



**FH Bielefeld**  
University of  
Applied Sciences



*Angewandte Mathematische Modellierung & Optimierung*

**AMMO – Berichte aus Forschung und Technologietransfer**

# **Lösungsansätze für Konfliktsituationen bei Feuerprozessen in kapazitierten Petri-Netzen**

Timo Kleine-Döpke, M. Sc.

Prof. Dr. rer. pol. Hermann-Josef Kruse

**Veröffentlichungsreihe (Onlinepublikation):**  
**AMMO – Berichte aus Forschung und Technologietransfer**

**ISSN**  
2198-4824

**Erscheinungsort**  
<http://www.fh-bielefeld.de/ammo/veroeffentlichungen/ammo-berichte-aus-forschung-und-technologietransfer>

**Herausgeber**  
Sprecher FSP AMMO, Fachhochschule Bielefeld

Fachhochschule Bielefeld  
Fachbereich Ingenieurwissenschaften und Mathematik  
FSP Angewandte Mathematische Modellierung und Optimierung  
Am Stadtholz 24  
33609 Bielefeld

## Autoren



### **Timo Kleine-Döpke (M.Sc.)<sup>1</sup>**

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

timo.kleine-doepke@fh-bielefeld.de

**Fachgebiete:** Petri-Netze, Personalplanung, Echtzeitsysteme

Hat sein Master-Studium im Fach *Optimierung und Simulation* und seinen Bachelor in der *Angewandten Mathematik* an der Fachhochschule Bielefeld absolviert. Arbeitete seit der Mitte seines Bachelor-Studiums als SHK im AMMO-Projekt „Coloured Petri Nets“ (CPN) von Prof. Dr. Hermann-Josef Kruse und hat sich in den letzten Jahren unter anderem in dieser Fachrichtung spezialisiert. Arbeitet an der Fachhochschule Bielefeld als wissenschaftlicher Mitarbeiter im it´s OWL-Innovationsprojekt „eXtreme Fast Automation“ (efa) und ist seit Januar 2015 ein Mitglied des FSP AMMO.



### **Prof. Dr. rer. pol. Hermann-Josef Kruse**

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

hermann-josef.kruse@fh-bielefeld.de

**Fachgebiete:** Wirtschaftsmathematik, Operations Research

Seit 1995 als Professor für Wirtschaftsmathematik (insbesondere Operations Research) an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. F&E-Projekte im Bereich der Optimierung und Simulation diskreter Systeme zur Entscheidungsunterstützung bei betrieblichen Problemstellungen.

---

<sup>1</sup>Die Beteiligung an diesem Artikel wurde zum Teil mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 02PQ2313 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

# Vorwort

Das CPN-Team<sup>1</sup> des Forschungsschwerpunktes *Angewandte Mathematische Modellierung und Optimierung* (AMMO) hat 2014 als Band 2 der Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld [ISSN 2196-6192] Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen veröffentlicht [BKK<sup>+</sup>14].

In diesem Heft findet eine Erweiterung dieser Ansätze auf kapazitierte Petri-Netze statt. Hierbei liegt das Hauptaugenmerk auf der Behandlung von sogenannten Input- und Output-Konflikten, die auf kanonische Weise durch die Kapazitierung der Plätze in Petri-Netzen entstehen. In enger Anlehnung an die Lösungsansätze in [BKK<sup>+</sup>14] werden analoge Auflösungskonzepte für kapazitierte (gefärbte) Petri-Netze entwickelt. Dabei wird auch der dort eingeführte Formalismus zur Beschreibung von gefärbten Petri-Netzen übernommen und problemspezifisch weiterentwickelt.

Aufgrund der engen Anlehnung an [BKK<sup>+</sup>14] wird es für einen Leser dieses Heftes sicherlich eine Hilfe sein, sich das Buch zwecks Nachschlagewerk herunterzuladen:

<http://www.fh-bielefeld.de/fb3/presse/veroeffentlichungen>.

---

<sup>1</sup>CPN steht für **C**oloured **P**etri **N**et.

# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Einführung in Petri-Netze</b>  | <b>1</b>  |
| <b>2</b> | <b>Kapazitierte Petri-Netze</b>   | <b>10</b> |
| 2.1      | Formale Definition . . . . .  | 12        |
| 2.2      | Bewertungsfunktionen bei Petri-Netzen mit Kapazitäten . . . . .   | 17        |
| 2.3      | Algorithmische Beschreibung von lokalen Freigabeprozessen bei Petri-Netzen<br>mit Kapazitäten . . . . . | 20        |
| 2.4      | Petri-Netz mit Kapazitäten und Konfliktlösung . . . . .   | 28        |
| 2.5      | Unabhängige Konfliktlösungen . . . . .  | 30        |
| 2.6      | Vorrang-Konfliktlösungen . . . . .  | 32        |
| 2.7      | Kooperative Konfliktlösungen . . . . .  | 36        |
| <b>3</b> | <b>Kapazitierte gefärbte Petri-Netze</b>  | <b>41</b> |
| 3.1      | Formale Definition gefärbter Petri-Netze mit Kapazitäten . . . . .                                      | 46        |
| 3.2      | Konfliktlösung bei gefärbten Petri-Netzen mit Kapazitäten . . . . .                                     | 57        |
| <b>4</b> | <b>Ausblick</b>   | <b>60</b> |
|          | <b>Literaturverzeichnis</b>   | <b>62</b> |
|          | <b>Anhang: Multimengen</b>  | <b>64</b> |



# 1 Einführung in Petri-Netze

Es werden zunächst die wichtigsten Grundbegriffe aus dem Gebiet der Petri-Netze in enger Anlehnung an den Formalismus aus [BKK<sup>+</sup>14] eingeführt.

## 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  $(p, t) \in F$  bzw.  $(t, p) \in B$  eine natürliche Zahl als das zugehörige **Gewicht**  $f(p, t)$  bzw.  $f(t, p)$  zuweist.<sup>1</sup>

Der entsprechende gewichtete Digraph  $G = (V, E, f)$  lässt sich als ein (ungewichteter) Multidigraph<sup>2</sup>  $G^*$  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.<sup>3</sup>

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

<sup>2</sup>Zum Begriff Multidigraph siehe einschlägige Literatur (u.a. [Die10] und [Vol96]).

<sup>3</sup>Im Folgenden wird ausschließlich die gewichtete Form von Petri-Netzen verwendet.

**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.<sup>1</sup>

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

**Beispiel:**

In Abb. 1.1 ist ein Petri-Netz 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.  $t_1\bullet = \{p_4\}$ ,  $t_2\bullet = \{p_5\}$ ,  $t_3\bullet = \{p_1, p_2\}$ ,  $t_4\bullet = \{p_2, p_3\}$ .

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

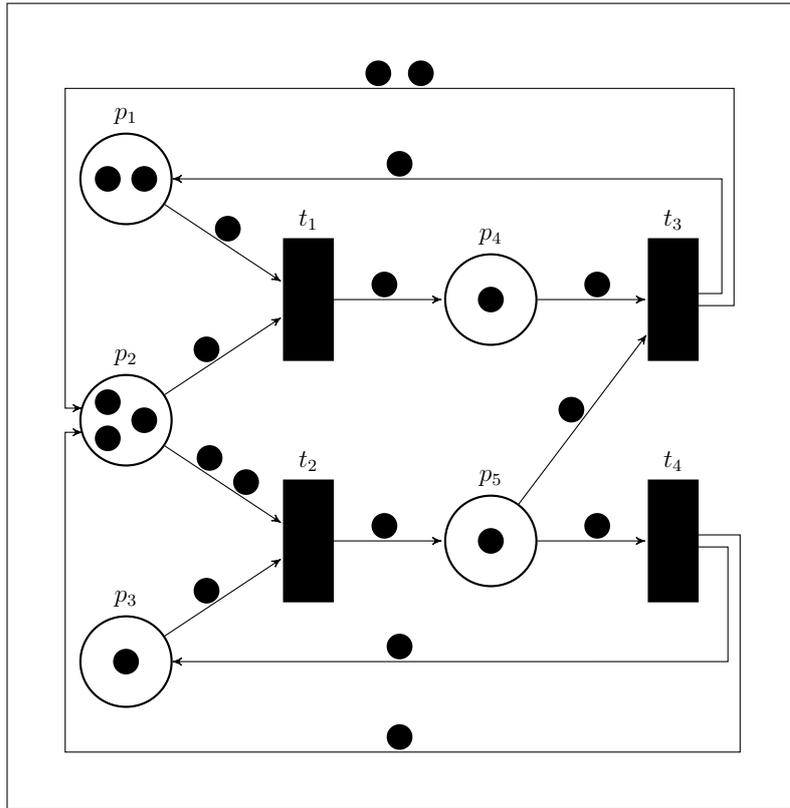


Abbildung 1.1: Beispiel eines gewichteten Petri-Netzes

**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  **$\mathbf{z}$ -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 für  $i = 1, \dots, m$  gilt:

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

**Beispiel:**

Im obigen Beispiel (Abb. 1.1) sind aktuell alle Transitionen aktiviert. 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.

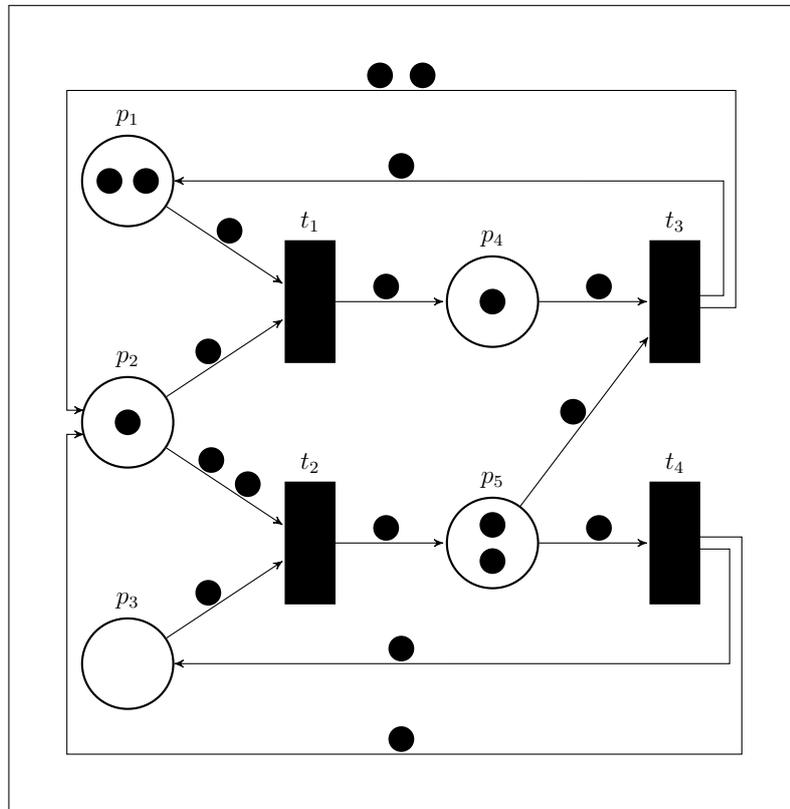


Abbildung 1.2: Zustand des Petri-Netzes aus Abb. 1.1 nach Feuern der Transition  $t_2$

Neben dem seriellen Feuern, bei dem selbst bei seriellen Rechnungen<sup>1</sup> 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.

<sup>1</sup>Eine Rechnung ist eine Folge mehrerer Feuerungen und den daraus resultierenden Zustandsübergängen. Für eine genauere Erläuterung sei auf [BKK<sup>+</sup>14, Abschn. 1.2.1 und Abschn. 1.2.2] verwiesen.

**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.<sup>1</sup> 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. 1.2) sind aktuell die Transitionen  $t_1, t_3$  und  $t_4$  nebenläufig feuerbar.

Werden nun die Transitionen  $t_1, t_3$  und  $t_4$  nebenläufig gefeuert, so wird vom Zustand  $\mathbf{z}' =$

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

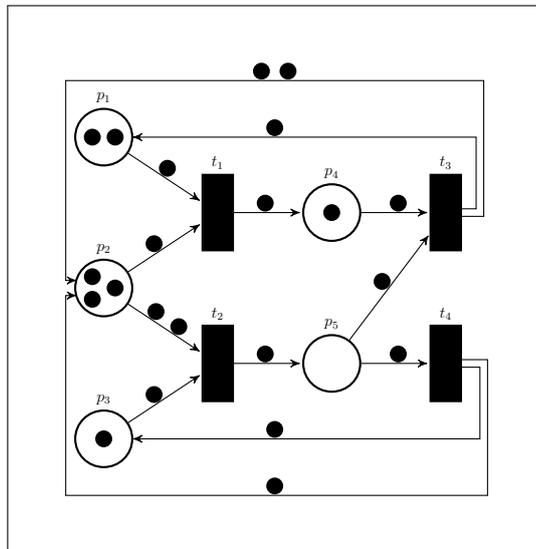


Abbildung 1.3: Zustand des Petri-Netzes aus Abb. 1.2 nach nebenläufigem Feuern der Transitionen  $t_1, t_3$  und  $t_4$

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

## 1 Einführung in Petri-Netze

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 so genannten *allgemeinen Konfliktsituationen* kommen, wie das folgende Beispiel in Abbildung 1.4 zeigt (vgl. [BKK<sup>+</sup>14, S.13]):

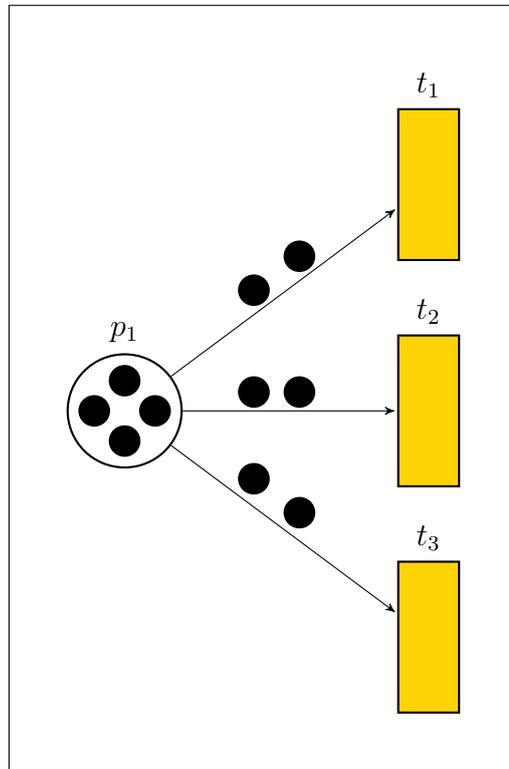


Abbildung 1.4: Allgemeine Konfliktsituation

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, einen Teil der Transitionenmenge oder sogar maximal viele Transitionen trotzdem zu feuern. Im obigen Beispiel wären jeweils zwei Transitionen nebenläufig feuierbar, 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 Abbildung 1.5 exemplarisch dargestellt wird (vgl. [BKK<sup>+</sup>14, S.14]).

