsknoten)

$\mathbf{g}_0 := \mathbf{0}$ mit $|\mathbf{g}_0| = |\mathcal{K}|$ (Gewicht der Teillösung [Vektor])

$n_0 := \sum_{j=1}^{\kappa(\mathcal{K})} \psi(p, t_j)$ (theoretischer Nutzen des Problems)

$N := 0$ (Effizienzschranke (Bound))

$TP := \{X_0\}$ (Menge aller Teilknoten)

$s := 0$ (Zähler der Teilknoten)

$m_0 := 0$ (Stufe des Teilknotens)

$L := \emptyset$ (Lösung)

1. Schritt: Falls $TP = \emptyset$, terminiere.

Wähle X_r aus TP mit $n_r = \max(n_i \mid X_i \in TP)$ (bei Mehrdeutigkeit wähle willkürlich)

2. Schritt: Falls $m_r < \kappa(\mathcal{K})$, führe aus:

Setze:

$$X_{s+1} := X_r \text{ mit } x_{m_r+1} = 1; X_{s+2} := X_r \text{ mit } x_{m_r+1} = 0;$$

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

$$n_{s+1} := n_r; n_{s+2} := n_r - \psi(p, t_{m_r+1});$$

$$g_{(s+1)_i} := \begin{cases} g_{r_i} + f(p_i, t_{m_r+1}) & \text{falls } t_{m_r+1} \in p_i \bullet \\ g_{r_i} & \text{falls } t_{m_r+1} \notin p_i \bullet \end{cases}$$

$$\mathbf{g}_{s+2} := \mathbf{g}_r;$$

$$m_{s+1} := m_{s+2} := m_r + 1;$$

$$TPg\ddot{u}l\text{t}ig := \{X_{s+i} \mid \mathbf{g}_{s+i} \leq \mathbf{z}(\mathcal{K}) \wedge n_{s+i} > N, i = 1, 2\}$$

$$TP := (TP \setminus \{X_r\}) \cup TPg\ddot{u}l\text{t}ig;$$

Falls $(m_r = \kappa(\mathcal{K}) - 1) \wedge (TPg\ddot{u}l\text{t}ig \neq \emptyset)$, führe aus:

Wähle X_i aus $TPg\ddot{u}l\text{t}ig$ mit $n_i = \max(n_i \mid X_i \in TPg\ddot{u}l\text{t}ig)$

Setze:

$$N := n_i$$

$$L := X_i$$

Für alle $X_i \in TP$ führe aus

$$\text{Falls } n_i \leq N, \text{ setze: } TP := TP \setminus \{X_i\};$$

Gehe zum Schritt 1.

Ergebnis: Das binäre Tupel $(x_1, x_2, \dots, x_{\kappa(\mathcal{K})}) \in L$ ist folgendermaßen zu interpretieren:

$$x_i = 1 \equiv \text{Transition } t_i \text{ wird freigegeben,}$$

$$x_i = 0 \equiv \text{Transition } t_i \text{ wird nicht freigegeben,}$$

so dass gilt:

$$TE_{out}(\mathcal{K}) := \{t_i \in TA_{out}(\mathcal{K}) \mid x_i = 1\}.$$

Die Menge $TE_{out}(\mathcal{K})$ beinhaltet nun alle aktivierten Transitionen, die vom Kollektiv \mathcal{K} zum Feuern freigegeben werden.

2.3.4 Probleme bei verallgemeinerter Kollektiventscheidung

Mit Kollektiven in dieser Form ist allerdings nur eine eingeschränkte Modellierung möglich, wie das Beispiel in Abb. 2.20 zeigt. Hiernach sollen die Plätze p_1, \dots, p_5 , die kein Kollektiv gemäß Def. 2.13 bilden, dennoch gemeinsam entscheiden, und zwar anhand der Transitionenbewertungen, während der Platz p_6 von dieser Kollektiventscheidung ausgeschlossen bleibt.

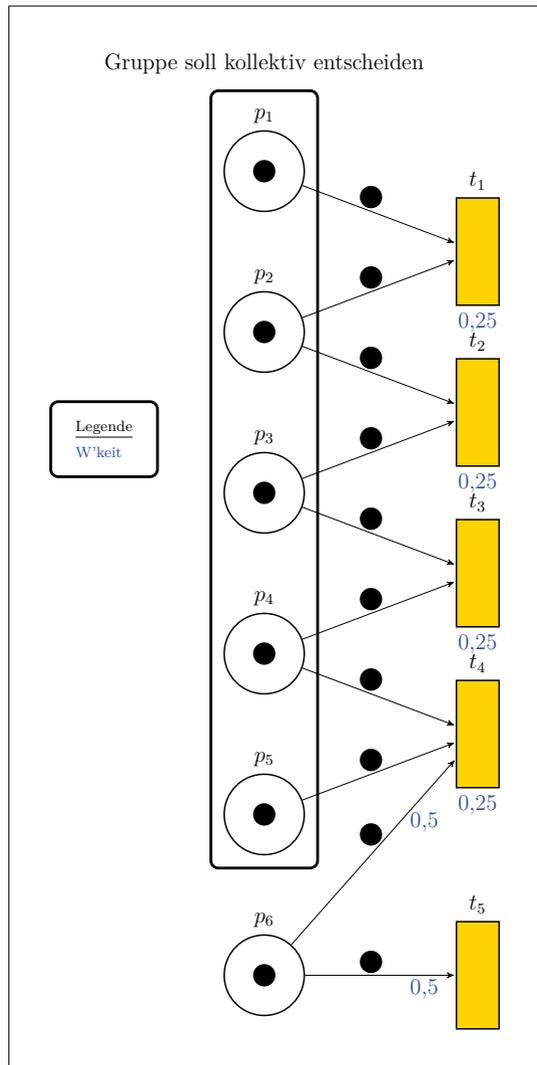


Abbildung 2.20: Modellierungsproblem

Um aber auch für derartige (Modell-)Konstrukte („kollektive Gruppen“) geeignete Freigabe-prozeduren für allgemeine Konfliktfälle angeben zu können, wenn z.B. in einem 6er-Team 5 Mitglieder kooperieren, während sich die sechste Person nicht mit dem Team abspricht, bietet sich die Technik der *Faltung* von Petri-Netzen in gefärbte Petri-Netze an. Bei dieser

2 Allgemeine Konfliktlösungen bei Feuerprozessen von Petri-Netzen

Technik wird jedem Platz innerhalb einer kollektiven Gruppe eine andere Farbe zugewiesen, z.B.

p_1	→	GRÜN
p_2	→	ROT
p_3	→	PINK
p_4	→	BLAU
p_5	→	VIOLETT

Nun wird jedes Token, das in einem Platz als Zustand oder an den Pfeilen als Gewichtung steht, in dieser Farbe gefärbt. Danach werden alle Plätze der kollektiven Gruppe zu einem *Superplatz* zusammengefasst; dementsprechend werden auch alle Pfeilgewichte aggregiert. Die Transitionenbewertungen werden nun auf die Pfeile des Superplatzes übertragen. Man erhält das gefärbte Petri-Netz in Abb. 2.21.

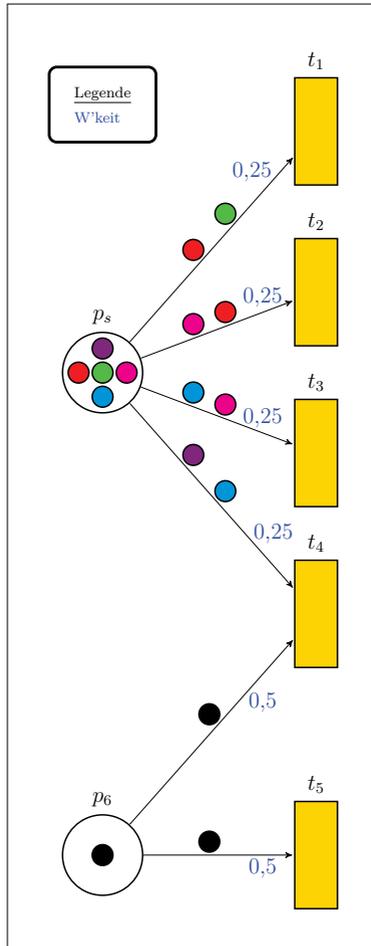


Abbildung 2.21: Modellierungslösung mit gefärbtem Petri-Netz

Für weiterführende Erläuterungen über gefärbte Petri-Netze sei auf Kapitel 4 verwiesen.

3 Feuerprozess-Szenarien

Zu Beginn von Kapitel 2 wurde bereits herausgestellt, dass im Fokus der Betrachtungen die Auflösung von allgemeinen Konfliktsituationen steht, damit es möglich ist, einen interaktionslosen, automatisierten und konfliktfreien Ablauf in Petri-Netz-Modellen zu garantieren, etwa zu Simulationszwecken. Hierbei spielt der konkrete Feuerprozess, der in jedem Zustand \mathbf{z} des Petri-Netzes ansteht und schließlich vollzogen wird, eine untergeordnete Rolle. Zwar ist der durchgeführte Feuerprozess für den Zustandsübergang ($\mathbf{z} \rightarrow \mathbf{z}'$) von entscheidender Bedeutung, die Analyse der Konfliktsituationen ist allerdings eher von statischer Natur, d.h. eine allein vom Zustand \mathbf{z} abhängige Bestimmung eines (methodisch) gelösten Petri-Netzes. Es lassen sich verschiedene Feuerprozess-Szenarien unterscheiden. Diese liefern weitere Modelleigenschaften, die der Modellierer für das Petri-Netz-Modell festlegt.

Dazu soll als erstes auf den einfachsten Feuerprozess eingegangen werden, nämlich auf den seriellen Feuerprozess.

Szenarium 1:

Der *serielle Feuerprozess* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge TA aller aktiven Transitionen (gemäß Def. 2.1)
2. Bestimmung einer Transition $t \in TA$
3. Feuern der Transition t

Es sei angemerkt, dass es bei einem seriellen Feuerprozess keine allgemeinen Konflikte geben kann, weil immer nur genau eine Transition $t \in TA$ gefeuert wird. Des Weiteren wurde im oberen Verfahren nicht beschrieben, wie in Schritt 2 die Transition $t \in TA$ bestimmt wird. Diese Festlegung wird bewusst ausgelassen, weil es hier die unterschiedlichsten Auswahlverfahren gibt, z.B.:

- Willkürliche Wahl einer Transition $t \in TA$ (z.B. mittels Zufallsgenerator)

- Wahl der zu feuern Transition durch außenstehenden Akteur/Benutzer (Möglichkeit der Interaktion)
- Wahl aufgrund einer Bewertung der Transition
- ...

Für die hier behandelten Konfliktsituationen sind allerdings nur nebenläufige Feuerprozesse relevant, die in unterschiedlichsten Varianten auftreten (und modelliert werden) können. Als „Standardvariante“ wird bezeichnet, dass möglichst alle aktiven bzw. freigegebenen Transitionen gefeuert werden.¹

Szenarium 2:

Der *nebenläufige Feuerprozess (Standardvariante)* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge TA aller aktiven Transitionen (gemäß Def. 2.1)
2. Ermittlung aller lokalen Freigabemengen $TE_{out}(p)$ (via Algorithmus aus Kap. 2)
3. Ermittlung der globalen Freigabemenge TE (gemäß Def. 2.12)
4. Nebenläufiges Feuern der Menge TE

Der serielle und der nebenläufige Feuerprozess, letzterer in der Standardvariante, stellen somit zwei entgegengesetzte Spezialfälle dar; beim ersten Szenarium wird immer nur *genau eine* Transition $t \in TA$ gefeuert, wohingegen beim zweiten Szenarium versucht wird, *alle* Transitionen $t \in TA$ zu feuern. Daher liegt der Unterschied in der *Menge der zu feuern Transitionen*. Aus dieser Betrachtung könnte man nun einen verallgemeinerten Feuerprozess definieren, der zusätzliche Auswahlsschritte enthält. In einem Schritt könnte aus der Menge TA wahlweise eine Menge TW bestimmt werden, welche dann versucht wird zu feuern. Diese Auswahlmenge greift vor dem Konfliktlösungsprozess in die Bestimmung der Freigabemenge TE ein. Des Weiteren wäre auch eine Wahlmöglichkeit darin gegeben, nicht alle Transitionen der globalen Freigabemenge TE zu feuern, sondern nur eine Teilmenge TF der schließlich und endlich gefeuerten Transitionen. Damit lassen sich nun beliebige spezielle Feuerprozesse darstellen (und modellieren); daher wird er hier auch als „*allgemeiner Feuerprozess*“ bezeichnet:

¹In der Literatur auch unter dem Begriff *Maximalschaltregel* zu finden (u.a. [Ger04]).

Szenarium 3:

Der *allgemeine Feuerprozess* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge TA aller aktiven Transitionen (gemäß Def. 2.1)
2. Bestimmung einer Menge TW von Transitionen aus TA
3. Ermittlung aller lokalen Freigabemengen $TE_{out}(p)$ aus TW (via Algorithmus aus Kap. 2)
4. Ermittlung der globalen Freigabemenge TE (gemäß Def. 2.12)
5. Bestimmung einer Menge TF von Transitionen aus TE
6. Nebenläufiges Feuern der Menge TF

Wird nun die spezielle Anforderung gestellt, dass die Mächtigkeit von TW gleich 1 ist, erhält man daraus den seriellen Feuerprozess; wird im Gegensatz dazu die Anforderung $TW = TA$ und $TF = TE$ gestellt, erhält man den nebenläufigen Feuerprozess in seiner Standardvariante. Dabei könnten nun spezifizierte Wahlmethoden für TW verwendet werden, wie z.B.:

- Willkürliche Wahl von Transitionen $t \in TA$ (z.B. mittels Zufallsgenerator) solange, bis TW eine bestimmte Mächtigkeit erreicht
- Wahl der Menge TW durch außenstehenden Akteur/Benutzer (Interaktion)
- Bestimmung von Transitionen $t \in TA$ aufgrund einer Bewertung der Transitionen solange, bis TW eine bestimmte Mächtigkeit erreicht
- ...

Analog lassen sich diese spezifizierten Wahlmethoden auch zur Bestimmung der Menge TF aus TE anwenden.

Beispiel:

Im Petri-Netz mit Konfliktlösung in Abb. 3.1 soll aus dem Zustand \mathbf{z} unter Anwendung des allgemeinen Feuerprozesses in einen Zustand \mathbf{z}' übergegangen werden. Die Plätze p_1, p_3 und p_4 können keinen allgemeinen Konflikt haben, weil sich jeweils nur eine Transition im Nachbereich der Plätze befindet, weswegen diese Plätze keiner Konfliktlösung unterzogen werden müssen (vgl. Kap. 2). Allerdings können die Plätze p_2 und p_5 einen allgemeinen Konflikt haben und daher eine Konfliktlösung benötigen. Beide Plätze lösen ihre Konflikte mit Wahrscheinlichkeiten, wobei die Wahrscheinlichkeiten für alle Transitionen gleich sind. Bei den Auswahlprozessen für TW oder TF wird im Beispiel von einem Benutzer ausgegangen, der interaktiv die Transitionen wählt.

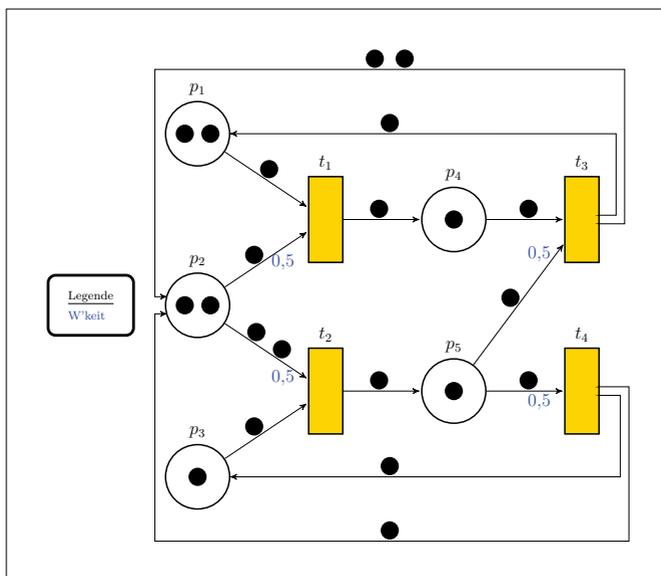


Abbildung 3.1: Petri-Netz-Beispiel für einen allgemeinen Feuerprozess im Zustand \mathbf{z}

1. Schritt:

Es ist leicht zu erkennen, dass alle Transitionen im Petri-Netz aktiviert sind und daher gilt:

$$TA = \{t_1, t_2, t_3, t_4\}$$

2. Schritt:

Der Benutzer wählt die Transitionen t_1 , t_2 und t_3 aus; daher ist:

$$TW = \{t_1, t_2, t_3\}$$

3. Schritt:

Bei der lokalen Konfliktlösung seien die folgenden lokalen Freigabemengen entstanden:

$$TE_{out}(p_1) = \{t_1\}$$

$$TE_{out}(p_2) = \{t_1\}$$

$$TE_{out}(p_3) = \{t_2\}$$

$$TE_{out}(p_4) = \{t_3\}$$

$$TE_{out}(p_5) = \{t_3\}$$

3 Feuerprozess-Szenarien

4. Schritt:

Daraus ergibt sich als globale Freigabemenge:

$$TE = \{t_1, t_3\}$$

5. Schritt:

Hiervon entscheidet der Benutzer, dass er willkürlich nur die Transition t_3 feuern möchte:

$$TF = \{t_3\}$$

6. Schritt:

Es wird somit die Menge TF gefeuert, woraus sich der Zustand \mathbf{z}' in Abb. 3.2 ergibt.

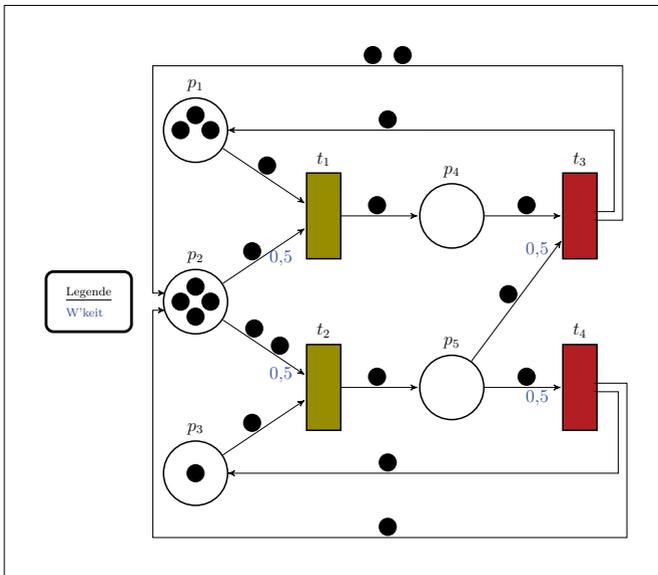


Abbildung 3.2: Petri-Netz-Beispiel für einen allgemeinen Feuerprozess im Folgezustand \mathbf{z}'

4 Einführung in gefärbte Petri-Netze

Zunächst wird eine schematische Herleitung der gefärbten Petri-Netze (CPN¹) aus den Petri-Netzen (PN) an einem einfachen Beispiel demonstriert.²

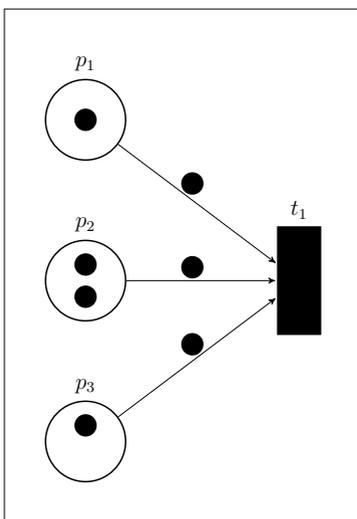


Abbildung 4.1: „Normales“ Petri-Netz

Betrachtet man das Petri-Netz aus Abb. 4.1, so sind alle Token im Netz „gleich“ und an sich nicht unterscheidbar; eine Unterscheidung ist nur möglich anhand der Plätze bzw. Pfeile, in bzw. an denen sie sich befinden. In der Anwendung könnten mögliche Interpretationen beispielsweise sein: p_1 enthält Schrauben, p_2 Holzbretter und p_3 Muttern, was durch eine extra Beschriftung klar gemacht werden müsste. Daher stellt sich die Frage: Wenn die Token unterschiedliche Objekte darstellen, warum unterscheiden sich die einzelnen Token nicht

¹Die Abkürzung CPN stammt vom engl. Begriff coloured Petri net.

²Für eine ausführlichere Herleitung sei hier auf [Ger04] verwiesen.

von andersartigen Token? Man stelle sich vor, dass nicht nur drei Komponenten in einem Kleinteilelager vorhanden sind, sondern dass man 500 oder mehr verschiedenartige Komponenten für unterschiedliche Produktionen zu bevorraten hat, wozu bei einer Modellierung von entsprechenden Produktionszusammenhängen ein relativ großes und unübersichtliches PN entstehen würde. Genau diese Probleme beheben CPN. Um aus einem PN ein CPN zu erhalten, werden mehrere gleiche Teile des Netzes zu einer vereinfachten Struktur durch „Faltung“ zusammengefasst, z.B. die drei Plätze aus Abb. 4.1. Damit aber die Unterscheidbarkeit erhalten bleibt, werden die Token „gefärbt“. Dafür wird jedem Platz eine andere Farbe zugewiesen, z.B.

$$\begin{aligned} p_1 &\rightarrow \text{ROT} \\ p_2 &\rightarrow \text{BLAU} \\ p_3 &\rightarrow \text{GRÜN} \end{aligned}$$

Nun wird jedes Token, das in einem Platz als Zustand oder an den Pfeilen als Gewichtung steht, in dieser Farbe gefärbt. Danach werden alle Plätze zu einem „gefärbten“ Platz zusammengefasst; dementsprechend werden auch alle Pfeilgewichte aggregiert (vgl. Abb. 4.2).

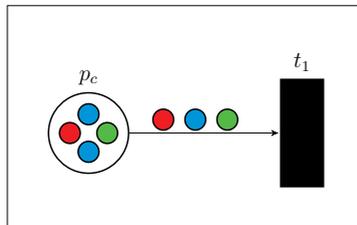


Abbildung 4.2: Gefärbtes Petri-Netz nach Faltung

In dieser Art kann man sich zunächst einmal den originären Zusammenhang zwischen PN und CPN bildlich vorstellen. Formal lassen sich CPN in unterschiedlichen Komplexitätsstufen definieren.

Definition 4.1:

Ein *einfach attribuiertes gefärbtes Petri-Netz (SCP N^t)* ist ein Tupel $N = (P, T, F, B, \mathbf{f}, C)$, wobei

- das 4-Tupel (P, T, F, B) ein Netz gemäß Def. 1.1 bildet;
- die Menge $C = \{c_1, \dots, c_k\}$ eine endliche, nichtleere Menge von sog. **Farben** c_1, \dots, c_k ist und auch **Farbenmenge (colour set)** genannt wird; dabei sei auf

kanonische Weise eine Anordnung der Farben in Form eines k -Tupels (c_1, \dots, c_k) festgelegt;

- die Abbildung $\mathbf{f} : F \cup B \rightarrow \mathbb{N}_0^k \setminus \{\mathbf{0}\}$ (**Gewichtungsfunktion**) jedem Pfeil $(p, t) \in F$ bzw. $(t, p) \in B$ einen Zeilenvektor aus nicht-negativen ganzen Zahlen als **Gewicht** zuweist (allerdings ungleich dem Nullvektor).

Die Farbenmenge wird auch in verallgemeinerter Weise (vgl. Def. 4.4) als *Attribut* bezeichnet und die einzelnen Farben als *Ausprägungen* dieses Attributes.

Beispiel:

In diesem Beispiel wird ein einfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, \mathbf{f}, C)$ konstruiert:

1. Das Netz $N = (P, T, F, B)$ mit:

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2\}$$

$$F = \{(p_1, t_1), (p_2, t_1), (p_3, t_2)\}$$

$$B = \{(t_1, p_3), (t_2, p_4), (t_2, p_5)\}$$

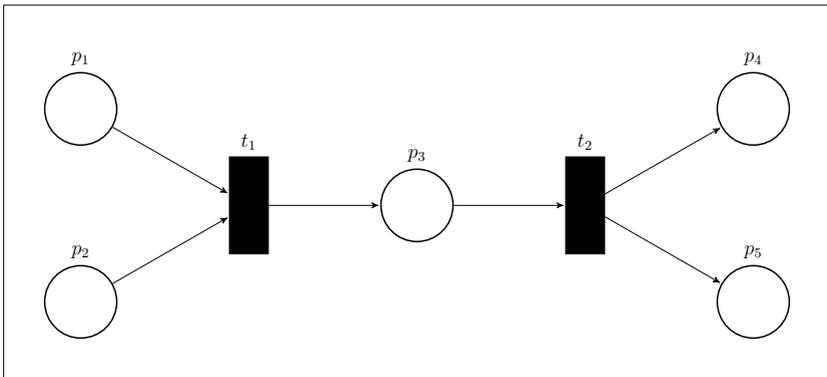


Abbildung 4.3: Beispiel eines Netzes

2. Es gibt die Farbenmenge $C = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$; dabei sei auf kanonische Weise eine Anordnung der Farben in Form eines 3-Tupels $(\text{ROT}, \text{BLAU}, \text{GRÜN})$ festgelegt.

¹Die Abkürzung SCPN stammt vom engl. Begriff single-attributed coloured Petri net.

3. Die Gewichtung der Pfeile sei wie folgt festgelegt:

$$f(p_1, t_1) = (1, 0, 1) \text{ für 1 rotes und 1 grünes Token}$$

$$f(p_2, t_1) = (0, 2, 0) \text{ für 2 blaue Token}$$

$$f(t_1, p_3) = (1, 2, 1) \text{ für 1 rotes, 2 blaue und 1 grünes Token}$$

$$f(p_3, t_2) = (2, 2, 2) \text{ für 2 rote, 2 blaue und 2 grüne Token}$$

$$f(t_2, p_4) = (0, 1, 2) \text{ für 1 blaues und 2 grüne Token}$$

$$f(t_2, p_5) = (2, 1, 0) \text{ für 2 rote und 1 blaues Token}$$

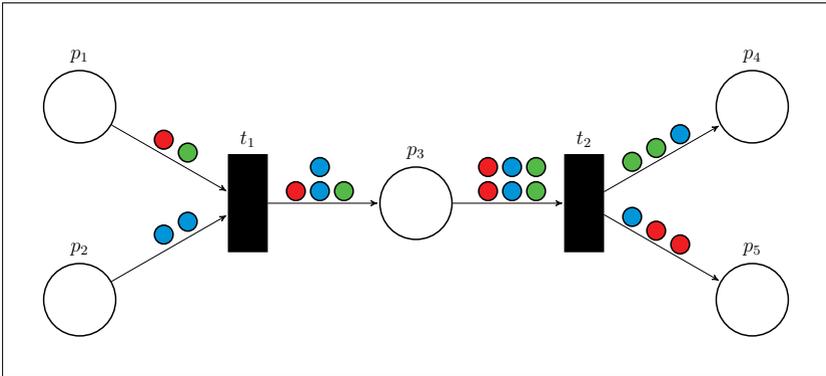


Abbildung 4.4: Beispiel eines einfach attribuiertes gefärbtes Petri-Netzes

Definition 4.2:

Es sei ein einfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, \mathbf{f}, C)$ gegeben. Eine Abbildung $\mathbf{z} : P \rightarrow \mathbb{N}_0^k$ heißt **Zustand** oder **Markierung** von N , wobei \mathbf{z} eindeutig als Vektor $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_m)^T$ mit $\mathbf{z}_i = \mathbf{z}(p_i)$ darstellbar ist, welcher als eine Matrix interpretiert werden kann.

Beispiel:

Ein Beispiel einer Markierung des SCPN von oben ist gemäß der obigen Vereinbarung:

$$\mathbf{z}(p_1) = \mathbf{z}_1 = (1, 1, 1) \text{ für 1 rotes, 1 blaues und 1 grünes Token}$$

$$\mathbf{z}(p_2) = \mathbf{z}_2 = (1, 2, 0) \text{ für 1 rotes und 2 blaue Token}$$

$$\mathbf{z}(p_3) = \mathbf{z}_3 = (1, 0, 1) \text{ für 1 rotes und 1 grünes Token}$$

$$\mathbf{z}(p_4) = \mathbf{z}_4 = (0, 0, 0) \text{ für kein Token}$$

$$\mathbf{z}(p_5) = \mathbf{z}_5 = (0, 0, 0) \text{ für kein Token}$$

oder als Vektor/Matrix:

$$\mathbf{z} = \begin{pmatrix} (1, 1, 1) \\ (1, 2, 0) \\ (1, 0, 1) \\ (0, 0, 0) \\ (0, 0, 0) \end{pmatrix} \equiv \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

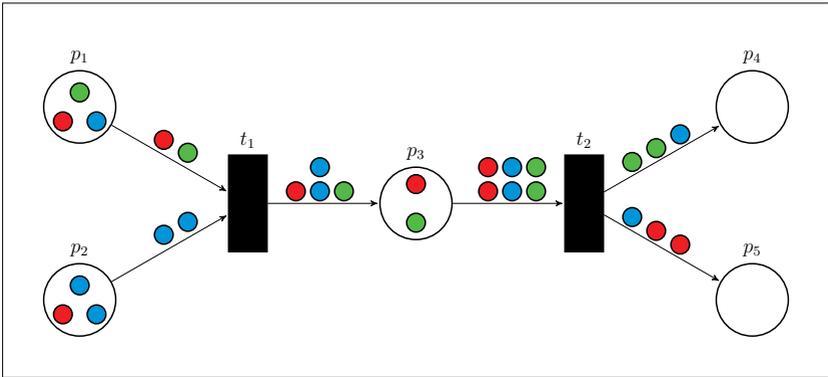


Abbildung 4.5: Beispiel einer Markierung eines SCPN

Definition 4.3:

Es seien ein einfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, \mathbf{f}, C)$ und eine Markierung \mathbf{z} von N gegeben.

- a) Eine Transition $t \in T$ von N heißt **aktiviert** oder **seriell feuerbar** im Zustand \mathbf{z} , wenn für alle $p \in \bullet t$ gilt: $\mathbf{f}(p, t) \leq \mathbf{z}(p)$.¹
- b) Eine im Zustand \mathbf{z} aktivierte Transition t wird kurz auch **\mathbf{z} -feuerbar** genannt.
- c) Das **Feuern** einer \mathbf{z} -feuerbaren Transition $t \in T$ von N ist der Übergang vom Zustand $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_m)^T$ in den Zustand $\mathbf{z}' = (\mathbf{z}'_1, \dots, \mathbf{z}'_m)^T$, wobei gilt:

$$\mathbf{z}'_i = \begin{cases} \mathbf{z}_i - \mathbf{f}(p_i, t) + \mathbf{f}(t, p_i) & \text{falls } p_i \in \bullet t \wedge p_i \in t \bullet \\ \mathbf{z}_i - \mathbf{f}(p_i, t) & \text{falls } p_i \in \bullet t \wedge p_i \notin t \bullet \\ \mathbf{z}_i + \mathbf{f}(t, p_i) & \text{falls } p_i \notin \bullet t \wedge p_i \in t \bullet \\ \mathbf{z}_i & \text{falls } p_i \notin \bullet t \wedge p_i \notin t \bullet \end{cases} \text{ für } i = 1, \dots, m.$$

Beispiel:

Überprüft man die Transitionen des obigen SCPN-Beispiels, dann gilt:

- t_1 ist aktiviert bzw. seriell feuierbar:

$$\mathbf{f}(p_1, t_1)^T = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \mathbf{z}(p_1)^T$$

$$\mathbf{f}(p_2, t_1)^T = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} = \mathbf{z}(p_2)^T$$

- t_2 ist nicht aktiviert:

$$\mathbf{f}(p_3, t_2)^T = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \not\leq \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \mathbf{z}(p_3)^T$$

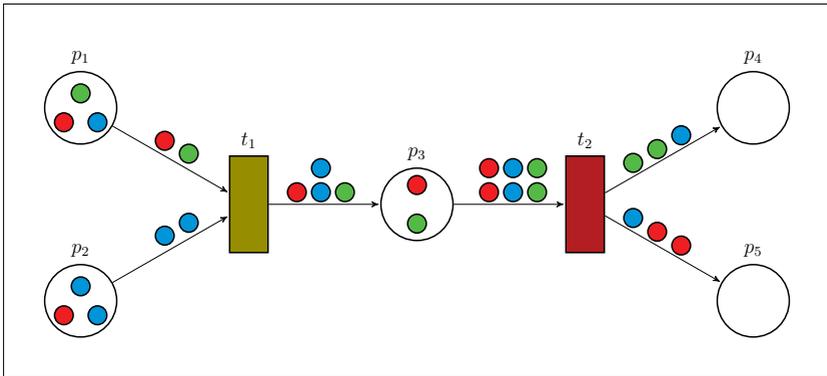


Abbildung 4.6: Feuerbarkeit des SCPN-Beispiels

Wird nun t_1 gefeuert, wird aus Zustand

$$\mathbf{z} = \begin{pmatrix} (1, 1, 1) \\ (1, 2, 0) \\ (1, 0, 1) \\ (0, 0, 0) \\ (0, 0, 0) \end{pmatrix} \equiv \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

in den Zustand

¹Mit dem Symbol \leq zwischen Vektoren wird der komponentenweise Vergleich verstanden.

$$\mathbf{z}' = \begin{pmatrix} (1, 1, 1) - (1, 0, 1) \\ (1, 2, 0) - (0, 2, 0) \\ (1, 0, 1) + (1, 2, 1) \\ (0, 0, 0) \\ (0, 0, 0) \end{pmatrix} = \begin{pmatrix} (0, 1, 0) \\ (1, 0, 0) \\ (2, 2, 2) \\ (0, 0, 0) \\ (0, 0, 0) \end{pmatrix} \equiv \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 2 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

übergegangen.