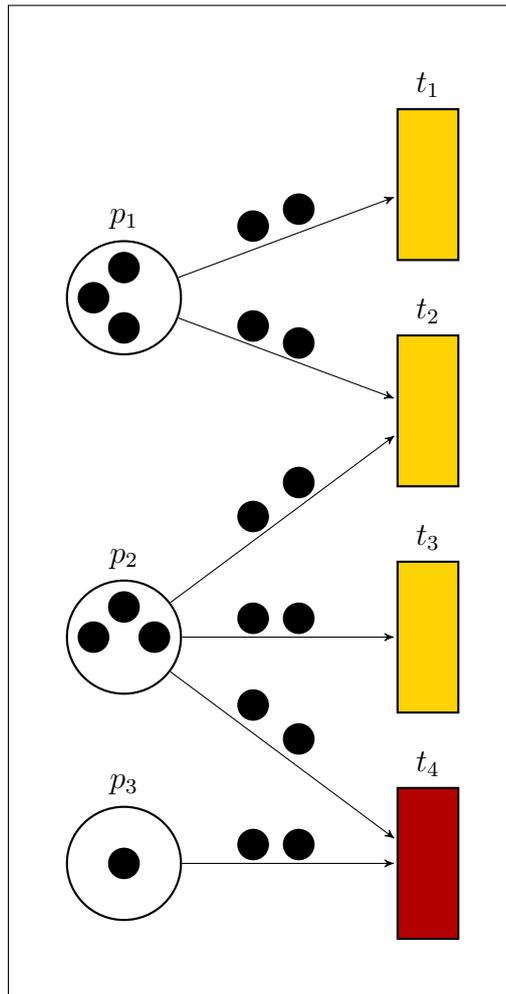


Abbildung 1.5: Mehrere allgemeine Konflikte mit gegenseitiger Beeinflussung

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 freigeben; 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.

Die formale Definition eines allgemeinen Konfliktes ergibt sich wie folgt (vgl. [BKK<sup>+</sup>14, S.16]):

**Definition 1.7:**

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 1.8:**

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 (aktivierte) Transition enthält.

**Definition 1.9:**

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. 1.6 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.

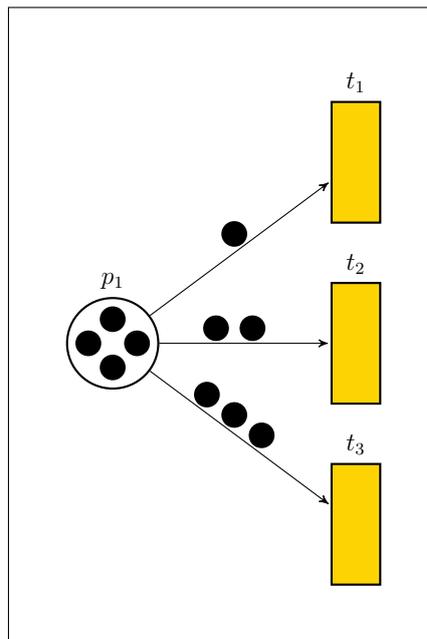


Abbildung 1.6: Maximal feuerbare Transitionenmengen

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.

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 in [BKK<sup>+</sup>14] verschiedene Lösungsansätze für allgemeine Konfliktsituationen vorgestellt. Diese werden im Folgenden auf kapazitive Konfliktsituationen übertragen.

## 2 Kapazitierte 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] und [Pro14]). Diese Petri-Netze haben eine *Minimalkapazität*  $C_{min}$  und eine *Maximalkapazität*  $C_{max}$ , welche die minimale bzw. maximale Anzahl von Token angibt, die im jeweiligen Platz enthalten sein kann.

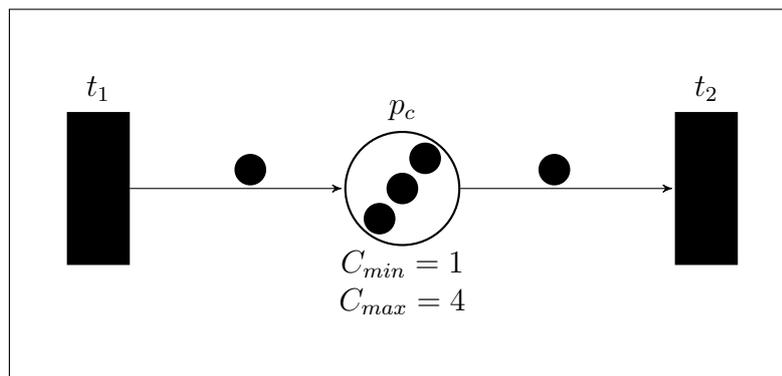


Abbildung 2.1: Petri-Netz mit Kapazitäten

Im obigen Beispiel (Abb. 2.1) muss der Platz  $p_c$  mindestens ein Token und darf 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 wird, ist diese Transition nicht aktiviert. Des Weiteren nehmen diese Bedingungen auch Einfluss auf die Konfliktsituation in einem Petri-Netz, denn es entstehen zwei neue Konfliktsituationen und zwar der *kapazitive Input-Konflikt* und der *kapazitive Output-Konflikt*.

Kapazitäten in Petri-Netzen sind eine pragmatische Erweiterung, um Petri-Netz-Modelle in einer einfachen Form darstellen zu können, denn ein Platz mit Kapazitäten müsste in einem „normalen“ Petri-Netz durch drei „normale“ (sprich: unkapazitierte) Plätze dargestellt werden, wie in Abb. 2.2 veranschaulicht wird.

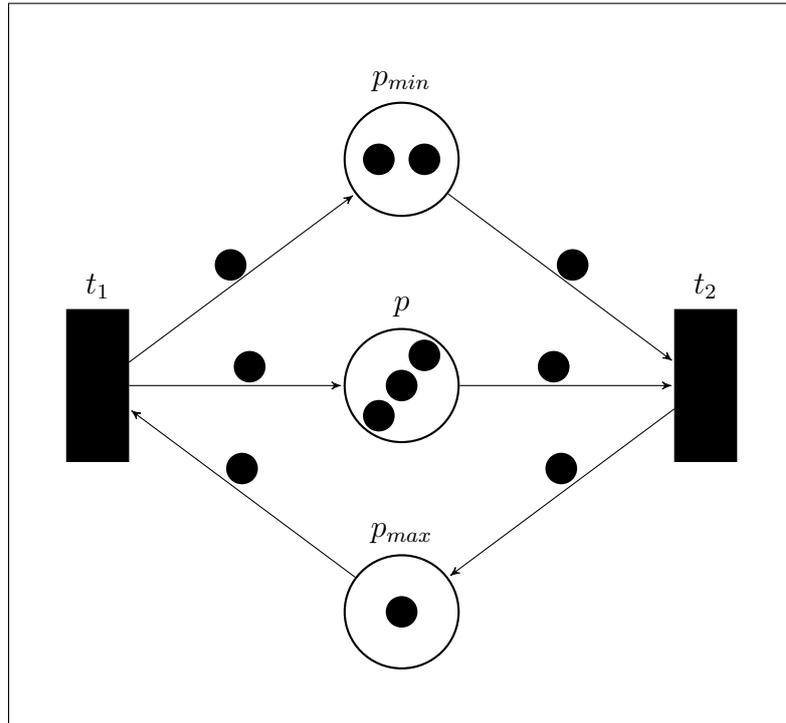


Abbildung 2.2: Äquivalentes Petri-Netz-Modell ohne Kapazitäten

Der Platz  $p$  ist dabei der Platz, welcher den Zustand des Platzes  $p_c$  mit Kapazitäten aus Abb. 2.1 wiedergibt. Die beiden kapazitiven Bedingungen werden durch die Zustände der Plätze  $p_{min}$  für die Minimal-Bedingung und  $p_{max}$  für die Maximal-Bedingung abgebildet; dazu muss der Initialzustand des Platzes mit Kapazitäten betrachtet werden, um hiermit die Zustände von  $p_{min}$  und  $p_{max}$  zu bestimmen:

$$\begin{aligned} \mathbf{z}(p_{max}) &= C_{max}(p_c) - \mathbf{z}(p_c), \\ \mathbf{z}(p_{min}) &= \mathbf{z}(p_c) - C_{min}(p_c). \end{aligned}$$

Die beiden Plätze  $p$  und  $p_{min}$  haben denselben Vor- bzw. Nachbereich und dieselben Pfeilgewichtungen wie der Platz mit Kapazitäten. Beim Platz  $p_{max}$  werden Vor- und Nachbereich des Platzes mit Kapazitäten vertauscht, die Pfeilgewichtung bleibt allerdings auch hier gleich.

## 2.1 Formale Definition

### Definition 2.1:

Ein **Petri-Netz mit Kapazitäten** ist ein Tupel  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z})$ , wobei

- das Tupel  $(P, T, F, B, f)$  ein Petri-Netz ist (vgl. Def. 1.2);
- $C_{max} : P \rightarrow (\mathbb{N} \cup \infty)$  eine Abbildung ist, welche jedem  $p \in P$  eine **Maximal-Kapazität**  $C_{max}(p)$  zuordnet;
- $C_{min} : P \rightarrow \mathbb{N}_0$  eine Abbildung ist, welche jedem  $p \in P$  eine **Minimal-Kapazität**  $C_{min}(p)$  zuordnet;
- $\mathbf{z}$  ein Zustand ist (vgl. Def. 1.4), für den gilt:

$$C_{min}(p) \leq \mathbf{z}(p) \leq C_{max}(p) \text{ für alle } p \in P.$$

### Bemerkung 2.2:

Für gewisse Anwendungen könnte es vorteilhaft sein, die Zustände eines PN mit Kapazitäten als zulässige oder unzulässige Zustände unterscheiden zu können. Dazu böte sich an, den Zustandsbegriff in Def. 2.1 allgemeiner zu fassen, indem die letzten Ungleichungen zunächst weggelassen werden. Dann könnte man den Zustand  $\mathbf{z}$  eines Platzes als zulässig (bzw. unzulässig) bezeichnen, wenn die Nebenbedingungen erfüllt sind (oder nicht).

### Bemerkung 2.3:

Soll ein Platz ohne Kapazitäten (oder mit nur einer Kapazitätsschranke) durch ein Petri-Netz mit Kapazitäten dargestellt werden, kann man dies durch die Zuordnung einer Minimal-Kapazität  $C_{min}(p) = 0$  bzw. einer Maximal-Kapazität  $C_{max}(p) = \infty$  erreichen. Zur besseren Übersichtlichkeit kann in diesen Fällen die Kapazitäten-Beschriftung am Platz auch entfallen.

### Beispiel:

In Abbildung 2.3 haben die Plätze  $p_1$ ,  $p_2$  und  $p_3$  folgende Minimal- und Maximal-Kapazitäten:

$$\begin{aligned} C_{min}(p_1) &= 2, C_{max}(p_1) = \infty, \\ C_{min}(p_2) &= 1, C_{max}(p_2) = 4, \\ C_{min}(p_3) &= 0, C_{max}(p_3) = 4. \end{aligned}$$

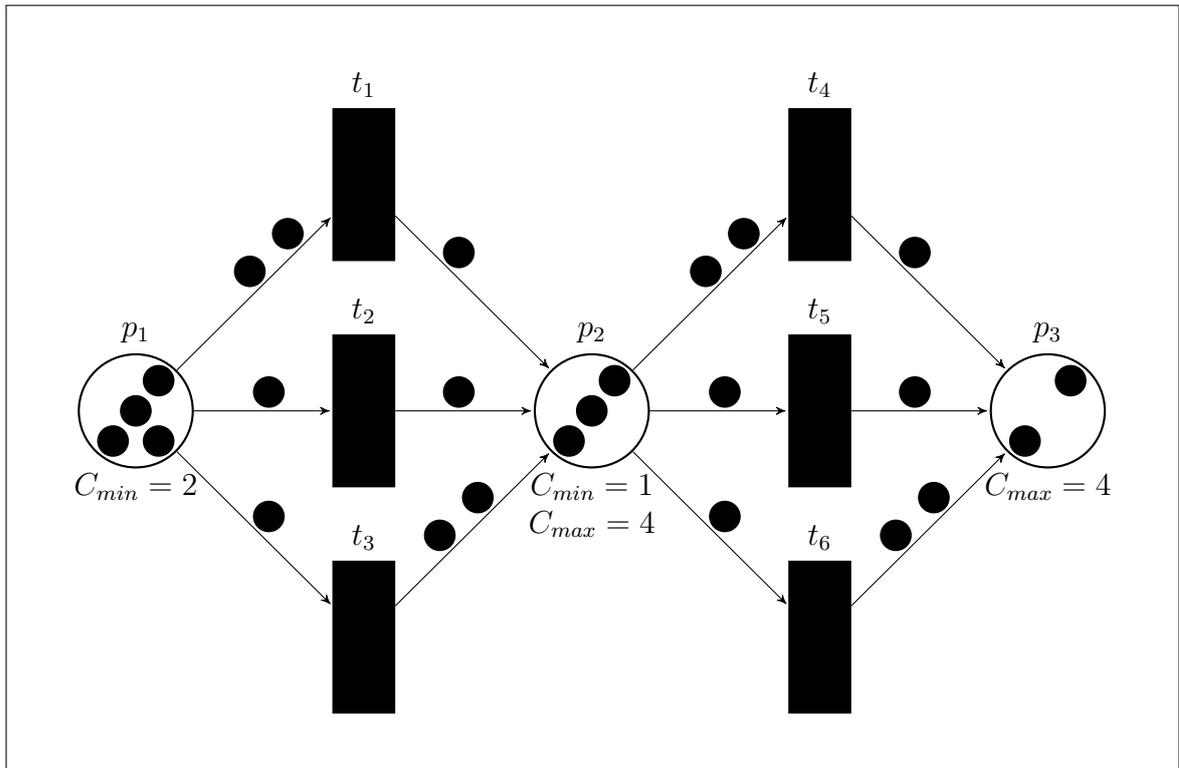


Abbildung 2.3: Petri-Netz mit Kapazitäten

**Definition 2.4:**

Es sei  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z})$  ein Petri-Netz mit Kapazitäten. Eine Transition  $t \in T$  von  $N$  heißt **aktiviert** oder **seriell feuierbar** im Zustand  $\mathbf{z}$ , wenn:

$$\forall p \in \bullet t \text{ gilt: } \mathbf{z}(p) - f(p, t) \geq C_{min}(p)$$

und

$$\forall p \in t \bullet \text{ gilt: } \mathbf{z}(p) + f(t, p) \leq C_{max}(p)$$

gilt. Demnach ist die Menge aller aktivierten Transitionen  $TA(\mathbf{z})$  des Petri-Netzes:

$$TA(\mathbf{z}) := \{t \in T \mid (\forall p \in \bullet t : \mathbf{z}(p) - f(p, t) \geq C_{min}(p)) \wedge (\forall p \in t \bullet : \mathbf{z}(p) + f(p, t) \leq C_{max}(p))\}.$$

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

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

Mit  $TA_{out}(p, \mathbf{z})$  bezeichnet man 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 2.5:**

Es sei  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z})$  ein Petri-Netz mit Kapazitäten und  $\{t_1, \dots, t_r\}$  eine Menge von Transitionen.<sup>1</sup> 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. Mit

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

werde der **gemeinsame Nachbereich** der Transitionen  $t_1, \dots, t_r$  bezeichnet. Die Transitionenmenge  $\{t_1, \dots, t_r\}$  heißt **nebenläufig feuierbar** im Zustand  $\mathbf{z}$ , wenn alle Transitionen aus  $\{t_1, \dots, t_r\}$  aktiviert sind<sup>2</sup> und für alle  $p \in \bullet\{t_1, \dots, t_r\}$  gilt:

$$\mathbf{z}(p) - \sum_{t \in \{t_1, \dots, t_r\} \cap p\bullet} f(p, t) \geq C_{min}(p)$$

sowie für alle  $p \in \{t_1, \dots, t_r\}\bullet$  gilt:

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

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

<sup>2</sup>Gemäß Def. 2.4

**Definition 2.6:**

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

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

$$\mathbf{z}(p) - \sum_{t \in TA_{out}(p)} f(p, t) < C_{min}(p).$$

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

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

**Bemerkung 2.7:**

Man beachte, dass die allgemeine Konfliktsituation eines Platzes  $p$  ein Spezialfall der Output-Konfliktsituation ist, nämlich für den Fall, dass  $C_{min}(p) = 0$  ist.