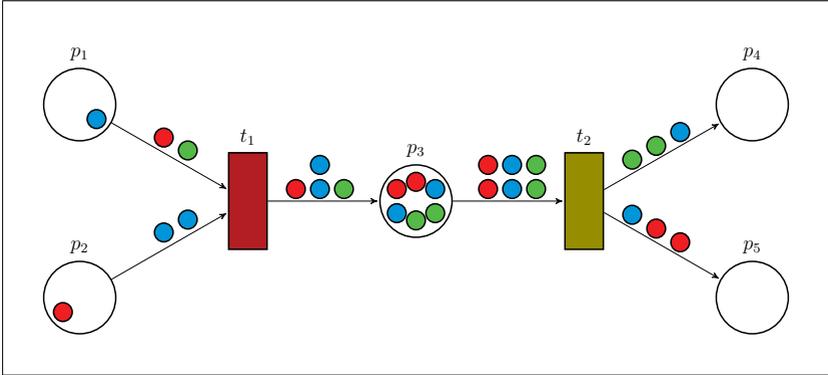


Abbildung 4.7: Zustandsübergang am SCPN-Beispiel

Im Folgenden werden die Token nicht nur mit einem einzigen Attribut (nämlich mit dem Attribut „Farbe“) versehen, sondern es werden mehrere verschiedene Attribute definiert und den einzelnen Token auf den Plätzen bzw. Pfeilen zugewiesen (vgl. [KD13]). Hierbei wird der Begriff der *Multimenge* verwendet (Grundlagen hierzu siehe Anhang).

Definition 4.4:

Ein *mehrfach attribuiertes gefärbtes Petri-Netz (MCPN¹)* ist ein Tupel $N = (P, T, F, B, f, A, K, k)$, wobei

- das Tupel (P, T, F, B) ein Netz gemäß Def. 1.1 bildet;
- $A = \{A_1, \dots, A_y\}$ eine endliche, nichtleere Menge von sog. **Attributen** ist, wobei die Attribute A_i ihrerseits endliche, nichtleere Mengen sind, die auf kanonische Weise geordnet sind und als Tupel dargestellt werden können: $A_i \equiv (a_1^i, \dots, a_{|A_i|}^i)$, $i = 1, \dots, y$, wobei die Elemente der Attribute auch **Ausprägungen** des jeweiligen Attributes genannt werden;

¹Die Abkürzung MCPN stammt vom engl. Begriff multi-attributed coloured Petri net.

- $K = \{K_1, \dots, K_x\}$ eine endliche, nichtleere Menge von sog. **Klassen** K_j ist, wobei K_j ein kartesisches Produkt aus Attributen einer Teilmenge von A ist, d.h. $K_j = A_{j_1} \times \dots \times A_{j_s}$ mit $\{j_1, \dots, j_s\} \subseteq \{1, \dots, y\}$ für $j = 1, \dots, x$; dabei heißt $\mathbb{K} := \bigcup_{K_j \in K} K_j$ die **Vereinigungsmenge aller Klassen**;
- $k : P \rightarrow K$ eine Abbildung ist (**Klassifizierungsfunktion**), die jedem Platz $p \in P$ eine Klasse aus der Klassenmenge K zuweist;
- $f : (F \cup B) \rightarrow \mathbb{M}(\mathbb{K})$ eine Abbildung ist (**Gewichtungsfunktion**), die jedem Pfeil $(p, t) \in F$ bzw. $(t, p) \in B$ eine der jeweiligen Klasse von p entsprechende Multimenge als **Gewicht** zuweist, d.h. $f(p, t)$ bzw. $f(t, p)$ ist eine Multimenge M in der Form $M = \{(a, h(a)) \mid a \in \mathbb{K}, h(a) \in \mathbb{N}_0\}$, wobei für die Stützmenge von M gilt:

$$\text{supp}(M) \subseteq k(p) \subseteq \mathbb{K}.^1$$

Bemerkung 4.5:

Aufgrund von großen Redundanzen ist es in einer grafischen Darstellung des MCPN empfehlenswert, wenn die Gewichte der einzelnen Pfeile nur aus der Stützmenge gebildet werden.

Beispiel:

In diesem Beispiel wird ein mehrfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, f, A, K, k)$ konstruiert:

1. Das Netz $N = (P, T, F, B)$ mit:

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2\}$$

$$F = \{(p_1, t_1), (p_2, t_1), (p_3, t_2)\}$$

$$B = \{(t_1, p_3), (t_2, p_4), (t_2, p_5)\}$$

¹Man beachte, dass $\text{supp}(M) \neq \emptyset$ sein muss.

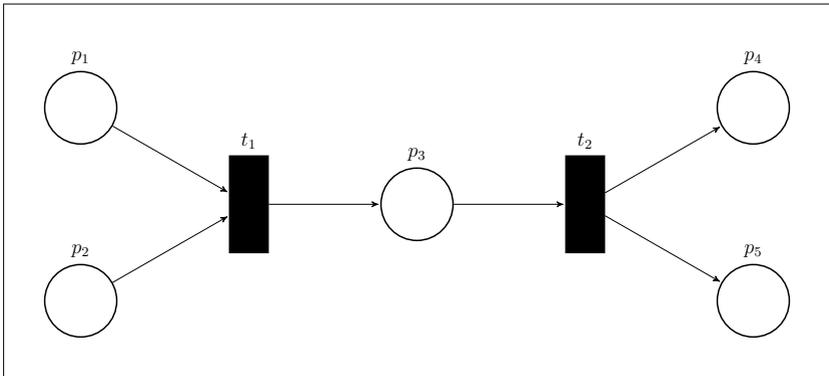


Abbildung 4.8: Beispiel eines Netzes

2. Die Klassen und Attribute

- Es gibt zwei Attribute in $A = \{\text{COL}, \text{NUM}\}$ mit:

$$\text{COL} = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$$

$$\text{NUM} = \{1, 2, 3\}$$

- Hieraus lassen sich auf kanonische Weise die folgenden drei Klassen erzeugen:
 $K = \{\text{FARBE}, \text{ZAHL}, \text{BUNTEZAHL}\}$ mit:

$$\text{FARBE} = \{((\text{ROT}), (\text{BLAU}), (\text{GRÜN}))\} \equiv \text{COL} = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$$

$$\text{ZAHL} = \{(1), (2), (3)\} \equiv \text{NUM} = \{1, 2, 3\}$$

$$\text{BUNTEZAHL} = \text{COL} \times \text{NUM} = \left\{ \begin{array}{l} (\text{ROT}, 1), (\text{ROT}, 2), (\text{ROT}, 3), \\ (\text{BLAU}, 1), (\text{BLAU}, 2), (\text{BLAU}, 3), \\ (\text{GRÜN}, 1), (\text{GRÜN}, 2), (\text{GRÜN}, 3) \end{array} \right\}$$

3. Die Klassifizierungsfunktion der Plätze:

$$k(p_1) = \text{FARBE}$$

$$k(p_2) = \text{ZAHL}$$

$$k(p_3) = \text{BUNTEZAHL}$$

$$k(p_4) = \text{ZAHL}$$

$$k(p_5) = \text{FARBE}$$

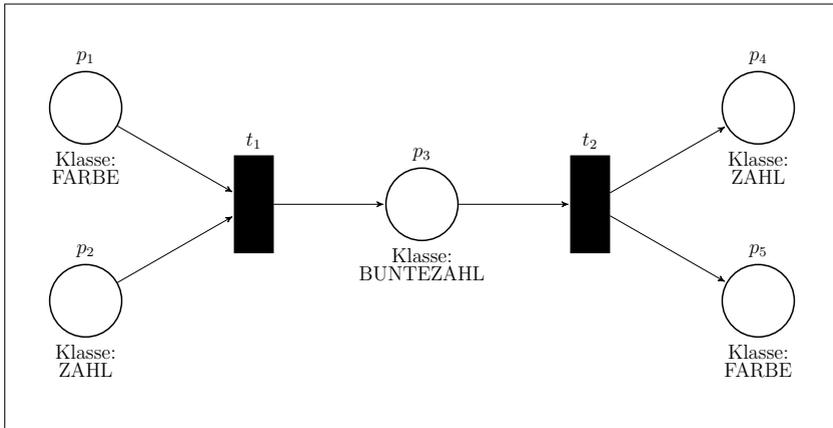


Abbildung 4.9: Beispiel eines Netzes mit Klassen

4. Die Gewichtung der Pfeile

$$f(p_1, t_1) \equiv \{(\text{ROT}, 1), (\text{GRÜN}, 1)\}$$

$$f(p_2, t_1) \equiv \{(1, 2)\}$$

$$f(t_1, p_3) \equiv \{((\text{ROT}, 1), 1), ((\text{GRÜN}, 1), 1)\}$$

$$f(p_3, t_2) \equiv \{((\text{ROT}, 1), 1), ((\text{BLAU}, 3), 1), ((\text{GRÜN}, 1), 1)\}$$

$$f(t_2, p_4) \equiv \{(1, 2), (3, 1)\}$$

$$f(t_2, p_5) \equiv \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\}$$

Eigentlich hätte in diesem Beispiel jede Gewichtung die Form

$$f(\dots) = \left\{ \begin{array}{l} (1, h(1)), (2, h(2)), (3, h(3)), \\ (\text{ROT}, h(\text{ROT})), (\text{BLAU}, h(\text{BLAU})), (\text{GRÜN}, h(\text{GRÜN})), \\ ((\text{ROT}, 1), h(\text{ROT}, 1)), ((\text{ROT}, 2), h(\text{ROT}, 2)), ((\text{ROT}, 3), h(\text{ROT}, 3)), \\ ((\text{BLAU}, 1), h(\text{BLAU}, 1)), ((\text{BLAU}, 2), h(\text{BLAU}, 2)), ((\text{BLAU}, 3), h(\text{BLAU}, 3)), \\ ((\text{GRÜN}, 1), h(\text{GRÜN}, 1)), ((\text{GRÜN}, 2), h(\text{GRÜN}, 2)), ((\text{GRÜN}, 3), h(\text{GRÜN}, 3)) \end{array} \right\}$$

haben müssen, aber wie man sieht, ist diese Form sehr umständlich und unübersichtlich, daher wird hier die reduzierte Entsprechungsform gemäß Bemerkung 4.5 verwendet.

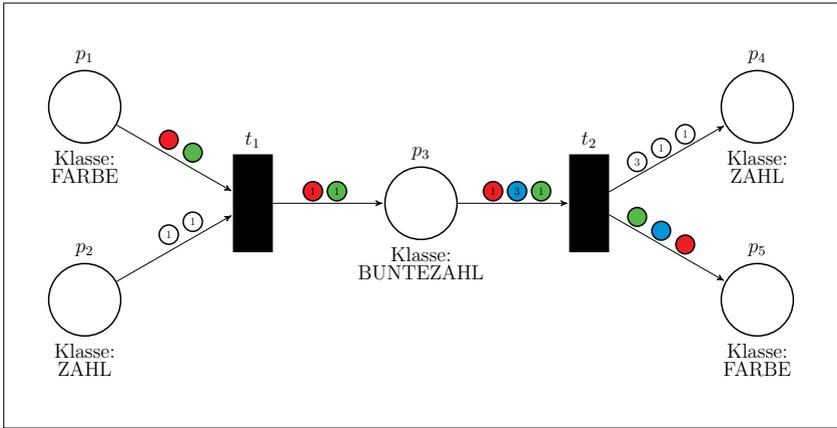


Abbildung 4.10: Beispiel eines gewichteten MCPN

Demnach ist ein SCPN ein MCPN mit genau einem Attribut ($A_1 = C$), dessen Ausprägungen „Farben“ genannt werden, und die Multimenge könnte man als einen Vektor aus nicht-negativen ganzen Zahlen interpretieren.

Definition 4.6:

Es sei ein mehrfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, f, A, K, k)$ gegeben. Eine Abbildung $\mathbf{z} : P \rightarrow \mathbb{M}(\mathbb{K})$, die jedem Platz $p \in P$ eine der Klasse $k(p)$ entsprechende Multimenge als Gewicht zuweist, heißt **Zustand** oder **Markierung** von N , wobei \mathbf{z} eindeutig als Vektor $\mathbf{z} = (z_1, \dots, z_m)$ mit $z_i = \mathbf{z}(p_i)$ darstellbar ist, d.h. $\mathbf{z}(p_i)$ ist eine Multimenge M_i in der Form $M_i = \{(a, h_i(a)) \mid a \in \mathbb{K}, h_i(a) \in \mathbb{N}_0\}$, wobei für die Stützmeng gilt: $\text{supp}(M_i) \subseteq k(p_i)$.

Bemerkung 4.7:

Aufgrund von großen Redundanzen ist es in einer grafischen Darstellung des mehrfach attribuierten gefärbten Petri-Netz empfehlenswert, wenn die Markierungen der einzelnen Plätze nur aus der Stützmeng gebildet werden (vgl. Bemerkung 4.5).

Beispiel:

Ein Beispiel einer Markierung des obigen MCPN ist:

$$\begin{aligned} \mathbf{z}(p_1) = z_1 &\equiv \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \mathbf{z}(p_5) = z_5 &\equiv \{(1, 4), (2, 1), (3, 1)\} \\ \mathbf{z}(p_2) = z_2 &\equiv \{((\text{BLAU}, 3), 1)\} \\ \mathbf{z}(p_4) = z_4 &\equiv \{(1, 1), (2, 1)\} \\ \mathbf{z}(p_3) = z_3 &\equiv \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{aligned}$$

4 Einführung in gefärbte Petri-Netze

oder als Vektor:

$$\mathbf{z} \equiv \begin{pmatrix} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \{(1, 4), (2, 1), (3, 1)\} \\ \{(\text{BLAU}, 3), 1\} \\ \{(1, 1), (2, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{pmatrix}$$

Eigentlich hätte in diesem Beispiel jede Markierung in der Form

$$\mathbf{z}(p) = \left\{ \begin{array}{l} (1, h(1)), (2, h(2)), (3, h(3)), \\ (\text{ROT}, h(\text{ROT})), (\text{BLAU}, h(\text{BLAU})), (\text{GRÜN}, h(\text{GRÜN})), \\ ((\text{ROT}, 1), h(\text{ROT}, 1)), ((\text{ROT}, 2), h(\text{ROT}, 2)), ((\text{ROT}, 3), h(\text{ROT}, 3)), \\ ((\text{BLAU}, 1), h(\text{BLAU}, 1)), ((\text{BLAU}, 2), h(\text{BLAU}, 2)), ((\text{BLAU}, 3), h(\text{BLAU}, 3)), \\ ((\text{GRÜN}, 1), h(\text{GRÜN}, 1)), ((\text{GRÜN}, 2), h(\text{GRÜN}, 2)), ((\text{GRÜN}, 3), h(\text{GRÜN}, 3)) \end{array} \right\}$$

dargestellt werden müssen, aber wie man sieht, ist diese Form sehr umständlich und unübersichtlich, daher wird hier die reduzierte Entsprechungsform gemäß Bemerkung 4.7 verwendet.

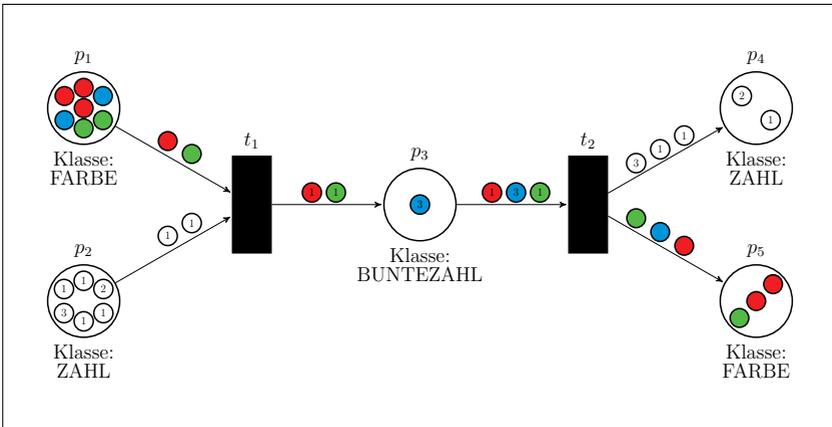


Abbildung 4.11: Beispiel einer Markierung eines MCPN

Die in Kapitel 2 vorgestellten Prozesse der Aktivierung und Feuerung von Transitionen in PN sollen auf MCPN übertragen werden. Hierbei liegt die Hauptschwierigkeit darin, die allgemeinen Operatoren $+$, $-$, \leq in entsprechende Multimengenoperatoren umzuwandeln.

Dazu wird folgendes Schema verwendet:

$$\begin{aligned} + &\rightarrow \uplus \\ - &\rightarrow \setminus \\ \leq &\rightarrow \subseteq \end{aligned}$$

Definition 4.8:

Es sei ein mehrfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, f, A, K, k)$ und eine Markierung \mathbf{z} von N gegeben.

- a) Eine Transition $t \in T$ von N heißt **aktiviert** oder **seriell feuerbar** im Zustand \mathbf{z} , wenn für alle $p \in \bullet t$ gilt: $f(p, t) \subseteq \mathbf{z}(p)$.
- b) Eine im Zustand \mathbf{z} aktivierte Transition t wird kurz auch **\mathbf{z} -feuerbar** genannt.
- c) Das **Feuern** einer \mathbf{z} -feuerbaren Transition $t \in T$ von N ist der Übergang vom Zustand $\mathbf{z} = (z_1, \dots, z_m)$ in den Zustand $\mathbf{z}' = (z'_1, \dots, z'_m)$, wobei gilt:

$$z'_i = \left\{ \begin{array}{ll} (z_i \setminus f(p_i, t)) \uplus f(t, p_i) & \text{falls } p_i \in \bullet t \wedge p_i \in t \bullet \\ z_i \setminus f(p_i, t) & \text{falls } p_i \in \bullet t \wedge p_i \notin t \bullet \\ z_i \uplus f(t, p_i) & \text{falls } p_i \notin \bullet t \wedge p_i \in t \bullet \\ z_i & \text{falls } p_i \notin \bullet t \wedge p_i \notin t \bullet \end{array} \right\} \text{ für } i = 1, \dots, m.$$

Beispiel:

Überprüft man die Transitionen des obigen MCPN-Beispiels, dann ist:

- t_1 aktiviert bzw. seriell feuerbar:

$$f(p_1, t_1) \subseteq \mathbf{z}(p_1)$$

$$f(p_2, t_1) \subseteq \mathbf{z}(p_2)$$

- t_2 nicht aktiviert, weil $f(p_3, t_2)$ keine Teilmenge von $\mathbf{z}(p_3)$ ist.

4 Einführung in gefärbte Petri-Netze

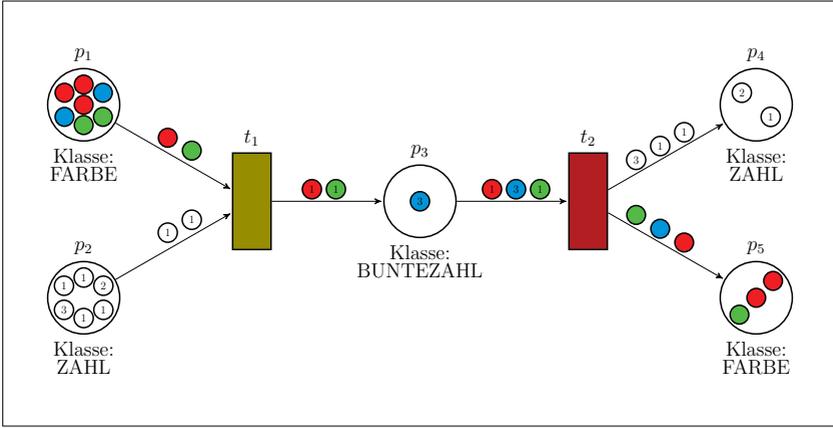


Abbildung 4.12: Feuerbarkeit des MCPN-Beispiels

Wird nun t_1 gefeuert, wird aus Zustand

$$\mathbf{z} \equiv \left(\begin{array}{l} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \{(1, 4), (2, 1), (3, 1)\} \\ \{((\text{BLAU}, 3), 1)\} \\ \{(1, 1), (2, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{array} \right)$$

in den Zustand

$$\mathbf{z}' \equiv \left(\begin{array}{l} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \setminus \{(\text{ROT}, 1), (\text{GRÜN}, 1)\} \\ \{(1, 4), (2, 1), (3, 1)\} \setminus \{(1, 2)\} \\ \{((\text{BLAU}, 3), 1)\} \uplus \{((\text{ROT}, 1), 1), ((\text{GRÜN}, 1), 1)\} \\ \{(1, 1), (2, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{array} \right)$$

$$= \left(\begin{array}{l} \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \\ \{(1, 2), (2, 1), (3, 1)\} \\ \{((\text{ROT}, 1), 1), ((\text{BLAU}, 3), 1), ((\text{GRÜN}, 1), 1)\} \\ \{(1, 1), (2, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{array} \right)$$

übergangen.

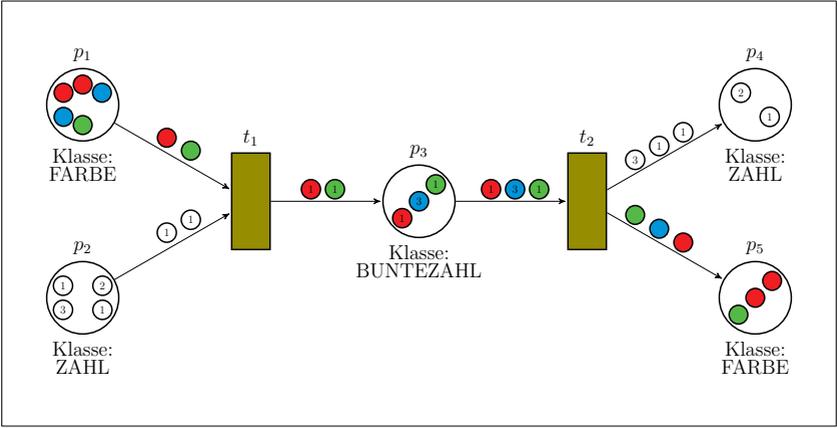


Abbildung 4.13: Zustandsübergang am MCPN-Beispiel

In diesem Zustand wären sowohl t_1 als auch t_2 aktiviert.

Neben dem seriellen Feuern, bei dem selbst bei seriellen Rechnungen immer nur genau eine aktivierte Transition bzgl. eines Zustandes \mathbf{z} für einen Zustandsübergang herangezogen wird, soll nun auch *nebenläufiges* Feuern von mehreren Transitionen formalisiert werden. Hierzu muss der Begriff der Feuerbarkeit verallgemeinert werden.

Definition 4.9:

Es sei ein mehrfach attribuiertes gefärbtes Petri-Netz $N = (P, T, F, B, f, A, K, k)$ und \mathbf{z} eine Markierung von N gegeben und $\{t_1, \dots, t_r\}$ eine Menge von Transitionen.¹ Mit $\bullet\{t_1, \dots, t_r\}$ werde der gemeinsame Vorbereich der Transitionen t_1, \dots, t_r bezeichnet. Die Transitionenmenge $\{t_1, \dots, t_r\}$ heißt *nebenläufig feuerbar* im Zustand \mathbf{z} , wenn alle Transitionen aus $\{t_1, \dots, t_r\}$ aktiviert sind² und für alle $p \in \bullet\{t_1, \dots, t_r\}$ gilt:

$$\biguplus_{t \in \{t_1, \dots, t_r\} \cap p \bullet} f(p, t) \subseteq \mathbf{z}(p).$$

Der neue Zustand \mathbf{z}'_i eines Platzes $p_i \in P$ wird nach der folgenden Gleichung bestimmt:

$$\mathbf{z}'_i = \left(\mathbf{z}_i \setminus \biguplus_{t \in \{t_1, \dots, t_r\} \cap p_i \bullet} f(p_i, t) \right) \uplus \biguplus_{t \in \{t_1, \dots, t_r\} \cap \bullet p_i} f(t, p_i).$$

¹Um eine Doppelindizierung zu vermeiden, wird o.B.d.A. angenommen, dass es sich um die r ersten Transitionen aus T handelt.

²Gemäß Def. 4.8

Beispiel:

Betrachtet man das MCPN-Beispiel im Zustand \mathbf{z}' (Abb. 4.13), ist es offensichtlich, dass die Transitionen t_1 und t_2 aktiviert und nebenläufig feuierbar sind:

Wird nun t_1 und t_2 gefeuert, wird aus Zustand

$$\mathbf{z}' \equiv \begin{pmatrix} \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \\ \{(1, 2), (2, 1), (3, 1)\} \\ \{((\text{ROT}, 1), 1), ((\text{BLAU}, 3), 1), ((\text{GRÜN}, 1), 1)\} \\ \{(1, 1), (2, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \end{pmatrix}$$

in den Zustand

$$\begin{aligned} \mathbf{z}'' &\equiv \begin{pmatrix} \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \setminus \{(\text{ROT}, 1), (\text{GRÜN}, 1)\} \\ \{(1, 2), (2, 1), (3, 1)\} \setminus \{(1, 2)\} \\ \left[\{((\text{ROT}, 1), 1), ((\text{BLAU}, 3), 1), ((\text{GRÜN}, 1), 1)\} \setminus \{((\text{ROT}, 1), 1), ((\text{BLAU}, 3), 1), ((\text{GRÜN}, 1), 1)\}\} \right. \\ \quad \left. \uplus \{((\text{ROT}, 1), 1), ((\text{GRÜN}, 1), 1)\} \right] \\ \{(1, 1), (2, 1)\} \uplus \{(1, 2), (3, 1)\} \\ \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \uplus \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \end{pmatrix} \\ &= \begin{pmatrix} \{(\text{ROT}, 1), (\text{BLAU}, 2)\} \\ \{(2, 1), (3, 1)\} \\ \{((\text{ROT}, 1), 1), ((\text{GRÜN}, 1), 1)\} \\ \{(1, 3), (2, 1), (3, 1)\} \\ \{(\text{ROT}, 3), (\text{BLAU}, 1), (\text{GRÜN}, 2)\} \end{pmatrix} \end{aligned}$$

übergegangen.

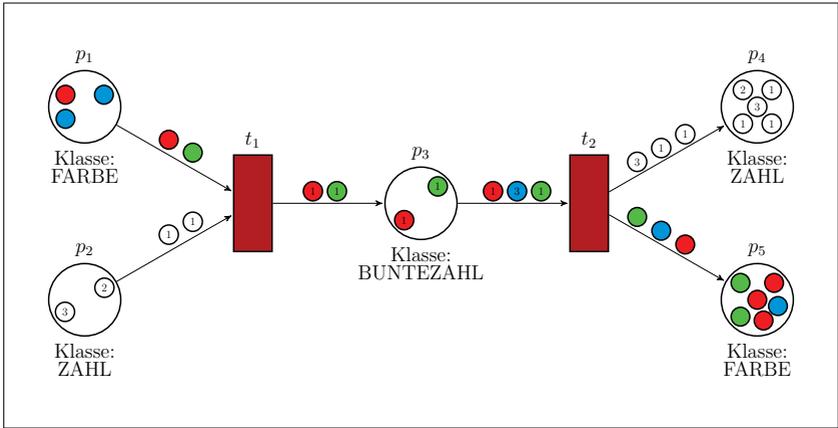


Abbildung 4.14: Zustand z'' des MCPN-Beispiels (nach nebenläufigem Feuern)

Des Weiteren lässt sich für das MCPN-Beispiel die folgende Rechnung ausführen:

$$\begin{pmatrix}
 \{(ROT, 3), (BLAU, 2), (GRÜN, 2)\} \\
 \{(1, 4), (2, 1), (3, 1)\} \\
 \{(BLAU, 3), 1\} \\
 \{(1, 1), (2, 1)\} \\
 \{(ROT, 2), (GRÜN, 1)\}
 \end{pmatrix}
 < t_1 >
 \begin{pmatrix}
 \{(ROT, 2), (BLAU, 2), (GRÜN, 1)\} \\
 \{(1, 2), (2, 1), (3, 1)\} \\
 \{(ROT, 1), 1, (BLAU, 3), 1, (GRÜN, 1), 1\} \\
 \{(1, 1), (2, 1)\} \\
 \{(ROT, 2), (GRÜN, 1)\}
 \end{pmatrix}$$

$$< t_1, t_2 >
 \begin{pmatrix}
 \{(ROT, 1), (BLAU, 2)\} \\
 \{(2, 1), (3, 1)\} \\
 \{(ROT, 1), 1, (GRÜN, 1), 1\} \\
 \{(1, 3), (2, 1), (3, 1)\} \\
 \{(ROT, 3), (BLAU, 1), (GRÜN, 2)\}
 \end{pmatrix}$$

Diese Rechnung kann im Endzustand nicht fortgesetzt werden, weil hier keine Transition mehr feuierbar ist.

In diesem Kapitel wurden bis jetzt nur CPN mit einer *konstanten* Pfeilgewichtung behandelt, was vor allem aus didaktischen Gründen geschieht. Die allgemeinere und in der Praxis häufiger benutzte Form von CPN mit *variabler* Pfeilgewichtung wird im Kapitel 6 beschrieben. In Kapitel 5 folgt zunächst die Anwendung von Konfliktlösungsstrategien auf die MCPN.

5 Allgemeine Konfliktlösungen bei Feuerprozessen von gefärbten Petri-Netzen

Als Ausgangspunkt betrachte man die *allgemeine Konfliktsituation* bei Petri-Netzen (vgl. Kap. 2). Hierbei liegt der Konflikt darin begründet, dass bei einer Teilmenge von Transitionen aus T jede einzelne zwar aktiviert und somit auch seriell feuierbar ist, aber aufgrund von zu wenigen Token im gemeinsamen Vorbereich nicht alle diese Transitionen nebenläufig feuierbar sind. Bei gefärbten Petri-Netzen ist aber nicht nur auf die Anzahl aller Token, sondern zusätzlich noch auf die Tokenanzahl in den einzelnen „Farben“ zu achten.

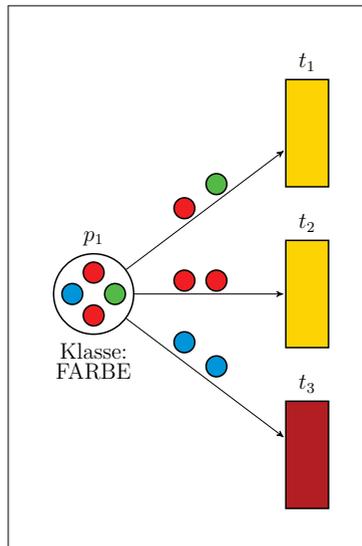


Abbildung 5.1: Allgemeiner Konflikt bei einem gefärbten Petri-Netz (vgl. Abb. 2.1)

Die in Kapitel 2 vorgestellten Konzepte zur Auflösung der allgemeinen Konfliktsituationen bei PN sollen auf MCPN übertragen werden. Wie bereits in Kapitel 4 erwähnt müssen dazu

5.1 Verallgemeinerung eines allgemeinen Konflikts für MCPN

die allgemeinen Operatoren $+$, $-$, \leq in Multimengenoperatoren nach dem Schema

$$\begin{aligned} + &\rightarrow \uplus \\ - &\rightarrow \setminus \\ \leq &\rightarrow \subseteq \end{aligned}$$

umgewandelt werden. Diese Umwandlung betrachten wir nun im folgenden Abschnitt (vgl. auch [KD13]).

5.1 Verallgemeinerung eines allgemeinen Konflikts für MCPN

Zunächst werden folgende Bezeichnungen verallgemeinert:

Definition 5.1 (vgl. Def. 2.1):

Das Tupel $(P, T, F, B, f, A, K, k, \mathbf{z})$ sei ein MCPN mit der Markierung \mathbf{z} .

a) $TA(\mathbf{z})$ bezeichnet die Menge aller aktivierten Transitionen des MCPN:

$$TA(\mathbf{z}) := \{t \in T \mid \forall p \in \bullet t : f(p, t) \subseteq \mathbf{z}(p)\}.$$

b) $TA_{in}(p, \mathbf{z})$ bezeichnet die Menge der aktivierten Transitionen des Vorbereichs eines Platzes $p \in P$:

$$TA_{in}(p, \mathbf{z}) := \{t \in \bullet p \mid t \in TA(\mathbf{z})\} = TA(\mathbf{z}) \cap \bullet p.$$

c) $TA_{out}(p, \mathbf{z})$ bezeichnet die Menge der aktivierten Transitionen des Nachbereichs eines Platzes $p \in P$:

$$TA_{out}(p, \mathbf{z}) := \{t \in p\bullet \mid t \in TA(\mathbf{z})\} = TA(\mathbf{z}) \cap p\bullet.$$

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch TA statt $TA(\mathbf{z})$ bzw. $TA_{in}(p)$ statt $TA_{in}(p, \mathbf{z})$ bzw. $TA_{out}(p)$ statt $TA_{out}(p, \mathbf{z})$ geschrieben.

Definition 5.2 (vgl. Def. 2.2):

Das Tupel $(P, T, F, B, f, A, K, k, \mathbf{z})$ sei ein MCPN mit der Markierung \mathbf{z} .

Ein Platz $p \in P$ induziert eine **allgemeine Konfliktsituation** bezüglich \mathbf{z} , wenn gilt:

$$\bigcup_{t \in TA_{out}(p)} f(p, t) \not\subseteq \mathbf{z}(p).$$

Man mache sich klar, dass ein Platz p nur dann einen allgemeinen Konflikt auslösen kann, wenn die Menge $TA_{out}(p)$ mehr als eine (aktivierte) Transition enthält.

Beispiel:

Die Plätze p_1 und p_2 des MCPN in Abb. 5.2 haben einen allgemeinen Konflikt. Der Platz p_1 kann entweder t_1 oder t_2 zum Feuern freigeben; der Platz p_2 kann entweder t_2 oder t_3 zum Feuern freigeben. Die Transition t_4 ist keine Option, weil diese aufgrund fehlender Token in p_3 nicht aktiviert ist.

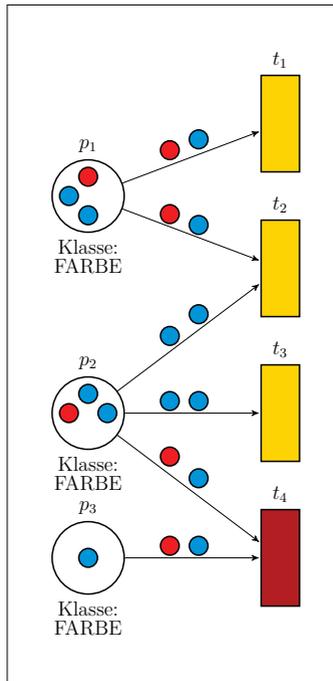


Abbildung 5.2: Allgemeine Konfliktsituation im MCPN

5.1 Verallgemeinerung eines allgemeinen Konflikts für MCPN

An dieser Stelle sei auch angemerkt, dass bei den normalen Petri-Netzen für die Bedingung des allgemeinen Konfliktes sowohl

$$\sum_{t \in TA_{out}(p)} f(p, t) > \mathbf{z}(p)$$

als auch

$$\sum_{t \in TA_{out}(p)} f(p, t) \not\leq \mathbf{z}(p)$$

gilt, zumal es sich um äquivalente Ausdrücke handelt. Zwischen den analogen Ausdrücken bei CPN, nämlich

$$\biguplus_{t \in TA_{out}(p)} f(p, t) \not\subseteq \mathbf{z}(p)$$

und

$$\biguplus_{t \in TA_{out}(p)} f(p, t) \supset \mathbf{z}(p)$$

besteht allerdings keine Äquivalenz mehr.

Beispiel:

Offensichtlich besteht im folgenden MCPN ein allgemeiner Konflikt gemäß obiger Definition.

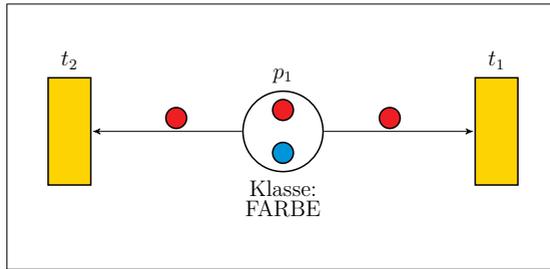


Abbildung 5.3: Allgemeiner Konflikt im MCPN

Dieser Sachverhalt wird durch die Bedingung

$$\biguplus_{t \in TA_{out}(p)} f(p, t) \not\subseteq \mathbf{z}(p)$$

wiedergegeben, zumal $\{(ROT, 2)\} \not\subseteq \{(ROT, 1), (BLAU, 1)\}$ gilt. Allerdings würde nach der Bedingung

$$\biguplus_{t \in TA_{out}(p)} f(p, t) \supset \mathbf{z}(p)$$

vermeintlich kein allgemeiner Konflikt bestehen.

Definition 5.3 (vgl. Def. 2.3):

Das Tupel $(P, T, F, B, f, A, K, k, \mathbf{z})$ sei ein MCPN mit der Markierung \mathbf{z} .

Eine Menge von Transitionen $\{t_1, \dots, t_r\}$ aus der Menge $TA_{out}(p)$ eines Platzes $p \in P$ ist **maximal feuerbar**, wenn folgende Bedingung für alle $t_0 \in (TA_{out}(p) \setminus \{t_1, \dots, t_r\})$ gilt:

$$\left(\bigoplus_{t \in \{t_1, \dots, t_r\}} f(p, t) \subseteq \mathbf{z}(p) \right) \wedge \left(f(p, t_0) \not\subseteq \bigoplus_{t \in \{t_1, \dots, t_r\}} f(p, t) \right)$$

Beispiel:

In Abb. 5.4 sind die einelementigen Transitionenmengen $\{t_1\}$, $\{t_2\}$ und $\{t_3\}$ jeweils zwar feuerbar, aber nicht maximal feuerbar. Die zweielementigen Transitionenmengen $\{t_1, t_2\}$ und $\{t_1, t_3\}$ sind maximal feuerbar, während die Transitionenmenge $\{t_2, t_3\}$ nicht feuerbar ist.

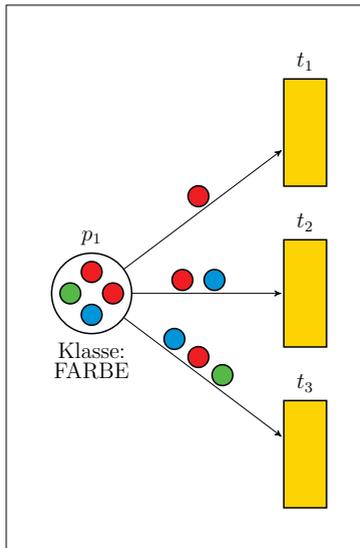


Abbildung 5.4: Maximal feuerbare Transitionenmengen eines MCPN

5.2 Lokale Lösungsansätze für allgemeine Konflikte bei MCPN

Zunächst sollen die Konfliktsituationen, wie bei den Petri-Netzen, lokal behandelt werden, d.h. es wird ein einzelner Platz p betrachtet. Bei dieser Betrachtungsweise wird eine gegenseitige Beeinflussung von Konflikten noch nicht berücksichtigt (vgl. Abschn. 2.2).

Definition 5.4 (vgl. Def. 2.4):

Das Tupel $(P, T, F, B, f, A, K, k, \mathbf{z})$ sei ein MCPN mit der Markierung \mathbf{z} .

Mit $TE_{out}(p, \mathbf{z})$ werde die **Menge der freigegebenen Transitionen** des Nachbereichs eines Platzes $p \in P$ bezeichnet. Selbstverständlich muss diese Transitionenmenge feuerbar sein, d.h. es muss gelten:

$$\biguplus_{t \in TE_{out}(p, \mathbf{z})} f(p, t) \subseteq \mathbf{z}(p)$$

und somit

$$TE_{out}(p, \mathbf{z}) \subseteq TA_{out}(p, \mathbf{z}).$$

Ist die Markierung \mathbf{z} fest vorgegeben bzw. als bekannt vorauszusetzen, wird vereinfacht auch $TE_{out}(p)$ statt $TE_{out}(p, \mathbf{z})$ geschrieben.

5.2.1 Gefärbte Petri-Netze mit Bewertungen

Es werden drei verschiedene Bewertungen von den Petri-Netzen (vgl. Abschn. 2.2.1) auf die MCPN übertragen:

1. Allgemeine Bewertungen
2. Prioritäten (insbesondere normierte Prioritäten)
3. Wahrscheinlichkeiten (insbesondere aktualisierte Wahrscheinlichkeiten)

Man beachte allerdings, dass diese Übertragung trivial ist, da die Bewertungsfunktionen äquivalent wie bei den Petri-Netzen definiert werden. Daher sei an dieser Stelle nur die allgemeine Bewertungsfunktion exemplarisch neu definiert.¹

Definition 5.5 (vgl. Def. 2.5):

Gegeben sei ein MCPN $N = (P, T, F, B, f, A, K, k)$. Eine Abbildung $\sigma : F \rightarrow \mathbb{R}_{\geq}$ heißt **allgemeine Bewertungsfunktion** von N , wobei jedem Pfeil $(p, t) \in F$ eine *allgemeine Bewertung* $\sigma(p, t)$ zugewiesen wird.

Beispiel:

Die Pfeile aus Abb. 5.5 haben folgende allgemeine Bewertungen:

$$\begin{aligned}\sigma(p_1, t_1) &= 6 \\ \sigma(p_1, t_2) &= 5 \\ \sigma(p_1, t_3) &= 4711 \\ \sigma(p_1, t_4) &= 4,5\end{aligned}$$

¹Eine ausführlichere Beschreibung hierzu findet man in [KD13].

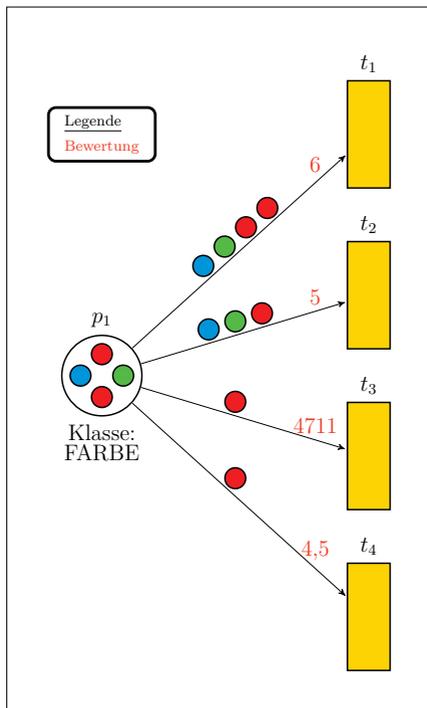


Abbildung 5.5: MCPN mit Bewertungen

Als weitere Beispiele seien hier auch noch zwei MCPN mit Prioritäten bzw. mit Wahrscheinlichkeiten dargestellt.

Beispiel:

Die Pfeile aus Abb. 5.6 haben folgende normierte Prioritäten:

$$\begin{aligned} \varrho(p_1, t_1) &= 1 \\ \varrho(p_1, t_2) &= 3 \\ \varrho(p_1, t_3) &= 2 \end{aligned}$$

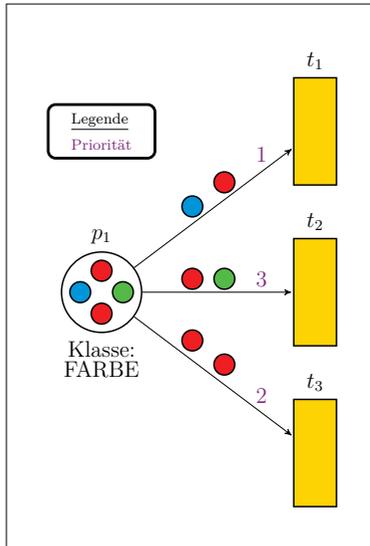


Abbildung 5.6: MCPN mit normierten Prioritäten

Beispiel:

Die Pfeile aus Abb. 5.7 haben folgende Wahrscheinlichkeiten:

$$\begin{aligned}
 \omega(p_1, t_1) &= 0,2 & \omega_z(p_1, t_1) &= 0,4 \\
 \omega(p_1, t_2) &= 0,1 & \omega_z(p_1, t_2) &= 0,2 \\
 \omega(p_1, t_3) &= 0,2 & \omega_z(p_1, t_3) &= 0,4 \\
 \omega(p_1, t_4) &= 0,5 & &
 \end{aligned}$$

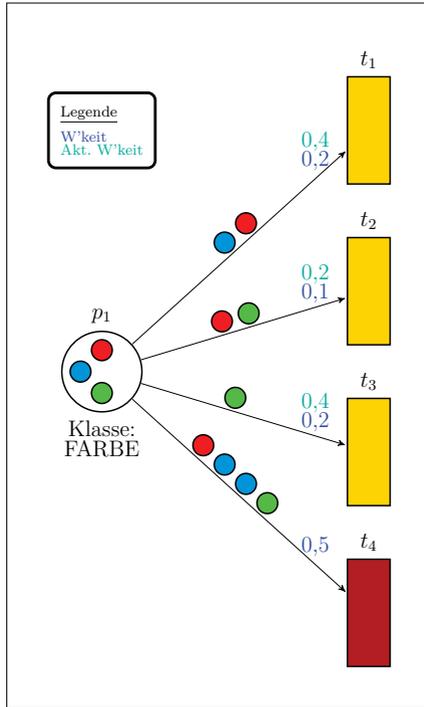


Abbildung 5.7: MCPN mit Wahrscheinlichkeiten

5.2.2 Beispiele für algorithmische Beschreibungen von lokalen Freigabeprozessen

Für die lokale Freigabe von Transitionen bei MCPN lassen sich dieselben Algorithmen verwenden wie bei den PN (vgl. Abschn. 2.2.2). Man beachte allerdings, dass hierbei die algorithmischen Beschreibungen auf Multimengen verallgemeinert werden müssen; die daraus hervorgehenden Änderungen sind in roter Schrift hervorgehoben.

Für Prioritäten

Gegeben sei ein MCPN $N = (P, T, F, B, f, A, K, k, \mathbf{z}, \varrho)$ mit Prioritätsfunktion ϱ . Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Prioritäten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben

ist und zudem die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ in aufsteigender Reihenfolge ihrer Prioritäten geordnet ist, d.h.

$$\varrho(p, t_i) < \varrho(p, t_j) \text{ für alle } t_i, t_j \in TA_{out}(p) \text{ mit } i < j.$$

Algorithmus 5.6: Lokale Freigabe mittels Prioritätenverfahren für MCPN (vgl. Algorithmus 2.8)

Setze $TE_{out}(p) := \emptyset$.

Für $i = 1, \dots, \kappa(p)$ führe aus:

Falls gilt: $f(p, t_i) \uplus \biguplus_{t \in TE_{out}(p)} f(p, t) \subseteq \mathbf{z}(p)$,

setze: $TE_{out}(p) := TE_{out}(p) \cup \{t_i\}$.

Ergebnis: Die Menge $TE_{out}(p)$ beinhaltet alle aktivierten Transitionen, die vom Platz p zum Feuern freigegeben werden. Bei der Bildung der Transitionenmenge $TE_{out}(p)$ gehen die Prioritäten der Transitionen bzgl. p maßgeblich ein.

Beispiel:

In Abb. 5.8 hat p_1 einen allgemeinen Konflikt.

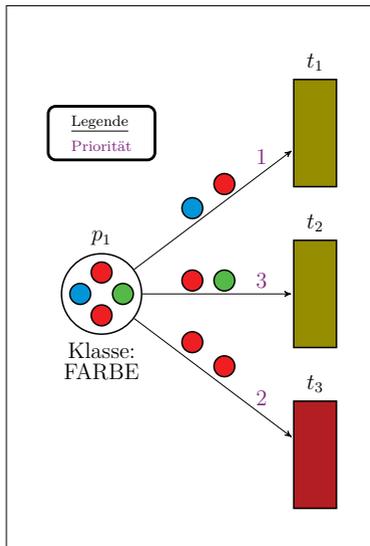


Abbildung 5.8: Lokale Freigabe bei MCPN mit Prioritätenverfahren

Es erfolgt die Freigabe nach Prioritäten; dabei wird in der folgenden Reihenfolge geprüft:

Setzung: $TE_{out}(p_1) := \emptyset$.

Prüfungen:

t_1 wg. $\varrho(p_1, t_1) = 1$:

$$\begin{aligned} f(p_1, t_1) &= \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \\ \Rightarrow TE_{out}(p_1) &:= TE_{out}(p_1) \cup \{t_1\} = \{t_1\} \end{aligned}$$

t_3 wg. $\varrho(p_1, t_3) = 2$:

$$\begin{aligned} f(p_1, t_1) \uplus f(p_1, t_3) &= \{(\text{ROT}, 3), (\text{BLAU}, 1)\} \not\subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \\ \Rightarrow t_3 &\text{ kann nicht (zusätzlich) freigegeben werden.} \end{aligned}$$

t_2 wg. $\varrho(p_1, t_2) = 3$:

$$\begin{aligned} f(p_1, t_1) \uplus f(p_1, t_2) &= \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \\ \Rightarrow TE_{out}(p_1) &:= TE_{out}(p_1) \cup \{t_2\} = \{t_1, t_2\} \end{aligned}$$

Ergebnis: $TE_{out}(p_1) = \{t_1, t_2\}$, d.h. es werden die Transitionen t_1 und t_2 bzgl. Platz p_1 freigegeben.

Für Wahrscheinlichkeiten

Gegeben sei ein MCPN $N = (P, T, F, B, f, A, K, k, \mathbf{z}, \omega)$ mit Wahrscheinlichkeitsfunktion ω . Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Wahrscheinlichkeiten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben ist. Durch eine Indizierung der Elemente $t_i \in TA_{out}(p)$, $i = 1, \dots, \kappa(p)$, ist eine Anordnung der Transitionen vorgegeben, so dass jeder Transition $t_i \in TA_{out}(p)$, $i = 1, \dots, \kappa(p)$, auf kanonische Weise die kumulierte Wahrscheinlichkeit

$$\omega_{\Sigma}(p, t_i) = \sum_{j=1}^i \omega_{\mathbf{z}}(p, t_j)$$

zugewiesen wird.

Algorithmus 5.7: Lokale Freigabe mittels Wahrscheinlichkeiten-Verfahren für MCPN (vgl. Algorithmus 2.9)

Setze:

$$TE_{out}(p) := \emptyset$$

$$L := TA_{out}(p) \text{ (Kandidatenliste)}$$

$$\omega_L := 1 \text{ (Summierte Restkandidaten-W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(0) := 0 \text{ (Hilfs-W'keit für Monte-Carlo-Methode)}$$

Für $i = 1, \dots, \kappa(p)$ führe aus:

Setze:

$$\pi(i) := i \text{ (Reihenfolgezähler)}$$

$$\omega^{aktuell}(\pi(i)) := \omega_z(p, t_i) \text{ (aktualisierte W'keit)}$$

$$\omega_{\Sigma}^{aktuell}(\pi(i)) := \omega_{\Sigma}(p, t_i) \text{ (aktualisierte kumulierte W'keit)}$$

Solange $L \neq \emptyset$ führe aus:

Ermittle Zufallszahl $\rho \in (0; 1]$.

Bestimme $t_i \in L$ mit $\rho \in (\omega_{\Sigma}^{aktuell}(\pi(i) - 1); \omega_{\Sigma}^{aktuell}(\pi(i))]$.

Falls gilt: $f(p, t_i) \uplus \biguplus_{t \in TE_{out}(p)} f(p, t) \subseteq \mathbf{z}(p)$,

$$\text{setze: } TE_{out}(p) := TE_{out}(p) \cup \{t_i\}.$$

Setze: $L := L \setminus \{t_i\}$; $\omega_L := \omega_L - \omega_z(p, t_i)$.

Für $j = i + 1, \dots, \kappa(p)$ setze: $\pi(j) := \pi(j) - 1$.

Für alle $t_j \in L$ führe aus:

Setze:

$$\omega^{aktuell}(\pi(j)) := \frac{\omega_z(p, t_j)}{\omega_L};$$

$$\omega_{\Sigma}^{aktuell}(\pi(j)) := \omega_{\Sigma}^{aktuell}(\pi(j) - 1) + \omega^{aktuell}(\pi(j)).$$

Ergebnis: Die Menge $TE_{out}(p)$ beinhaltet alle aktivierten Transitionen, die vom Platz p zum Feuern freigegeben werden.

Beispiel:

In Abb. 5.9 hat Platz p_1 einen allgemeinen Konflikt; die Freigabe erfolgt diesmal nach Wahrscheinlichkeiten. Es wird so vorgegangen:

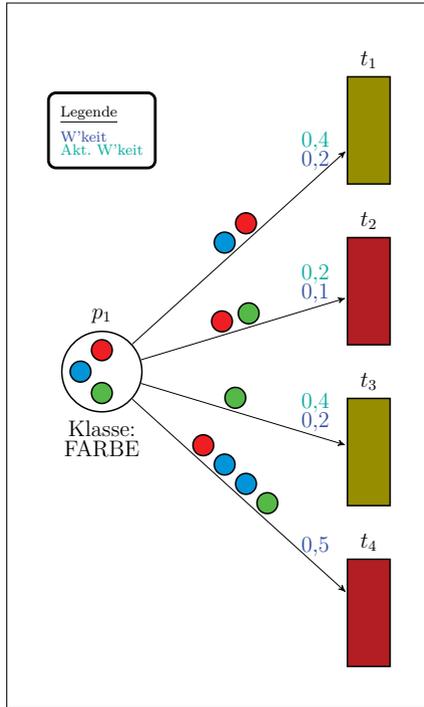


Abbildung 5.9: Lokale Freigabe bei MCPN mit Wahrscheinlichkeitsverfahren

Berechnung der aktualisierten und kumulierten Wahrscheinlichkeiten:

$$\begin{aligned} \omega_z(p_1, t_1) &= \frac{0,2}{0,5} = 0,4; \quad \omega_\Sigma(p_1, t_1) = 0,4; \\ \omega_z(p_1, t_2) &= \frac{0,1}{0,5} = 0,2; \quad \omega_\Sigma(p_1, t_2) = 0,4 + 0,2 = 0,6; \\ \omega_z(p_1, t_3) &= \frac{0,2}{0,5} = 0,4; \quad \omega_\Sigma(p_1, t_3) = 0,4 + 0,2 + 0,4 = 1. \end{aligned}$$

Setzungen: $TE_{out}(p_1) := \emptyset; L := TA_{out}(p_1) = \{t_1, t_2, t_3\}; \omega_L := 1; \omega_\Sigma^{aktuell}(0) := 0$

i	1	2	3
$\pi(i)$	1	2	3
$\omega^{aktuell}(\pi(i))$	0,4	0,2	0,4
$\omega_\Sigma^{aktuell}(\pi(i))$	0,4	0,6	1

Schritt 1:

$$L = \{t_1, t_2, t_3\} \neq \emptyset$$

$$\rho = 0,3 \text{ (Zufallszahl)}$$

5.2 Lokale Lösungsansätze für allgemeine Konflikte bei MCPN

$\rho \in (0; 0,4] \Rightarrow$ wähle $t_1 \Rightarrow i = 1$

$f(p_1, t_1) = \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \subseteq \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1)$

$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_1\} = \{t_1\}$

$L := \{t_1, t_2, t_3\} \setminus \{t_1\} = \{t_2, t_3\}$

$\omega_L := 1 - 0,4 = 0,6$

i	2	3
$\pi(i)$	1	2
$\omega^{\text{aktuell}}(\pi(i))$	$\frac{1}{3}$	$\frac{2}{3}$
$\omega_{\Sigma}^{\text{aktuell}}(\pi(i))$	$\frac{1}{3}$	1

Schritt 2:

$L = \{t_2, t_3\} \neq \emptyset$

$\rho = 0,1$ (Zufallszahl)

$\rho \in (0; \frac{1}{3}] \Rightarrow$ wähle $t_2 \Rightarrow i = 2$

$f(p_1, t_1) \uplus f(p_1, t_2) = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \not\subseteq$

$\{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1)$

$L := \{t_2, t_3\} \setminus \{t_2\} = \{t_3\}$

$\omega_L := 0,6 - 0,2 = 0,4$

i	3
$\pi(i)$	1
$\omega^{\text{aktuell}}(\pi(i))$	1
$\omega_{\Sigma}^{\text{aktuell}}(\pi(i))$	1

Schritt 3:

$L = \{t_3\} \neq \emptyset$

$\rho = 0,9$ (Zufallszahl)

$\rho \in (0; 1] \Rightarrow$ wähle $t_3 \Rightarrow i = 3$

$f(p_1, t_1) \uplus f(p_1, t_3) = \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \subseteq$

$\{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1)$

$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_3\} = \{t_1, t_3\}$

$L := \{t_3\} \setminus \{t_3\} = \emptyset$

$\omega_L := 0,4 - 0,4 = 0$

Schritt 4:

$L = \emptyset \Rightarrow$ Terminiere

Ergebnis: $TE_{out}(p_1) = \{t_1, t_3\}$; d.h. p_1 gibt t_1 und t_3 frei.

Eine alternative Lösung ist in Abb. 5.10 dargestellt. Diese ergibt sich, wenn beispielsweise die erste Zufallszahl 0,5 ist (alle anderen Transitionen würden dadurch nicht mehr freigebbar sein):

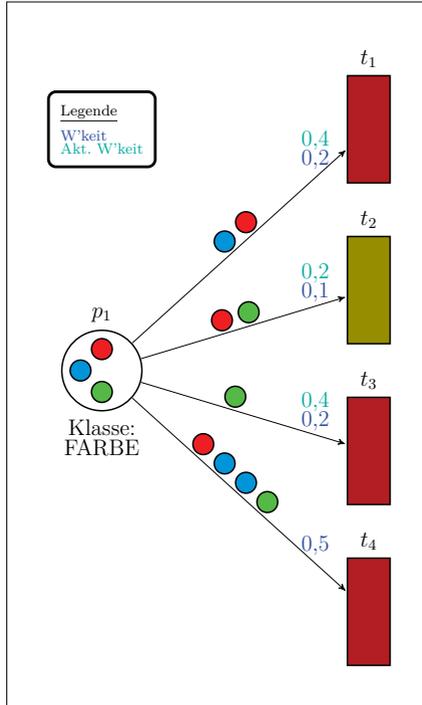


Abbildung 5.10: Lokale Freigabe bei MCPN mit Wahrscheinlichkeitsverfahren (alternative Lösung)

Für Nutzen

Gegeben sei ein MCPN $N = (P, T, F, B, f, A, K, k, z, \psi)$ im Zustand \mathbf{z} mit allgemeiner Bewertungsfunktion $\psi : F \rightarrow \mathbb{R}_{\geq}$, die als Nutzen aufgefasst wird. Ein Platz $p \in P$ erteilt in Bezug auf den Zustand \mathbf{z} einer Teilmenge $TE_{out}(p)$ von aktivierten Transitionen aus $TA_{out}(p)$ die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge** $TE_{out}(p)$ **mittels Nutzen**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge $TA_{out}(p)$ mit $\kappa(p) \in \mathbb{N}$ gegeben ist.

Hier werden nun die drei Algorithmen aus Abschnitt 2.2.2.2 verallgemeinert.

Der erste ist die Greedy-Heuristik für Nutzen, welche immer versucht, die Transition mit dem jeweils noch größten Nutzen freizugeben. Dafür seien die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ o.B.d.A. in absteigender Reihenfolge ihres Nutzens $\psi(p, t_j)$ geordnet, d.h.

$$\psi(p, t_1) \geq \psi(p, t_2) \geq \dots \geq \psi(p, t_{\kappa(p)}) \text{ für alle } t_j \in TA_{out}(p).$$

Der zweite Algorithmus ist der obige Greedy-Algorithmus, angewandt auf die **Nutzenquotienten** η_j , wobei die Transitionen $t_1, \dots, t_{\kappa(p)} \in TA_{out}(p)$ nun o.B.d.A. in absteigender Reihenfolge ihrer Nutzenquotienten η_j geordnet sind, d.h.

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_{\kappa(p)} \text{ für alle } t_j \in TA_{out}(p) \text{ mit } \eta_j = \frac{\psi(p, t_j)}{|f(p, t_j)|}.$$

Da der Nutzenquotient als Nutzen pro verbrauchten Token zu interpretieren ist, wird hier die Mächtigkeit der Pfeilgewichtung verwendet. Beide Algorithmen sind im Prinzip das Prioritätenverfahren mit einer anderen Anordnung der Transitionen, wobei $\psi(p, t_j)$ oder η_j „verallgemeinerte Prioritäten“ wären (wohlgemerkt: i.d.R. $\psi(p, t_j) \notin \mathbb{N}$ und $\eta_j \notin \mathbb{N}$).

Der folgende Algorithmus arbeitet nach dem Branch- & Bound-Prinzip, wobei die Transitionen wieder nach dem Nutzenquotienten geordnet sind.

Algorithmus 5.8: Lokale Freigabe mittels B&B (vgl. Algorithmus 2.10)

Setze:

$X_0 := \{(x_1, x_2, \dots, x_{\kappa(p)}) \mid x_j \in \{0, 1\}, j = 1, \dots, \kappa(p)\}$ (Ursprungsknoten)

$g_0 := \emptyset$ (Gewicht der Teillösung)

$n_0 := \sum_{j=1}^{\kappa(p)} \psi(p, t_j)$ (theoretischer Nutzen des Problems)

$N := 0$ (Effizienzschranke (Bound))

$TP := \{X_0\}$ (Menge aller Teilknoten)

$s := 0$ (Zähler der Teilknoten)

$m_0 := 0$ (Stufe des Teilknotens)

$L := \emptyset$ (Lösung)

1. Schritt: Falls $TP = \emptyset$ terminiere.

Wähle X_r aus TP mit $n_r = \max(n_i \mid X_i \in TP)$ (bei Mehrdeutigkeit wähle willkürlich den Index r)

2. Schritt: Falls $m_r < \kappa(p)$, führe aus:

Setze:

$$X_{s+1} := X_r \text{ mit } x_{m_r+1} = 1; X_{s+2} := X_r \text{ mit } x_{m_r+1} = 0;$$

$$n_{s+1} := n_r; n_{s+2} := n_r - \psi(p, t_{m_r+1});$$

$$g_{s+1} := g_r \uplus f(p, t_{m_r+1}); g_{s+2} := g_r;$$

$$m_{s+1} := m_{s+2} := m_r + 1;$$

$$TPg\ddot{u}ltig := \{X_{s+i} \mid g_{s+i} \subseteq \mathbf{z}(p) \wedge n_{s+i} > N, i = 1, 2\}$$

$$TP := (TP \setminus \{X_r\}) \cup TPg\ddot{u}ltig;$$

Falls $(m_r = \kappa(p) - 1) \wedge (TPg\ddot{u}ltig \neq \emptyset)$, führe aus:

Wähle X_l aus $TPg\ddot{u}ltig$ mit $n_l = \max(n_i \mid X_i \in TPg\ddot{u}ltig)$

Setze:

$$N := n_l$$

$$L := X_l$$

Für alle $X_i \in TP$ führe aus:

Falls $n_i \leq N$, setze:

$$TP := TP \setminus \{X_i\};$$

Gehe zum Schritt 1.

Ergebnis: Das binäre Tupel $(x_1, x_2, \dots, x_{\kappa(p)}) \in L$ ist folgendermaßen zu interpretieren:

$$x_i = 1 \equiv \text{Transition } t_i \text{ wird freigegeben,}$$

$$x_i = 0 \equiv \text{Transition } t_i \text{ wird nicht freigegeben,}$$

so dass gilt:

$$TE_{out}(p) := \{t_i \in TA_{out}(p) \mid x_i = 1\}.$$

Die Menge $TE_{out}(p)$ beinhaltet nun alle aktivierten Transitionen, die vom Platz p zum Feu-
ern freigegeben werden.

Auch hier bietet es sich wieder an, eine erste Startlösung schon als Effizienzschranke N
zu benutzen (vgl. Diving B&B in [Kä06, S. 25f.]).

Beispiel:

In Abb. 5.11 ist ein MCPN mit Nutzen abgebildet. In diesem MCPN hat Platz p_1 einen allgemeinen Konflikt; die Freigabe erfolgt nach Nutzenwerten. Je nach Freigabe-Prozedur ergeben sich drei Lösungsmöglichkeiten:

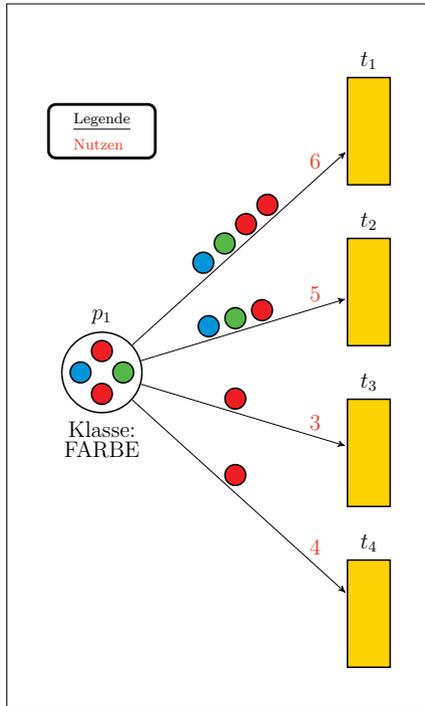


Abbildung 5.11: MCPN mit Nutzen

1. Lösung mit Greedy-Heuristik für Nutzen:

Hier werden die Transitionen nach ihrem Nutzen in absteigender Reihenfolge geordnet, so dass sich die Reihenfolge t_1, t_2, t_4, t_3 ergibt. Nun wird folgendermaßen verfahren:

Setzung:

$$TE_{out}(p_1) := \emptyset.$$

5 Allgemeine Konfliktlösungen bei Feuerprozessen von gefärbten Petri-Netzen

Prüfungen:

t_1 wg. $\psi(p_1, t_1) = 6$:

$$\begin{aligned} f(p_1, t_1) &= \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \\ &\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_1\} = \{t_1\} \end{aligned}$$

t_2 wg. $\psi(p_1, t_2) = 5$:

$$\begin{aligned} f(p_1, t_1) + f(p_1, t_2) &= \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \not\subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \Rightarrow t_2 \text{ wird nicht freigegeben.} \end{aligned}$$

t_4 wg. $\psi(p_1, t_4) = 4$:

$$\begin{aligned} f(p_1, t_1) + f(p_1, t_4) &= \{(\text{ROT}, 3), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \not\subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \Rightarrow t_4 \text{ wird nicht freigegeben.} \end{aligned}$$

t_3 wg. $\psi(p_1, t_3) = 3$:

$$\begin{aligned} f(p_1, t_1) + f(p_1, t_3) &= \{(\text{ROT}, 3), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \not\subseteq \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} &= \mathbf{z}(p_1) \Rightarrow t_3 \text{ wird nicht freigegeben.} \end{aligned}$$

Ergebnis: $TE_{out}(p_1) = \{t_1\}$, d.h. Platz p_1 gibt allein die Transition t_1 frei. Der gesamte Nutzen dieser Freigabe beträgt $N = 6$.

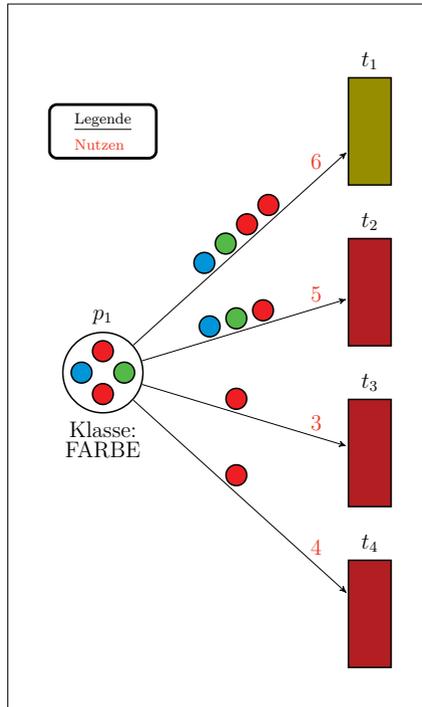


Abbildung 5.12: Lokale Freigabe bei MCPN mit Greedy-Heuristik für Nutzen

2. Lösung mit Greedy-Heuristik für Nutzenquotient:

Hier werden die Transitionen nun nach ihren Nutzenquotienten in absteigender Reihenfolge geordnet, so dass sich die Reihenfolge t_4, t_3, t_2, t_1 auf Grund der zugehörigen Nutzenquotienten $\frac{4}{1} = 4, \frac{3}{1} = 3, \frac{5}{3} = 1,\bar{6}$ und $\frac{6}{4} = 1,5$ ergibt. Nun wird folgendermaßen verfahren:

Setzung:

$$TE_{out}(p_1) := \emptyset.$$

Prüfungen:

$$t_4 \text{ wg. } \frac{\psi(p_1, t_4)}{|f(p_1, t_4)|} = \frac{4}{1} = 4:$$

$$f(p_1, t_4) = \{(ROT, 1)\} \subseteq \{(ROT, 2), (BLAU, 1), (GRÜN, 1)\} = \mathbf{z}(p_1)$$

$$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_4\} = \{t_4\}$$

5 Allgemeine Konfliktlösungen bei Feuerprozessen von gefärbten Petri-Netzen

$$t_3 \text{ wg. } \frac{\psi(p_1, t_3)}{|f(p_1, t_3)|} = \frac{3}{1} = 3:$$

$$f(p_1, t_4) + f(p_1, t_3) = \{(\text{ROT}, 2)\} \subseteq \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1)$$

$$\Rightarrow TE_{out}(p_1) := TE_{out}(p_1) \cup \{t_3\} = \{t_4, t_3\}$$

$$t_2 \text{ wg. } \frac{\psi(p_1, t_2)}{|f(p_1, t_2)|} = \frac{5}{3} = 1,\bar{6}:$$

$$f(p_1, t_4) + f(p_1, t_3) + f(p_1, t_2) = \{(\text{ROT}, 3), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \not\subseteq$$

$$\{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1) \Rightarrow t_2 \text{ wird nicht freigegeben.}$$

$$t_1 \text{ wg. } \frac{\psi(p_1, t_1)}{|f(p_1, t_1)|} = \frac{6}{4} = 1,5:$$

$$f(p_1, t_4) + f(p_1, t_3) + f(p_1, t_1) = \{(\text{ROT}, 4), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \not\subseteq$$

$$\{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} = \mathbf{z}(p_1) \Rightarrow t_1 \text{ wird nicht freigegeben.}$$

Ergebnis: $TE_{out}(p_1) = \{t_3, t_4\}$, d.h. Platz p_1 gibt t_3 und t_4 frei. Der gesamte Nutzen dieser Freigabe wäre $N = 7$.