**Beispiel:**

Im folgenden Petri-Netz mit Kapazitäten aus Abbildung 2.4 soll veranschaulicht werden, welchen Einfluss Kapazitäten darauf haben, ob Transitionen aktiviert sind bzw. ob Plätze einen Konflikt haben.

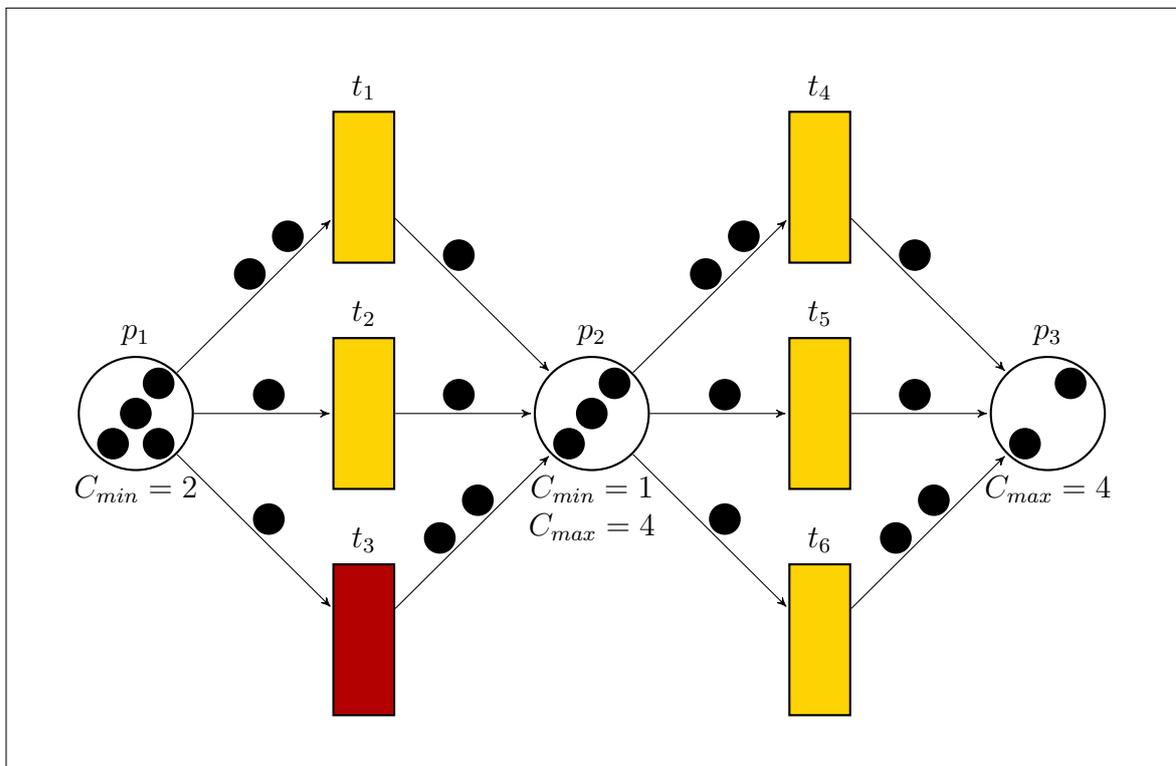


Abbildung 2.4: Aktiviertheit und Konflikte eines Petri-Netzes mit Kapazitäten

## 2 Kapazitierte Petri-Netze

Die Transitionen  $t_1, t_2, t_4, t_5$  und  $t_6$  sind aktivierbar, die Transition  $t_3$  allerdings auf Grund der Maximal-Kapazität von  $p_2$  nicht, da

$$\mathbf{z}(p_2) + f(t_3, p_2) = 3 + 2 > 4 = C_{max}(p_2)$$

ist. Der Platz  $p_1$  induziert eine Output-Konfliktsituation, da

$$\mathbf{z}(p_1) - \sum_{t \in TA_{out}(p_1)} f(p_1, t) = 4 - (2 + 1) = 1 < 2 = C_{min}(p_1)$$

ist. Der Platz  $p_2$  induziert sowohl eine Input- als auch eine Output-Konfliktsituation, da

$$\mathbf{z}(p_2) + \sum_{t \in TA_{in}(p_2)} f(t, p_2) = 3 + (1 + 1) = 5 > 4 = C_{max}(p_2)$$

und

$$\mathbf{z}(p_2) - \sum_{t \in TA_{in}(p_2)} f(p_2, t) = 3 - (2 + 1 + 1) = -1 < 1 = C_{min}(p_2)$$

ist. Der Platz  $p_3$  induziert eine Input-Konfliktsituation, da

$$\mathbf{z}(p_3) + \sum_{t \in TA_{in}(p_3)} f(t, p_3) = 2 + (1 + 1 + 2) = 6 > 4 = C_{max}(p_3)$$

ist.

### Definition 2.8:

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

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

$$\left( \mathbf{z}(p) - \sum_{t \in \{t_1, \dots, t_r\}} f(p, t) \geq C_{min}(p) \right) \wedge \left( \mathbf{z}(p) - f(p, t_0) - \sum_{t \in \{t_1, \dots, t_r\}} f(p, t) < C_{min}(p) \right).$$

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

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

In den nächsten Abschnitten sollen die Konfliktlösungskonzepte aus [BKK<sup>+</sup>14, Kap. 2] und die Feuerszenarien [BKK<sup>+</sup>14, Kap. 3] auf Petri-Netze mit Kapazitäten angepasst werden. Grundsätzlich gibt es bei den Petri-Netzen mit Kapazitäten mehrere Vorgehensweisen zur Lösung eines Konfliktes, von denen einige hier vorgestellt werden. Dafür müssen die Bewertungsfunktionen aus [BKK<sup>+</sup>14, Abschn. 2.2.1] verallgemeinert werden.

## 2.2 Bewertungsfunktionen bei Petri-Netzen mit Kapazitäten

### Definition 2.9:

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

Gegenüber unkapazitierten Petri-Netzen (vgl. [BKK<sup>+</sup>14, S. 19f]) werden nun auch den Pfeilen aus  $B$  Bewertungen zugewiesen.

### Beispiel:

In Abbildung 2.5 ist ein Petri-Netz mit Kapazitäten und allgemeiner Bewertungsfunktion dargestellt, wobei die Bewertung als Nutzen aufgefasst werden könnte.

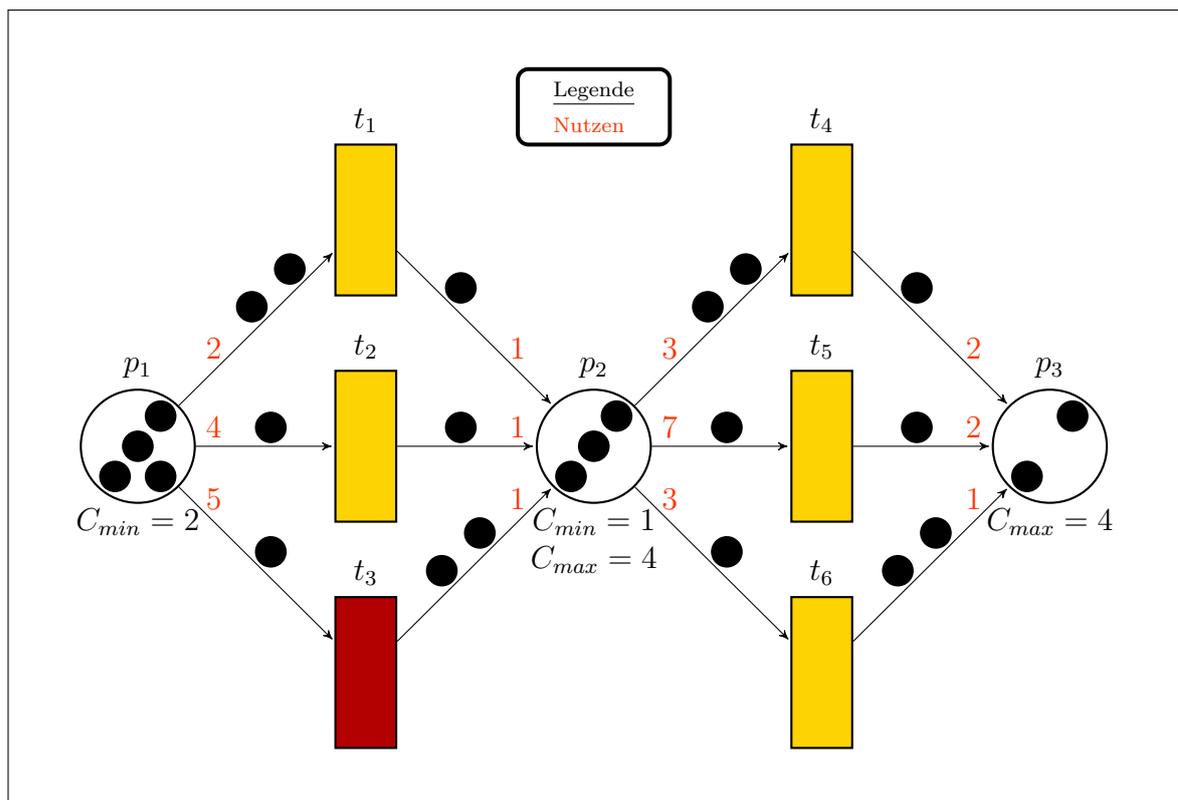


Abbildung 2.5: Petri-Netz mit Kapazitäten und Bewertung (Nutzen)

**Definition 2.10:**

Gegeben sei ein Petri-Netz mit Kapazitäten  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z})$ . Eine Abbildung  $\varrho : (F \cup B) \rightarrow \mathbb{N}$  heißt (**allgemeine**) **Prioritätsfunktion** von  $N$  und  $\varrho(p, t)$  bzw.  $\varrho(t, p)$  die **Priorität** des Pfeils  $(p, t) \in F$  bzw.  $(t, p) \in B$ , 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

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

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

**Beispiel:**

In Abbildung 2.6 ist ein Petri-Netz mit Kapazitäten und normierter Prioritätsfunktion dargestellt.

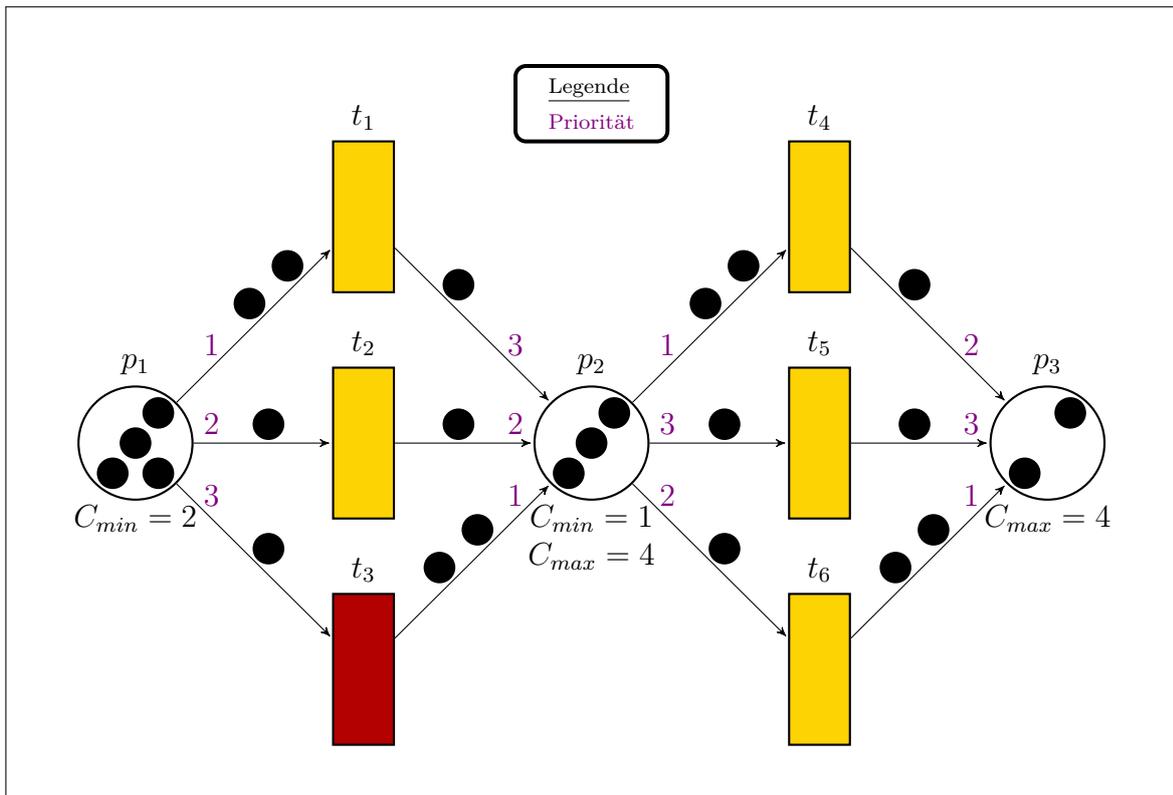


Abbildung 2.6: Petri-Netz mit Kapazitäten und Prioritäten

**Definition 2.11:**

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

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

und

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

Unter Berücksichtigung des Zustandes  $\mathbf{z}$  und den daraus folgenden Mengen  $F_z := \{(p, t) \in F \mid t \in TA(\mathbf{z})\}$  und  $B_z := \{(t, p) \in B \mid t \in TA(\mathbf{z})\}$  bezeichnet man die Abbildung  $\omega_z : (F_z \cup B_z) \rightarrow [0; 1]$  mit

$$\omega_z(x, y) = \begin{cases} \frac{\omega(x, y)}{\sum_{t_j \in TA_{out}(x)} \omega(x, t_j)} & \text{für } (x, y) \in F_z \\ \frac{\omega(x, y)}{\sum_{t_j \in TA_{in}(y)} \omega(t_j, y)} & \text{für } (x, y) \in B_z \end{cases}$$

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

**Beispiel:**

In Abbildung 2.7 ist ein Petri-Netz mit Kapazitäten und allgemeiner Wahrscheinlichkeitsfunktion dargestellt. Zur besseren Übersichtlichkeit wurde auf die Darstellung der aktualisierten Wahrscheinlichkeiten hier verzichtet.