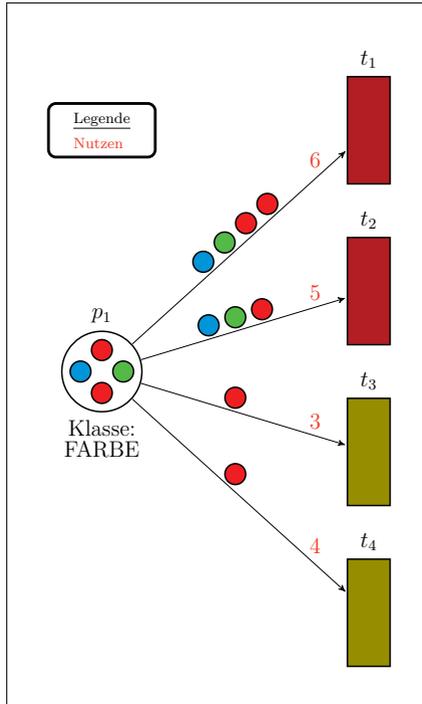


Abbildung 5.13: Lokale Freigabe bei MCPN mit Greedy-Heuristik für Nutzenquotient

3. Lösung mit Branch & Bound:

Hier werden die Transitionen zunächst nach ihren Nutzenquotienten in absteigender Reihenfolge geordnet, so dass sich wieder die Reihenfolge t_4, t_3, t_2, t_1 ergibt. Des Weiteren wird hier die Methode mit einer schon vorgegebenen Startlösung von $N = 7$ aus dem oberen Greedy-Verfahren verwendet. Die einzelnen Schritte des B&B-Verfahrens lassen sich in Abb. 5.14 verfolgen. Dabei ist zu beachten, dass die vereinfachte Schreibweise $R \equiv \text{ROT}, B \equiv \text{BLAU}, G \equiv \text{GRÜN}$ verwendet wird.

Am Anfangsknoten 0 ist der theoretisch noch zu erzielende Gesamtnutzen $n = 18$ vermerkt. Dieser ergibt sich aber nur, wenn das Tupel $(1, 1, 1, 1)$ realisierbar ist, d.h. $x_1 = x_2 = x_3 = x_4 = 1$ und $TE_{out}(p_1) = \{t_1, t_2, t_3, t_4\}$. Zudem wird die Anzahl g (dies ist eine Multimenge) der Tokens, die in Platz p_1 zur Verfügung stehen müssen, vermerkt. Da zu Beginn des Verfahrens noch keine Transitionen freigegeben sind (d.h. $x_1 = x_2 = x_3 = x_4 = 0$ bzw. $TE_{out}(p_1) = \emptyset$), gilt $g = \emptyset$.

Es wird die Transition mit dem größten Nutzenquotienten gewählt, also t_4 , und es werden die beiden Fälle $x_1 = 1$ und $x_1 = 0$ analysiert. Durch entsprechende Verzweigung (branching) ergeben sich die beiden Knoten 1 und 2. Am Knoten 1 sind die Konsequenzen aus der Entscheidung $x_1 = 1$ bzw. $TE_{out}(p_1) := \{t_4\}$ vermerkt, d.h. der theoretisch noch maximal mögliche Gesamtnutzen ist unverändert geblieben, während sich wegen $f(p_1, t_4) = \{(\text{ROT}, 1)\}$ ein „Verbrauch“ von einem roten Token eingestellt hat. Solange $g \subseteq \mathbf{z}(p_1)$ gilt (*Zulässigkeitsgrenze*), ist der Knoten für weitere Verzweigungen zulässig. Am Knoten 2 sind die Konsequenzen aus der Entscheidung $x_1 = 0$ bzw. weiterhin $TE_{out}(p_1) = \emptyset$ vermerkt, d.h. der theoretisch noch mögliche Gesamtnutzen hat sich um $\psi(p_1, t_4) = 4$ verringert, also $n = 14$, während die Tokenanzahl g unverändert bleibt. In der Folge wird immer an demjenigen Knoten weiter verzweigt, der den maximalen Wert n aufweist.

Eine erste zulässige Lösung ist am Knoten 10 erzielt: Das zugehörige binäre Tupel $(1, 0, 1, 0)$ entspricht der Freigabemenge $TE_{out}(p_1) = \{t_2, t_4\}$ mit dem Gesamtnutzen $N = 9$. Im weiteren Verlauf des Verfahrens wird diese Zwischenlösung $N = 9$ als untere Grenze (*Effizienzgrenze*) für das weitere Verzweigen eingesetzt, indem von zulässigen Knoten, deren Nutzenwert n nicht größer als die untere Grenze ist, auch nicht weiter verzweigt wird. Die Abbrüche auf Grund von Zulässigkeits- oder Effizienzgrenzen sind in Abb. 5.14 durch rote Striche für *nicht zulässig* bzw. blaue Striche für *nicht effizient* gekennzeichnet. Das Erreichen von zulässigen Lösungen wird durch grüne Striche verdeutlicht.

5 Allgemeine Konfliktlösungen bei Feuerprozessen von gefärbten Petri-Netzen

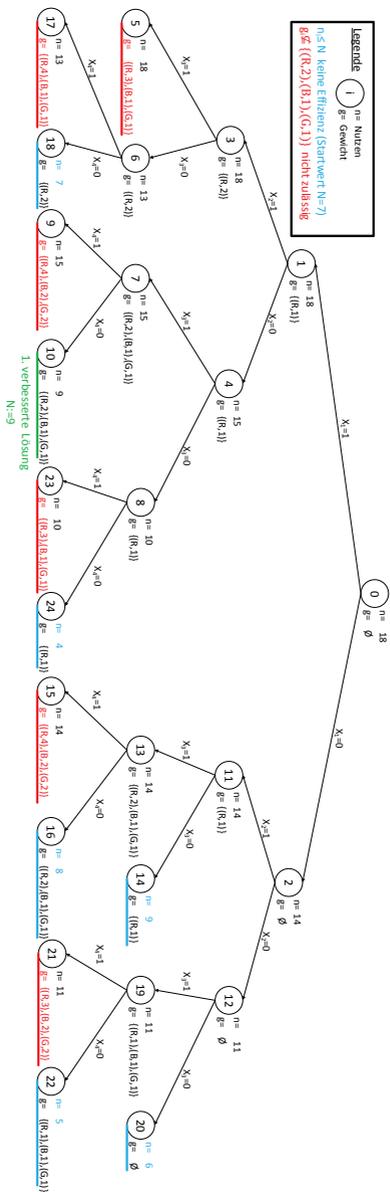


Abbildung 5.14: Lösung nach Branch & Bound

Insgesamt ergibt sich ein Baum mit 25 Knoten, wobei sich keine weitere Verbesserung ergibt, d.h. Platz p_1 gibt die Transitionen t_2 und t_4 frei. Der gesamte Nutzen dieser Freigabe beträgt $N = 9$. Es ist leicht nachzuvollziehen, dass hiermit der maximal mögliche Gesamtnutzen für eine Freigabe bzgl. p_1 erzielt ist.

5.3 MCPN mit Konfliktlösung

Nun soll hier in Analogie zu den PN aus den lokalen Freigabemengen eine globale Freigabemenge entstehen. Dazu wird auch hier jedem Platz $p \in P$ eine Konfliktlösung zugeordnet, so dass man ein MCPN mit Konfliktlösung erhält (vgl. Abschn. 2.2.3).

Definition 5.9 (vgl. Def. 2.11):

Das Tupel $(P, T, F, B, f, A, K, k, \mathbf{z}, \alpha, \delta, \zeta)$ heißt ein **MCPN mit Konfliktlösung**, wenn gilt:

- $(P, T, F, B, f, A, K, k, \mathbf{z})$ ist ein MCPN im Zustand \mathbf{z}
- Die Abbildung $\alpha : P \rightarrow \{\text{Prio}, \text{W'keit}, \text{Nutzen}\}$ ist eine **Bewertungsartfunktion**, welche jedem Platz $p \in P$ eine bestimmte Art von Bewertung zuordnet.

- Die Abbildung $\delta : F \rightarrow \left\{ \begin{array}{ll} \mathbb{N} & \text{für } \alpha(p) = \text{Prio} \\ [0, 1] & \text{für } \alpha(p) = \text{W'keit} \\ \mathbb{R}_{\geq} & \text{für } \alpha(p) = \text{Nutzen} \end{array} \right\}$ ist eine **Bewertungsfunktion**, welche jedem Pfeil von einem Platz $p \in P$ zu einer Transition $t \in p \bullet$ eine Bewertung $\delta(p, t)$ entsprechend der Bewertungsart des Platzes und unter den folgenden Bedingungen zuordnet:

– dass jede Priorität nur einmal von jedem Platz p benutzt wird

$$\delta(p, t_j) \neq \delta(p, t_k) \quad \forall t_j, t_k \in p \bullet \quad \text{mit } t_j \neq t_k \text{ falls } \alpha(p) = \text{Prio},$$

– dass die Summe der Wahrscheinlichkeiten gleich 1 ist

$$\sum_{t \in p \bullet} \delta(p, t) = 1 \quad \forall p \text{ falls } \alpha(p) = \text{W'keit}.$$

Die Plätze aus Abb. 5.15 haben folgende Lösungsstrategien:

$\zeta(p_1) =$ lokales Prioritäten-Verfahren

$\zeta(p_2) =$ lokales W'keiten-Verfahren

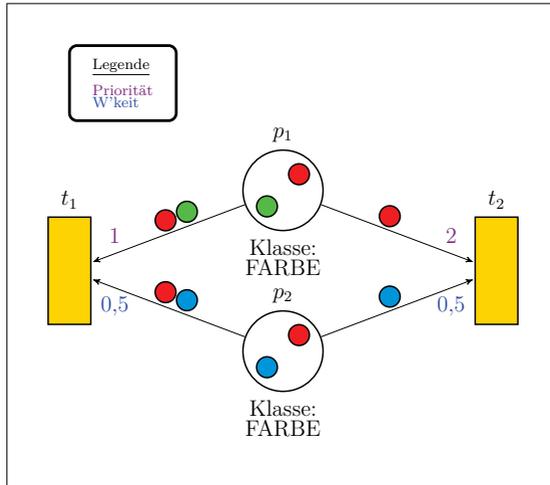


Abbildung 5.15: Beispiel für ein MCPN mit Konfliktlösung

In Abb. 5.15 hat sowohl p_1 als auch p_2 einen allgemeinen Konflikt. Die Freigabe erfolgt bei p_1 nach Prioritäten und bei p_2 nach Wahrscheinlichkeiten.

Es können nun zwei Fälle eintreten.

1. Fall:

p_1 wird t_1 freigegeben: $TE_{out}(p_1) = \{t_1\}$.

p_2 wird t_1 freigegeben: $TE_{out}(p_2) = \{t_1\}$.

Daraus ergibt sich die Freigabemenge $TE = \{t_1\}$, weil alle Plätze im Vorbereich der Transition t_1 die Freigabe erteilt haben.

2. Fall:

p_1 wird t_1 freigegeben: $TE_{out}(p_1) = \{t_1\}$.

p_2 wird t_2 freigegeben: $TE_{out}(p_2) = \{t_2\}$.

$\Rightarrow TE = \emptyset$, weil keine Transition von allen Plätzen in ihrem Vorbereich freigegeben wird.

Dieses Beispiel zeigt, dass trotz der Konfliktlösung immer noch der Fall eintreten kann,

dass keine Transition feuerbar ist.

Am Ende von Kapitel 2 wurde veranschaulicht, dass kollektive Lösungsansätze zur Behandlung von Konflikten bei Feuerprozessen in Petri-Netzen durch Faltung (und „Einfärben“) auf lokale Lösungsansätze zur Behandlung von Konflikten in gefärbten Petri-Netzen überführt werden können. Entsprechend lassen sich prinzipiell kollektive Lösungsansätze zur Behandlung von Konflikten bei Feuerprozessen in gefärbten Petri-Netzen durch Faltung wieder auf lokale Lösungsansätze abbilden. Daher wird auf eine ausführliche Ausarbeitung an dieser Stelle verzichtet.

6 Gefärbte Petri-Netze mit Variablen

In der Praxis werden zumeist CPN mit variablen Pfeilgewichtungen verwendet (vgl. [JK09], [Ger04]). Die dazu eingeführten Variablen in den CPN lassen sich auf ähnliche Weise durch die Faltung herleiten wie die Farben der Tokens (vgl. Kap. 4). Dies soll am folgenden Beispiel demonstriert werden.

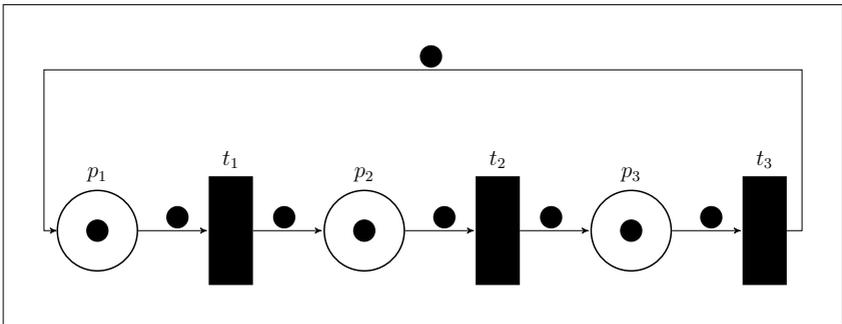


Abbildung 6.1: Nicht gefaltetes Petri-Netz

Das Petri-Netz in Abb. 6.1 stellt in seiner Graphenstruktur offensichtlich einen Zyklus dar. Durch „Konzentration“ der Plätze und der Transitionen entsteht eine vereinfachte Petri-Schlinge, indem die Plätze p_1 , p_2 und p_3 zu einem „konzentrierten“ Platz p_c (vgl. Abb. 6.2) und die Transitionen t_1 , t_2 und t_3 zu einer „konzentrierten“ Transition t_c „zusammengefaltet“ werden (siehe Abb. 6.3). Um bei diesem Faltvorgang die wesentlichen Informationen über die Struktur des ursprünglichen (nicht gefalteten) Petri-Netzes zu erhalten, werden die Token aus den Plätzen p_1 , p_2 und p_3 und an den zugehörigen Pfeilen von p_i nach t_i ($i = 1, 2, 3$) nach dem Schema

- $p_1 \rightarrow$ ROT
- $p_2 \rightarrow$ BLAU
- $p_3 \rightarrow$ GRÜN

6 Gefärbte Petri-Netze mit Variablen

gefärbt; man erhält das MCPN aus Abb. 6.2.

Um dabei die Reihenfolge im Zyklus des ursprünglichen Petri-Netzes erkenntlich zu erhalten, werden die Token an den Pfeilen (t, p) mit der jeweiligen Farbe des Platzes aus dem Nachbereich t gefärbt. Damit lässt sich serielles Feuern im MCPN (Abb. 6.2) auch umkehrbar als serielles Feuern im Petri-Netz (Abb. 6.1) nachvollziehen. Beispielsweise wird beim Feuern von t_2 im MCPN dem Platz p_c ein blaues Token entzogen, dafür ein grünes Token hinzugefügt. Die neue Markierung des gefärbten Platzes p_c (1 rotes und 2 grüne Token) entspricht im nicht gefärbten Petri-Netz dem analogen Zustand, dass p_1 ein Token, p_2 kein Token und p_3 zwei Token hat (vgl. Abb. 6.2).

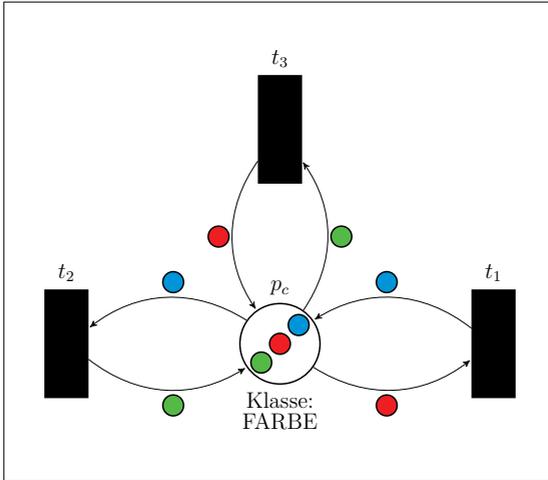


Abbildung 6.2: Zu einem MCPN gefaltetes PN

Im nächsten Faltungsschritt werden nun auch die Transitionen zusammengefasst. Es entsteht das gefaltete CPN mit Variablen in Abb. 6.3. Dabei ist zu beachten, dass die Pfeilgewichtungen der ehemals einzelnen Transitionen verschieden voneinander sind, was dazu führt, dass bei der Faltung der Transitionen zwangsläufig Variablen als Pfeilgewichtungen zu verwenden sind.

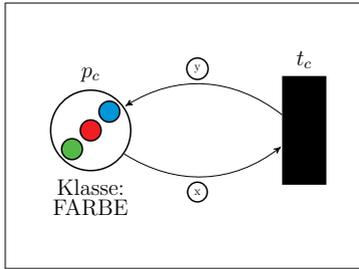


Abbildung 6.3: Gefaltetes CPN mit Variablen

Alle Variablen auf den mit einer Transition positiv oder negativ inzidenten Pfeilen bilden die *Variablenmenge* $Var(t)$ der Transition t . Im obigen Beispiel ist dies

$$Var(t_c) = \{x, y\}.$$

Nun werden diesen Variablen in passender Weise Ausprägungen aus der Farbenmenge zugeordnet. Eine solche Zuordnung nennt man *Modus* (vgl. [Rei10]). Sie ordnet jeder Variablen auf den Pfeilen einer Transition eine Ausprägung zu. Wie man aber erkennt, dürften im oberen Beispiel nur bestimmte Modi zugelassen werden, um die Äquivalenz zum ursprünglichen Petri-Netz zu erhalten. Diese grundsätzlich möglichen Modi einer Transition werden in der *Modus-Menge* $m(t)$ zusammengefasst. Die Modus-Menge für das obige Beispiel ist:

$$m(t_c) = \{\beta_1, \beta_2, \beta_3\} \text{ mit } \beta_i : \{x, y\} \rightarrow \{\text{ROT, BLAU, GRÜN}\} \text{ für } i = 1, 2, 3,$$

wobei konkret gilt:

$$\begin{aligned} \beta_1(x) &= \text{ROT}, & \beta_1(y) &= \text{BLAU}, \\ \beta_2(x) &= \text{BLAU}, & \beta_2(y) &= \text{GRÜN}, \\ \beta_3(x) &= \text{GRÜN}, & \beta_3(y) &= \text{ROT}. \end{aligned}$$

Dabei erweist sich die folgende Schreibweise in Anlehnung an [Rei10, S. 28] für die einzelnen Modi als praktikabel:

$$\begin{aligned} \beta_1 &: \{x = \text{ROT}, y = \text{BLAU}\}, \\ \beta_2 &: \{x = \text{BLAU}, y = \text{GRÜN}\}, \\ \beta_3 &: \{x = \text{GRÜN}, y = \text{ROT}\}. \end{aligned}$$

Soll nun die Transition t_c gefeuert werden, müsste zuvor ein Modus $\beta_i \in m(t_c)$ ausgewählt werden. Als Beispiel soll nun die Transition t_c im Modus β_1 gefeuert werden.

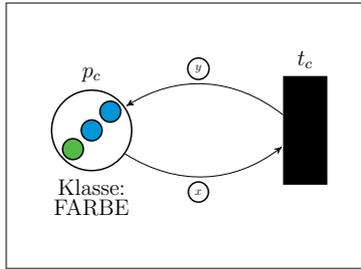


Abbildung 6.4: Zustand des CPN nach dem Feuern von Transition t_c im Modus β_1

Es ist einfach nachzuvollziehen, dass diese konkrete Moduswahl dem Feuern der Transition t_1 im ursprünglichen Petri-Netz entspricht (vgl. Abb. 6.1).

An dieser Stelle entsteht nun die Problematik, ob eine Transition überhaupt in einem bestimmten Modus feuerbar ist oder ob überhaupt ein feuerbarer Modus¹ existiert. Wie man im oberen Beispiel erkennt, ist es nicht möglich, in diesem Zustand die Transition t_c ein weiteres Mal im Modus β_1 zu feuern.

Dieses grundlegende Konzept der CPN mit Variablen soll nun im ersten Abschnitt dieses Kapitels behandelt und formal definiert werden. Der zweite Abschnitt befasst sich mit der Modus-Wahl und ihrer Problematik. Hierbei wird aus der sehr umfangreichen Thematik nur ein begrenzter Teilbereich näher behandelt und ein Ausblick auf weiterführende Fragestellungen gegeben (vgl. auch [KD13]).

6.1 CPN mit einfachen Variablen

Definition 6.1:

Ein **gefärbtes Petri-Netz mit Variablen** ist ein Tupel $N = (P, T, F, B, f, K, \mathbb{V}, k)$, wobei

- das Tupel (P, T, F, B) ein Netz gemäß Def. 1.1 bildet;
- $K = \{K_1, \dots, K_x\}$ eine endliche, nichtleere Menge von sog. **Klassen** K_i ist, wobei die Klassen K_i ihrerseits endliche, nichtleere Mengen sind mit $K_i = \{a_1^i, \dots, a_{|K_i|}^i\}$,

¹Die Ausdrucksweise, dass ein Modus feuerbar ist, mag zunächst ungewöhnlich erscheinen, da ja eigentlich eine Transition in einem Modus feuert. Wenn man aber bedenkt, dass ein Modus einer Transition im ursprünglichen PN einer bestimmten Transition entspricht, ist diese Ausdrucksweise durchaus eine angebrachte Sprechweise.

$i = 1, \dots, x$, wobei die Elemente der Klassen auch **Ausprägungen** der jeweiligen Klasse genannt werden; dabei bildet $\mathbb{K} := \bigcup_{K_i \in K} K_i$ die **Vereinigungsmenge aller Klassen**;

- $\mathbb{V} = \{v_1, \dots, v_n\}$ eine endliche Menge von sog. **Variablen** v_n ist;
- $k : (P \cup \mathbb{V}) \rightarrow K$ eine Abbildung ist (**Klassifizierungsfunktion**), die jedem Platz $p \in P$ und jeder Variable $v_n \in \mathbb{V}$ eine Klasse aus der Klassenmenge K zuweist;
- $f : (F \cup B) \rightarrow \mathbb{M}(\mathbb{K} \cup \mathbb{V})$ eine Abbildung ist (**Gewichtungsfunktion**), die jedem Pfeil $(p, t) \in F$ bzw. $(t, p) \in B$ eine der jeweiligen Klasse von p entsprechende Multimenge als **Gewicht** zuweist, indem $f(p, t)$ bzw. $f(t, p)$ eine Multimenge M in der Form $M = M_c \uplus M_v$ mit $M_c = \{(a, h(a)) \mid a \in \mathbb{K}, h(a) \in \mathbb{N}_0\}$ und $M_v = \{(a, h(a)) \mid a \in \mathbb{V}, h(a) \in \mathbb{N}_0\}$ ist, wobei für die Stützmenge von M_c gilt:

$$\text{supp}(M_c) \subseteq k(p)$$

und $\forall a \in \text{supp}(M_v)$ gilt:

$$k(a) = k(p).$$

An dieser Stelle sei angemerkt, dass für gefärbte Petri-Netze mit Variablen eine Klasse nur aus einem Attribut besteht und daher hier Einfachheit halber die Begrifflichkeit der Attribute weggelassen wird und Attribute direkt als Klasse beschrieben werden. Es wäre aber prinzipiell auch möglich, die Definition mit Attributen zu beschreiben (vgl. Def. 4.4).

Beispiel:

In diesem Beispiel wird schrittweise ein gefärbtes Petri-Netz mit Variablen $(P, T, F, B, f, K, \mathbb{V}, k)$ konstruiert:

1. Das Netz $N = (P, T, F, B)$ mit:

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$

$$T = \{t_1, t_2\}$$

$$F = \{(p_1, t_1), (p_2, t_1), (p_3, t_2)\}$$

$$B = \{(t_1, p_3), (t_2, p_4), (t_2, p_5)\}$$

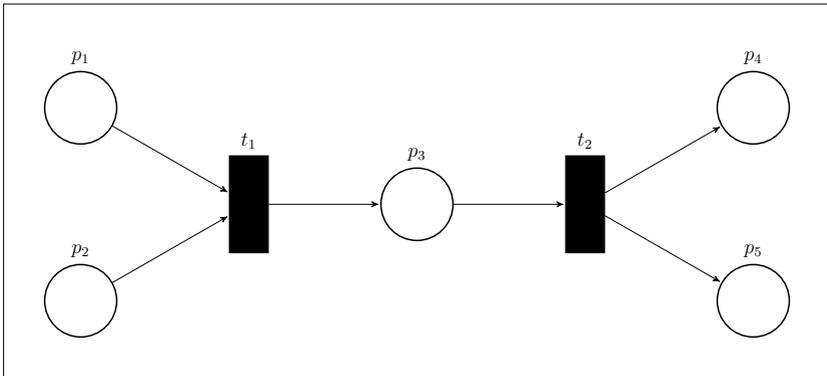


Abbildung 6.5: Beispiel eines Netzes

2. Die Klassen aus $K = \{\text{FARBE}, \text{ZAHL}\}$ mit:

$\text{FARBE} = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$

$\text{ZAHL} = \{1, 2, 3\}$

3. Die Variablen aus $\mathbb{V} = \{n, x, y, z\}$.

4. Die Klassifizierung der Plätze und Variablen:

Plätze

$k(p_1) = \text{FARBE}$

$k(p_2) = \text{FARBE}$

$k(p_3) = \text{FARBE}$

$k(p_4) = \text{ZAHL}$

$k(p_5) = \text{FARBE}$

Variablen

$k(n) = \text{ZAHL}$

$k(x) = \text{FARBE}$

$k(y) = \text{FARBE}$

$k(z) = \text{FARBE}$

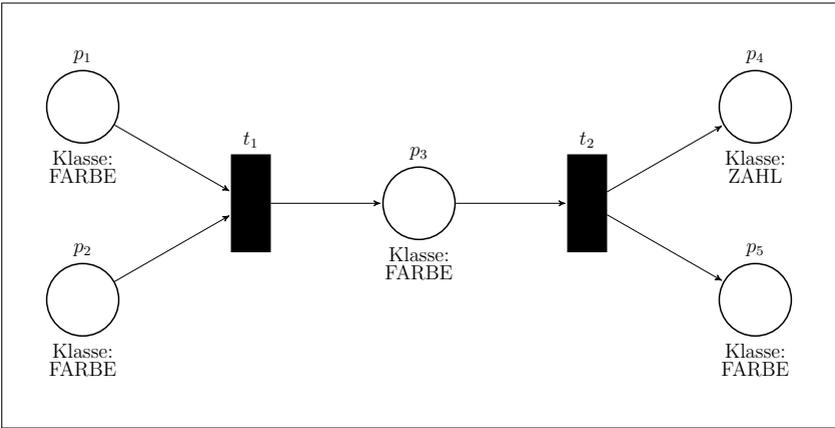


Abbildung 6.6: Beispiel eines Netzes mit Klassen

5. Die Gewichtung der Pfeile

$$\begin{aligned}
 f(p_1, t_1) &\equiv \{(x, 1), (y, 1)\} \\
 f(p_2, t_1) &\equiv \{(x, 1), (z, 1)\} \\
 f(t_1, p_3) &\equiv \{(x, 2), (y, 1), (z, 1)\} \\
 f(p_3, t_2) &\equiv \{(x, 3), (\text{ROT}, 1)\} \\
 f(t_2, p_4) &\equiv \{(n, 3)\} \\
 f(t_2, p_5) &\equiv \{(x, 1)\}
 \end{aligned}$$

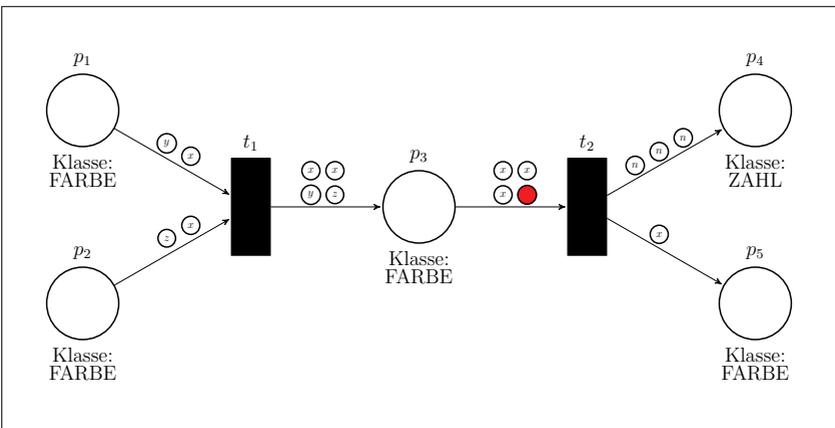


Abbildung 6.7: Beispiel eines mit Variablen gewichteten CPN

Definition 6.2:

Es sei ein gefärbtes Petri-Netz mit Variablen $N = (P, T, F, B, f, K, \mathbb{V}, k)$ gegeben. Eine Abbildung $\mathbf{z} : P \rightarrow \mathbb{M}(\mathbb{K})$, die jedem Platz $p \in P$ eine der Klasse $k(p)$ entsprechende Multimenge als Gewicht zuweist, heißt **Zustand** oder **Markierung** von N , wobei \mathbf{z} eindeutig als Vektor $\mathbf{z} = (z_1, \dots, z_m)$ mit $z_i = \mathbf{z}(p_i)$ darstellbar ist, d.h. $\mathbf{z}(p_i)$ ist eine Multimenge M_i in der Form $M_i = \{(a, h_i(a)) \mid a \in \mathbb{K}, h_i(a) \in \mathbb{N}_0\}$, wobei für die Stützmenge gilt: $\text{supp}(M_i) \subseteq k(p_i)$.

Beispiel:

Ein Beispiel einer Markierung des CPN mit Variablen von oben ist:

$$\begin{aligned} \mathbf{z}(p_1) = z_1 &\equiv \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \mathbf{z}(p_5) = z_5 &\equiv \{(\text{ROT}, 2), (\text{BLAU}, 3), (\text{GRÜN}, 1)\} \\ \mathbf{z}(p_2) = z_2 &\equiv \{(\text{GRÜN}, 1)\} \\ \mathbf{z}(p_4) = z_4 &\equiv \emptyset \\ \mathbf{z}(p_3) = z_3 &\equiv \{(\text{BLAU}, 1)\} \end{aligned}$$

oder als Vektor:

$$\mathbf{z} \equiv \begin{pmatrix} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 3), (\text{GRÜN}, 1)\} \\ \{(\text{GRÜN}, 1)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{pmatrix}$$

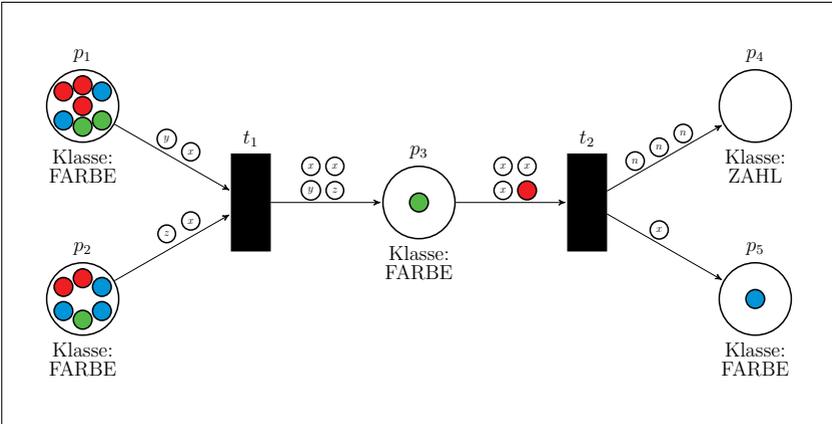


Abbildung 6.8: Beispiel einer Markierung eines CPN mit Variablen

Definition 6.3:

Es sei ein gefärbtes Petri-Netz mit Variablen $N = (P, T, F, B, f, K, \mathbb{V}, k)$ gegeben. Eine Abbildung $Var : T \rightarrow \wp(\mathbb{V})^1$, die jeder Transition $t \in T$ eine Menge von Variablen $Var(t) \subseteq \mathbb{V}$ zuordnet, so dass alle Variablen, die auf den positiv oder negativ inzidenten Pfeilen von t als Gewicht stehen, diese Menge bilden, d.h.

$$Var(t) = \left(\bigcup_{f(p,t) \text{ mit } p \in \bullet t} \text{supp}(f(p,t)) \cup \bigcup_{f(t,p) \text{ mit } p \in t \bullet} \text{supp}(f(t,p)) \right) \cap \mathbb{V}.$$

Die Menge $Var(t)$ bezeichnet man als die **Variablenmenge** von t , entsprechend heißen die Elemente von $Var(t)$ die **Variablen** von t .

Man beachte, dass mit obiger Definition gewährleistet ist, dass die Menge $Var(t)$ nur Variablen, jedoch keine Konstanten enthält.

Beispiel:

Für das Beispiel aus Abb. 6.7 sind die Variablen der Transition t_1

$$Var(t_1) = \{x, y, z\}$$

und von Transition t_2

$$Var(t_2) = \{x, n\}.$$

Definition 6.4:

Es sei ein gefärbtes Petri-Netz mit Variablen $N = (P, T, F, B, f, K, \mathbb{V}, k)$ gegeben. Eine Abbildung $\beta : \mathbb{V} \rightarrow \mathbb{K}$, die jeder Variablen $v \in \mathbb{V}$ eine Ausprägung $a \in \mathbb{K}$ zuordnet, so dass $a \in k(v)$ gilt, bezeichnet man als **Modus**.

Beispiel:

Für das Beispiel aus Abb. 6.7 können insgesamt die folgenden 81 Modi gebildet werden:

¹Mit $\wp(\mathbb{V})$ ist die Potenzmenge von \mathbb{V} bezeichnet.

Tabelle 6.1: Modus-Menge mit Mächtigkeit 81

β	x	y	z	n	β	x	y	z	n	β	x	y	z	n
1	ROT	ROT	ROT	1	28	ROT	ROT	ROT	2	55	ROT	ROT	ROT	3
2	ROT	ROT	BLAU	1	29	ROT	ROT	BLAU	2	56	ROT	ROT	BLAU	3
3	ROT	ROT	GRÜN	1	30	ROT	ROT	GRÜN	2	57	ROT	ROT	GRÜN	3
4	ROT	BLAU	ROT	1	31	ROT	BLAU	ROT	2	58	ROT	BLAU	ROT	3
5	ROT	BLAU	BLAU	1	32	ROT	BLAU	BLAU	2	59	ROT	BLAU	BLAU	3
6	ROT	BLAU	GRÜN	1	33	ROT	BLAU	GRÜN	2	60	ROT	BLAU	GRÜN	3
7	ROT	GRÜN	ROT	1	34	ROT	GRÜN	ROT	2	61	ROT	GRÜN	ROT	3
8	ROT	GRÜN	BLAU	1	35	ROT	GRÜN	BLAU	2	62	ROT	GRÜN	BLAU	3
9	ROT	GRÜN	GRÜN	1	36	ROT	GRÜN	GRÜN	2	63	ROT	GRÜN	GRÜN	3
10	BLAU	ROT	ROT	1	37	BLAU	ROT	ROT	2	64	BLAU	ROT	ROT	3
11	BLAU	ROT	BLAU	1	38	BLAU	ROT	BLAU	2	65	BLAU	ROT	BLAU	3
12	BLAU	ROT	GRÜN	1	39	BLAU	ROT	GRÜN	2	66	BLAU	ROT	GRÜN	3
13	BLAU	BLAU	ROT	1	40	BLAU	BLAU	ROT	2	67	BLAU	BLAU	ROT	3
14	BLAU	BLAU	BLAU	1	41	BLAU	BLAU	BLAU	2	68	BLAU	BLAU	BLAU	3
15	BLAU	BLAU	GRÜN	1	42	BLAU	BLAU	GRÜN	2	69	BLAU	BLAU	GRÜN	3
16	BLAU	GRÜN	ROT	1	43	BLAU	GRÜN	ROT	2	70	BLAU	GRÜN	ROT	3
17	BLAU	GRÜN	BLAU	1	44	BLAU	GRÜN	BLAU	2	71	BLAU	GRÜN	BLAU	3
18	BLAU	GRÜN	GRÜN	1	45	BLAU	GRÜN	GRÜN	2	72	BLAU	GRÜN	GRÜN	3
19	GRÜN	ROT	ROT	1	46	GRÜN	ROT	ROT	2	73	GRÜN	ROT	ROT	3
20	GRÜN	ROT	BLAU	1	47	GRÜN	ROT	BLAU	2	74	GRÜN	ROT	BLAU	3
21	GRÜN	ROT	GRÜN	1	48	GRÜN	ROT	GRÜN	2	75	GRÜN	ROT	GRÜN	3
22	GRÜN	BLAU	ROT	1	49	GRÜN	BLAU	ROT	2	76	GRÜN	BLAU	ROT	3
23	GRÜN	BLAU	BLAU	1	50	GRÜN	BLAU	BLAU	2	77	GRÜN	BLAU	BLAU	3
24	GRÜN	BLAU	GRÜN	1	51	GRÜN	BLAU	GRÜN	2	78	GRÜN	BLAU	GRÜN	3
25	GRÜN	GRÜN	ROT	1	52	GRÜN	GRÜN	ROT	2	79	GRÜN	GRÜN	ROT	3
26	GRÜN	GRÜN	BLAU	1	53	GRÜN	GRÜN	BLAU	2	80	GRÜN	GRÜN	BLAU	3
27	GRÜN	GRÜN	GRÜN	1	54	GRÜN	GRÜN	GRÜN	2	81	GRÜN	GRÜN	GRÜN	3

Dabei sind natürlich nicht unbedingt alle Modi feuerbar.

Definition 6.5:

Es sei ein gefärbtes Petri-Netz mit Variablen $N = (P, T, F, B, f, K, \mathbb{V}, k)$ gegeben, zudem ein Modus β und eine Transition $t \in T$. Dann bezeichnet man mit

$$f_\beta(p, t) = \{(a, h(a)) \mid (a, h(a)) \in f(p, t) \wedge a \in k(p)\} \uplus \{(\beta(a), h(a)) \mid (a, h(a)) \in f(p, t) \wedge a \in \text{Var}(t)\}$$

bzw.

$$f_{\beta}(t, p) = \{(a, h(a)) \mid (a, h(a)) \in f(t, p) \wedge a \in k(p)\} \uplus \\ \{(\beta(a), h(a)) \mid (a, h(a)) \in f(t, p) \wedge a \in \text{Var}(t)\}$$

ein **Pfeilgewicht im Modus β bzgl. Transition t** .

Beispiel:

Für das Beispiel aus Abb. 6.7 ist das Pfeilgewicht

$$f(t_1, p_3) = \{(x, 2), (y, 1), (z, 1)\}$$

im Modus

$$\beta_6 : \{x = \text{ROT}, y = \text{BLAU}, z = \text{GRÜN}, n = 1\}$$

gegeben als

$$f_{\beta_6}(t_1, p_3) = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\}.$$

Man beachte, dass man ein MCPN erhält, wenn für jede Transition genau ein Modus gewählt wird und dadurch alle zugehörigen Pfeile in diesen Modi sind. Wenn nur für eine Transition ein Modus gewählt wird, ergibt sich auf kanonische Weise ein durch β und t **induziertes MCPN**, kurz (β, t) -MCPN.

Beispiel:

Für das Beispiel aus Abb. 6.7 sei der Modus

$$\beta_6 : \{x = \text{ROT}, y = \text{BLAU}, z = \text{GRÜN}, n = 1\}$$

für Transition t_1 gewählt, daher erhält man das folgende (β_6, t_1) -MCPN:

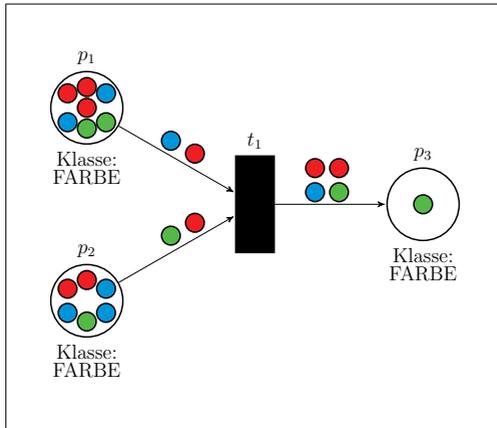


Abbildung 6.9: Durch Modus-Wahl entstandenes (β_6, t_1) -MCPN aus CPN mit Variablen in Abb. 6.8

Nun soll den Transitionen eines CPN mit Variablen eine Menge von Modi mitgegeben werden, um nur bestimmte Zuordnungen zu erlauben, ähnlich wie in dem einleitenden Beispiel dieses Kapitels. Dafür sei hier nun ein CPN mit Variablen mit einer solchen Modus-Menge definiert.

Definition 6.6:

Ein *gefärbtes Petri-Netz mit Variablen und Modus-Menge* ist ein Tupel $N = (P, T, F, B, f, K, \mathbb{V}, k, \mathbb{B}, m)$, wobei

- das Tupel $(P, T, F, B, f, K, \mathbb{V}, k)$ ein CPN mit Variablen gemäß Def. 6.1 bildet;
- $\mathbb{B} = \{\beta_1, \dots, \beta_b\}$ eine endliche Menge von Modi ist und als *Modus-Menge von N* bezeichnet wird;
- $m : T \rightarrow \wp(\mathbb{B})$ eine Abbildung ist (*Modus-Menge einer Transition*), die jeder Transition $t \in T$ eine Teilmenge der Modusmenge \mathbb{B} zuweist.

Man beachte, dass eine Modus-Menge \mathbb{B} nicht unbedingt die Menge aller möglichen Modi darstellt, die über die Variablen \mathbb{V} gebildet werden kann, sondern durchaus eine Teilmenge sein darf.

Allerdings wird in der Regel eine solche (Gesamt-)Modus-Menge bei einem CPN nicht explizit angegeben, weil es sich dabei um die komplette Enumeration aller Kombinationen handelt. Dies sind im obigen Beispiel 81 Modi. Es ist daher einfacher, diese nicht explizit zu

benennen, zumal es - außer bei „kleinen“ Modus-Mengen - einfachere Methoden gibt, eine solche Beschränkung zu beschreiben (vgl. [JK09]).

Beispiel:

Für das Beispiel aus Abb. 6.7 könnte folgende Modus-Menge gegeben sein:

$$\mathbb{B} = \left\{ \begin{array}{l} \beta_1, \beta_2, \beta_3, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{19}, \beta_{20}, \beta_{21}, \\ \beta_{31}, \beta_{32}, \beta_{33}, \beta_{40}, \beta_{41}, \beta_{42}, \beta_{49}, \beta_{50}, \beta_{51}, \\ \beta_{61}, \beta_{62}, \beta_{63}, \beta_{70}, \beta_{71}, \beta_{72}, \beta_{79}, \beta_{80}, \beta_{81} \end{array} \right\}.$$

Daraus ist es möglich, folgende Modus-Mengen für die Transitionen t_1 und t_2 zu bilden: $m(t_1) = \{\beta_{12}, \beta_{20}, \beta_{33}, \beta_{49}, \beta_{62}, \beta_{71}\}$ bzw. $m(t_2) = \{\beta_1, \beta_{41}, \beta_{81}\}$.

Es sei an dieser Stelle noch angemerkt, dass man mit der Modus-Menge $m(t)$ einer Transition t bestimmte Bedingungen oder Abhängigkeiten modellieren kann. Zum Beispiel ist mit der Modus-Menge $m(t_1)$ die Bedingung erzeugt worden, dass alle Variablen unterschiedliche Werte annehmen müssen, demzufolge gilt:

$$x \neq y \neq z \neq x.$$

Zudem wird mit der Modus-Menge $m(t_2)$ eine bestimmte Abhängigkeit der Variable n von Variable x erzeugt:

$$\begin{aligned} x = \text{ROT} &\rightarrow n = 1 \\ x = \text{BLAU} &\rightarrow n = 2 \\ x = \text{GRÜN} &\rightarrow n = 3 \end{aligned}$$

Als nächstes soll nun auf das Feuern eines CPN mit Variablen eingegangen werden. Hierbei ist zu beachten, dass an dieser Stelle immer davon ausgegangen wird, dass ein Benutzer einen Modus für jede Transition auswählt und nicht durch eine Strategie eine automatische Auswahl getroffen wird wie in Abschnitt 6.2.

Definition 6.7:

Das Tupel $N = (P, T, F, B, f, K, \mathbb{V}, k, \mathbb{B}, m)$ sei ein gefärbtes Petri-Netz mit Variablen und Modus-Menge und \mathbf{z} eine Markierung von N .

- a) Eine Transition $t \in T$ von N heißt **aktiviert im Modus β** und Zustand \mathbf{z} , wenn $\beta \in m(t)$ ist und für alle $p \in \bullet t$ gilt: $f_\beta(p, t) \subseteq \mathbf{z}(p)$.

- b) Eine Transition $t \in T$ von N heißt **aktivierbar**, wenn ein $\beta \in m(t)$ existiert, womit t im Modus β aktiviert ist.
- c) Eine im Zustand \mathbf{z} und Modus β aktivierte Transition t wird kurz auch **(\mathbf{z}, β) -feuerbar** genannt.
- d) Das **Feuern** einer (\mathbf{z}, β) -feuerbaren Transition $t \in T$ von N ist der Übergang vom Zustand $\mathbf{z} = (z_1, \dots, z_m)$ in den Zustand $\mathbf{z}' = (z'_1, \dots, z'_m)$, wobei gilt:

$$z'_i = \left\{ \begin{array}{ll} (z_i \setminus f_\beta(p_i, t)) \uplus f_\beta(t, p_i) & \text{falls } p_i \in \bullet t \wedge p_i \in t \bullet \\ z_i \setminus f_\beta(p_i, t) & \text{falls } p_i \in \bullet t \wedge p_i \notin t \bullet \\ z_i \uplus f_\beta(t, p_i) & \text{falls } p_i \notin \bullet t \wedge p_i \in t \bullet \\ z_i & \text{falls } p_i \notin \bullet t \wedge p_i \notin t \bullet \end{array} \right\} \text{ für } i = 1, \dots, m.$$

Es sei angemerkt, dass man allgemein eine Transition, die aktiviert im Modus β ist, auch als **seriell feuerbar im Modus β** bezeichnet oder eine aktivierbare als **seriell feuerbar** bezeichnet. Des Weiteren ist der Begriff „aktivierbar“ eingeführt worden, damit auch Aussagen darüber möglich sind, ob eine Transition überhaupt aktiviert sein kann, ohne einen Modus spezifizieren zu müssen.

Beispiel:

Überprüft man die Transitionen des CPN mit Variablen aus Abb. 6.8 mit der Modus-Menge

$$\mathbb{B} = \left\{ \begin{array}{l} \beta_1, \beta_2, \beta_3, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{19}, \beta_{20}, \beta_{21}, \\ \beta_{31}, \beta_{32}, \beta_{33}, \beta_{40}, \beta_{41}, \beta_{42}, \beta_{49}, \beta_{50}, \beta_{51}, \\ \beta_{61}, \beta_{62}, \beta_{63}, \beta_{70}, \beta_{71}, \beta_{72}, \beta_{79}, \beta_{80}, \beta_{81} \end{array} \right\}$$

und den Modus-Mengen der Transitionen:

$$m(t_1) = \mathbb{B},$$

$$m(t_2) = \{\beta_1, \beta_{10}, \beta_{19}, \beta_{31}, \beta_{40}, \beta_{49}, \beta_{61}, \beta_{70}, \beta_{79}\},$$

dann ist:

- a) t_1 aktivierbar und somit seriell feuerbar, weil ein Modus β existiert, so dass für alle $p \in \bullet t_1$ gilt:

$$f_\beta(p, t_1) \subseteq \mathbf{z}(p).$$

Beispielsweise gilt für den Modus

$$\beta_{33} : \{x = \text{ROT}, y = \text{BLAU}, z = \text{GRÜN}, n = 2\}$$

die Bedingung:

$$f_{\beta_{33}}(p_1, t_1) = \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \subseteq \mathbf{z}(p_1),$$

$$f_{\beta_{33}}(p_2, t_1) = \{(\text{ROT}, 1), (\text{GRÜN}, 1)\} \subseteq \mathbf{z}(p_2).$$

Trotzdem können immer auch Modi existieren, in denen t_1 nicht feuerebar ist, z.B. im Modus

$$\beta_{81} : \{x = \text{GRÜN}, y = \text{GRÜN}, z = \text{GRÜN}, n = 3\},$$

denn es gilt:

$$f_{\beta_{81}}(p_2, t_1) = \{(\text{GRÜN}, 2)\} \not\subseteq \mathbf{z}(p_2).$$

b) t_2 nicht aktivierbar, weil kein Modus $\beta \in m(t_2)$ existiert, bei dem für alle $p \in \bullet t_2$ gilt:

$$f_{\beta}(p, t_2) \subseteq \mathbf{z}(p).$$

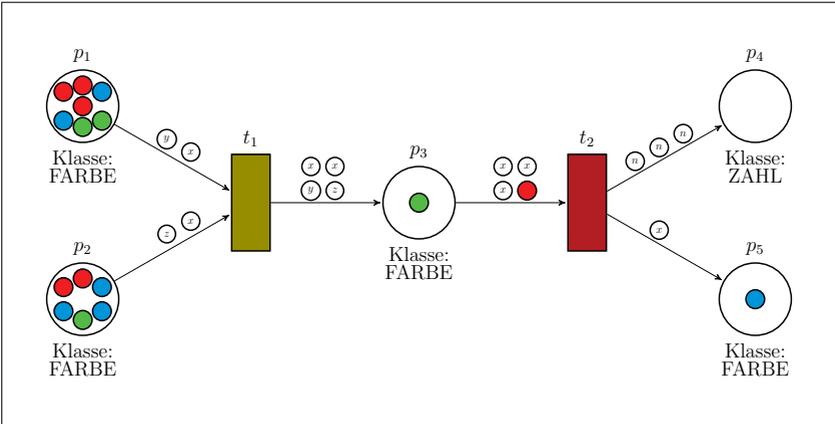


Abbildung 6.10: Feuerbarkeit des CPN mit Variablen

Wird nun t_1 im Modus β_{33} gefeuert, wird aus Zustand

$$\mathbf{z} \equiv \begin{pmatrix} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 3), (\text{GRÜN}, 1)\} \\ \{(\text{GRÜN}, 1)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{pmatrix}$$

6 Gefärbte Petri-Netze mit Variablen

in den Zustand

$$\begin{aligned}
 \mathbf{z}' &\equiv \begin{pmatrix} \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 2)\} \setminus \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 3), (\text{GRÜN}, 1)\} \setminus \{(\text{ROT}, 1), (\text{GRÜN}, 1)\} \\ \{(\text{GRÜN}, 1)\} \uplus \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{pmatrix} \\
 &= \begin{pmatrix} \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\} \\ \{(\text{ROT}, 1), (\text{BLAU}, 3)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{pmatrix}
 \end{aligned}$$

übergegangen.

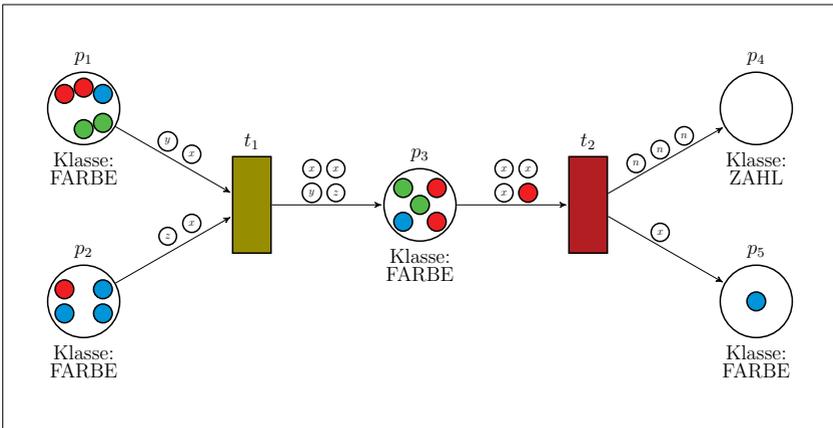


Abbildung 6.11: Zustandsübergang beim Feuern der Transition t_1 im Modus β_{33}

In diesem Zustand wäre t_1 immer noch aktivierbar, aber t_2 wäre dies noch nicht.

Feuert man nun t_1 erneut, diesmal aber im Modus

$$\beta_{71} : \{x = \text{BLAU}, y = \text{GRÜN}, z = \text{BLAU}, n = 3\},$$

dann erhält man den Zustand

$$\mathbf{z}'' \equiv \begin{pmatrix} \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \\ \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 4), (\text{GRÜN}, 3)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{pmatrix}.$$

Nun wäre sowohl t_1 als auch t_2 aktivierbar.

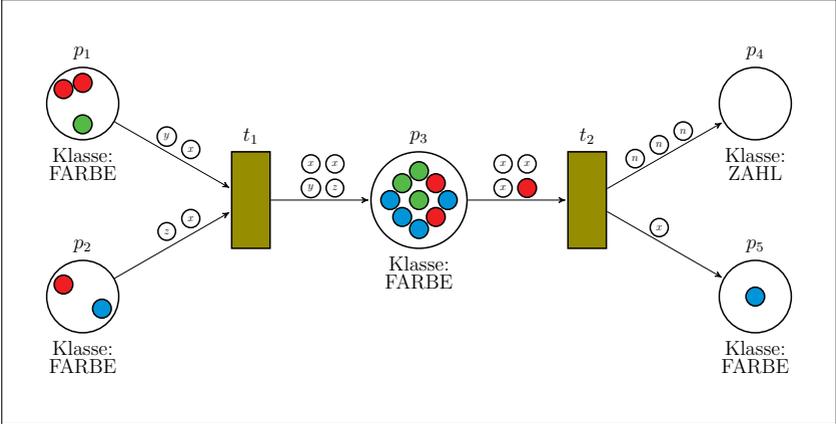


Abbildung 6.12: Feuern der Transition t_1 im Modus β_{71} und Zustand \mathbf{z}'

Die Zustandsübergänge lassen sich wie folgt beschreiben (vgl. Abschn. 1.2.1):

Die Symbolik $\mathbf{z} \langle (t, \beta) \rangle \mathbf{z}'$ bedeutet, dass die Transition t eines im Zustand \mathbf{z} vorliegenden CPN im Modus β aktiviert ist und nach dem Feuern der Transition t in den Zustand \mathbf{z}' übergeht. Man sagt auch kurz: t feuert im Modus β von \mathbf{z} nach \mathbf{z}' .

Mittels dieser Schreibweise lassen sich nun auch allgemeinere Zustandsübergangsketten beschreiben, die man auch als (*serielle*) **Rechnungen** bezeichnet (vgl. Abschn. 1.2.1):

$$\mathbf{z}^1 \langle (t_{j_1}, \beta_{x_1}) \rangle \mathbf{z}^2 \langle (t_{j_2}, \beta_{x_2}) \rangle \mathbf{z}^3 \dots \mathbf{z}^r \langle (t_{j_r}, \beta_{x_r}) \rangle \mathbf{z}^{r+1},$$

wobei $j_k \in \{1, \dots, m\}$ und $x_k \in m(t_{j_k})$ für $k = 1, \dots, r$ gilt. Man beachte, dass sichergestellt ist, dass jede Transition t_{j_k} auch (z^k, β_{x_k}) -feuerbar ist, und eine Transition mehrfach in einer Rechnung vorkommen kann. Man mache sich ebenfalls klar, dass jede in einer Rechnung vorkommende Transition erst dann aktiviert sein muss, wenn ihr Feuern ansteht.

Beispiel:

Die Rechnung zum obigen Beispiel ist:

$$\mathbf{z}\langle(t_1, \beta_{33})\rangle\mathbf{z}'\langle(t_1, \beta_{71})\rangle\mathbf{z}''.$$

Neben dem seriellen Feuern, bei dem selbst bei seriellen Rechnungen immer nur genau eine aktivierte Transition bzgl. eines Zustandes \mathbf{z} für einen Zustandsübergang herangezogen wird, soll nun auch gleichzeitiges (*nebenläufiges*) Feuern von mehreren Transitionen formalisiert werden. Hierzu muss der Begriff der Feuerbarkeit verallgemeinert werden.

Definition 6.8 (vgl. Def. 1.6 bzw. Def. 4.9):

Das Tupel $N = (P, T, F, B, f, K, \mathbb{V}, k, \mathbb{B}, m)$ sei ein gefärbtes Petri-Netz mit Variablen und Modus-Menge und \mathbf{z} eine Markierung von N und $\{t_1, \dots, t_r\}$ eine Menge von Transitionen¹. Mit $\bullet\{t_1, \dots, t_r\}$ werde der gemeinsame Vorbereich der Transitionen t_1, \dots, t_r bezeichnet (vgl. Def. 1.6). Des Weiteren ist eine Menge von Modi $\{\beta_{t_1}, \dots, \beta_{t_r}\}$ gegeben, so dass gilt: $\beta_{t_k} \in m(t_k)$ für $k = 1, \dots, r$. Die Transitionenmenge $\{t_1, \dots, t_r\}$ heißt **in den Modi $\{\beta_{t_1}, \dots, \beta_{t_r}\}$ nebenläufig feuierbar** im Zustand \mathbf{z} , wenn für alle Transitionen $t \in \{t_1, \dots, t_r\}$ gilt:

$$f_{\beta_t}(p, t) \subseteq \mathbf{z}(p) \text{ für alle } p \in \bullet t,$$

wodurch die Transitionen in ihrem jeweiligen Modus β_t aktiviert sind² und für alle $p \in \bullet\{t_1, \dots, t_r\}$ gilt:

$$\biguplus_{t \in \{t_1, \dots, t_r\} \cap p \bullet} f_{\beta_t}(p, t) \subseteq \mathbf{z}(p).$$

Der neue Zustand z'_i eines Platzes $p_i \in P$ wird nach der folgenden Gleichung bestimmt:

$$z'_i = \left(z_i \setminus \biguplus_{t \in \{t_1, \dots, t_r\} \cap p_i \bullet} f_{\beta_t}(p_i, t) \right) \uplus \biguplus_{t \in \{t_1, \dots, t_r\} \cap \bullet p_i} f_{\beta_t}(t, p_i).$$

¹Um eine Doppelindizierung zu vermeiden, wird o.B.d.A. angenommen, dass es sich um die r ersten Transitionen aus T handelt.

²Gemäß Def. 6.7.

Beispiel:

Betrachtet man das obige Beispiel eines CPN mit Variablen im Zustand \mathbf{z}'' , ist es offensichtlich, dass die Transitionen t_1 und t_2 aktivierbar sind:

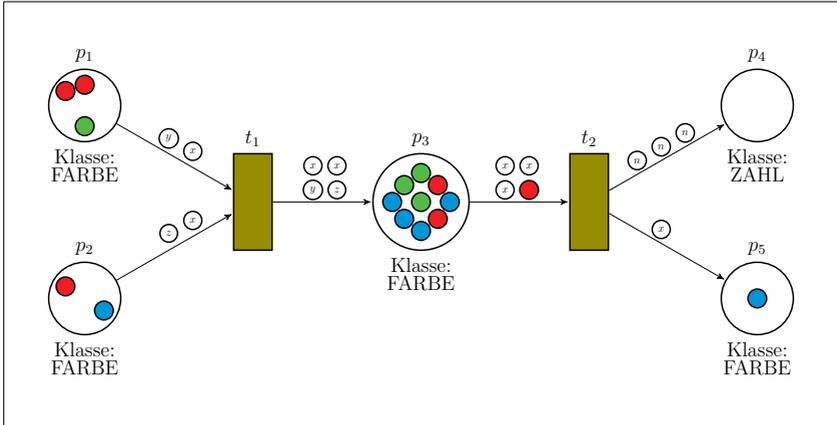


Abbildung 6.13: CPN mit Variablen im Zustand \mathbf{z}''

Nun sollen t_1 und t_2 nebenläufig gefeuert werden. Dazu wird für t_1 der Modus

$$\beta_{62} : \{x = \text{ROT}, y = \text{GRÜN}, z = \text{BLAU}, n = 3\}$$

und für t_2 der Modus

$$\beta_{25} : \{x = \text{GRÜN}, y = \text{GRÜN}, z = \text{ROT}, n = 1\}$$

gewählt. Es ist offensichtlich, dass dadurch die Bedingung des nebenläufigen Feuerns erfüllt ist; es wird aus dem Zustand

$$\mathbf{z}'' \equiv \left(\begin{array}{c} \{(\text{ROT}, 2), (\text{GRÜN}, 1)\} \\ \{(\text{ROT}, 1), (\text{BLAU}, 1)\} \\ \{(\text{ROT}, 2), (\text{BLAU}, 4), (\text{GRÜN}, 3)\} \\ \emptyset \\ \{(\text{BLAU}, 1)\} \end{array} \right)$$

6 Gefärbte Petri-Netze mit Variablen

in den Zustand

$$\mathbf{z}''' \equiv \begin{pmatrix} \{(ROT, 2), (GRÜN, 1)\} \setminus \{(ROT, 1), (GRÜN, 1)\} \\ \{(ROT, 1), (BLAU, 1)\} \setminus \{(ROT, 1), (BLAU, 1)\} \\ \{((ROT, 2), (BLAU, 4), (GRÜN, 3)) \setminus ((ROT, 1), (GRÜN, 3))\} \uplus \{(ROT, 2), (BLAU, 1), (GRÜN, 1)\} \\ \emptyset \uplus \{(1, 3)\} \\ \{(BLAU, 1)\} \uplus \{(GRÜN, 1)\} \end{pmatrix} \\ = \begin{pmatrix} \{(ROT, 1)\} \\ \emptyset \\ \{(ROT, 3), (BLAU, 5), (GRÜN, 1)\} \\ \{(1, 3)\} \\ \{(BLAU, 1), (GRÜN, 1)\} \end{pmatrix}$$

übergangen.

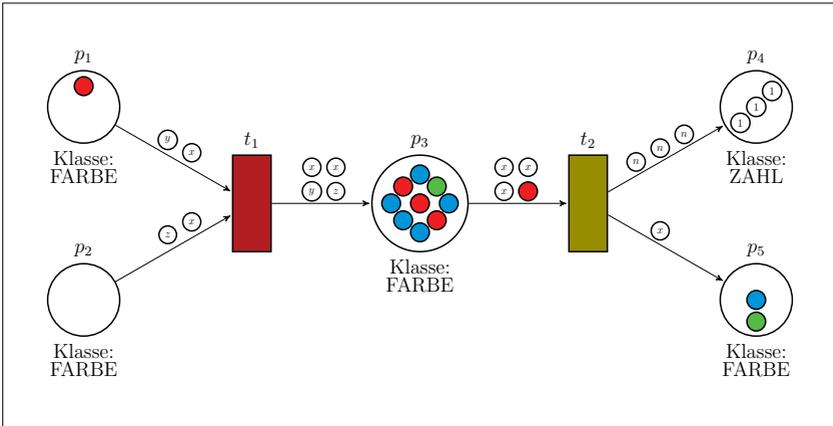


Abbildung 6.14: CPN mit Variablen im Zustand \mathbf{z}'''

Die Symbolik $\mathbf{z}\langle(t_1, \beta_x), (t_2, \beta_y)\rangle\mathbf{z}'$ bedeutet, dass ein nebenläufiges Feuern der Transitionen t_1 und t_2 eines im Zustand \mathbf{z} vorliegenden CPN möglich ist und dabei in den Zustand \mathbf{z}' übergeht. Die allgemeine Darstellung für einen Zustandsübergang durch nebenläufiges Feuern von k Transitionen ist $\mathbf{z}\langle(t_{j_1}, \beta_{x_1}), (t_{j_2}, \beta_{x_2}), \dots, (t_{j_k}, \beta_{x_k})\rangle\mathbf{z}'$. Mittels dieser Schreibweise lassen sich nun auch allgemeinere Zustandsübergangsketten beschreiben, die man auch als **Rechnungen** (vgl. Abschn. 1.2.2) bezeichnet:

$$\mathbf{z}^1\langle(t_{j_1}, \beta_{x_1}), (t_{j_2}, \beta_{x_2}), \dots, (t_{j_r}, \beta_{x_r})\rangle\mathbf{z}^2\langle(t_{k_1}, \beta_{y_1}), (t_{k_2}, \beta_{y_2}), \dots, (t_{k_s}, \beta_{y_s})\rangle\mathbf{z}^3 \dots \\ \dots \mathbf{z}^q\langle(t_{l_1}, \beta_{z_1}), (t_{l_2}, \beta_{z_2}), \dots, (t_{l_v}, \beta_{z_v})\rangle\mathbf{z}^{q+1}.$$

Da nun die Grundlagen der CPN mit Variablen gelegt sind, soll im nächsten Abschnitt auf mögliche Erweiterungen und Probleme dieser CPN mit Variablen eingegangen werden.

Für eine weiterführende Behandlung der CPN mit Variablen sei auf [JK09] und Kapitel 9 verwiesen.

6.2 Problematik der Modus-Wahl

Wie schon unter 6.1 beschrieben, ist eine häufig anzutreffende Modus-Menge die Menge aller möglichen Modi, die über die Variablen aus \mathbb{V} gebildet werden kann. Solange man einen „Benutzer“ wie in 6.1 hat, der (ggf. interaktiv) einen Modus wählt, in welchem ge- feuert werden soll, besteht kein explizites Auswahlproblem. Andererseits könnte aber der Untersuchungsgegenstand eines Problems darin bestehen, dass die Modus-Wahl automa- tisch vollzogen wird bzw. werden soll. Dabei gibt es unterschiedlichste Strategien, ähnlich wie bei den Konfliktlösungen. Diese Strategien lassen sich in zwei Kategorien unterteilen, und zwar in Strategien, bei denen die Häufigkeit der einzelnen Tokens berücksichtigt wird, und in solche, bei denen dies nicht geschieht.

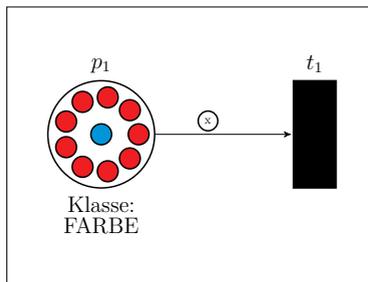


Abbildung 6.15: Problematik der Modus-Wahl

Im obigen Beispiel würden die Strategien mit Berücksichtigung der Häufigkeit darauf eingehen, dass die Anzahl der mit ROT gefärbten Tokens 9 und die der mit BLAU gefärbten nur 1 ist, und dementsprechend der Variablen x in 90% der Fälle ROT zuweisen und in 10% BLAU.

Bevor allerdings eine Modus-Wahl stattfinden kann, muss zuerst einmal geprüft werden, ob eine Transition überhaupt aktivierbar und daher feuerebar ist. Dies kann im schlimmsten Fall eine vollständige Suche zur Folge haben. Dazu wird das Beispiel mit der Klasse $ZAHL := \{1, \dots, 100\}$ und der Modus-Menge $m(t_1)$ aller möglichen Kombinationen betrachtet.