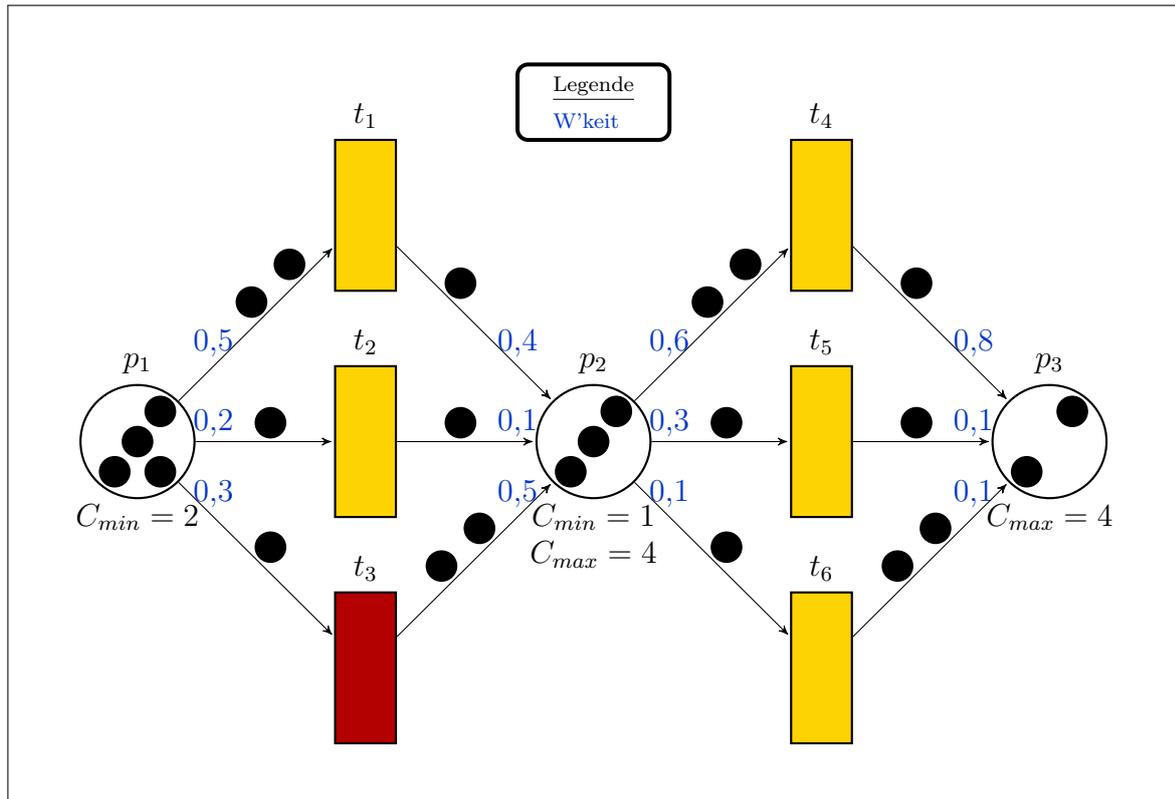


Abbildung 2.7: Petri-Netz mit Kapazitäten und Wahrscheinlichkeiten

### 2.3 Algorithmische Beschreibung von lokalen Freigabeprozessen bei Petri-Netzen mit Kapazitäten

Zur Lösung von Output- bzw. Input-Konflikten können prinzipiell die Algorithmen aus [BKK<sup>+</sup>14, Abschn. 2.2.2] verwendet werden. Es müssen allerdings die Zulässigkeit-Bedingungen (und die verwendeten Transitionen beim Input-Konflikt) angepasst werden; dies wird im Folgenden explizit nachvollzogen.

#### Prioritäten

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

$t_1, \dots, t_{\kappa_{in}(p)} \in TA_{in}(p)$  in aufsteigender Reihenfolge ihrer Prioritäten geordnet ist, d.h.

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

bzw.

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

**Algorithmus 2.12: Lokale Freigabe mittels Prioritätenverfahren bei Output-Konflikt**

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

Für  $i = 1, \dots, \kappa_{out}(p)$  führe aus:

$$\text{Falls gilt: } \mathbf{z}(p) - f(p, t_i) - \sum_{t \in TE_{out}(p)} f(p, t) \geq C_{min}(p),$$

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

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

**Algorithmus 2.13: Lokale Freigabe mittels Prioritätenverfahren bei Input-Konflikt**

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

Für  $i = 1, \dots, \kappa_{in}(p)$  führe aus:

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

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

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

**Beispiel:**

Für das Petri-Netz mit Kapazitäten und Prioritäten aus Abb. 2.6 ergibt sich:

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

$$TE_{out}(p_2) = \{t_4\},$$

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

bzw.

$$\begin{aligned} TE_{in}(p_1) &= \emptyset, \\ TE_{in}(p_2) &= \{t_2\}, \\ TE_{in}(p_3) &= \{t_6\}. \end{aligned}$$

**Bemerkung 2.14:**

Die lokalen Freigabe-Prozeduren für Petri-Netze mit Kapazitäten und Bewertungen (Nutzen) sind analog zu konzipieren, indem die aktivierten Transitionen aus  $TA_{out}(p)$  bzw.  $TE_{out}(p)$  in absteigender Reihenfolge ihrer Nutzenwerte anzuordnen sind, wobei für den Fall gleicher Nutzenwerte eine zusätzliche Entscheidungsregel zu Grunde gelegt werden müsste.

**Wahrscheinlichkeiten**

Gegeben sei ein Petri-Netz  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z}, \omega)$  mit Wahrscheinlichkeitsfunktion  $\omega$ . Ein Platz  $p \in P$  erteilt in Bezug auf den Zustand  $\mathbf{z}$  einer Teilmenge  $TE_{out}(p)$  bzw.  $TE_{in}(p)$  von aktivierten Transitionen aus  $TA_{out}(p)$  bzw.  $TA_{in}(p)$  die Freigabe zum Feuern durch die im Folgenden beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge**  $TE_{out}(p)$  bzw.  $TE_{in}(p)$  **mittels Wahrscheinlichkeiten**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge  $TA_{out}(p)$  mit  $\kappa_{out}(p) \in \mathbb{N}$  bzw. die Mächtigkeit der Menge  $TA_{in}(p)$  mit  $\kappa_{in}(p) \in \mathbb{N}$  gegeben ist. Durch eine Indizierung der Elemente  $t_1, \dots, t_{\kappa_{out}(p)} \in TA_{out}(p)$  bzw.  $t_1, \dots, t_{\kappa_{in}(p)} \in TA_{in}(p)$  ist eine Anordnung der Transitionen vorgegeben, so dass jeder Transition  $t_i \in TA_{out}(p)$ ,  $i = 1, \dots, \kappa_{out}(p)$ , bzw.  $t_i \in TA_{in}(p)$ ,  $i = 1, \dots, \kappa_{in}(p)$ , auf kanonische Weise die kumulierte Wahrscheinlichkeit

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

bzw.

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

zugewiesen wird.

**Algorithmus 2.15: Lokale Freigabe mittels Wahrscheinlichkeiten-Verfahren bei Output-Konflikt**

Setze:

$$\begin{aligned} TE_{out}(p) &:= \emptyset \\ L &:= TA_{out}(p) \text{ (Kandidatenliste)} \\ \omega_L &:= 1 \text{ (Summierte Restkandidaten-W'keit)} \end{aligned}$$

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

F'ur  $i = 1, \dots, \kappa_{\text{out}}(p)$  f'uhre aus:

Setze:

$$\pi(i) := i \text{ (Reihenfolgez'ahler)}$$

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

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

Solange  $L \neq \emptyset$  f'uhre aus:

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

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

Falls gilt:  $\mathbf{z}(p) - f(p, t_i) - \sum_{t \in TE_{\text{out}}(p)} f(p, t) + \geq C_{\text{min}}(p)$ ,

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

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

F'ur  $j = i + 1, \dots, \kappa(p)_{\text{out}}$  setze:  $\pi(j) := \pi(j) - 1$ .

F'ur alle  $t_j \in L$  f'uhre aus:

Setze:

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

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

Ergebnis: Die Menge  $TE_{\text{out}}(p)$  beinhaltet alle aktivierten Transitionen, die vom Platz  $p$  bei der L'osung des Output-Konfliktes zum Feuern freigegeben werden.

### Algorithmus 2.16: Lokale Freigabe mittels Wahrscheinlichkeiten-Verfahren bei Input-Konflikt

Setze:

$$TE_{\text{in}}(p) := \emptyset$$

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

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

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

F'ur  $i = 1, \dots, \kappa_{\text{in}}(p)$  f'uhre aus:

Setze:

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

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

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

Solange  $L \neq \emptyset$  führe aus:

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

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

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

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

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

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

Für alle  $t_j \in L$  führe aus:

Setze:

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

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

Ergebnis: Die Menge  $TE_{in}(p)$  beinhaltet alle aktivierten Transitionen, die vom Platz  $p$  bei der Lösung des Input-Konfliktes zum Feuern freigegeben werden.

### Beispiel:

Für das Petri-Netz mit Kapazitäten und Wahrscheinlichkeiten aus Abb. 2.7 ergibt sich:

$$TE_{out}(p_1) = \{t_1\} \text{ (bei } \rho = 0,4),$$

$$TE_{out}(p_2) = \{t_5, t_6\} \text{ (bei } \rho = 0,7 \text{ und } \rho = 0,9),$$

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

bzw.

$$TE_{in}(p_1) = \emptyset,$$

$$TE_{in}(p_2) = \{t_1\} \text{ (bei } \rho = 0,2),$$

$$TE_{in}(p_3) = \{t_4, t_5\} \text{ (bei } \rho = 0,5 \text{ und } \rho = 0,3).$$

### Nutzen

Gegeben sei ein Petri-Netz  $N = (P, T, F, B, f, C_{max}, C_{min}, \mathbf{z}, \psi)$  im Zustand  $\mathbf{z}$  mit allgemeiner Bewertungsfunktion  $\psi : (F \cup B) \rightarrow \mathbb{R}_{\geq}$ , die als Nutzen aufgefasst wird. Ein Platz  $p \in P$  erteilt in Bezug auf den Zustand  $\mathbf{z}$  einer Teilmenge  $TE_{out}(p)$  bzw.  $TE_{in}(p)$  von aktivierten Transitionen aus  $TA_{out}(p)$  bzw.  $TA_{in}(p)$  die Freigabe zum Feuern durch die im

Folgende beschriebene Bildungsvorschrift dieser **lokalen Freigabemenge**  $TE_{out}(p)$  bzw.  $TE_{in}(p)$  **mittels Nutzen**. Dabei werde o.B.d.A. vorausgesetzt, dass die Mächtigkeit der Menge  $TA_{out}(p)$  mit  $\kappa_{out}(p) \in \mathbb{N}$  bzw. die Mächtigkeit der Menge  $TA_{in}(p)$  mit  $\kappa_{in}(p) \in \mathbb{N}$  gegeben ist.

Der folgende Algorithmus arbeitet nach dem Branch- & Bound-Prinzip. Dafür seien die Transitionen  $t_1, \dots, t_{\kappa_{out}(p)} \in TA_{out}(p)$  bzw.  $t_1, \dots, t_{\kappa_{in}(p)} \in TA_{in}(p)$  o.B.d.A. in absteigender Reihenfolge ihrer Nutzenquotienten  $\eta_j$  geordnet, d.h.

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

bzw.

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

### Algorithmus 2.17: Lokale Freigabe mittels Branch & Bound bei Output-Konflikt

Setze:

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

$g_0 := 0$  (Gewicht der Teillösung)

$n_0 := \sum_{j=1}^{\kappa_{out}(p)} \psi(t_j, p)$  (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  $m_r < \kappa_{out}(p)$ , führe aus:

Setze:

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

$$\begin{aligned}
 n_{s+1} &:= n_r; n_{s+2} := n_r - \psi(p, t_{m_r+1}); \\
 g_{s+1} &:= g_r + f(p, t_{m_r+1}); g_{s+2} := g_r; \\
 m_{s+1} &:= m_{s+2} := m_r + 1; \\
 TP_{\text{g\u00fcltig}} &:= \{X_{s+i} \mid \mathbf{z}(p) - g_{s+i} \geq C_{\min}(p) \wedge n_{s+i} > N, i = 1, 2\} \\
 TP &:= (TP \setminus \{X_r\}) \cup TP_{\text{g\u00fcltig}};
 \end{aligned}$$

**3. Schritt:** Falls  $(m_r = \kappa_{\text{out}}(p) - 1) \wedge (TP_{\text{g\u00fcltig}} \neq \emptyset)$ , f\u00fchre aus:

W\u00e4hle  $X_l$  aus  $TP_{\text{g\u00fcltig}}$  mit  $n_l = \max(n_i \mid X_i \in TP_{\text{g\u00fcltig}})$

Setze:

$$N := n_l$$

$$L := X_l$$

F\u00fcr alle  $X_i \in TP$  f\u00fchre aus:

Falls  $n_i \leq N$ , setze:

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

Gehe zum Schritt 1.

Ergebnis: Das bin\u00e4re Tupel  $(x_1, x_2, \dots, x_{\kappa(p)}) \in L$  ist folgenderma\u00dfen zu interpretieren:

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

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

so dass gilt:

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

### Algorithmus 2.18: Lokale Freigabe mittels Branch & Bound bei Input-Konflikt

Setze:

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

$g_0 := 0$  (Gewicht der Teill\u00f6sung)

$n_0 := \sum_{j=1}^{\kappa_{\text{in}}(p)} \psi(t_j, p)$  (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  $m_r < \kappa_{in}(p)$ , führe aus:  
 Setze:  
 $X_{s+1} := X_r$  mit  $x_{m_r+1} = 1$ ;  $X_{s+2} := X_r$  mit  $x_{m_r+1} = 0$ ;  
 $n_{s+1} := n_r$ ;  $n_{s+2} := n_r - \psi(t_{m_r+1}, p)$ ;  
 $g_{s+1} := g_r + f(t_{m_r+1}, p)$ ;  $g_{s+2} := g_r$ ;  
 $m_{s+1} := m_{s+2} := m_r + 1$ ;  
 $TP_{gültig} := \{X_{s+i} \mid \mathbf{z}(p) + g_{s+i} \leq C_{max}(p) \wedge n_{s+i} > N, i = 1, 2\}$   
 $TP := (TP \setminus \{X_r\}) \cup TP_{gültig}$ ;

**3. Schritt:** Falls  $(m_r = \kappa_{in}(p) - 1) \wedge (TP_{gültig} \neq \emptyset)$ , führe aus:  
 Wähle  $X_l$  aus  $TP_{gültig}$  mit  $n_l = \max(n_i \mid X_i \in TP_{gültig})$   
 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:

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

### Bemerkung 2.19:

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:

Für das Petri-Netz mit Kapazitäten und Bewertung (Nutzen) aus Abb. 2.5 ergibt sich:

$$\begin{aligned}TE_{out}(p_1) &= \{t_2\}, \\TE_{out}(p_2) &= \{t_5, t_6\}, \\TE_{out}(p_3) &= \emptyset\end{aligned}$$

bzw.

$$\begin{aligned}TE_{in}(p_1) &= \emptyset, \\TE_{in}(p_2) &= \{t_1\} \text{ oder } \{t_2\}, \\TE_{in}(p_3) &= \{t_4, t_5\}.\end{aligned}$$

## 2.4 Petri-Netz mit Kapazitäten und Konfliktlösung

In ähnlicher Weise, wie ein Petri-Netz mit Konfliktlösung definiert ist in [BKK<sup>+</sup>14, Def. 2.11], kann man auch ein Petri-Netz mit Kapazitäten und Konfliktlösung definieren, welches verschiedene Bewertungsarten wie in Abb. 2.8 enthält.