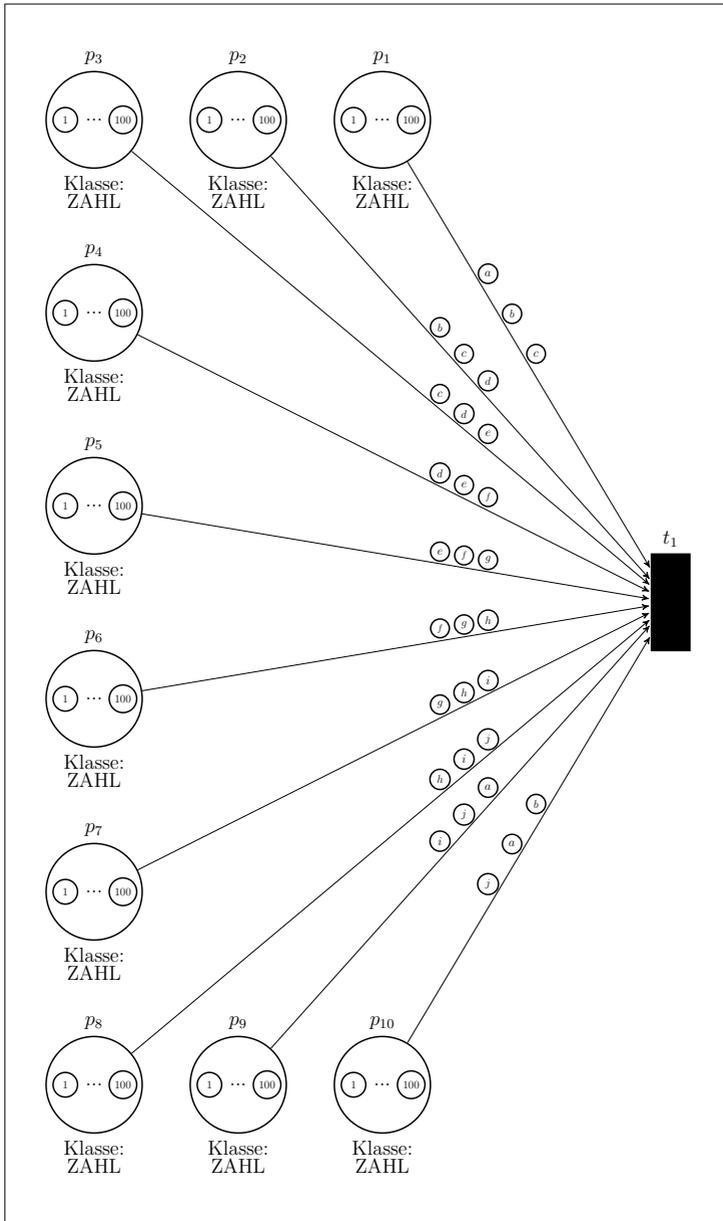


Abbildung 6.16: Problematik der vollständigen Suche bei der Modus-Wahl

Bei diesem Beispiel müssten $100^{10} = 100.000.000.000.000.000.000$ (100 Trillionen) Kombinationen geprüft werden. Natürlich gibt es eine Menge von Vereinfachungs-/Reduktionsstrategien wie zum Beispiel die **Pattern-Matching-Methode** nach [JK09], die man anwenden könnte, aber selbst bei einer Reduktion um 99% sind immer noch 1 Trillion und daher zu viele Optionen übrig.

Da die Ermittlung der Menge aller feuerbaren Modi einer Transition bereits in Methoden wie das Pattern-Matching integriert ist, wird die Untersuchung, ob ein Modus feuerbar ist, als gegeben angesehen.

Nun sollen exemplarisch ein paar Wahl-Strategien vorgestellt werden, wobei zu beachten ist, dass dies nur eine Auswahl möglicher Strategien ist und es noch viele andere Möglichkeiten gibt.

6.2.1 Wahlstrategien ohne Berücksichtigung der Häufigkeit

Für diese Strategien werden hier zwei Kategorien vorgestellt, einerseits eine *Modi-Wahrscheinlichkeitswahl* und andererseits eine *Wahl nach Token-Nutzen*. Dazu betrachten wir das folgende Beispiel:

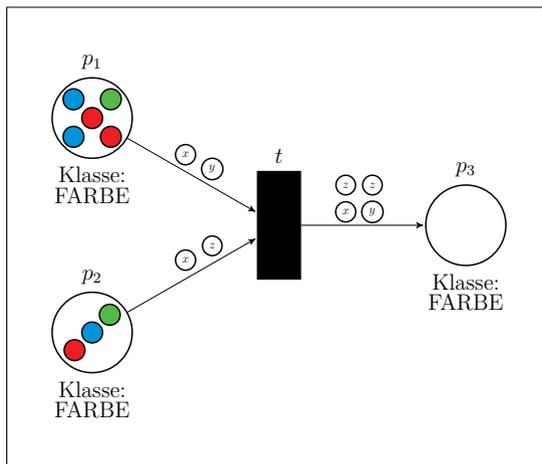


Abbildung 6.17: Wahlstrategien ohne Berücksichtigung der Häufigkeit

Zur Bildung der Modus-Menge $m(t)$ wird die vollständige Enumeration zugelassen, womit man folgende Modi erhält:

Tabelle 6.2: Modus-Menge mit Mächtigkeit 27

β	x	y	z	β	x	y	z	β	x	y	z
1	ROT	ROT	ROT	10	BLAU	ROT	ROT	19	GRÜN	ROT	ROT
2	ROT	ROT	BLAU	11	BLAU	ROT	BLAU	20	GRÜN	ROT	BLAU
3	ROT	ROT	GRÜN	12	BLAU	ROT	GRÜN	21	GRÜN	ROT	GRÜN
4	ROT	BLAU	ROT	13	BLAU	BLAU	ROT	22	GRÜN	BLAU	ROT
5	ROT	BLAU	BLAU	14	BLAU	BLAU	BLAU	23	GRÜN	BLAU	BLAU
6	ROT	BLAU	GRÜN	15	BLAU	BLAU	GRÜN	24	GRÜN	BLAU	GRÜN
7	ROT	GRÜN	ROT	16	BLAU	GRÜN	ROT	25	GRÜN	GRÜN	ROT
8	ROT	GRÜN	BLAU	17	BLAU	GRÜN	BLAU	26	GRÜN	GRÜN	BLAU
9	ROT	GRÜN	GRÜN	18	BLAU	GRÜN	GRÜN	27	GRÜN	GRÜN	GRÜN

Modi-Wahrscheinlichkeitswahl (vgl. Wahrscheinlichkeiten bei Konflikten):

Hier wird jedem Modus eine von einem Benutzer (oder vom Problem an sich) vorgegebene Wahrscheinlichkeit zugewiesen, dass dieser Modus gewählt wird; hier der Einfachheit halber jedem Modus dieselbe Wahrscheinlichkeit:

$$\text{Wahrscheinlichkeit}(\beta_i) := \frac{1}{27} \text{ für } i = 1, \dots, 27.$$

Allerdings sind je nach Zustand nicht alle Modi feuerebar, z.B. sind hier nur die folgenden 16 Modi feuerebar:

$$\beta_2, \beta_3, \beta_5, \beta_6, \beta_8, \beta_9, \beta_{10}, \beta_{12}, \beta_{13}, \beta_{15}, \beta_{16}, \beta_{18}, \beta_{19}, \beta_{20}, \beta_{22}, \beta_{23}.$$

Dies bedeutet, dass die Wahrscheinlichkeiten aktualisiert werden müssten: $\text{W}^{\text{keit}}_{\text{aktuell}}(\beta_i) := \frac{1}{16}$ für $i \in \{2, 3, 5, 6, 8, 9, 10, 12, 13, 15, 16, 18, 19, 20, 22, 23\}$ bzw. 0 sonst. Nach diesen aktualisierten Wahrscheinlichkeiten könnte nun per Monte-Carlo-Methode ein Modus ausgewählt werden, der gefeuert wird.

Token-Nutzen:

Jeder Ausprägung der Klasse FARBE wird ein bestimmter Ertrag (Output) bzw. Aufwand (Input) zugewiesen und somit jedem entsprechend gefärbten Token. Auf kanonische Weise wird dies auf die Pfeilgewichte $f_\beta(p, t)$ und $f_\beta(t, p)$ übertragen; z.B. mit

$$\text{Aufwand}(f_\beta(p, t)) := 1 \cdot h(\text{ROT}) + 3 \cdot h(\text{BLAU}) + 5 \cdot h(\text{GRÜN})$$

und

$$\text{Ertrag}(f_\beta(t, p)) := 1 \cdot h(\text{ROT}) + 3 \cdot h(\text{BLAU}) + 5 \cdot h(\text{GRÜN}).$$

In diesem Beispiel sind Ertrag und Aufwand gleich, könnten aber theoretisch auch unterschiedlich sein:

$$\begin{aligned}\text{Aufwand}((\text{ROT}, 1)) &= \text{Ertrag}((\text{ROT}, 1)) = 1 \\ \text{Aufwand}((\text{BLAU}, 1)) &= \text{Ertrag}((\text{BLAU}, 1)) = 3 \\ \text{Aufwand}((\text{GRÜN}, 1)) &= \text{Ertrag}((\text{GRÜN}, 1)) = 5\end{aligned}$$

Nun stellt man für jeden Modus bzgl. der Transition t seinen Nutzen fest, und zwar mit:

$$\text{Nutzen}(\beta_i) := \text{Gesamtertrag}(\beta_i) - \text{Gesamtaufwand}(\beta_i)$$

wobei

$$\text{Gesamtertrag}(\beta_i) := \sum_{p \in t \bullet} \text{Ertrag}(f_{\beta_i}(t, p))$$

und

$$\text{Gesamtaufwand}(\beta_i) := \sum_{p \in \bullet t} \text{Aufwand}(f_{\beta_i}(p, t))$$

ist.

Tabelle 6.3: Token-Nutzen-Zusammenstellung für alle Modi

i	Gesamtertrag(β_i)	Gesamtaufwand(β_i)	Nutzen(β_i)	feuerbar
1	4	4	0	Nein
2	8	6	2	Ja
3	12	8	4	Ja
4	6	6	0	Nein
5	10	8	2	Ja
6	14	10	4	Ja
7	8	8	0	Nein
8	12	10	2	Ja
9	16	12	4	Ja
10	6	8	2	Ja
11	10	10	0	Nein
12	14	12	2	Ja
13	8	10	2	Ja
14	12	12	0	Nein
15	16	14	2	Ja
16	10	12	-2	Ja
17	14	14	0	Nein
18	18	16	2	Ja
19	8	12	-4	Ja
20	12	14	-2	Ja
21	16	16	0	Nein
22	10	14	-4	Ja
23	14	16	-2	Ja
24	18	18	0	Nein
25	12	16	-4	Nein
26	16	18	-2	Nein
27	20	20	0	Nein

Anschließend wird aus den Modi, bei denen die Transition feuerbar ist, der mit dem größten Nutzen ausgewählt. Im Beispiel haben die Modi β_3 , β_6 und β_9 den größten Nutzen; daher wird hier zufällig oder willkürlich bestimmt, welcher von den drei Modi gewählt wird.

Grundsätzlich ist im Vorfeld zu überlegen, ob man einmal für alle Modi den Nutzen am Anfang berechnet und speichert oder jedes Mal für die feuerbaren Modi den Nutzen neu berechnet.¹ Die Entscheidung sollte davon abhängig gemacht werden, ob man eine große Menge feuerbarer Modi erwartet oder nicht.

¹Selbstverständlich sind zusätzlich zu diesen beiden „Extremstrategien“ effizientere Vorgehensweisen denkbar

Anstelle des Wahlkriteriums „größter Nutzen“ könnte man auch nach dem größten Gesamtertrag oder dem kleinsten Gesamtaufwand eine Modus-Wahl treffen.

6.2.2 Wahlstrategien mit Berücksichtigung der Häufigkeit

Bei diesen Strategien¹ wird im Speziellen darauf eingegangen, dass unterschiedliche Token in Abhängigkeit von den Zuständen des Netzes unterschiedlich oft in den einzelnen Plätzen des Vorbereiches einer Transition vorkommen. Um dieser Dynamik bei der Modi-Wahl Rechnung zu tragen, wäre es eine sinnvolle Strategie, Token, die häufiger in einem Platz vorkommen, mit einer größeren Wahrscheinlichkeit einer Variablen zu versehen als solche, die weniger häufig vorkommen. Eine derartige den Zuständen angepasste Strategie eignet sich sehr gut zur Modellierung von Verhaltensmustern in einer Vielzahl von realen Problemstellungen.

Eine solche Strategie anzuwenden ist zwar bei nur einem Platz und nur einer Variablen wie in Abb. 6.15 noch recht klar und eindeutig, aber bei mehreren Plätzen und Variablen ist dies schon schwieriger, wie im folgenden Beispiel verdeutlicht wird.

¹Diese Strategien sind in Anlehnung an das CPN-Simulationstool "CPN Tools" entstanden (<http://cpntools.org/>).

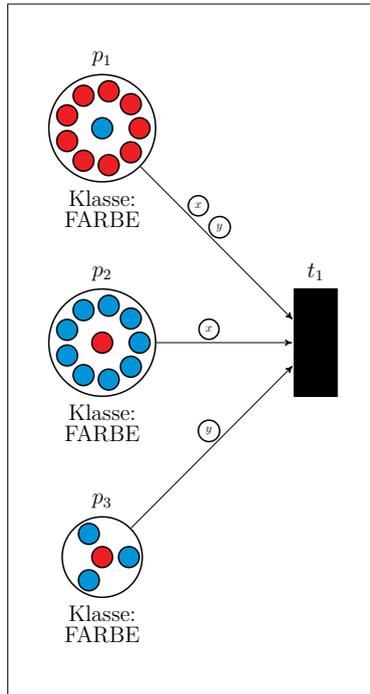


Abbildung 6.18: Probleme bei der Modi-Wahl unter Berücksichtigung der Häufigkeit

Betrachtet man die Variable x und die Plätze p_1 und p_2 , dann ist die Häufigkeitsverteilung der Tokens mit der Farbe ROT oder BLAU sehr unterschiedlich. Daher stellt sich die Frage: Wie sollte hier nach den Häufigkeiten vorgegangen werden? Es werden hier die folgenden drei Möglichkeiten vorgeschlagen:

1. Möglichkeit

Die Summen der einzelnen Tokens werden über alle Plätze, auf deren Pfeile die Variable x steht, gebildet und diese Summenhäufigkeit wird dazu verwendet, dass hier die Wahrscheinlichkeit für $x = \text{ROT}$ bei 50%, für $x = \text{BLAU}$ bei 50% und für $x = \text{GRÜN}$ bei 0% liegt.

2. Möglichkeit

Für jede Variable wird ein „Repräsentanzplatz“ festgelegt, auf dessen Grundlage die Zuordnungswahrscheinlichkeiten der einzelnen Variablen bestimmt werden, allerdings unter der Voraussetzung, dass die jeweilige Variable auch auf dem Pfeil des Repräsentanzplatzes als Gewicht steht. Hätte z.B. die Variable x den Platz p_2 als Repräsentanzplatz, dann würde hier

die Wahrscheinlichkeit für $x = \text{ROT}$ bei 10%, für $x = \text{BLAU}$ bei 90% und für $x = \text{GRÜN}$ bei 0% liegen.

3. Möglichkeit

Bei jedem Feuern wird zufällig ein Platz bestimmt, dessen Häufigkeit zur Wahrscheinlichkeitsbestimmung herangezogen wird, wiederum unter der Voraussetzung, dass die Variable auch auf dem Pfeil des Platzes als Gewicht steht.

Dieses gewählte Verfahren müsste nun für jede Variable durchgeführt werden, und man würde so für jede Zuordnung eine Wahrscheinlichkeit dafür erhalten, dass diese gewählt wird. Aus diesen Wahrscheinlichkeiten könnte man eine Gesamtwahrscheinlichkeit der Zuordnung berechnen. Hier könnte nun das Verfahren der Modi-Wahrscheinlichkeitswahl angewendet werden.

Eine erste praktikable Vereinfachung des Problems ist es, wenn nicht über alle Tokenfarben geprüft würde, sondern nur über die Tokenfarben, deren Häufigkeit in jedem der Plätze größer 0 ist.

Beispiel:

Für das CPN aus Abb. 6.18 soll eine automatisierte Modus-Wahl in Abhängigkeit der Tokenhäufigkeit stattfinden. Dazu wird für die Variable x der Repräsentanzplatz p_2 gewählt und für Variable y der Repräsentanzplatz p_3 , so dass die Zuordnungs-W'keiten geben sind mit:

$$\text{Zuordnungs-W'keit}(x = \text{BLAU}) := 0,9$$

$$\text{Zuordnungs-W'keit}(x = \text{ROT}) := 0,1$$

$$\text{Zuordnungs-W'keit}(y = \text{BLAU}) := 0,75$$

$$\text{Zuordnungs-W'keit}(y = \text{ROT}) := 0,25.$$

Mit diesen Zuordnungs-W'keiten lassen sich für die folgenden Modi

$$\beta_1 : \{x = \text{ROT}, y = \text{ROT}\},$$

$$\beta_2 : \{x = \text{ROT}, y = \text{BLAU}\},$$

$$\beta_3 : \{x = \text{BLAU}, y = \text{ROT}\},$$

$$\beta_4 : \{x = \text{BLAU}, y = \text{BLAU}\}$$

6 Gefärbte Petri-Netze mit Variablen

die Modi-W'keiten

$$W'keit(\beta_1) := 0,1 \cdot 0,25 = 0,025$$

$$W'keit(\beta_2) := 0,1 \cdot 0,75 = 0,075$$

$$W'keit(\beta_3) := 0,9 \cdot 0,25 = 0,225$$

$$W'keit(\beta_4) := 0,9 \cdot 0,75 = 0,675$$

bestimmen.

Allerdings wäre der Modus β_4 nicht feuerbar, so dass aktualisierte Wahrscheinlichkeiten berechnet werden müssten:

$$W'keit_{\text{aktuell}}(\beta_1) := \frac{0,025}{0,325} = \frac{1}{13}$$

$$W'keit_{\text{aktuell}}(\beta_2) := \frac{0,075}{0,325} = \frac{3}{13}$$

$$W'keit_{\text{aktuell}}(\beta_3) := \frac{0,225}{0,325} = \frac{9}{13}$$

Nach diesen aktualisierten Wahrscheinlichkeiten könnte nun eine Monte-Carlo-Methode verwendet werden, um einen Modus auszuwählen.

7 Konfliktlösung bei CPN mit Variablen

In diesem Kapitel soll die Konfliktlösung des nebenläufigen Feuerprozesses bei CPN mit konstanten Pfeilgewichtungen aus Kapitel 5 und die CPN mit variabler Pfeilgewichtung aus Kapitel 6 zusammengeführt werden, um so eine Konfliktlösung für CPN mit variablen Pfeilgewichten zu erhalten. Diese zu Beginn recht schwierig erscheinende Aufgabenstellung ist jedoch einfach nachzuvollziehen, wenn man bedenkt, in welchem Schritt überhaupt erst ein Konflikt auftritt. Dazu betrachte man das folgende CPN-Beispiel mit den Modus-Mengen:

$$m(t_1) = m(t_2) = \{\beta_1, \beta_2, \beta_3\},$$

wobei

$$\beta_1 : \{x = \text{ROT}, y = \text{ROT}\},$$

$$\beta_2 : \{x = \text{BLAU}, y = \text{BLAU}\},$$

$$\beta_3 : \{x = \text{GRÜN}, y = \text{GRÜN}\}.$$

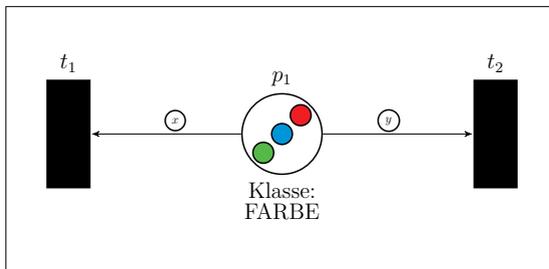


Abbildung 7.1: Konflikt bei gefärbten Petri-Netzen mit variabler Pfeilbewertung

In diesem CPN ist zu erkennen, dass beide Transitionen t_1 und t_2 aktivierbar sind. Sollen allerdings beide Transitionen nebenläufig gefeuert werden, kann es passieren, dass ein Konflikt auftritt, falls beiden Variablen dieselbe Farbe zugeordnet wird. Da vor der Modi-Wahl noch nicht feststeht, ob ein Konflikt in p_1 vorliegt, müssen erst die Modi gewählt werden, in

denen die jeweiligen Transitionen gefeuert werden sollen.

Grundsätzlich gilt der folgende Sachverhalt: Wenn die Modi für alle zufeuernden Transitionen bestimmt sind, erhält man ein induziertes MCPN, in dem überhaupt erst eine Aussage darüber gemacht werden kann, ob ein Konflikt vorliegt. Falls anschließend ein Konflikt vorliegt, könnten grundsätzlich alle Lösungsstrategien aus Kapitel 5 angewendet werden. Daher sind keine neuen oder modifizierten Konfliktlösmethoden notwendig.

Dazu sei eine Verallgemeinerung des nebenläufigen Feuerprozesses bei MCPN auf CPN mit Variablen beschrieben.

Szenarium 4 (vgl. Szenarium 3 in Kap. 3):

Der **allgemeine Feuerprozess bei CPN mit Variablen** verfährt nach den folgenden Schritten:

1. Ermittlung der Menge TA aller gemäß Def. 6.7 (b) aktiven Transitionen:

$$TA := \{t \in T \mid \forall p \in \bullet t : f_\beta(p, t) \subseteq \mathbf{z}(p) \text{ mit } \beta \in m(t)\}$$

2. Bestimmung einer Menge TW von Transitionen aus TA
3. Bestimmung eines Modus für jede Transition aus TW
(via Verfahren aus Kap. 6.2 oder durch Wahl eines Benutzers)
4. Ermittlung aller lokalen Freigabemengen $TE_{out}(p)$ aus TW
(via Algorithmus aus Kap. 5)
5. Ermittlung der globalen Freigabemenge TE (gemäß Def. 5.10)
6. Bestimmung einer Menge TF von Transitionen aus TE
7. Nebenläufiges Feuern der Menge TF

Beispiel:

Gegeben sei das folgende CPN mit Variablen, wobei für den Platz p_2 die Konfliktlösung nach Prioritäten gilt. Bei den Auswahlprozessen für TW oder TF wird im Beispiel von einem Benutzer ausgegangen, der interaktiv die Transitionen wählt.

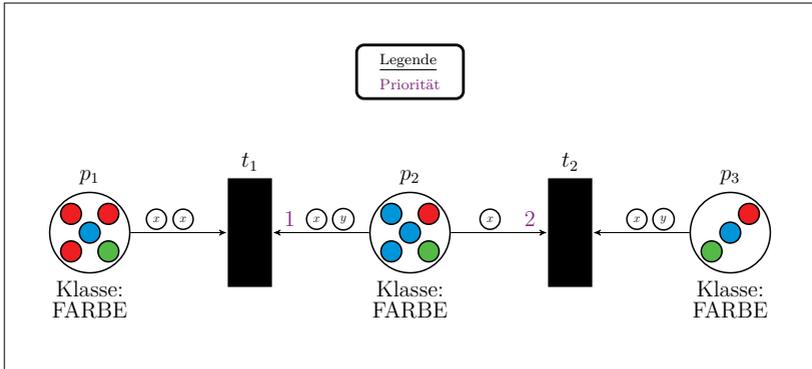


Abbildung 7.2: CPN mit Variablen und Prioritäten

Die Transitionen t_1 und t_2 haben jeweils die Modus-Menge:

$$m(t_i) := \mathbb{B} = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9\}, \quad i = 1, 2;$$

mit

$$\begin{aligned} \beta_1 &: \{x = \text{ROT}, y = \text{ROT}\}, \\ \beta_2 &: \{x = \text{ROT}, y = \text{BLAU}\}, \\ \beta_3 &: \{x = \text{ROT}, y = \text{GRÜN}\}, \\ \beta_4 &: \{x = \text{BLAU}, y = \text{ROT}\}, \\ \beta_5 &: \{x = \text{BLAU}, y = \text{BLAU}\}, \\ \beta_6 &: \{x = \text{BLAU}, y = \text{GRÜN}\}, \\ \beta_7 &: \{x = \text{GRÜN}, y = \text{ROT}\}, \\ \beta_8 &: \{x = \text{GRÜN}, y = \text{BLAU}\}, \\ \beta_9 &: \{x = \text{GRÜN}, y = \text{GRÜN}\}. \end{aligned}$$

Schritt 1: Bestimmung von TA

Beide Transitionen sind aktivierbar, da t_1 in den Modi β_2, β_3 und Transition t_2 in den Modi $\beta_2, \beta_3, \beta_4, \beta_6, \beta_7, \beta_8$ feuert. Daher ist die Menge der aktivierbaren Transitionen $TA := \{t_1, t_2\}$ und in Abb. 7.3 blau markiert. Diese Menge soll nun nebenläufig gefeuert werden.

7 Konfliktlösung bei CPN mit Variablen

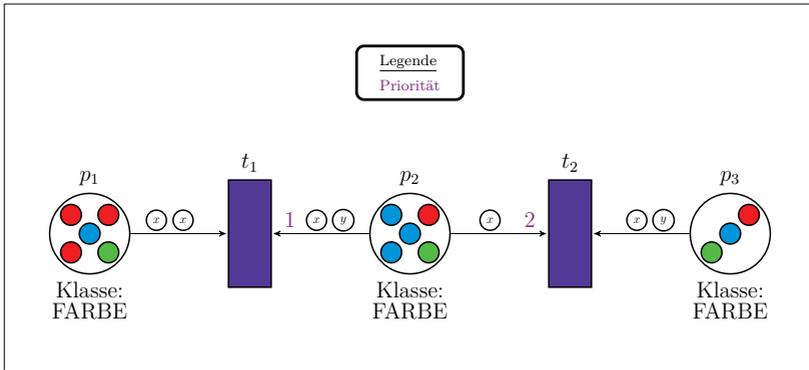


Abbildung 7.3: Aktivierbare Transitionen eines CPN mit Variablen

Schritt 2: Bestimmung von TW

Der Benutzer wählt $TW = T_A$.

Schritt 3: Wahl der Modi

An dieser Stelle könnte nun für jede Transition jeweils ein Modus gewählt werden, und zwar nach einer Methode aus Abschnitt 6.2, z.B. durch die Modi-Wahrscheinlichkeitswahl mit den Wahrscheinlichkeiten:

$$W^{\text{keit}}(\beta_i) := \frac{1}{9} \text{ für } i = 1, \dots, 9.$$

Allerdings sind nicht alle Modi feuerbar, z.B. sind hier für t_1 nur die folgenden 2 Modi feuerbar:

$$\beta_2, \beta_3,$$

und für t_2 die folgenden 6 Modi:

$$\beta_2, \beta_3, \beta_4, \beta_6, \beta_7, \beta_8.$$

Dies bedeutet, dass die Wahrscheinlichkeiten aktualisiert werden müssten, und zwar für t_1 :

$$W^{\text{keit}}_{\text{aktuell}}(\beta_i) := \frac{1}{2} \text{ für } i \in \{2, 3\} \text{ bzw. } 0 \text{ sonst,}$$

und für t_2 :

$$W'keit_{aktuell}(\beta_i) := \frac{1}{6} \text{ für } i \in \{2, 3, 4, 6, 7, 8\} \text{ bzw. } 0 \text{ sonst,}$$

Für diese aktualisierten Wahrscheinlichkeiten sei für Transition t_1 der Modus

$$\beta_3 : \{x = \text{ROT}, y = \text{GRÜN}\}$$

und für Transition t_2 der Modus

$$\beta_8 : \{x = \text{GRÜN}, y = \text{BLAU}\}$$

gewählt worden. Da nun jeder Transition ein Modus zugeordnet wurde, erhält man das MCPN aus Abb. 7.4.

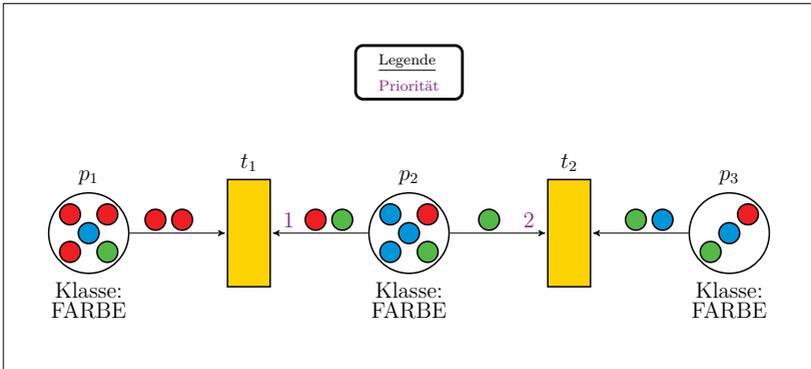


Abbildung 7.4: Durch Modus-Wahl entstandenes MCPN aus CPN mit Variablen in Abb. 7.3.

Schritt 4: Bestimmung von allen $TE_{out}(p)$

Da nur p_2 mehr als eine Transition im Nachbereich hat, muss auch nur die eine lokale Freigabemenge $TE_{out}(p_2)$ bestimmt werden. Dafür wird das lokale Prioritätenverfahren (siehe Algorithmus 5.6) verwendet:

Setzung: $TE_{out}(p_2) := \emptyset$.

Prüfungen:

$$t_1 \text{ wg. } \varrho(p_2, t_1) = 1:$$

$$f(p_2, t_1) = \{(\text{ROT}, 1), (\text{GRÜN}, 1)\} \subseteq$$

$$\{(\text{ROT}, 1), (\text{BLAU}, 3), (\text{GRÜN}, 1)\} = \mathbf{z}(p_2)$$

$$\Rightarrow TE_{out}(p_2) := TE_{out}(p_2) \cup \{t_1\} = \{t_1\}$$

7 Konfliktlösung bei CPN mit Variablen

t_2 wg. $\varrho(p_2, t_2) = 2$:

$$f(p_2, t_1) \uplus f(p_2, t_2) = \{(ROT, 1), (GRÜN, 2)\} \not\subseteq \\ \{(ROT, 1), (BLAU, 3), (GRÜN, 1)\} = \mathbf{z}(p_2)$$

$\Rightarrow t_2$ kann nicht (zusätzlich) freigegeben werden.

Ergebnis: $TE_{out}(p_2) = \{t_1\}$, d.h. es wird die Transition t_1 von Platz p_2 freigegeben.

Schritt 5: Bestimmung von TE

Da hier nur p_2 einen Konflikt hat, weil die anderen Plätze lediglich eine Transition im Nachbereich besitzen, ist hier

$$TE := TE_{out}(p_2) = \{t_1\}.$$

Schritt 6: Bestimmung von TF

Der Benutzer wählt $TF = TE$.

Schritt 7: Nebenläufiges Feuern der Transitionen-Menge TF

Als letzter Schritt wird nun die Transition t_1 gefeuert und man erhält das CPN aus Abb. 7.5.

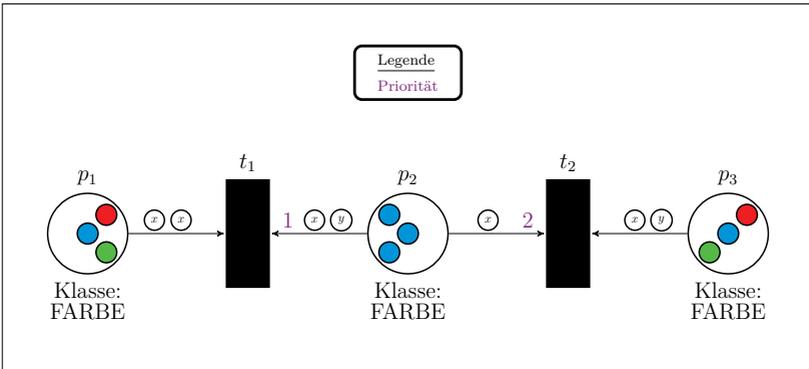


Abbildung 7.5: CPN mit Variablen nach dem Feuern der Transition t_1 im Modus β_3

Es hätten natürlich auch andere Modi für die Transitionen gewählt werden können, z.B. für Transition t_1 der Modus

$$\beta_2 : \{x = ROT, y = BLAU\}$$

und für Transition t_2 der Modus

$$\beta_4 : \{x = \text{BLAU}, y = \text{ROT}\}.$$

Daraus würde man das entsprechende MCPN aus Abb. 7.6 erhalten.

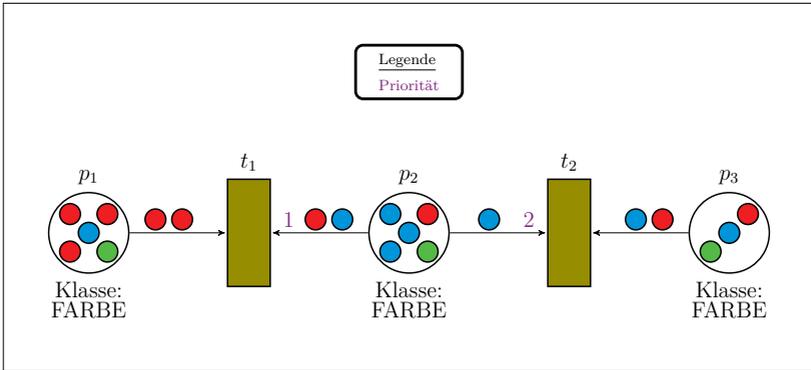


Abbildung 7.6: Durch alternative Modus-Wahl entstandenes MCPN aus CPN mit Variablen von Abb. 7.3.

In diesem MCPN liegt in p_2 nun kein Konflikt vor und beide Transitionen t_1 und t_2 können gefeuert werden. Ob ein Konflikt vorliegt, hängt somit davon ab, welche Modi für die einzelnen Transitionen gewählt werden. Hieraus könnte man eine Strategie zur Modus-Wahl herleiten, indem möglichst diejenigen Modi gewählt werden, die in Kombination miteinander keine Konflikte erzeugen. Solche Strategien nennt man **Modus-Wahl mit Konfliktprophylaxe** (vgl. [KD13]). Diese „Konfliktvermeidungs-Verfahren“ sind allerdings sehr rechenaufwändig, weil die einzelnen Transitionen und ihre Modus-Wahl in Abhängigkeit zueinander stehen. Da aber schon bei einer einzigen Transition schnell mehrere Trillionen Rechnungen gemacht werden müssen, um zulässige Modi zu finden, kann es hier passieren, dieser sowieso große Rechenaufwand mit der Anzahl der Transitionen noch potenziert wird. Zudem kann hierbei nicht einmal garantiert werden, dass zulässige Modi existieren, die keine Konflikte verursachen, wie das folgende Beispiel zeigt.

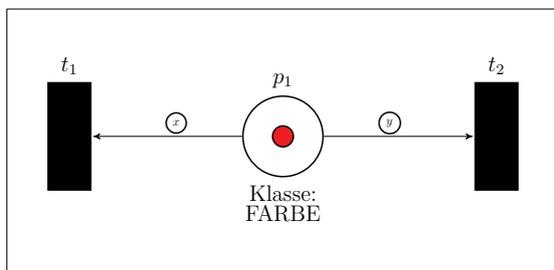


Abbildung 7.7: Beispiel für einen nicht zu verhindernden Konflikt

Wenn t_1 und t_2 nebenläufig gefeuert werden sollen, ist keine Verhinderung des Konflikts möglich, weil den Variablen x und y jeweils nur die Farbe ROT zugewiesen werden kann. Dieses Beispiel zeigt, dass eine Konfliktprophylaxe allein nicht ausreicht, um bei CPN mit Variablen eine allgemein gültige Auflösung von Konfliktsituationen herbeizuführen.