### Definition 2.20:

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

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

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

- dass jede Priorität jeweils (Input-Konflikt und Output-Konflikt) nur einmal von jedem Platz  $p$  benutzt wird:

$$\delta(p, t_i) \neq \delta(p, t_k) \text{ für alle } t_i, t_k \in p\bullet \text{ mit } t_i \neq t_k \text{ falls } \alpha(p) = \text{Prio},$$

und

$$\delta(t_i, p) \neq \delta(t_k, p) \text{ für alle } t_i, t_k \in \bullet p \text{ mit } t_i \neq t_k \text{ falls } \alpha(p) = \text{Prio},$$

- dass jeweils (Input-Konflikt und Output-Konflikt) die Summe der Wahrscheinlichkeiten gleich 1 ist:

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

und

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

- Die Abbildung

$$\zeta : P \rightarrow \left\{ \begin{array}{ll} \{\text{lokales Prioritäten-Verfahren}\} & \text{für } \alpha(p) = \text{Prio} \\ \{\text{lokales W'keiten-Verfahren}\} & \text{für } \alpha(p) = \text{W'keit} \\ \{\text{lokales B\&B-Verfahren}\} & \text{für } \alpha(p) = \text{Nutzen} \end{array} \right\}$$

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

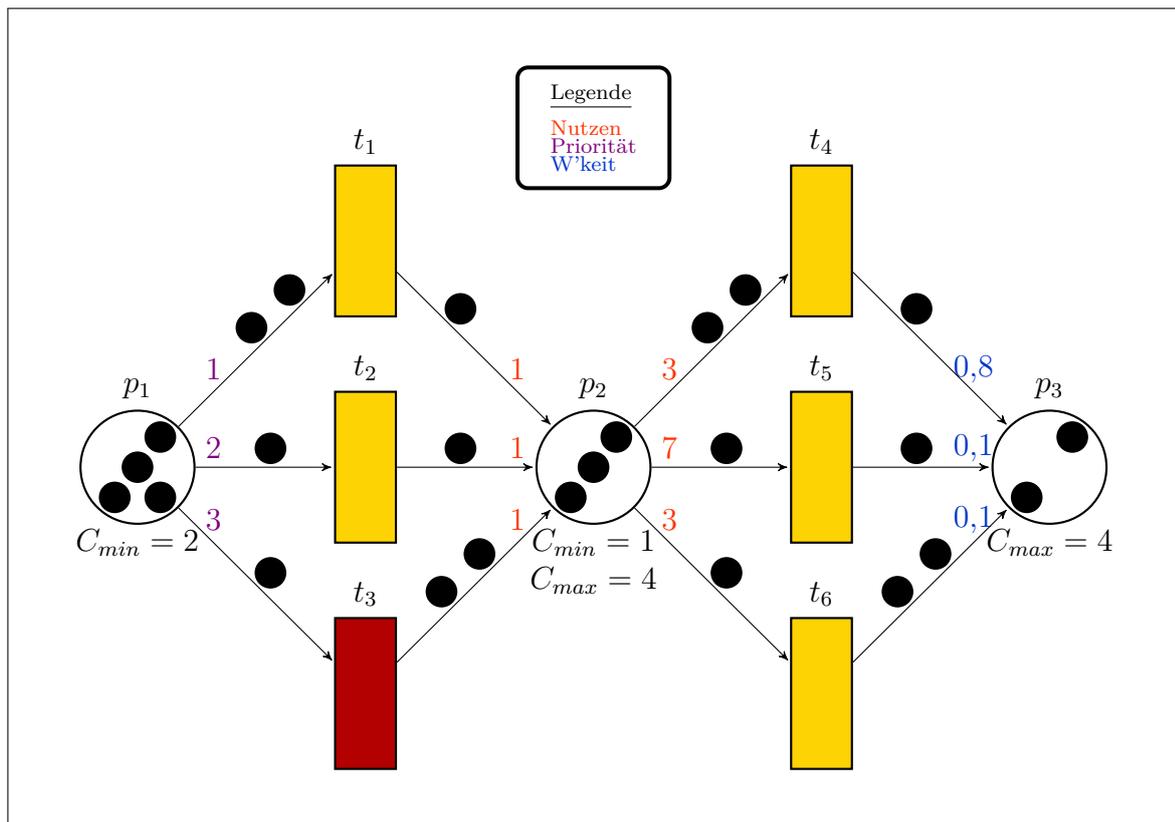


Abbildung 2.8: Petri-Netz mit Kapazitäten und verschiedenen Bewertungsarten

Die hier vorgestellten lokalen Freigabeprozeduren bei Output- bzw. Input-Konflikten werden im Folgenden dazu benutzt, den „Normalfall“, dass in Petri-Netzen mit Kapazitäten *beide* Konfliktsituationen auftreten, zu behandeln. Hierzu werden verschiedene Konfliktlösungskonzepte vorgestellt, die sich im Wesentlichen durch die Kombination der lokalen Freigabemengen unterscheiden.

## 2.5 Unabhängige Konfliktlösungen

Das Konzept der unabhängigen Konfliktlösungen ist die Übertragung des Konfliktlösungskonzeptes aus dem äquivalenten Petri-Netz ohne Kapazitäten. Man beachte, dass sowohl der Output- als auch der Input-Konflikt im äquivalenten Petri-Netz jeweils ein allgemeiner Konflikt wäre. Dementsprechend werden hier die Output- und Input-Konflikte unabhängig voneinander gelöst.

### Szenarium 1

Der *nebenläufige Feuerprozess (unabhängige Konfliktlösungen bei Petri-Netzen mit Kapazitäten)* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge  $TA$  aller aktiven Transitionen (gemäß Def. 2.4)
2. Ermittlung aller lokalen Output-Freigabemengen  $TE_{out}(p)$  aus jeweiligem  $TA_{out}(p)$  via Algorithmus aus Abschn. 2.3.
3. Ermittlung aller lokalen Input-Freigabemengen  $TE_{in}(p)$  aus jeweiligem  $TA_{in}(p)$  via Algorithmus aus Abschn. 2.3.
4. Ermittlung der globalen Freigabemenge  $TE$  mit:

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

5. Nebenläufiges Feuern der Menge  $TE$

**Beispiel:**

Am Petri-Netz mit Kapazitäten und Nutzen aus Abb. 2.9 soll der nebenläufige Feuerprozess mit unabhängigen Konfliktlösungen nachvollzogen werden.

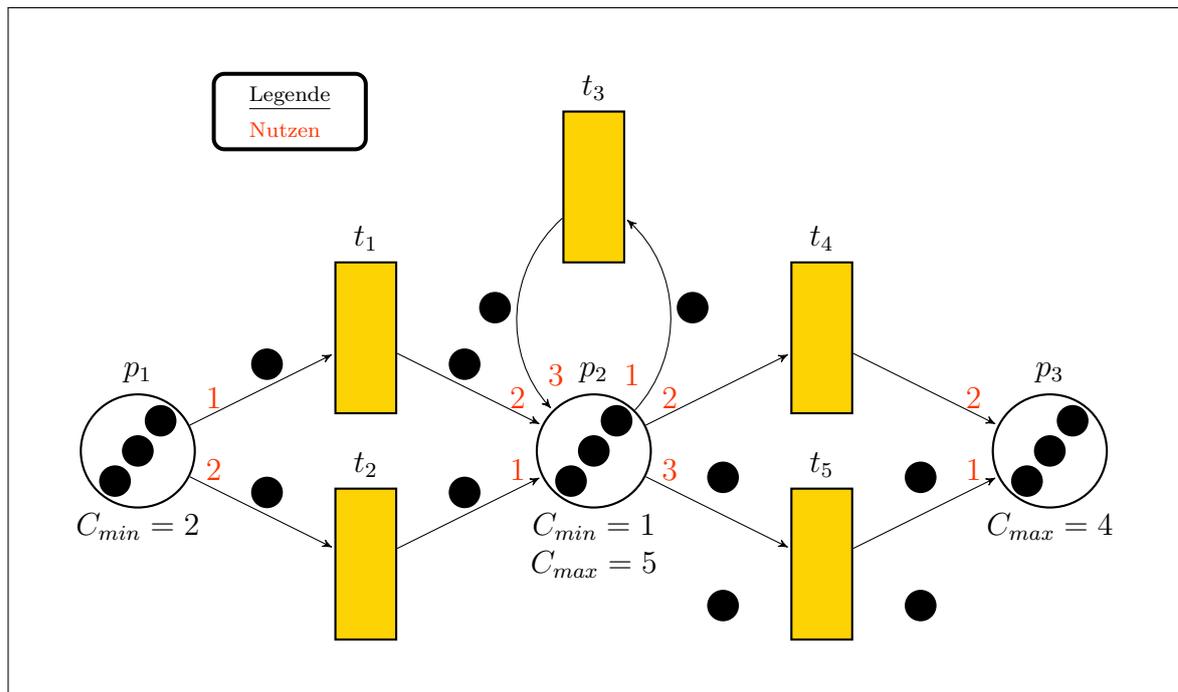


Abbildung 2.9: Petri-Netz mit Kapazitäten und Nutzen

1. Schritt:

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

$$TA = \{t_1, t_2, t_3, t_4, t_5\}$$

### 2. Schritt:

Bei der lokalen Output-Konfliktlösung sind die folgenden lokalen Freigabemengen entstanden:

$$TE_{out}(p_1) = \{t_2\}$$

$$TE_{out}(p_2) = \{t_4, t_5\}$$

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

### 3. Schritt:

Bei der lokalen Input-Konfliktlösung sind die folgenden lokalen Freigabemengen entstanden:

$$TE_{in}(p_1) = \emptyset$$

$$TE_{in}(p_2) = \{t_1, t_3\}$$

$$TE_{in}(p_3) = \{t_4\}$$

### 4. Schritt:

Daraus ergibt sich als globale Freigabemenge:

$$TE = \{t_4\}$$

### 5. Schritt:

Es wird somit die Menge  $TE = \{t_4\}$  gefeuert.

Die hierarchische Gleichstellung der beiden voneinander unabhängig vollzogenen Ermittlungen der lokalen Output- und Input-Freigabemengen für jeden Platz kann allerdings dazu führen, dass die Freigabemenge  $TE$  unbesetzt bleibt ( $TE = \emptyset$ ), obwohl es die Möglichkeit gäbe, Transitionen nebenläufig zu feuern. Daher sind auch andere Konfliktlösungskonzepte angedacht, die diesen „ungünstigen“ Fall weitgehend ausschalten.

## 2.6 Vorrang-Konfliktlösungen

Eine andere Möglichkeit zur Konfliktlösung wären die Vorrang-Konfliktlösungen. Hierbei werden erst alle Output- bzw. alle Input-Konflikte gelöst. Diejenigen Transitionen, die bei diesem ersten Schritt nicht freigegeben wurden, werden dann beim Lösen der Input- bzw. Output-Konflikte nicht mehr berücksichtigt. Hierdurch können beim Modellieren hierarchische („von oben nach unten“ bzw. „von unten nach oben“) Entscheidungs-Strukturen, die sich aus sachlogischen Zusammenhängen im Praxisprozess ergeben, abgebildet werden.

**Bemerkung 2.21:**

Bei den Vorrang-Konfliktlösungen können die Bewertungsarten aus Def. 2.9, Def. 2.10 und Def. 2.11 bzw. das Petri-Netz mit Kapazitäten und Konfliktlösung aus Def. 2.20 verwendet werden. Ebenfalls können wieder die angepassten Algorithmen aus Abschnitt 2.3 zum Zuge kommen. Hierbei muss nur beachtet werden, dass die Algorithmen nicht mehr nur auf die Menge  $TA_{out}(p)$  bzw.  $TA_{in}(p)$ , sondern auf spezielle Teilmengen  $TAe_{out}(p)$  bzw.  $TAe_{in}(p)$  angewendet werden.

**Szenarium 2**

Der *nebenläufige Feuerprozess (Output-Vorrang-Konfliktlösungen bei Petri-Netzen mit Kapazitäten)* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge  $TA$  aller aktiven Transitionen (gemäß Def. 2.4)
2. Ermittlung aller lokalen Output-Freigabemengen  $TE_{out}(p)$  aus jeweiligem  $TA_{out}(p)$  via Algorithmus aus Abschn. 2.3.
3. Ermittlung aller lokalen Input-Freigabemengen  $TE_{in}(p)$  aus jeweiligem  $TAe_{in}(p)$  mit

$$TAe_{in}(p) := \{t \in TA_{in}(p) \mid t \in TE_{out}(p_i) \forall p_i \in \bullet t\}$$

via Algorithmus aus Abschn. 2.3.

4. Ermittlung der globalen Freigabemenge  $TE$  mit:

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

5. Nebenläufiges Feuern der Menge  $TE$

**Beispiel:**

Am Petri-Netz mit Kapazitäten und Nutzen aus Abb. 2.9 soll der nebenläufige Feuerprozess mit Output-Vorrang-Konfliktlösung nachvollzogen werden.

1. Schritt:

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

$$TA = \{t_1, t_2, t_3, t_4, t_5\}$$

2. Schritt:

Bei der lokalen Output-Konfliktlösung sind die folgenden lokalen Freigabemengen entstanden:

## 2 Kapazitierte Petri-Netze

$$TE_{out}(p_1) = \{t_2\}$$

$$TE_{out}(p_2) = \{t_4, t_5\}$$

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

### 3. Schritt:

Es werden folgende Mengen gebildet:

$$TAe_{in}(p_1) = \emptyset$$

$$TAe_{in}(p_2) = \{t_2\}$$

$$TAe_{in}(p_3) = \{t_4\}$$

Bei der hieraus folgenden lokalen Input-Konfliktlösung ergeben sich die folgenden lokalen Freigabemengen:

$$TE_{in}(p_1) = \emptyset$$

$$TE_{in}(p_2) = \{t_2\}$$

$$TE_{in}(p_3) = \{t_4\}$$

### 4. Schritt:

Daraus ergibt sich als globale Freigabemenge:

$$TE = \{t_2, t_4\}$$

### 5. Schritt:

Es wird somit die Menge  $TE = \{t_2, t_4\}$  gefeuert.

### Szenarium 3

Der *nebenläufige Feuerprozess (Input-Vorrang-Konfliktlösungen bei Petri-Netzen mit Kapazitäten)* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge  $TA$  aller aktiven Transitionen (gemäß Def. 2.4)
2. Ermittlung aller lokalen Input-Freigabemengen  $TE_{in}(p)$  aus jeweiligem  $TA_{in}(p)$  via Algorithmus aus Abschn. 2.3.
3. Ermittlung aller lokalen Output-Freigabemengen  $TE_{out}(p)$  aus jeweiligem  $TA_{out}(p)$  mit

$$TAe_{out}(p) := \{t \in TA_{out}(p) \mid t \in TE_{in}(p_i) \forall p_i \in t\bullet\}$$

via Algorithmus aus Abschn. 2.3.

4. Ermittlung der globalen Freigabemenge  $TE$  mit:

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

5. Nebenläufiges Feuern der Menge  $TE$

**Beispiel:**

Am Petri-Netz mit Kapazitäten und Nutzen aus Abb. 2.9 soll der nebenläufige Feuerprozess mit Input-Vorrang-Konfliktlösung nachvollzogen werden.

1. Schritt:

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

$$TA = \{t_1, t_2, t_3, t_4, t_5\}$$

2. Schritt:

Bei der lokalen Input-Konfliktlösung sind die folgenden lokalen Freigabemengen entstanden:

$$\begin{aligned} TE_{in}(p_1) &= \emptyset \\ TE_{in}(p_2) &= \{t_1, t_3\} \\ TE_{in}(p_3) &= \{t_4\} \end{aligned}$$

3. Schritt:

Es werden folgende Mengen gebildet:

$$\begin{aligned} TAe_{out}(p_1) &= \{t_1\} \\ TAe_{out}(p_2) &= \{t_3, t_4\} \\ TAe_{out}(p_3) &= \emptyset \end{aligned}$$

Bei der hieraus folgenden lokalen Output-Konfliktlösung ergeben sich die folgenden lokalen Freigabemengen:

$$\begin{aligned} TE_{out}(p_1) &= \{t_1\} \\ TE_{out}(p_2) &= \{t_3, t_4\} \\ TE_{out}(p_3) &= \emptyset \end{aligned}$$

4. Schritt:

Daraus ergibt sich als globale Freigabemenge:

$$TE = \{t_1, t_3, t_4\}$$

5. Schritt:

Es wird somit die Menge  $TE = \{t_1, t_3, t_4\}$  gefeuert.

## 2.7 Kooperative Konfliktlösungen

Fasst man in einem Petri-Netz-Modell die einzelnen Plätze als altruistische Entscheidungsträger auf, die sich einem Gemeinwohl verpflichtet fühlen (z.B. maximaler Gesamtnutzen), wäre eine gemeinsame Lösung beider Konflikte, wie in einem kooperativen Spiel<sup>1</sup>, eine sinnvolle Vorgehensweise. Zudem haben die bis jetzt vorgestellten Konfliktlösungen eine gewisse Problematik mit Petri-Schlingen<sup>2</sup>, da es passieren kann, dass die zur Schlinge gehörige Transition nur von der Output- bzw. Input-Konfliktlösung freigegeben wird. Dies hat etwas von dem sprichwörtlichen „Sich selbst ein Bein stellen“, was grundsätzlich kein Problem darstellt, wenn im Realsystem solche Fälle gewollt sind, wohl aber unter obiger Kooperationsannahme.

Bei der kooperativen Konfliktlösung soll im Folgenden allerdings ausschließlich und exemplarisch auf Petri-Netze mit allgemeiner Bewertungsfunktion, deren Werte als Nutzen aufgefasst werden, eingegangen werden. Ähnliche Strategien wie für Nutzen können auch für Prioritäten bzw. Wahrscheinlichkeiten angewendet werden, auf deren explizite Ausformulierung hier aber verzichtet werden soll.

### Nutzen

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

Der folgende Algorithmus arbeitet nach dem Branch- & Bound-Prinzip. Dafür seien die Transitionen  $t_1, \dots, t_{\kappa(p)} \in TA_{lok}(p)$  o.B.d.A. in absteigender Reihenfolge ihrer Nutzenquo-

<sup>1</sup>Zum Begriff „kooperative Spiele“ siehe u.a. [HI09], [PS07], [Wie05] und [Win15].

<sup>2</sup>Zum Begriff „Petri-Schlinge“ siehe u.a. [BKK<sup>+</sup>14, S. 3].

tienten  $\eta_j$  geordnet, d.h.

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_{\kappa(p)} \text{ für alle } t_j \in TA_{lok}(p)$$

mit

$$\eta_j = \begin{cases} \frac{\psi(t_j, p)}{f(t_j, p)} & \text{falls } t_j \in \bullet p \wedge t_j \notin p \bullet \\ \frac{\psi(p, t_j)}{f(p, t_j)} & \text{falls } t_j \notin \bullet p \wedge t_j \in p \bullet \\ \frac{\psi(t_j, p) + \psi(p, t_j)}{f(t_j, p) + f(p, t_j)} & \text{falls } t_j \in \bullet p \wedge t_j \in p \bullet \end{cases}$$

### Algorithmus 2.22: Lokale Freigabe mittels Branch & Bound bei kooperativer Konfliktlösung

Setze:

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

$gIn_0 := 0$  (Input-Gewicht der Teillösung)

$gOut_0 := 0$  (Out-Gewicht der Teillösung)

$n_0 := \sum_{t \in TA_{out}(p)} \psi(t, p) + \sum_{t \in TA_{in}(p)} \psi(p, t)$  (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  $m_r < \kappa(p)$ , führe aus:

Setze:

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

$n_{s+1} := n_r$ ;

$$n_{s+2} := \begin{cases} n_r - \psi(t_{m_r+1}, p) & \text{falls } t_{m_r+1} \in \bullet p \wedge t_{m_r+1} \notin p \bullet \\ n_r - \psi(p, t_{m_r+1}) & \text{falls } t_{m_r+1} \notin \bullet p \wedge t_{m_r+1} \in p \bullet ; \\ n_r - \psi(t_{m_r+1}, p) - \psi(p, t_{m_r+1}) & \text{falls } t_{m_r+1} \in \bullet p \wedge t_{m_r+1} \in p \bullet \end{cases}$$

Falls  $t_{m_r+1} \in \bullet p$ , führe aus:

$$gIn_{s+1} := gIn_r + f(t_{m_r+1}, p);$$

$$gIn_{s+2} := gIn_r;$$

Falls  $t_{m_r+1} \in p\bullet$ , führe aus:

$$gOut_{s+1} := gOut_r + f(p, t_{m_r+1});$$

$$gOut_{s+2} := gOut_r;$$

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

$$TPg\ddot{u}l\text{t}ig :=$$

$$\{X_{s+i} \mid \mathbf{z}(p) - gOut_{s+i} \geq C_{min}(p) \wedge \mathbf{z}(p) + gIn_{s+i} \leq C_{max}(p) \wedge n_{s+i} > N, i = 1, 2\}$$

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

**3. Schritt:** Falls  $(m_r = \kappa(p) - 1) \wedge (TPg\ddot{u}l\text{t}ig \neq \emptyset)$ , führe aus:

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

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:

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

**Szenarium 4**

Der *nebenläufige Feuerprozess (kooperative Konfliktlösung bei Petri-Netzen mit Kapazitäten)* verfährt nach den folgenden Schritten:

1. Ermittlung der Menge  $TA$  aller aktiven Transitionen (gemäß Def. 2.4)
2. Ermittlung aller lokalen Freigabemengen  $TE_{lok}(p)$  aus jeweiligem  $TA_{lok}(p)$  via Alg. 2.22
3. Ermittlung der globalen Freigabemenge  $TE$  mit:

$$TE := \{t \in TA \mid (\forall p \in \bullet t : t \in TE_{lok}(p)) \wedge (\forall p \in t \bullet : t \in TE_{lok}(p))\}$$

4. Nebenläufiges Feuern der Menge  $TE$

**Beispiel:**

Am Petri-Netz mit Kapazitäten und Nutzen aus Abb. 2.9 soll der nebenläufige Feuerprozess mit kooperativer Konfliktlösung nachvollzogen werden.

1. Schritt:

Es gilt:

$$TA = \{t_1, t_2, t_3, t_4, t_5\}$$

2. Schritt:

Es werden folgende Mengen gebildet:

$$TA_{lok}(p_1) = \{t_1, t_2\}$$

$$TA_{lok}(p_2) = \{t_1, t_2, t_3, t_4, t_5\}$$

$$TA_{lok}(p_3) = \{t_4, t_5\}$$

Bei der hieraus folgenden lokalen Input-Konfliktlösung entstehen die folgenden lokalen Freigabemengen:

$$TE_{lok}(p_1) = \{t_2\}$$

$$TE_{lok}(p_2) = \{t_1, t_3, t_5\}$$

$$TE_{lok}(p_3) = \{t_4\}$$

3. Schritt:

Daraus ergibt sich als globale Freigabemenge:

$$TE = \{t_3\}$$

## 2 Kapazitierte Petri-Netze

### 4. Schritt:

Es wird somit die Menge  $TE = \{t_3\}$  gefeuert.

### 3 Kapazitierte gefärbte Petri-Netze

In diesem Kapitel sollen nun die Konzepte der kapazitierten Petri-Netze auf die gefärbten Petri-Netze mit Kapazitäten übertragen und erweitert werden. Zunächst wird eine schematische Herleitung der gefärbten Petri-Netze (CPN) aus den Petri-Netzen (PN) an einem einfachen Beispiel demonstriert.<sup>1</sup>

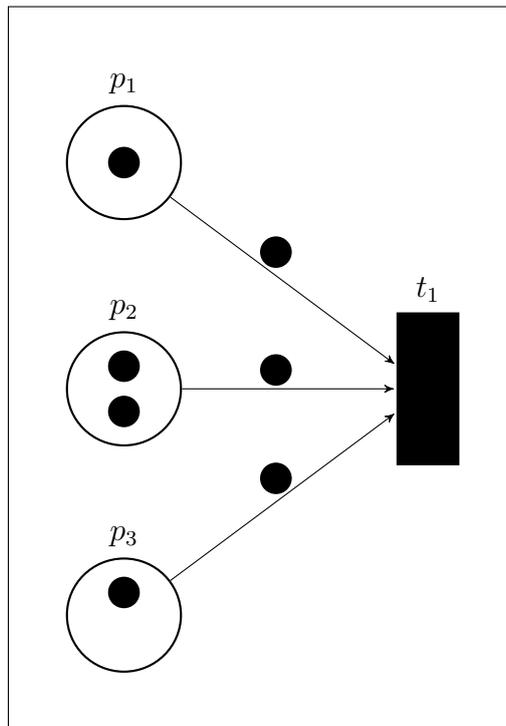


Abbildung 3.1: „Normales“ Petri-Netz

Betrachtet man das Petri-Netz aus Abb. 3.1, 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 klargemacht werden müsste. Daher stellt sich die Frage: Wenn die Token unterschiedliche Objekte darstellen, warum unterscheiden sich die einzelnen Token nicht

<sup>1</sup>Für eine ausführlichere Herleitung sei hier auf [Ger04] verwiesen.

### 3 Kapazitierte gefärbte Petri-Netze

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. 3.1. 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. 3.2).

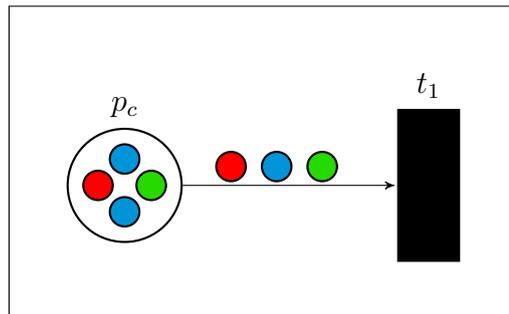


Abbildung 3.2: Gefärbtes Petri-Netz nach Faltung

In dieser Art kann man sich zunächst einmal den originären Zusammenhang zwischen PN und CPN bildlich vorstellen. Im Folgenden werden daher den Plätzen jeweils eine *Klasse* (nämlich mit der Klasse „FARBE“) zugeordnet. Ein Platz kann nur Token enthalten, welche eine Farbe haben, die in ihrer Klasse enthalten ist. Hierbei wird der Formalismus der *Multimenge* verwendet (Grundlagen hierzu siehe Anhang).

Das Prinzip der „Faltung“ lässt sich auf Petri-Netze mit Kapazitäten übertragen, indem die Kapazitätsbeschränkungen entsprechend der Farbzuordnung

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

aggregiert werden und zu einer Multimenge zusammengefasst werden (vgl. Abb 3.3 und Abb 3.4).

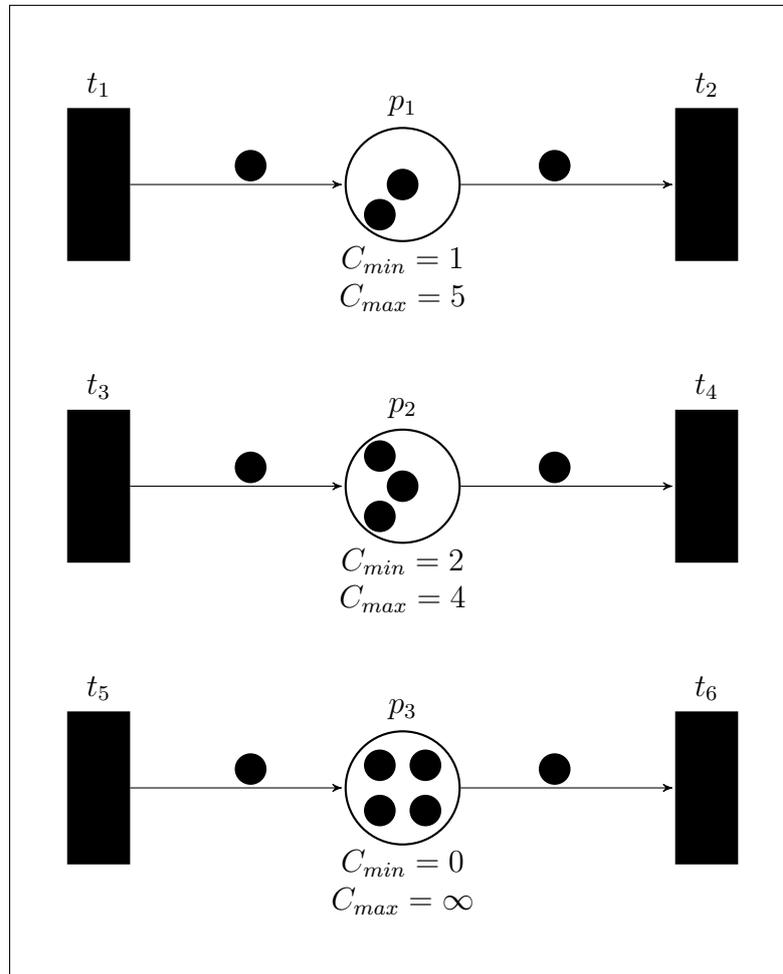


Abbildung 3.3: Mehrere kapazitierte Petri-Netze

Bei dem gefärbten Petri-Netz in Abb. 3.4 müsste der Zustand von  $p_c$  entsprechend der jeweiligen Kapazitätsgrenzen der einzelnen Plätze  $p_1$ ,  $p_2$  und  $p_3$  mindestens  $\{(ROT, 1), (BLAU, 2), (GRÜN, 0)\}$  und höchstens  $\{(ROT, 5), (BLAU, 4), (GRÜN, \infty)\}$  sein.