An dieser Stelle wird nun die formale und theoretische Behandlung der CPN und ihrer Konfliktlösungen abgeschlossen. In Kapitel 8 wird auf ein ausführliches Anwendungsbeispiel eingegangen, mit dem gezeigt wird, wie die entwickelte Theorie in eine Modellierung von realen Prozessen eingebunden werden kann.

8 Anwendungsbeispiel: Biergarten

Nun sollen die in der Theorie behandelten CPN und ihre Konfliktlösung in einer Anwendung verwendet werden. Hierfür soll ein Modell eines Biergartens vorgestellt werden, welches in Anlehnung an die Arbeit von [KC11] modelliert wurde.

In diesem Buch wird bei der Modellierung des Biergartens allerdings ausschließlich die Bar an sich ohne Gäste und Kellner betrachtet, damit das Modell in einem übersichtlichen Rahmen bleibt, aber dennoch die wesentlichen Aspekte, welche in diesem Buch im Fokus der Untersuchung stehen, berücksichtigt werden.

8.1 Problemstellung

In der Bar des Biergartens arbeiten zwei Barkeeper, der eine Barkeeper ist allerdings Alkoholiker und darf daher nicht mit alkoholischen Getränken in Kontakt kommen. Daher sind diese beiden Barkeeper voneinander getrennt in der Bar positioniert, so dass der eine nur Getränke ohne Alkohol abfüllt (**BARKEEPER OHNE ALC.**) und der andere nur Getränke mit Alkohol abfüllt (**BARKEEPER ALC.**). Beide Barkeeper erhalten Bestellungen aus einem Bestellmenge (**BESTELLUNG**) und die Gläser für die Getränke von einer Spüle (**SPÜLE**). Da aber öfters Glasknappheit herrscht (bei ausgelassen feiernden Gästen geht öfters etwas zu Bruch), können nicht immer alle Getränke sofort abgefüllt werden. Daher hat der Betreiber entschieden, dass bei Knappheit der Gläser alkoholische Getränke Vorrang haben sollen, weil mit diesen mehr Gewinn gemacht werden kann. Das Sortiment der nicht alkoholischen Getränke besteht aus Wasser, Limonade und Saft, das alkoholische Sortiment aus Bier, Spirituosen und Cocktails. Des Weiteren verfügt die Bar über die folgenden drei Gläserarten: Pinnchen, Standardglas und Cocktailglas. Die einzelnen Getränke werden in

8 Anwendungsbeispiel: Biergarten

die folgenden Gläser gefühlt:

Wasser	→	Standardglas
Limonade	→	Standardglas
Saft	→	Cocktailglas
Bier	→	Standardglas
Spirituosen	→	Pinnchen
Cocktail	→	Cocktailglas

Im Moment ist die folgende Bestellung zu erledigen:

2 Wasser, 1 Limonade, 1 Saft, 2 Bier, 1 Spirituose, 2 Cocktails.

Dabei sind die folgenden Gläser in der Spüle:

2 Pinnchen, 4 Standardgläser, 3 Cocktailgläser.

Wenn die Barkeeper Getränke fertig gemacht haben, werden diese auf Förderbänder (**FÖRDERBAND 1**, **FÖRDERBAND 2**) gestellt, um sie an eine gemeinsame Entnahmestelle (**TRESEN**) zu transportieren. Diese Förderbänder laufen in einer Kreuzung (**KREUZUNG**) zusammen, aber weil es hier öfter „Zusammenstöße“ zwischen den einzelnen Getränken gegeben hat, hat der Betreiber eine Steuerung (**STEUERUNG**) eingebaut. Diese Steuerung regelt mithilfe zweier Stopper (**STOPPER 1**, **STOPPER 2**) den Einlass der Getränke in die Kreuzung. Daher befindet sich immer nur genau ein Getränk in der Kreuzung, für die solange eine Eintrittsblockade gilt, bis ein Sensor (**SENSOR**) nach der Kreuzung passiert wurde und dadurch die Kreuzung wieder freigegeben wird. Allerdings steht in der Beschreibung für die Mechanik nicht angegeben, in welcher Weise diese Freigabe erfolgt. Beobachtungen ergaben aber, dass ungefähr eine 50 : 50-Chance besteht, welcher Stopper freigegeben wird. Auf dem nächsten Förderband (**FÖRDERBAND 3**) werden die Getränke nach vorne zu einem Greifer (**GREIFER**) transportiert, der die Getränke in Warteposition auf den Tresen abstellt und so für die Kellner erreichbar macht. Das Restaurant (**RESTAURANT**) soll an dieser Stelle nur stark vereinfacht dargestellt werden; und zwar soll nur abgebildet werden, dass die Getränke vom Tresen ins Restaurant gebracht werden und ein monetärer Rückfluss in die Kasse (**KASSE**) stattfindet, und zwar entsprechend den

Preisen des Biergartens. Die Preise (in Euro) auf der Karte sind:

Wasser	→	1
Limonade	→	3
Saft	→	3
Bier	→	3
Spirituosen	→	2
Cocktail	→	5

8.2 Modell eines gefärbten Petri-Netzes mit Variablen

Es soll nun für diese Problemstellung ein CPN mit Variablen modelliert werden. Dafür wird als Erstes eine Netzstruktur erzeugt. Hierfür muss bestimmt werden, welche Komponenten (Knoten) benötigt werden, um den Sachverhalt korrekt abzubilden. Dabei kommt es vor allem darauf an, wie genau das Modell sein soll, um zu bestimmen, wie viele und vor allem welche Komponenten benötigt werden. Einen Vorschlag für die Knoten bilden die fett markierten Komponenten in der obigen Problembeschreibung.

Nun soll bestimmt werden, welche Komponenten als Transitionen und welche als Plätze dargestellt werden sollen. An dieser Stelle können unterschiedliche Zuordnungen gemacht werden, je nachdem wie das Modell konstruiert werden soll. Empfehlenswert ist es jedoch, dass Komponenten, die Umwandlungen, Bearbeitungen, Produktionen oder Reduktionen abbilden, als Transitionen und Komponenten, die eine Lagerung, Anzeige, Speicherung oder Zustand wiedergeben, als Plätze dargestellt werden.

Demnach sollten die Komponenten:

- BARKEEPER OHNE ALC.
- BARKEEPER ALC.
- STOPPER 1
- STOPPER 2
- SENSOR
- GREIFER
- RESTAURANT

8 Anwendungsbeispiel: Biergarten

als Transitionen und die Komponenten:

- BESTELLUNG
- SPÜLE
- FÖRDERBAND 1
- FÖRDERBAND 2
- KREUZUNG
- STEUERUNG
- FÖRDERBAND 3
- TRESEN
- KASSE

als Plätze definiert werden.

Aus der obigen Problembeschreibung wird nun die Netzstruktur in Abb. 8.1 modelliert.

Als nächstes ist von Interesse zu bestimmen, welche Klassen für die Token benötigt werden. Ein Vorschlag ist, die folgenden Klassen mit den dazugehörigen Ausprägungen zu verwenden:

GETRÄNKE := $\{W, L, S, B, Sp, C\} \equiv \{\mathbf{W}$ asser, \mathbf{L} imonade, \mathbf{S} aft, \mathbf{B} ier, \mathbf{S} pirituose, \mathbf{C} ocktail $\}$,

GLÄSER := $\{P, G, Cg\} \equiv \{\mathbf{P}$ innchen, \mathbf{S} tandardglas, \mathbf{C} ocktailglas $\}$,

GELD := $\{1, 2, 3, 5\}$,

SIGNAL := $\{\bullet\}$.

Die Klasse SIGNAL wird verwendet, um Signale bei der Steuerungseinheit darzustellen. Im Prinzip ist ein Token mit der Farbe \bullet wie ein Token aus einem „normalen“ Petri-Netz zu betrachten.

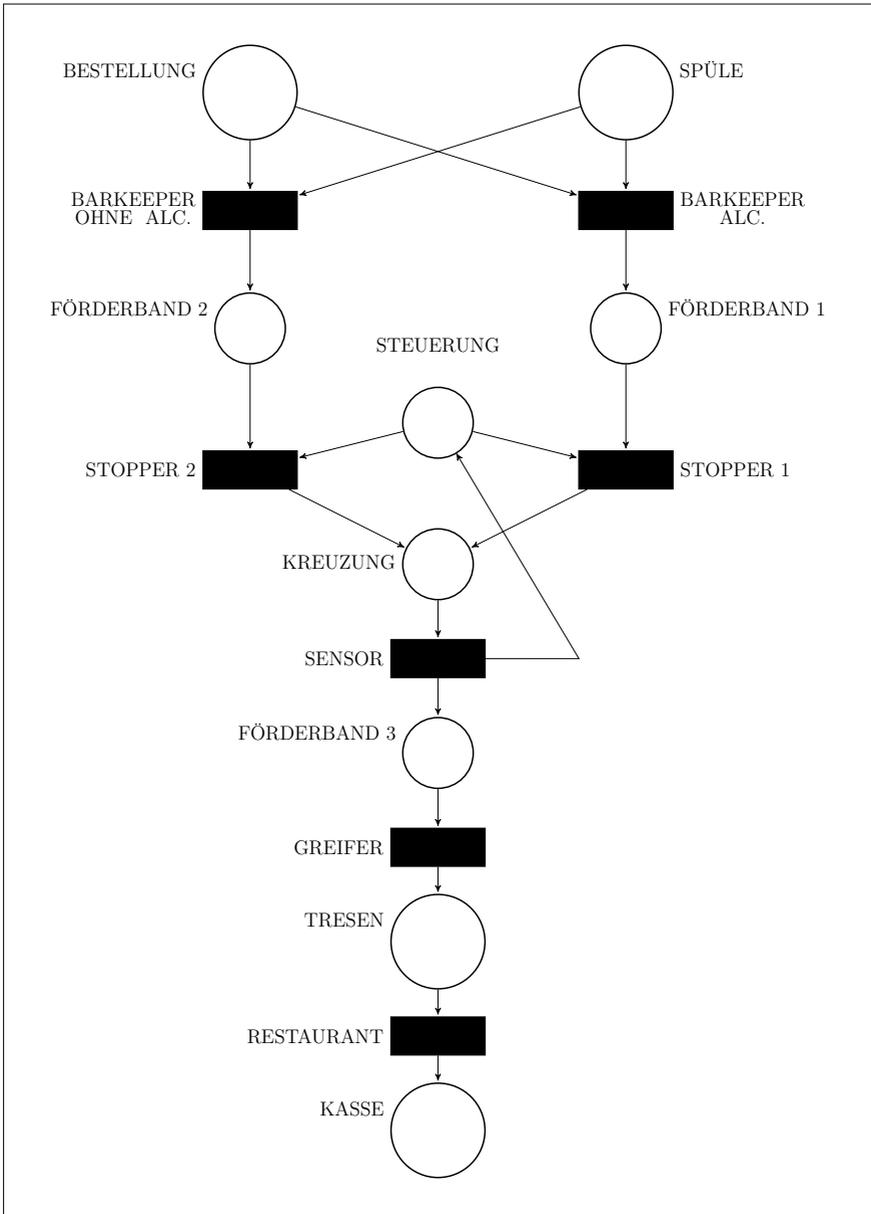


Abbildung 8.1: Netz zur Problemstellung „Biergarten“

8 Anwendungsbeispiel: Biergarten

Hiernach werden den einzelnen Plätzen folgende Klassen zugeordnet:

- $k(\text{BESTELLUNG}) = \text{GETRÄNKE}$
- $k(\text{SPÜLE}) = \text{GLÄSER}$
- $k(\text{FÖRDERBAND 1}) = \text{GETRÄNKE}$
- $k(\text{FÖRDERBAND 2}) = \text{GETRÄNKE}$
- $k(\text{KREUZUNG}) = \text{GETRÄNKE}$
- $k(\text{STEUERUNG}) = \text{SIGNAL}$
- $k(\text{FÖRDERBAND 3}) = \text{GETRÄNKE}$
- $k(\text{TRESEN}) = \text{GETRÄNKE}$
- $k(\text{KASSE}) = \text{GELD}$

Man erhält folgendes Netz mit in Klammern angegebenen Klassen in Abb. 8.2.

Zu den Klassen werden nun die folgenden Variablen

$$V := \{g, gl, \text{€}\},$$

mit den dazugehörigen Klassen

- $k(g) = \text{GETRÄNKE}$
- $k(gl) = \text{GLÄSER}$
- $k(\text{€}) = \text{GELD}$

definiert.

Werden dem Netz nun noch Gewichte an den Pfeilen hinzugefügt, so erhält man das CPN mit Variablen aus Abb. 8.3.

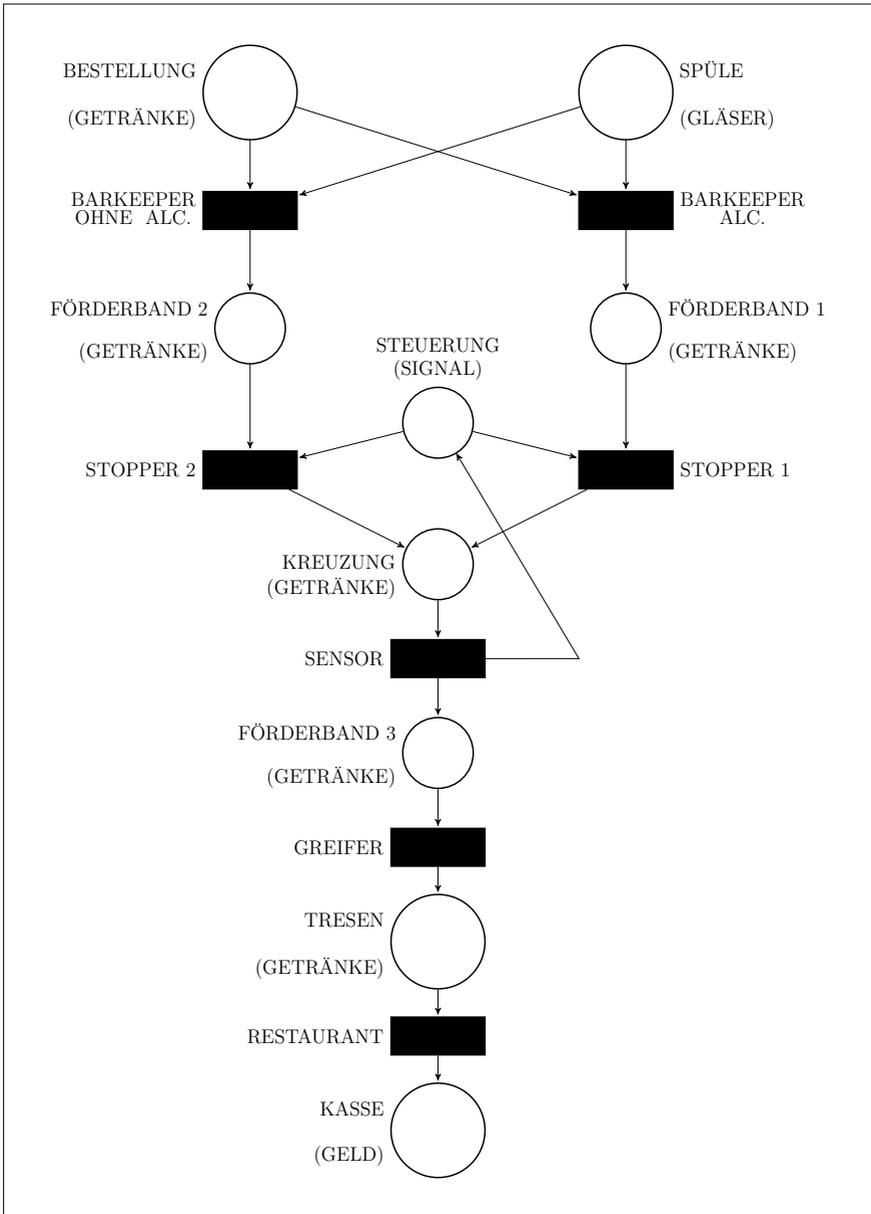


Abbildung 8.2: Netz mit Klassen zum Beispiel „Biergarten“

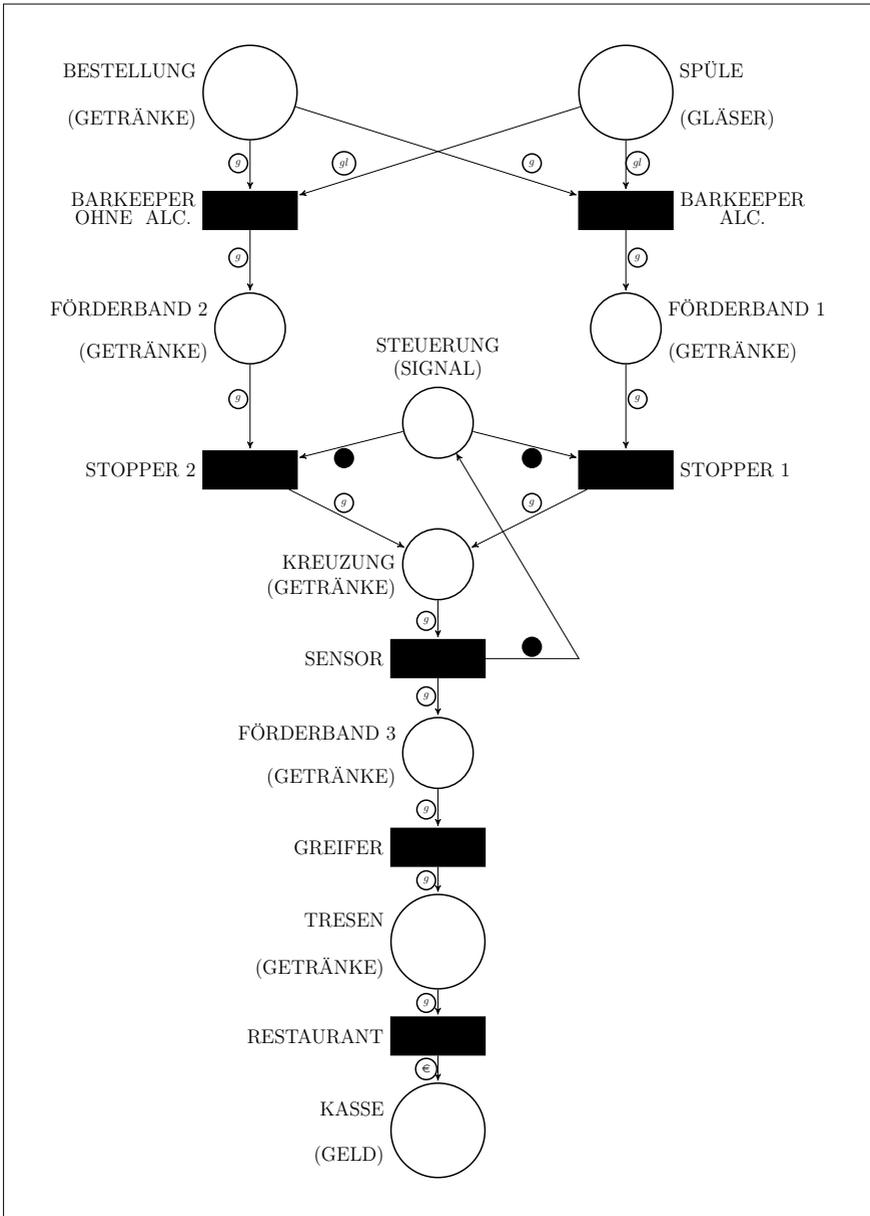


Abbildung 8.3: Netz mit Klassen und variabler Pfeilgewichtung für das Modell „Biergarten“

8.2 Modell eines gefärbten Petri-Netzes mit Variablen

Als Nächstes wird die Modus-Menge \mathbb{B} des Netzes bestimmt, die sich aus den Zuordnungen der Getränke zu Gläsern und Preisen wie folgt ergibt:

$$\mathbb{B} = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$$

mit

$$\beta_1 : \{g = W, gl = G, \epsilon = 1\},$$

$$\beta_2 : \{g = L, gl = G, \epsilon = 3\},$$

$$\beta_3 : \{g = S, gl = Cg, \epsilon = 3\},$$

$$\beta_4 : \{g = B, gl = G, \epsilon = 3\},$$

$$\beta_5 : \{g = Sp, gl = P, \epsilon = 2\},$$

$$\beta_6 : \{g = C, gl = Cg, \epsilon = 5\}.$$

Allerdings dürfen wiederum nicht alle Transitionen in allen diesen Modi feuern, so dass für jede Transition t eine jeweils eigene Modus-Menge $m(t)$ gebildet werden muss, die man der folgenden Tabelle entnehmen kann.

Tabelle 8.1: Zuordnung der einzelnen Modus-Mengen zu den Transitionen des Netzes „Biergarten“

Transition t	$Var(t) :=$	$m(t) :=$
BARKEEPER OHNE ALC.	$\{g, gl\}$	$\{\beta_1, \beta_2, \beta_3\}$
BARKEEPER ALC.	$\{g, gl\}$	$\{\beta_4, \beta_5, \beta_6\}$
STOPPER 1	$\{g\}$	$\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$
STOPPER 2	$\{g\}$	$\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$
SENSOR	$\{g\}$	$\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$
GREIFER	$\{g\}$	$\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$
RESTAURANT	$\{g, \epsilon\}$	$\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6\}$

Da nun für jede Transition eine Modus-Menge bestimmt ist, in der die Transitionen grundsätzlich gefeuert werden könnten, muss noch festgelegt werden, wie der Modus bestimmt wird, in dem gefeuert wird. Hierfür wird der Einfachheit halber die Modi-Wahrscheinlichkeitswahl verwendet, wobei für die Wahrscheinlichkeiten aller Modi innerhalb einer Modus-Menge der Gleichverteilungsansatz unterstellt wird.

Nun werde das spezielle Konfliktverhalten einiger Komponenten betrachtet.

Glasknappheit

Wenn zu wenige Gläser in der Spüle vorhanden sind, um alle bestellten Getränke abzufüllen, dann hat der „alkoholische“ Barkeeper Vorrang. Diese Entscheidung trifft im Modell die Spüle. Dazu wird eine Prioritätsfunktion ϱ definiert, die den Pfeilen (SPÜLE, BARKEEPER ALC.) und (SPÜLE, BARKEEPER OHNE ALC.) die folgenden Prioritäten zuordnet:

$$\begin{aligned} \varrho(\text{SPÜLE, BARKEEPER ALC.}) &:= 1 \\ \varrho(\text{SPÜLE, BARKEEPER OHNE ALC.}) &:= 2 \end{aligned}$$

Mit diesen Prioritäten könnte nun mittels lokalem Prioritätenverfahren eine lokale Lösung des Konfliktes bestimmt werden. Hiermit wäre ein praxisadäquates Verhalten des CPN abgebildet.

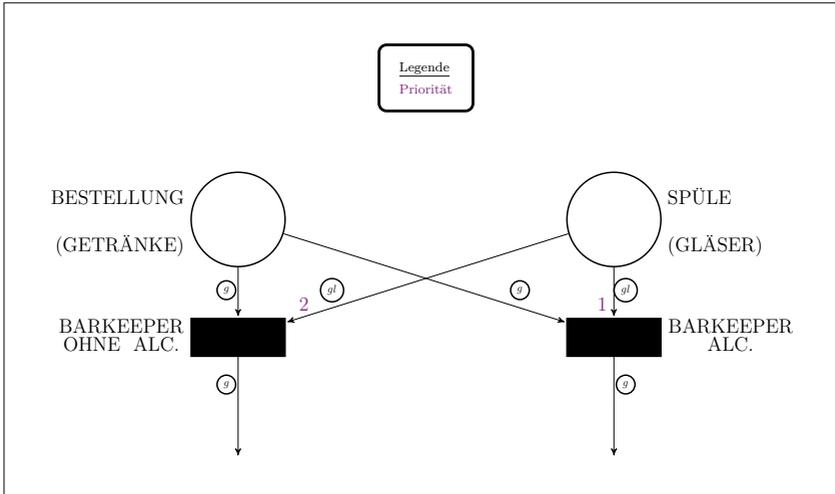


Abbildung 8.4: Konfliktsituation Glasknappheit im Modell „Biergarten“

Kreuzungsengpass

Diese Konfliktsituation tritt ein, wenn zwei Getränke gleichzeitig die Kreuzung passieren sollen. Hierbei wird laut Beobachtung zufällig entschieden, welches Getränk passieren darf und welches erst später. Die Entscheidung darüber trifft die Steuerung der Stopper. Dies wird dadurch erreicht, dass der Platz STEUERUNG nur mit einem einzigen Token

$$z(\text{STEUERUNG}) := \{(\bullet, 1)\}$$

8.2 Modell eines gefärbten Petri-Netzes mit Variablen

markiert ist, so entsteht automatisch ein Konflikt in der STEUERUNG, wenn auf beiden Förderbändern ein Getränk steht, welches die Kreuzung passieren soll. Um diesen Konflikt zu lösen, wird eine Wahrscheinlichkeitsfunktion ω definiert, die den Pfeilen (STEUERUNG, STOPPER 1) und (STEUERUNG, STOPPER 2) dieselben Wahrscheinlichkeiten zuweist:

$$\omega(\text{STEUERUNG, STOPPER 1}) = 0,5,$$

$$\omega(\text{STEUERUNG, STOPPER 2}) = 0,5.$$

Mit diesen Wahrscheinlichkeiten kann nun mittels lokalem Wahrscheinlichkeitsverfahren eine lokale Lösung des Konfliktes bestimmt werden. Durch die Rückführung eines Token durch die Transition SENSOR ist gewährleistet, dass die STEUERUNG die KREUZUNG wieder frei gibt. Hiermit wäre wieder ein praxisadäquates Verhalten des Petri-Netzes abgebildet.

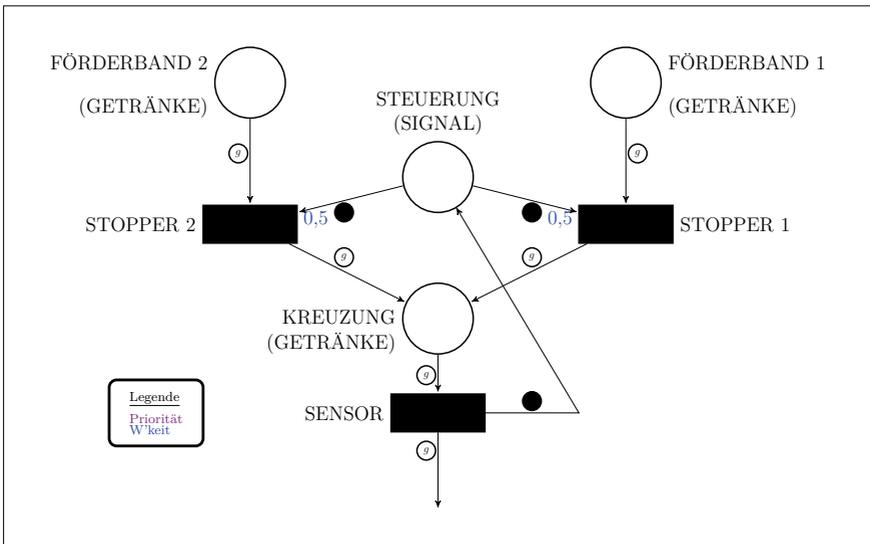


Abbildung 8.5: Kreuzungsengpass im Modell „Biergarten“

8 Anwendungsbeispiel: Biergarten

Als letztes muss nun noch für jeden Platz ein Anfangszustand bestimmt werden, der sich aus den aktuellen Bestellungen und verfügbaren Gläsern in der Spüle wie folgt ergibt:

$$\begin{aligned}z(\text{BESTELLUNG}) &:= \{(W, 2), (L, 1), (S, 1), (B, 2), (Sp, 1), (C, 2)\} \\z(\text{SPÜLE}) &:= \{(P, 2), (G, 4), (Cg, 3)\} \\z(\text{FÖRDERBAND 1}) &:= \emptyset \\z(\text{FÖRDERBAND 2}) &:= \emptyset \\z(\text{KREUZUNG}) &:= \emptyset \\z(\text{STEUERUNG}) &:= \{(\bullet, 1)\} \\z(\text{FÖRDERBAND 3}) &:= \emptyset \\z(\text{TRESEN}) &:= \emptyset \\z(\text{KASSE}) &:= \emptyset.\end{aligned}$$

Hierdurch erhält man das endgültige Modell aus Abb. 8.6.

Mit einer Erweiterung dieses Modells könnte der Ablauf der Getränkeabfüllung in der Bar an einem Abend simuliert werden. Allgemein können Petri-Netz-Modelle Aufschluss über komplexe Zusammenhänge in einem Prozess geben, indem man die Startbedingungen oder bestimmte Verhaltensregeln im Modell ändert und deren Auswirkungen simuliert. Zum Beispiel könnte eine Untersuchung der Konfliktlösungen des Modells stattfinden, und zwar in Bezug darauf, ob effizientere Strategien existieren und um damit eine Optimierung des Prozesses zu erwirken. Des Weiteren könnte dieses Modell „beliebig“ erweitert werden, z.B. um ein Zurückkommen von gebrauchten Gläsern zur Spüle, Neubestellungen von den Gästen oder eine Knappheit von „Rohstoffen“ zur Getränkeabfüllung. Interessant wäre auch die Transition RESTAURANT durch Komponenten wie Kellner oder Gäste zu ersetzen und so ein genaueres Verhalten des Prozesses zu erzeugen.

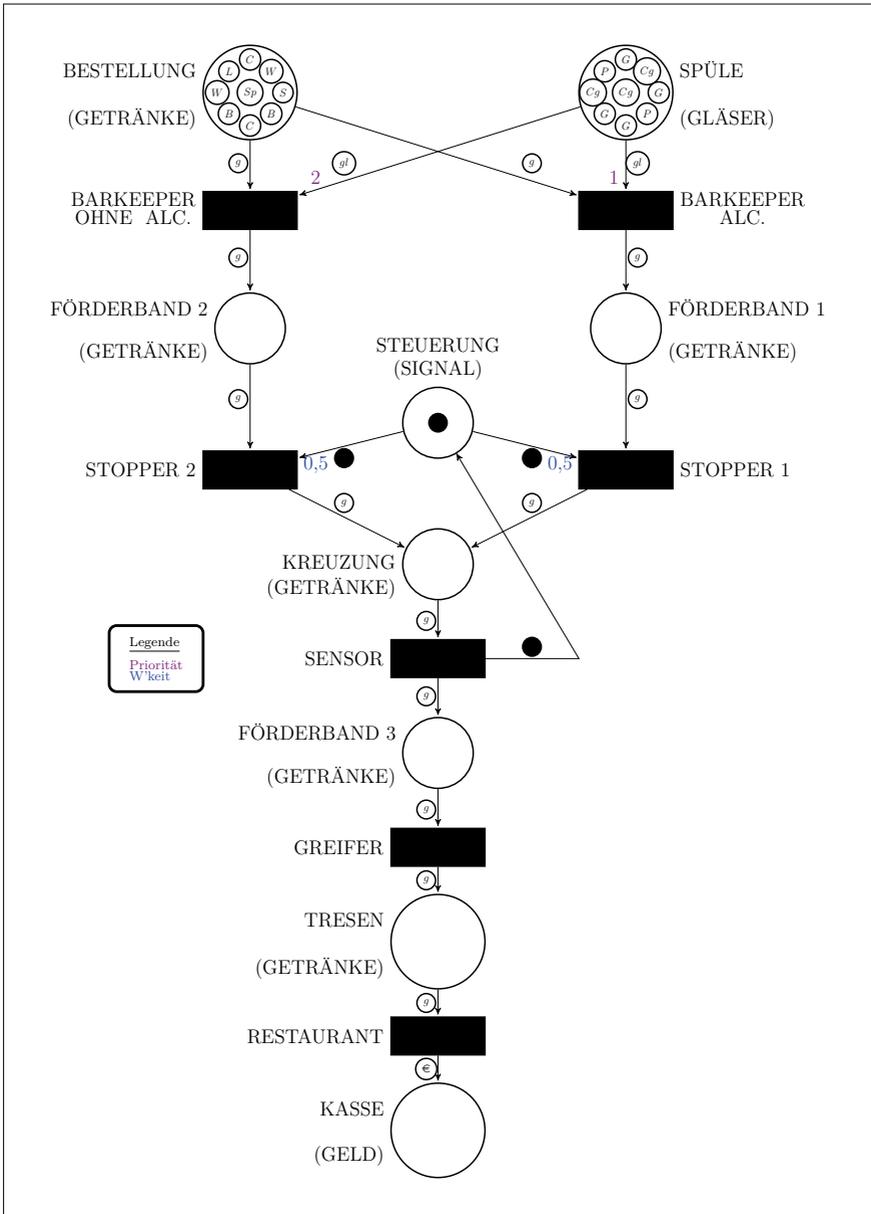


Abbildung 8.6: Endgültiges Modell „Biergarten“

9 Ausblick

In diesem Buch wurde eine anschauliche Einführung in das Gebiet der (gefärbten) Petri-Netze und ihrer Formalismen gegeben, die für eine „einfache“ Simulation benötigt werden. Allerdings sind mit den bis jetzt vorgestellten Elementen nur sehr eingeschränkte Petri-Netz-Modelle und darauf basierende Simulationen möglich. In nächster Zukunft werden im Rahmen des CPN-Projektes des Forschungs- und Entwicklungsschwerpunktes AMMO weitere Entwicklungen im Bereich der Petri-Netz-Simulation folgen.

Geplant sind die folgenden Erweiterungen:

- a) CPN mit Variablen sollen auf „*allgemeine*“ gefärbte Petri-Netze (vgl. [JK09]) erweitert werden, um so einen Formalismus zu erhalten, mit dem die meisten gängigen CPN-Modelle abgebildet werden können.
- b) Eine Einführung der Konzepte der *kapazitierten Petri-Netze* (vgl. [Pro13]) und deren Übertragung auf die gefärbten Petri-Netze.
- c) Eine Einführung der Konzepte der *erweiterten Petri-Netze* (vgl. [Pro13]) und deren Übertragung auf die gefärbten Petri-Netze.
- d) Für viele Simulationen ist der Begriff der „Zeit“ von entscheidender Bedeutung; daher sollen die PN und CPN um verschiedene Zeit-Konzepte (vgl. [Pro13] und [Ger04]) erweitert werden.

An dieser Stelle soll abschließend ein kurzer, anschaulicher Einblick auf die zukünftigen Erweiterungen gegeben werden.