Zusätzlich könnte man noch die Bedingungen einfügen, dass eine minimale und maximale Gesamtanzahl von Token in einem Platz enthalten sein muss, wobei die Farben der Token keine Rolle spielen. Beispielsweise soll im Platz  $p_c$  in Abb. 3.5 zusätzlich zu den Kapazitätseinschränkungen für die einzelnen Farben (vgl. Abb. 3.4) auch noch verlangt sein, dass mindestens fünf und höchstens zehn Token beliebiger Farbe enthalten sind (*Gesamtkapazitäten*).

### 3 Kapazitierte gefärbte Petri-Netze

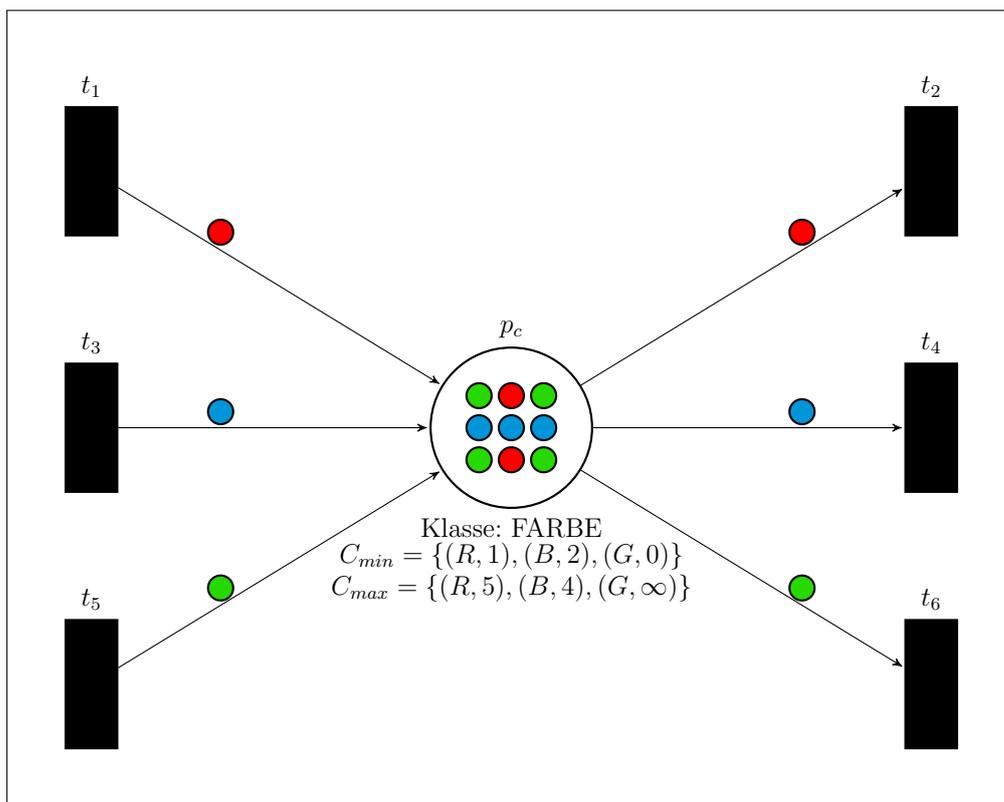


Abbildung 3.4: Kapazitiertes gefärbtes Petri-Netz

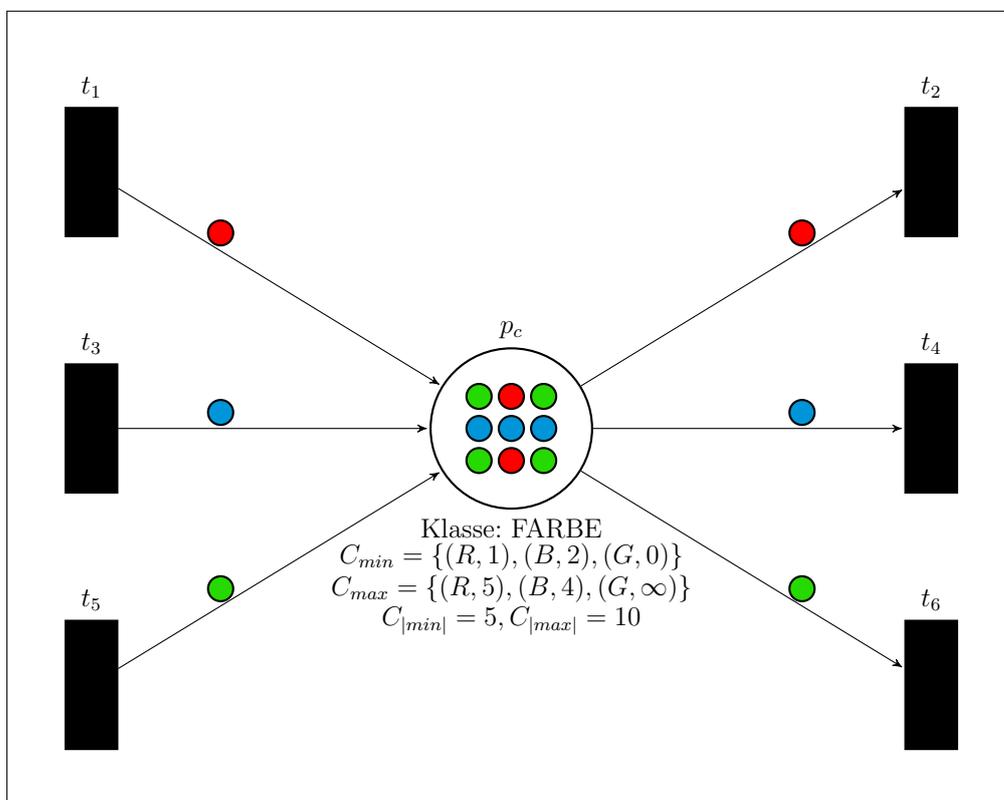


Abbildung 3.5: Kapazitiertes gefärbtes Petri-Netz mit Gesamtkapazität

Während die Kapazitätsgrenzen der einzelnen Klassenelemente (hier Farben) als Multimengen  $C_{min}$  und  $C_{max}$  formalisiert werden, sind die Gesamtkapazitäten  $C_{|min|}$  und  $C_{|max|}$  eines Platzes als „natürliche“ Zahlen angegeben, zumal  $C_{|min|} = 0$  und  $C_{|max|} = \infty$  der Einfachheit halber weggelassen werden (können).

Man beachte, dass die Gesamtkapazitäten  $C_{|min|}$  und  $C_{|max|}$  nur dann explizit gesetzt werden (müssen), wenn

$$C_{|min|} > |C_{min}|$$

bzw.

$$C_{|max|} < |C_{max}|$$

gilt; andernfalls wäre die zusätzliche Angabe redundant. Die Gesamtkapazität kann man sich wie in Abb. 3.6 dargestellt herleiten (vgl. auch Abb. 2.1 und Abb. 2.2).

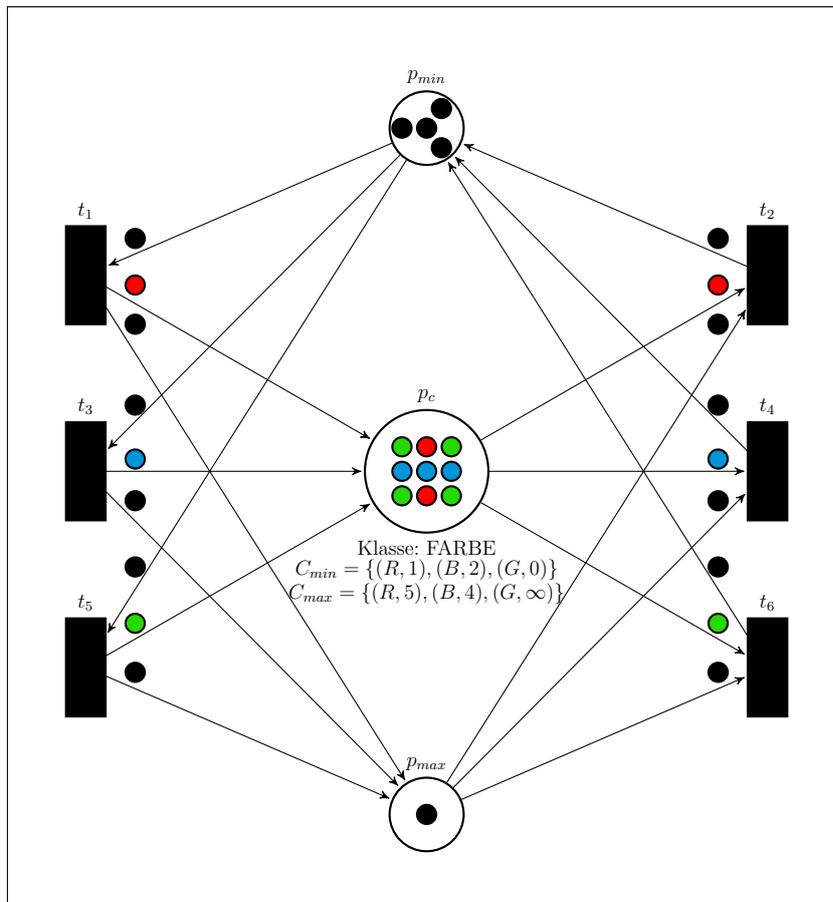


Abbildung 3.6: Äquivalentes Modell zu einem kapazitierten gefärbten Petri-Netz mit Gesamtkapazität (vgl. Abb. 3.5)

### 3.1 Formale Definition gefärbter Petri-Netze mit Kapazitäten

Formal lassen sich CPN in unterschiedlichen Komplexitätsstufen definieren (vgl. [BKK<sup>+</sup>14], [Ger04],[JK09]), auf denen Kapazitäten eingeführt werden könnten. In diesem Artikel wird als Basis-Definition das *mehrfach attribuierte gefärbte Petri-Netz* (MCPN<sup>1</sup>) aus [BKK<sup>+</sup>14, Def. 4.4] verwendet.

**Definition 3.1:**

Ein *mehrfach attribuiertes gefärbtes Petri-Netz mit Kapazitäten* ist ein Tupel  $N = (P, T, F, B, f, A, K, k, C_{max}, C_{min}, C_{|max|}, C_{|min|}, \mathbf{z})$ , 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;
- $K = \{K_1, \dots, K_x\}$  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 Stützmeng von  $M$  gilt:

$$supp(M) \subseteq k(p) \subseteq \mathbb{K};^2$$

<sup>1</sup>Die Abkürzung MCPN stammt vom engl. Begriff multi-attributed coloured Petri net.

<sup>2</sup>Man beachte, dass  $supp(M) \neq \emptyset$  sein muss.

- $C_{max} : P \rightarrow \mathbb{M}(\mathbb{K})$  eine Abbildung ist, welche jedem  $p \in P$  eine **Maximal-Kapazität**  $C_{max}(p)$  zuordnet, wobei für die Stützmenge von  $C_{max}(p)$  gilt:

$$supp(C_{max}(p)) \subseteq k(p);^1$$

- $C_{min} : P \rightarrow \mathbb{M}(\mathbb{K})$  eine Abbildung ist, welche jedem  $p \in P$  eine **Minimal-Kapazität**  $C_{min}(p)$  zuordnet, wobei für die Stützmenge von  $C_{min}(p)$  gilt:

$$supp(C_{min}(p)) \subseteq k(p);$$

- $C_{|max|} : P \rightarrow (\mathbb{N} \cup \infty)$  eine Abbildung ist, welche jedem  $p \in P$  eine **Gesamt-Maximal-Kapazität**  $C_{|max|}(p)$  zuordnet;

- $C_{|min|} : P \rightarrow \mathbb{N}_0$  eine Abbildung ist, welche jedem  $p \in P$  eine **Gesamt-Minimal-Kapazität**  $C_{|min|}(p)$  zuordnet;

- $\mathbf{z} : P \rightarrow \mathbb{M}(\mathbb{K})$  eine Abbildung ist, die jedem Platz  $p \in P$  eine der Klasse  $k(p)$  entsprechende Multimenge als Gewicht zuweist; diese 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:

$$supp(M_i) \subseteq k(p_i).$$

Zudem muss gelten:

$$C_{min}(p) \subseteq \mathbf{z}(p) \subseteq C_{max}(p) \text{ für alle } p \in P$$

und

$$C_{|min|}(p) \leq |\mathbf{z}(p)| \leq C_{|max|}(p) \text{ für alle } p \in P.$$

### Bemerkung 3.2:

Für gewisse Anwendungen könnte es vorteilhaft sein, die Zustände eines MCPN mit Kapazitäten als zulässige oder unzulässige Zustände unterscheiden zu können. Dazu böte sich an, den Zustandsbegriff in Def. 3.1 allgemeiner zu fassen und die letzten Forderungen zunächst wegzulassen. Dann könnte man den Zustand  $\mathbf{z}$  eines Platzes als zulässig (bzw. unzulässig)

<sup>1</sup>Man beachte, dass  $supp(C_{max}(p)) \neq \emptyset$  sein muss.

bezeichnen, wenn die Nebenbedingungen erfüllt sind (oder nicht). Darüber hinaus könnte auch noch eine Graduierung der Zulässigkeit eines Zustandes vorgenommen werden, indem dahingehend unterschieden wird, welche Bedingungen erfüllt sind oder nicht. Auf eine solche Verfeinerung des Zustandes wird hier allerdings verzichtet.

**Bemerkung 3.3:**

Aufgrund von großen Redundanzen ist es in einer grafischen Darstellung des MCPN mit Kapazitäten empfehlenswert, wenn die Gewichte der einzelnen Pfeile und die Markierungen bzw. Minimal-Kapazität der einzelnen Plätze nur aus der Stützmenge gebildet werden. Bei der Maximal-Kapazität kann man nur die Elemente aus  $k(p)$  darstellen, deren Häufigkeit kleiner  $\infty$  ist, wobei damit eine hinreichend große natürliche Zahl gemeint ist, die in der Realsituation nicht erreicht werden kann und somit eigentlich auch keine echte Beschränkung darstellt.

**Bemerkung 3.4:**

Soll ein Platz ohne Kapazitäten (oder nicht mit allen Kapazitätsschranken) durch ein MCPN mit Kapazitäten dargestellt werden, kann man dies durch die Zuordnung einer Minimal-Kapazität  $C_{min}(p) = \{(a, 0) \mid a \in \mathbb{K}\}$ , Gesamt-Minimal-Kapazität  $C_{|min|}(p) = 0$ , Maximal-Kapazität  $C_{max}(p) = \{(a, \infty) \mid a \in k(p)\}$  bzw. einer Gesamt-Maximal-Kapazität  $C_{|max|}(p) = \infty$  erreichen. Zur besseren Übersichtlichkeit kann in diesen Fällen die Kapazitäten-Beschriftung am Platz auch entfallen.