9.1 „Allgemeine“ gefärbte Petri-Netze

Die bis jetzt behandelten CPN mit Variablen (vgl. Kap. 6) bilden die Grundlage für die Theorie der gefärbten Petri-Netze mit Variablen und daher nur einen einfachen Spezialfall

der „allgemeinen“ CPN, ähnlich wie bei dem Zusammenhang von SCPN und MCPN (vgl. Kap. 4). Es existiert bereits eine Vielzahl von komplexen Erweiterungen in diesem Gebiet. Diese Erweiterungen werden in mancher Literatur auch als Prädikaten-Transitions-Netze (vgl. [Ger04]) bezeichnet. Solche Erweiterungen führen zum sog. "non-hierarchical Coloured Petri Net" von [JK09, S. 87].

Bei diesen CPN wird als Pfeilgewichtung keine Multimenge aus Konstanten und Variablen mehr verwendet, sondern ein jeweilig passender *Ausdruck*. Dafür wird eine *Ausdrucksmenge* definiert (vgl. [JK09, S.84]), welche alle Ausdrücke eines CPN enthält. Diese Ausdrücke werden in der Regel recht unmathematisch in einer gewissermaßen sprachlichen Algebra (vgl. [Ger04]) oder in einer Pseudoprogrammiersprache (vgl. [JK09]) definiert, wie das folgende Beispiel zeigt.

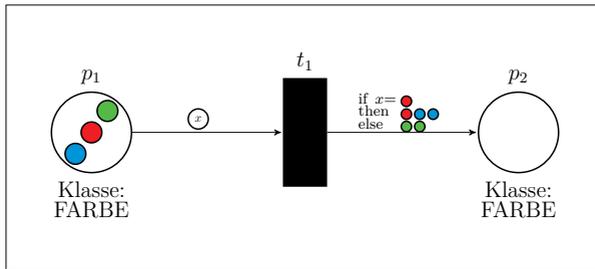


Abbildung 9.1: Allgemeines CPN mit Pseudoprogrammiersprache

Diese Ausdrücke stellen im mathematischen Sinne Abbildungen dar. Anders als zu vermuten sind diese Abbildungen nicht von den Variablen im Petri-Netz abhängig, sondern von dem Modus, in welchem die jeweilige Transition gefeuert wird. Bei dem obigen Beispiel würde für t_1 mit der Modus-Menge

$$m(t_1) = \{\beta_1, \beta_2, \beta_3\}$$

und den Modi

$$\begin{aligned} \beta_1 &: \{x = \text{ROT}\}, \\ \beta_2 &: \{x = \text{BLAU}\}, \\ \beta_3 &: \{x = \text{GRÜN}\}. \end{aligned}$$

der Ausdruck

if $x = \{(\text{ROT}, 1)\}$
 then $\{(\text{ROT}, 1), (\text{BLAU}, 2)\}$
 else $\{(\text{GRÜN}, 2)\}$

der folgenden Abbildung

$$y(\beta) = \begin{cases} \{(\text{ROT}, 1), (\text{BLAU}, 2)\} & \text{falls } \beta = \beta_1 \\ \{(\text{GRÜN}, 2)\} & \text{falls } \beta = \beta_2 \\ \{(\text{GRÜN}, 2)\} & \text{falls } \beta = \beta_3 \end{cases}$$

entsprechen.

Diese „allgemeinen“ CPN ermöglichen es, sehr komplexe Modelle recht übersichtlich zu modellieren und eine große Bandbreite von Problemstellungen zu bewältigen. Die Problematik an ihnen ist allerdings, dass immer eine *Formalsprache* für die Ausdrücke verwendet wird, die ein Benutzer lernen und auch verstehen muss.

9.2 Kapazitierte (gefärbte) Petri-Netze

In vielen Situationen des alltäglichen Lebens gibt es naturgemäß bestimmte Ober- und Unterschranken für die Anzahl oder Menge von bestimmten Elementen, wie z.B. dass maximal 10 Personen in einem Fahrstuhl fahren dürfen oder das mindestens 2 Arbeiter notwendig sind, um eine Tätigkeit auszuführen. Um diese Sachverhalte abbilden zu können, verwendet man die *kapazitierten Petri-Netze* (vgl. [Pro13]). Diese Petri-Netze haben eine *Minimalkapazität* und eine *Maximalkapazität*, welche die minimale bzw. maximale Anzahl von Token angibt, die im jeweiligen Platz enthalten sein kann.

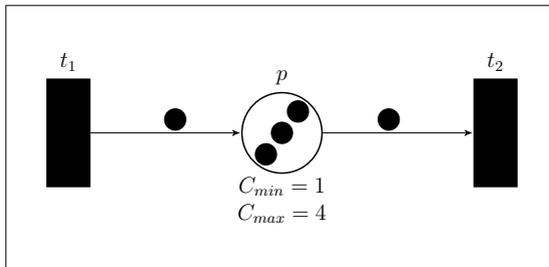


Abbildung 9.2: Petri-Netz mit Kapazitäten

Im obigen Beispiel müsste der Platz p mindestens ein Token und dürfte maximal vier Token enthalten. Diese Kapazitäten nehmen wesentlichen Einfluss darauf, ob eine Transition aktiviert ist. Denn wenn durch das Feuern einer Transition eine dieser Bedingungen verletzt würde, ist diese Transition nicht aktiviert. Dadurch entstehen zwei neue Konfliktsituationen und zwar der *kapazitive Input-Konflikt* und der *kapazitive Output-Konflikt*.

Die Konfliktlösungskonzepte (vgl. Kap. 2) und Feuerszenarien (vgl. Kap. 3) sollen auf kapazitierte PN angepasst und auf kapazitierte CPN übertragen werden.

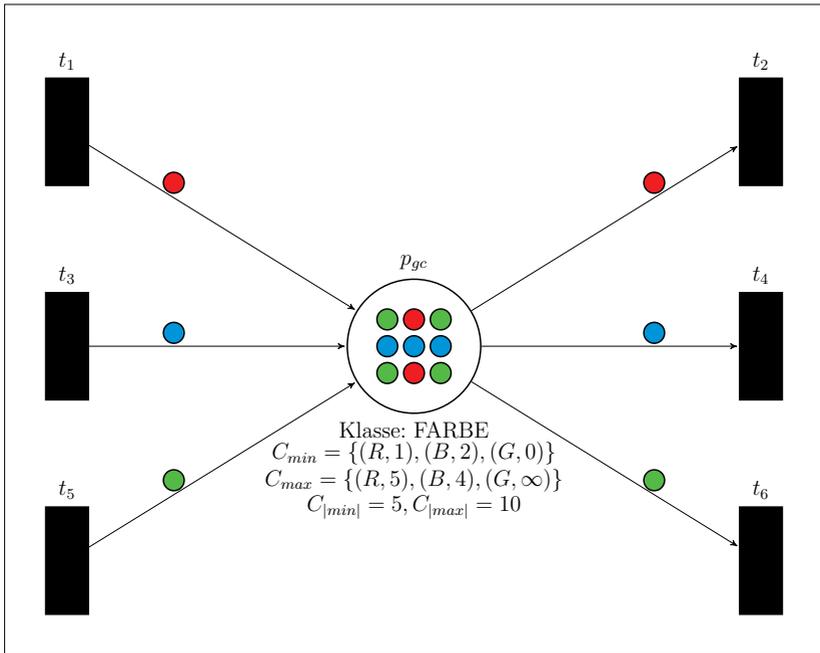


Abbildung 9.3: Kapazitiertes gefärbtes Petri-Netze mit Gesamtkapazität

Bei dem gefärbten Petri-Netz im obigen Beispiel müsste der Zustand von p_{gc} mindestens $\{(ROT, 1), (BLAU, 2), (GRÜN, 0)\}$ und maximal $\{(ROT, 5), (BLAU, 4), (GRÜN, \infty)\}$ sein. Zusätzlich könnte man noch die Bedingungen einfügen, dass eine minimale und maximale Gesamtanzahl von Tokens in einem Platz enthalten sein muss, z.B. in p_{gc} müssen mindestens fünf und dürfen maximal zehn Token beliebiger Farbe enthalten sein (*Gesamtkapazitäten*).

9.3 Erweiterte (gefärbte) Petri-Netze

Viele Prozesse in der Realität hängen davon ab, dass ein bestimmter Zustand eintritt, z.B. eine Ampel springt nur dann auf Grün um, falls ein Auto an der Kreuzung steht, oder ein Männerwitz wird nur erzählt, wenn sich keine Frau im selben Raum befindet. Bei diesen Situationen hängt die Durchführung eines Prozesses nur vom Zustand ab, der selber nicht verändert wird. Für solche Problemstellungen werden die *erweiterten Petri-Netze* (vgl. [Pro13]) verwendet, bei denen spezielle Pfeile eingeführt werden, und zwar *Testpfeile* (gestrichelte Pfeile) und *Hemmpfeile* (Pfeile mit Kreisspitze).

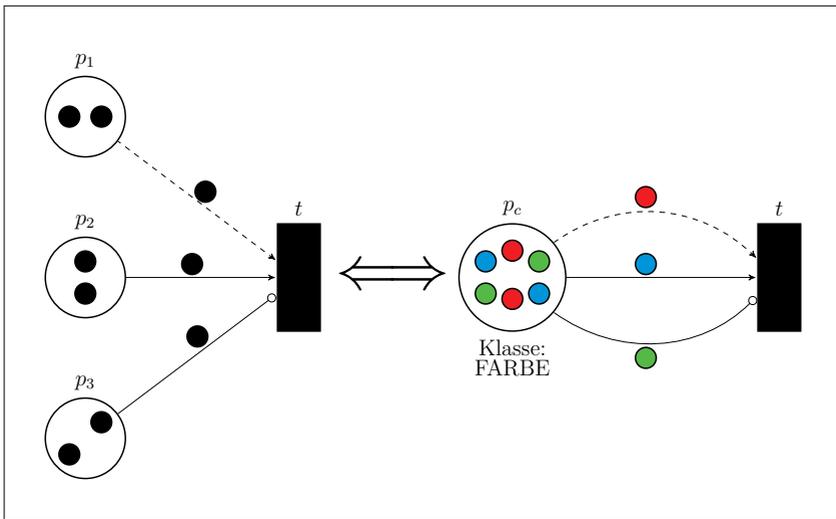


Abbildung 9.4: (Gefärbtes) Petri-Netz mit Testkante und Hemmkante

Testpfeile sorgen dafür, dass eine Transition nur aktiviert ist, wenn der Zustand des Platzes echt größer ist als das Pfeilgewicht des Testpfeils. Im obigen Beispiel kann t nur aktiviert sein, wenn in p_1 mehr als ein Token ist. Hemmpfeile sorgen dafür, dass eine Transition nur aktiviert ist, wenn der Zustand des Platzes echt kleiner ist als das Pfeilgewicht des Hemmpfeils. Im obigen Beispiel kann t nur aktiviert sein, wenn in p_3 weniger als ein Token ist. Bei beiden Pfeilsorten wird beim Feuern der Zustand des verbundenen Platzes nicht verändert.

Diese Konzepte lassen sich analog auch auf den gefärbten Fall übertragen, wie im obigen Beispiel angedeutet wird.

9.4 (Gefärbte) Petri-Netze mit Zeit

Die meisten Prozesse haben eine starke Zeitabhängigkeit. Eine Ampel ist beispielsweise 25 Sekunden grün, 5 Sekunden gelb und dann erst rot; eine Maschine, die im 2-Sekunden-Takt eine bestimmte Tätigkeit ausführt, oder das Befüllen eines Glases dauert 8 Sekunden. Daher braucht man für die Simulation solcher Prozesse auch einen Zeitbegriff. Für (gefärbte) Petri-Netze gibt es einige Zeitvarianten (vgl. [Pro13], [Ger04], [JK09]). Zu den bereits in der Modelica-Petri-Netz-Bibliothek PNlib vorhandenen sollen weitere Zeitvarianten implementiert werden.

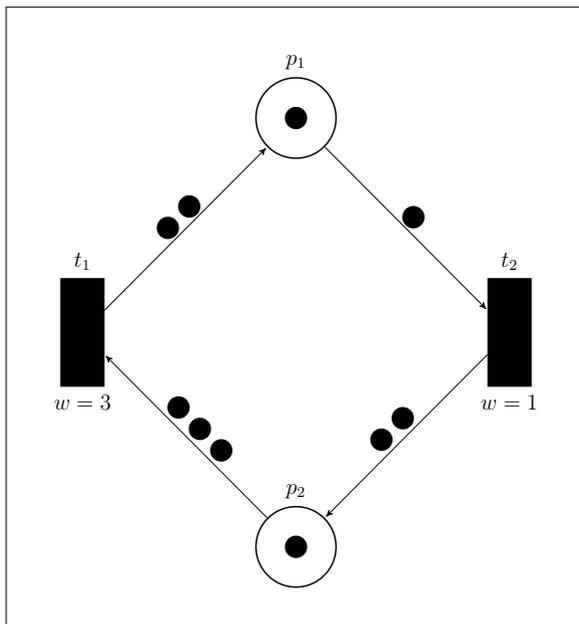


Abbildung 9.5: Petri-Netz mit Wartezeiten

Im obigen Petri-Netz sind den Transitionen *Wartezeiten* zugewiesen (vgl. [Pro13]); dies bedeutet, dass eine Transition erst feuert, wenn nach ihrer Aktivierung eine bestimmte Zeit abgelaufen ist. Im Beispiel feuert t_1 erst, wenn drei Zeiteinheiten abgelaufen sind, und t_2 erst, wenn eine Zeiteinheit abgelaufen ist. Der Zustandsverlauf der beiden Plätze p_1 und p_2 kann im unteren Diagramm verfolgt werden.

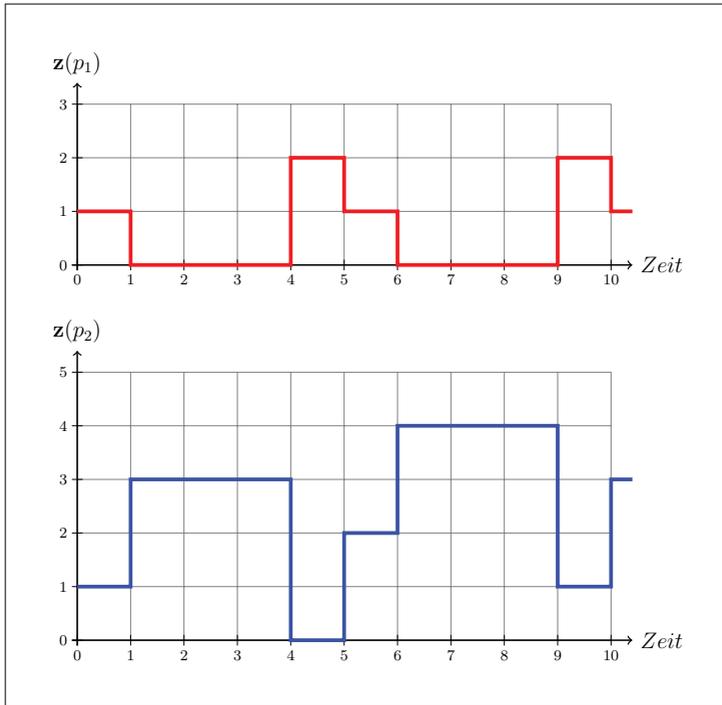


Abbildung 9.6: Zustandsverläufe des Petri-Netzes mit Wartezeiten

Symbolverzeichnis

A	Menge von Attributen, Seite 69
A_i	Attribut, Seite 69
a^i	Ausprägung von Attribut A_i , Seite 69
B	Menge von Pfeilen von T nach P , Seite 5
\mathbb{B}	Modus-Menge eines CPN, Seite 118
β bzw. β_i	Modus, Seite 115
η	Nutzenquotient, Seite 31
F	Menge von Pfeilen von P nach T , Seite 5
f_β	Gewichtungsfunktion im Modus β , Seite 116
f	Gewichtungsfunktion, Seite 6
h	Häufigkeitsfunktion einer Multimenge, Seite 170
K	Menge von Klassen, Seite 70
K_j	Klasse, Seite 70
\mathbb{K}	Vereinigungsmenge aller Klassen, Seite 70
k	Klassifizierungsfunktion, Seite 70
$\mathbb{M}(A)$	Menge aller Multimengen über eine Menge A , Seite 175
$m(t)$	Modus-Menge einer Transition $t \in T$, Seite 118
N	Netz bzw. Petri-Netz, Seite 5

Symbolverzeichnis

ω	Allgemeine Wahrscheinlichkeitsfunktion, Seite 22
ω_z	Aktualisierte Wahrscheinlichkeitsfunktion, Seite 22
P	Menge von Plätzen, Seite 5
p oder p_i	Platz, Seite 5
$\bullet p$	Vorbereich von Platz p , Seite 6
$p\bullet$	Nachbereich von Platz p , Seite 6
ψ	Bewertungsfunktion, Seite 31
ϱ	Prioritätsfunktion, Seite 19
σ	Allgemeine Bewertungsfunktion, Seite 19
$\text{supp}(M)$	Stützmenge der Multimenge M , Seite 171
T	Menge von Transitionen, Seite 5
t oder t_j	Transition, Seite 5
$\bullet t$	Vorbereich von Transition t , Seite 6
$t\bullet$	Nachbereich von Transition t , Seite 6
$\bullet\{t_1, \dots, t_r\}$	Gemeinsamer Vorbereich der Transitionen t_1, \dots, t_r , Seite 10
TA	Menge aller aktivierten Transitionen eines Petri-Netzes, Seite 16
TA_{in}	Menge der aktivierten Transitionen des Vorbereichs eines Platzes, Seite 16
TA_{out}	Menge der aktivierten Transitionen des Nachbereichs eines Platzes, Seite 16
TE	Menge der global freigegebenen Transitionen eines Petri-Netzes, Seite 41
TE_{out}	Menge der lokal freigegebenen Transitionen des Nachbereichs eines Platzes, Seite 18

\mathbb{V}	Menge von Variablen, Seite 111
v bzw. v_n	Variable, Seite 111
$Var(t)$	Variablenmenge einer Transition $t \in T$, Seite 115
\mathbf{z}	Zustand eines Petri-Netzes, Seite 6
z_i bzw. $\mathbf{z}(p_i)$	Zustand eines Platzes, Seite 6

Literaturverzeichnis

- [Bau97] BAUMGARTEN, B.: Petri-Netze – Grundlagen und Anwendungen. Heidelberg, 2. Auflage, 1997.
- [BM08] BONDY, J.A. und MURTY, U.S.R.: Graph Theory. Berlin-Heidelberg, 2008.
- [Bra94] BRANDSTÄDT, A.: Graphen und Algorithmen. Stuttgart, 1994.
- [BY07] BOSTAN-KORPEOGLU, B. und YAZICI, A.: A Fuzzy Petri Net Model for Intelligent Databases. *Data & Knowledge Engineering*, 62:219–247, 2007.
- [Die10] DIESTEL, R.: Graphentheorie. Stuttgart, 4. Auflage, 2010.
- [Ger04] GERBER, S.: Petri-Netze 2. Institut für Informatik, Universität Leipzig, Sommersemester 2004.
- [JK09] JENSEN, K. und KRISTENSEN, L. M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Berlin-Heidelberg, 2009.
- [Kä06] KÄMPF, M.: Probleme der Tourenbildung. Chemnitzer Informatik Bericht, Fakultät für Informatik, Technische Universität Chemnitz, 2006.
- [KC11] KRUSE, H.-J. und CAK, K.: Eine Einführung in die Simulationssoftware AutoMod. Version 0.1. Manuskript im Bachelor-Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2011.
- [KD13] KLEINE-DÖPKE, T.: Konfliktlösungen für den nebenläufigen Feuerprozess bei gefärbten Petri-Netzen und ihre Anwendungen. Bachelorarbeit im Studiengang Angewandte Mathematik der Fachhochschule Bielefeld, 2013.
- [Krc0J] KRCCMAR, H.: Vorlesung Petri-Netze. Lehrstuhl für Wirtschaftsinformatik, Universität Hohenheim, o.J.

- [KV12] KORTE, B. und VYGEN, J.: Kombinatorische Optimierung – Theorie und Algorithmen. 2. Auflage, 2012.
- [Lak95] LAKOS, C.: From Coloured Petri nets to Object Petri Nets. In: de Michelis G, Diaz M (eds): Applications and Theory of Petri Nets 1995. Lecture Notes in Computer Science, 935:278–297, 1995.
- [Mon01] MONRO, G.P.: The Concept of Multiset. In: Proceedings of the Workshop on Multiset Processing, 347-358, London 2001.
- [Nol76] NOLTEMEIER, H.: Graphentheorie mit Algorithmen und Anwendungen. Berlin-New York, 1976.
- [NS96] NÄGLER, G. und STOPP, F.: Graphen und Anwendungen. Stuttgart, 1996.
- [Pro13] PROSS, S.: Hybrid Modeling and Optimization of Biological Processes. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Band 1, 2013.
- [PW03] PRIESE, L. und WIMMEL, H.: Theoretische Informatik – Petri-Netze. Berlin-Heidelberg, 2003.
- [Rei82] REISIG, W.: Petrinetze – Eine Einführung. Berlin-Heidelberg, 1982.
- [Rei10] REISIG, W.: Petrinetze. Wiesbaden, 2010.
- [Sch03] SCHRIJVER, A.: Combinatorial Optimization. Berlin-Heidelberg, 2003.
- [Sta90] STARKE, P.H.: Analyse von Petri-Netz-Modellen. Stuttgart, 1990.
- [Syr87] SYROPOULOS, A.: Mathematics of Multisets. Zeitschrift f. math. Logik und Grundlagen d. Math., 33:171–178, 1987.

Anhang: Multimengen

Eine Multimenge ist eine modifizierte Form des Mengenbegriffs aus der klassischen Mengenlehre nach Cantor, indem es bei einer Multimenge erlaubt ist, ein und dasselbe Element mehrmals zu enthalten. Dementsprechend müssen auch die Mengenoperationen auf Multimengen modifiziert werden (hierzu siehe [Mon01], [Syr87]).

Definition A.1:

Eine **Multimenge** M über einer klassischen Menge A wird als eine klassische Menge geordneter Paare $(a, h(a))$ aufgefasst, wobei gilt:

$$\begin{aligned} a &\in A \\ h &: A \rightarrow \mathbb{N}_0 \end{aligned}$$

Dabei gibt $h(a)$ die Häufigkeit des Elementes a in M an. Die klassische Menge A wird auch als die **Grundmenge** der Multimenge M und h als die **Häufigkeitsfunktion** von M bezeichnet.

Beispiel:

(a) Ein Beispiel einer Multimenge über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ ist:

$$M = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\},$$

was auch interpretiert werden kann als

$$M = \{\text{ROT}, \text{ROT}, \text{ROT}, \text{BLAU}, \text{BLAU}, \text{GRÜN}\}$$

oder auch ungeordnet als

$$M = \{\text{ROT}, \text{BLAU}, \text{ROT}, \text{GRÜN}, \text{ROT}, \text{BLAU}\}.$$

(b) Ein weiteres Beispiel einer Multimenge über der Menge A ist:

$$M = \{(\text{ROT}, 2), (\text{BLAU}, 0), (\text{GRÜN}, 1)\},$$

was auch interpretiert werden kann als

$$M = \{\text{ROT}, \text{ROT}, \text{GRÜN}\}$$

oder auch ungeordnet als

$$M = \{\text{ROT}, \text{GRÜN}, \text{ROT}\} = \{\text{GRÜN}, \text{ROT}, \text{ROT}\}.$$

Definition A.2:

Es seien A eine klassische Menge und $M = \{(a, h(a)) \mid a \in A, h(a) \in \mathbb{N}_0\}$ eine Multimenge über A . Die klassische Teilmenge $\text{supp}(M) = \{a \in A \mid h(a) > 0\}$ von A wird *stützende Menge*, *Stützmenge* oder *reduzierte Grundmenge* der Multimenge M genannt.

Im Extremfall kann die stützende Menge einer Multimenge die Grundmenge selber sein ($\text{supp}(M) = A$), wenn es kein Element $a \in A$ mit $h(a) = 0$ gibt. Umgekehrt kann die stützende Menge aber im anderen Extremfall auch die leere Menge \emptyset sein, wenn nämlich jedem Element $a \in A$ die Häufigkeit $h(a) = 0$ zugewiesen wird.

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ sei die Multimenge:

$$M = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 0)\}.$$

Die Stützmenge von M ist

$$\text{supp}(M) = \{\text{ROT}, \text{BLAU}\}.$$

Definition A.3:

Zu einer Grundmenge A sei die Multimenge $M = \{(a, h(a)) \mid a \in A, h(a) \in \mathbb{N}_0\}$ über A gegeben. Dann ist die **Mächtigkeit der Multimenge** M definiert als:

$$|M| := \sum_{a \in A} h(a)$$

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ sei die Multimenge:

$$M = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 0)\}.$$

Die Mächtigkeit der Multimenge von M ist

$$|M| = h(\text{ROT}) + h(\text{BLAU}) + h(\text{GRÜN}) = 3 + 2 + 0 = 5.$$

Definition A.4:

Zu einer Grundmenge A seien die Multimengen $M_1 = \{(a, h_1(a)) \mid a \in A, h_1(a) \in \mathbb{N}_0\}$ und $M_2 = \{(a, h_2(a)) \mid a \in A, h_2(a) \in \mathbb{N}_0\}$ über A gegeben.

- M_1 heißt **Teilmenge** von M_2 ($M_1 \subseteq M_2$), wenn $\forall a \in A : h_1(a) \leq h_2(a)$.
- M_1 heißt **echte Teilmenge** von M_2 ($M_1 \subset M_2$), wenn $M_1 \subseteq M_2$ und $\exists a \in A : h_1(a) < h_2(a)$.
- M_1 und M_2 heißen **gleich** ($M_1 = M_2$), wenn $\forall a \in A : h_1(a) = h_2(a)$.

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ seien die drei Multimengen:

$$M_1 = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\},$$

$$M_2 = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\},$$

$$M_3 = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\}.$$

(a) M_2 ist eine (echte) Teilmenge von M_1 ($M_2 \subseteq M_1$):

$$h_2(\text{ROT}) = 2 \leq 3 = h_1(\text{ROT}),$$

$$h_2(\text{BLAU}) = 1 \leq 2 = h_1(\text{BLAU}),$$

$$h_2(\text{GRÜN}) = 1 \leq 1 = h_1(\text{GRÜN}).$$

(b) M_3 und M_1 sind gleich ($M_3 = M_1$):

$$\begin{aligned}h_3(\text{ROT}) &= 3 = h_1(\text{ROT}), \\h_3(\text{BLAU}) &= 2 = h_1(\text{BLAU}), \\h_3(\text{GRÜN}) &= 1 = h_1(\text{GRÜN}).\end{aligned}$$

Definition A.5:

Zu einer Grundmenge A seien die Multimengen $M_1 = \{(a, h_1(a)) \mid a \in A, h_1(a) \in \mathbb{N}_0\}$ und $M_2 = \{(a, h_2(a)) \mid a \in A, h_2(a) \in \mathbb{N}_0\}$ über A gegeben. Dann heißen die Multimengen

- $M_1 \uplus M_2 := \{(a, s(a)) \mid a \in A, s(a) = h_1(a) + h_2(a)\}$ die **Vereinigungsmenge** (oder **Summe**) von M_1 und M_2 ,
- $M_1 \setminus M_2 := \{(a, d(a)) \mid a \in A, d(a) = \max(h_1(a) - h_2(a); 0)\}$ die **Differenzmenge** (oder **Differenz**) von M_1 und M_2 .

Bemerkung:

Im Fall $M_2 \subseteq M_1$ ergibt sich die Differenz $M_1 \setminus M_2$ vereinfacht als

$$M_1 \setminus M_2 = \{(a, d(a)) \mid a \in A, d(a) = h_1(a) - h_2(a)\}.$$

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ seien die zwei Multimengen:

$$\begin{aligned}M_1 &= \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\}, \\M_2 &= \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\}.\end{aligned}$$

(a) Die Vereinigungsmenge von M_1 und M_2 ist

$$\begin{aligned}M_1 \uplus M_2 &= \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \uplus \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\} \\&= \{(\text{ROT}, 2+2), (\text{BLAU}, 2+1), (\text{GRÜN}, 1+2)\} \\&= \{(\text{ROT}, 4), (\text{BLAU}, 3), (\text{GRÜN}, 3)\}.\end{aligned}$$

(b) Die Differenzmenge von M_1 und M_2 ist

$$\begin{aligned} M_1 \setminus M_2 &= \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \setminus \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\} \\ &= \{(\text{ROT}, \max(2 - 2; 0)), (\text{BLAU}, \max(2 - 1; 0)), (\text{GRÜN}, \max(1 - 2; 0))\} \\ &= \{(\text{ROT}, 0), (\text{BLAU}, 1), (\text{GRÜN}, 0)\}. \end{aligned}$$

Definition A.6:

Zu zwei Grundmengen A und B seien die Multimengen $M_1 = \{(a, h_1(a)) \mid a \in A, h_1(a) \in \mathbb{N}_0\}$ über A und $M_2 = \{(a, h_2(a)) \mid a \in B, h_2(a) \in \mathbb{N}_0\}$ über B gegeben. Dann heißt die Multimenge

$$M_1 \uplus M_2 := \left\{ (a, s(a)) \mid a \in (A \cup B), s(a) = \begin{cases} h_1(a) + h_2(a) & \text{für } a \in A \wedge a \in B \\ h_1(a) & \text{für } a \in A \wedge a \notin B \\ h_2(a) & \text{für } a \notin A \wedge a \in B \end{cases} \right\}$$

die **Vereinigungsmenge** (oder **Summe**) von M_1 und M_2 .

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ sei die Multimenge

$$M_1 = \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\}$$

und über der Menge $B = \{\text{ROT}, \text{PINK}, \text{SCHWARZ}\}$ sei die Multimenge

$$M_2 = \{(\text{ROT}, 2), (\text{PINK}, 1), (\text{SCHWARZ}, 2)\}.$$

Die Vereinigungsmenge von M_1 und M_2 ist

$$\begin{aligned} M_1 \uplus M_2 &= \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\} \uplus \{(\text{ROT}, 2), (\text{PINK}, 1), (\text{SCHWARZ}, 2)\} \\ &= \{(\text{ROT}, 2 + 2), (\text{BLAU}, 2), (\text{GRÜN}, 1), (\text{PINK}, 1), (\text{SCHWARZ}, 2)\} \\ &= \{(\text{ROT}, 4), (\text{BLAU}, 2), (\text{GRÜN}, 1), (\text{PINK}, 1), (\text{SCHWARZ}, 2)\}. \end{aligned}$$

Definition A.7:

Es seien N Multimengen $M_i = \{(a, h_i(a)) \mid a \in A, h_i(a) \in \mathbb{N}_0\}$ von $i = 1, \dots, N$ über A gegeben. Dann heißt die Multimenge

$$\biguplus_{i=1, \dots, N} M_i := M_1 \uplus M_2 \uplus M_3 \uplus \dots \uplus M_{N-1} \uplus M_N$$

die **N -äre Vereinigungsmenge** (oder **N -äre Summe**) aller M_i .

Beispiel:

Gegeben über der Menge $A = \{\text{ROT}, \text{BLAU}, \text{GRÜN}\}$ seien die vier Multimengen:

$$M_1 = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\},$$

$$M_2 = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\},$$

$$M_3 = \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 0)\},$$

$$M_4 = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\}.$$

Die N -äre Vereinigungsmenge aller M_i ist

$$\biguplus_{i=1, \dots, 4} M_i = M_1 \uplus M_2 \uplus M_3 \uplus M_4 = \{(\text{ROT}, 9), (\text{BLAU}, 6), (\text{GRÜN}, 3)\}.$$

Definition A.8:

Es sei A eine klassische Menge. Dann heißt die Menge

$$\mathbb{M}(A) = \{M \mid M = \{(a, h(a)) \mid a \in A, h(a) \in \mathbb{N}_0\}\}$$

die **Menge aller Multimengen** über der Menge A .

Abschließend werde noch eine alternative Schreibweise für Multimengen eingeführt, die insbesondere bei ausführlichen Mengenausdrücken für mehr Übersichtlichkeit sorgen kann. Diese Darstellungsform lehnt sich an die Schreibweise für algebraische Potenzausdrücke und diesbezügliche Potenzregeln an, etwa: $a^3 \cdot b^2 \cdot c \equiv a^3 \cdot b^2 \cdot c^1 \cdot d^0$.

Über einer endlichen Grundmenge $A = \{a_i \mid i = 1, \dots, m\}$ sei $M = \{(a_i, h(a_i)) \mid i = 1, \dots, m\}$ eine Multimenge gemäß Def. A.1. Diese Multimenge lässt sich folgendermaßen

Anhang: Multimengen

auch in **Potenzdarstellung** schreiben:

$$M = \prod_{i=1}^m a_i^{h_i} = a_1^{h_1} \cdot a_2^{h_2} \cdot \dots \cdot a_m^{h_m} \text{ mit } h_i := h(a_i) \text{ f\"ur } i = 1, \dots, m.$$

Hieraus ergibt sich auf kanonische Weise eine vereinfachte Schreibweise f\"ur die in Def. A.5 eingef\"uhrt Summe f\"ur zwei Multimengen $M_1 = \{(a_i, h(a_i)) \mid i = 1, \dots, m\}$ und $M_2 = \{(a_i, g(a_i)) \mid i = 1, \dots, m\}$ \u00fcber der Grundmenge A , wobei h und g die H\u00e4ufigkeitsfunktionen von M_1 bzw. M_2 seien sowie $h_i := h(a_i)$ und $g_i := g(a_i)$ f\"ur $i = 1, \dots, m$ gelte:

$$M_1 \uplus M_2 := \prod_{i=1}^m a_i^{h_i} \cdot \prod_{i=1}^m a_i^{g_i} = \prod_{i=1}^m a_i^{h_i + g_i}.$$

Diese Schreibweise l\u00e4sst sich ohne Weiteres auch auf die N-\u00e4re Summe \u00fcbertragen.

Im Fall $M_2 \subset M_1$ l\u00e4sst sich die Differenz $M_1 \setminus M_2$ ebenfalls vereinfacht schreiben als:

$$M_1 \setminus M_2 = \frac{\prod_{i=1}^m a_i^{h_i}}{\prod_{i=1}^m a_i^{g_i}} = \prod_{i=1}^m a_i^{h_i - g_i}.$$



**Forschungsreihe des Fachbereichs
Ingenieurwissenschaften und Mathematik**

Herausgeber

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften
und Mathematik

Band 2

November 2014

ISSN 2196-6192