**Beispiel:**

In diesem Beispiel wird ein mehrfach attribuiertes gefärbtes Petri-Netz mit Kapazitäten  $N = (P, T, F, B, f, A, K, k, C_{max}, C_{min}, C_{|max|}, C_{|min|}, \mathbf{z})$  konstruiert:

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

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

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$$

$$F = \{(p_1, t_1), (p_1, t_2), (p_1, t_3), (p_2, t_4), (p_2, t_5), (p_2, t_6)\}$$

$$B = \{(t_1, p_2), (t_2, p_2), (t_3, p_2), (t_4, p_3), (t_5, p_3), (t_6, p_3)\}$$

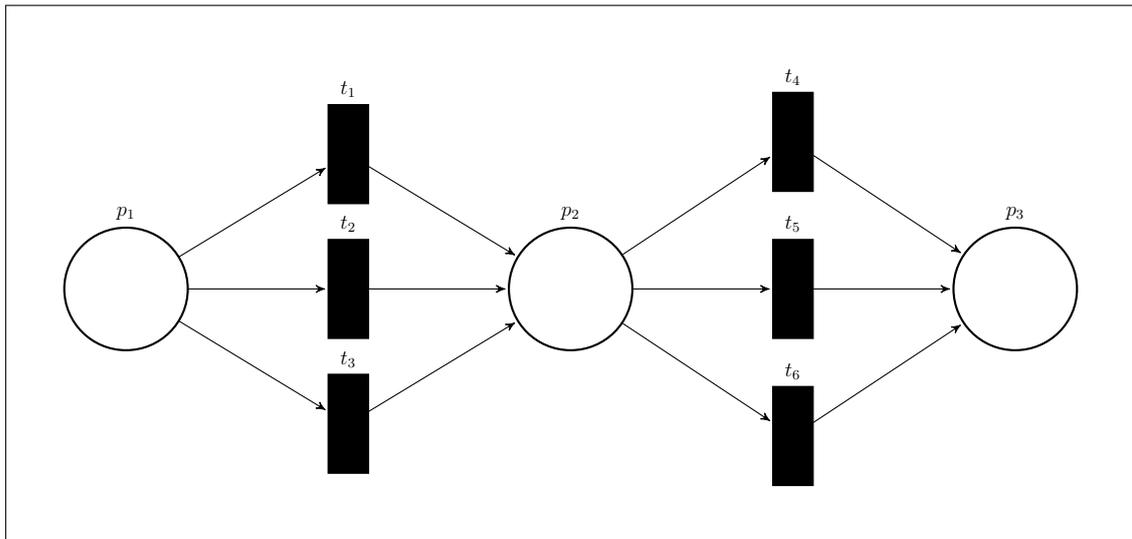


Abbildung 3.7: Beispiel eines Netzes

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:

$$K = \{\text{FARBE}, \text{ZAHL}, \text{BUNTEZAHL}\} \text{ mit:}$$

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

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

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

Auf Grund der Übersichtlichkeit werden die Attribute ROT, BLAU und GRÜN auch in der abgekürzten Form benutzt: R=ROT, B=BLAU und G= GRÜN.

3. Die Klassifizierungsfunktion der Plätze:

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

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

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

### 3 Kapazitierte gefärbte Petri-Netze

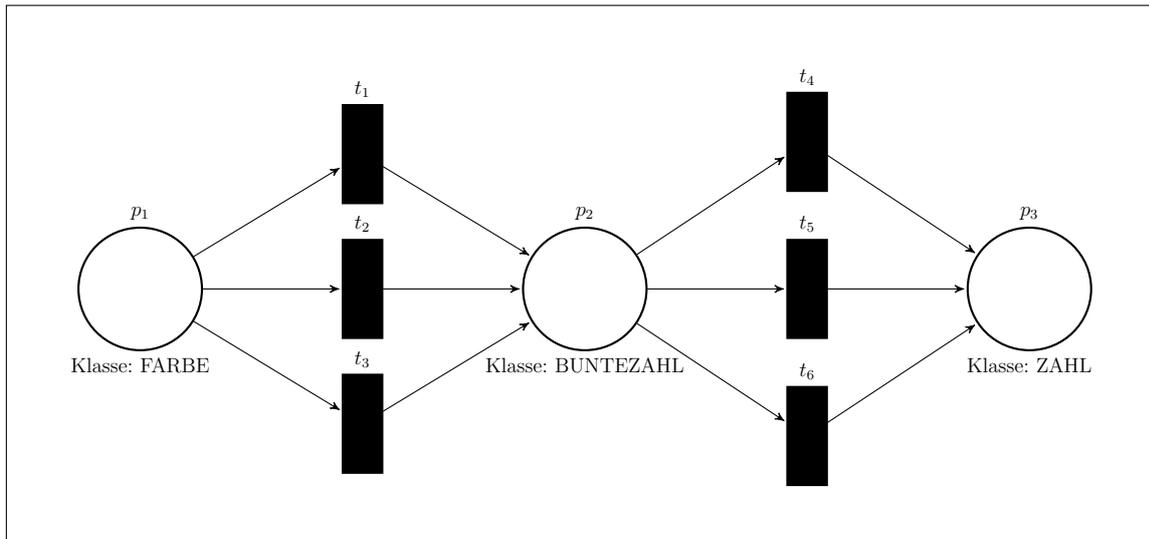


Abbildung 3.8: Beispiel eines Netzes mit Klassen

4. Die Gewichtung der Pfeile:

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

$$f(p_1, t_2) \equiv \{(\text{GRÜN}, 1)\}$$

$$f(p_1, t_3) \equiv \{(\text{BLAU}, 1)\}$$

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

$$f(t_2, p_2) \equiv \{((\text{GRÜN}, 3), 1)\}$$

$$f(t_3, p_2) \equiv \{((\text{BLAU}, 1), 1)\}$$

$$f(p_2, t_4) \equiv \{((\text{ROT}, 2), 1)\}$$

$$f(p_2, t_5) \equiv \{((\text{GRÜN}, 3), 1)\}$$

$$f(p_2, t_6) \equiv \{((\text{BLAU}, 1), 1)\}$$

$$f(t_4, p_3) \equiv \{(2, 1)\}$$

$$f(t_5, p_3) \equiv \{(3, 1)\}$$

$$f(t_6, p_3) \equiv \{(1, 1)\}$$

Eigentlich hätte in diesem Beispiel jede Pfeilgewichtung 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 3.3 verwendet.

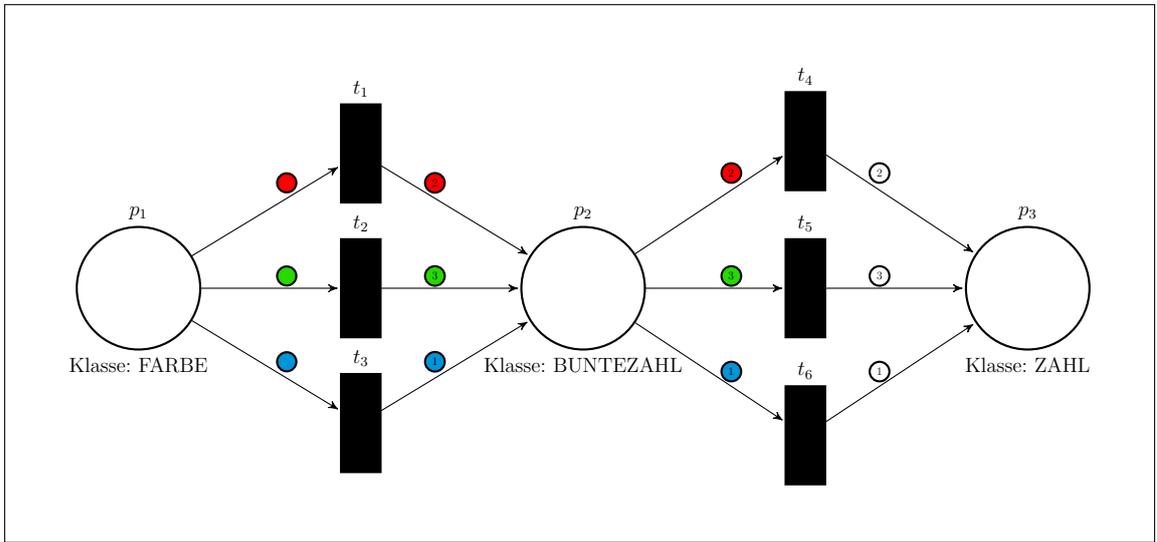


Abbildung 3.9: Beispiel eines gewichteten MCPN

#### 5. Kapazitäten der Plätze:

Für  $p_1$ :

$$C_{\min}(p_1) \equiv \{(\text{ROT}, 1), (\text{BLAU}, 1), (\text{GRÜN}, 2)\},$$

$$C_{\max}(p_1) \equiv \{(a, \infty) \mid a \in k(p_1)\},$$

$$C_{|\min|}(p_1) = 5,$$

$$C_{|\max|}(p_1) = \infty.$$

Für  $p_2$ :

$$\begin{aligned} C_{\min}(p_2) &\equiv \{((\text{ROT}, 2), 1), ((\text{BLAU}, 2), 1)\} \\ &\equiv \left\{ \begin{array}{l} ((\text{ROT}, 1), 0), ((\text{ROT}, 2), 1), ((\text{ROT}, 3), 0), \\ ((\text{BLAU}, 1), 0), ((\text{BLAU}, 2), 1), ((\text{BLAU}, 3), 0), \\ ((\text{GRÜN}, 1), 0), ((\text{GRÜN}, 2), 0), ((\text{GRÜN}, 3), 0) \end{array} \right\}, \\ C_{\max}(p_2) &\equiv \{((\text{BLAU}, 1), 2), ((\text{GRÜN}, 3), 1)\} \end{aligned}$$

### 3 Kapazitierte gefärbte Petri-Netze

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

$$C_{|\min|}(p_2) = 8,$$

$$C_{|\max|}(p_2) = 10.$$

Für  $p_3$ :

$$C_{\min}(p_3) = \{(a, 0) \mid a \in \mathbb{K}\},$$

$$C_{\max}(p_3) \equiv \{(1, 2), (2, 2), (3, 2)\},$$

$$C_{|\min|}(p_3) = 0,$$

$$C_{|\max|}(p_3) = 5.$$

Eigentlich hätte in diesem Beispiel jede Minimal- bzw. Maximal-Kapazität in der Form

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

bzw.

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

dargestellt werden müssen, aber wie man sieht, ist diese Form sehr umständlich und unübersichtlich, daher wird hier die reduzierte Entsprechungsform gemäß Bemerkung 3.3 verwendet.

#### 6. Der Zustand:

Ein Beispiel einer Markierung des obigen MCPN ist:

$$\mathbf{z}(p_1) = z_1 \equiv \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 2)\}$$

$$\mathbf{z}(p_2) = z_2 \equiv \left\{ \begin{array}{l} ((\text{ROT}, 1), 1), ((\text{ROT}, 2), 1), ((\text{ROT}, 3), 1), \\ ((\text{BLAU}, 1), 1), ((\text{BLAU}, 2), 1), ((\text{BLAU}, 3), 1), \\ ((\text{GRÜN}, 1), 1), ((\text{GRÜN}, 2), 1), ((\text{GRÜN}, 3), 1) \end{array} \right\}$$

$$\mathbf{z}(p_3) = z_3 \equiv \{(1, 1), (2, 1), (3, 1)\}$$

oder als Vektor:

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

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 3.3 verwendet.

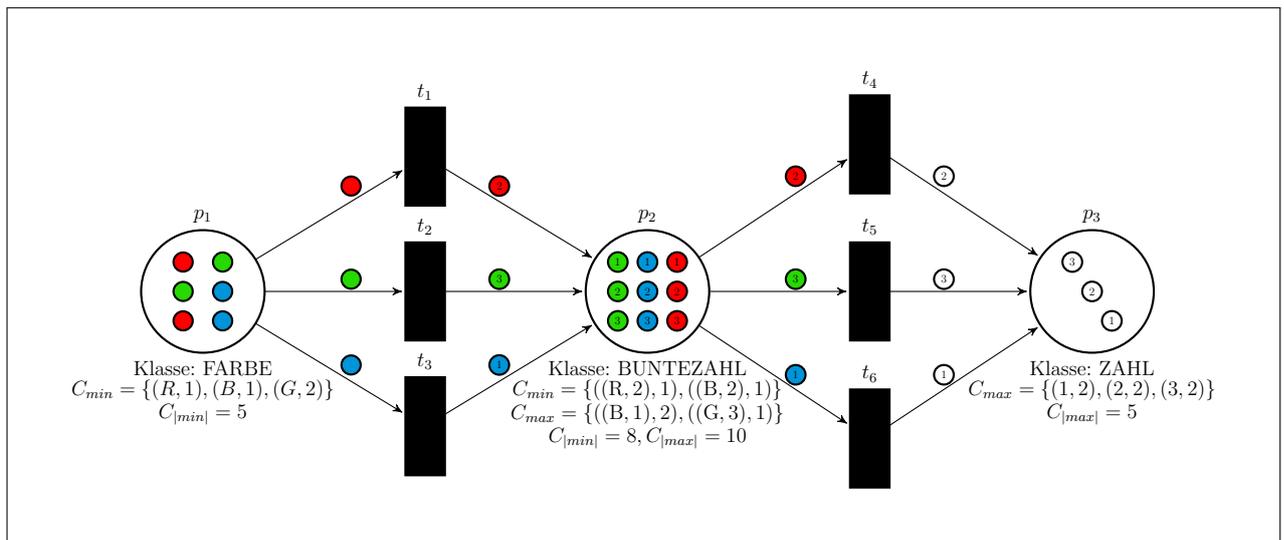


Abbildung 3.10: Vollständiges MCPN mit Kapazitäten

Die in Kapitel 1 und 2 vorgestellten Prozesse der Aktivierung und Feuerung von Transitionen in PN (ohne bzw. mit Kapazitäten) sollen auf MCPN mit Kapazitäten übertragen werden. Hierbei liegt die Hauptschwierigkeit darin, die allgemeinen Operatoren  $+$ ,  $-$ ,  $\leq$  in entspre-

chende Multimengenoperatoren umzuwandeln.<sup>1</sup> Dazu wird folgendes Schema verwendet:

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

**Definition 3.5:**

Es sei  $N = (P, T, F, B, f, A, K, k, C_{max}, C_{min}, C_{|max|}, C_{|min|}, \mathbf{z})$  ein mehrfach attribuiertes gefärbtes Petri-Netz mit Kapazitäten. Eine Transition  $t \in T$  von  $N$  heißt **aktiviert** oder **seriell feuierbar** im Zustand  $\mathbf{z}$ , wenn:

$$\forall p \in \bullet t \text{ gilt: } (\mathbf{z}(p) \setminus f(p, t) \supseteq C_{min}(p)) \wedge (|\mathbf{z}(p)| - |f(p, t)| \geq C_{|min|}(p))$$

und

$$\forall p \in t \bullet \text{ gilt: } (\mathbf{z}(p) \uplus f(t, p) \subseteq C_{max}(p)) \wedge (|\mathbf{z}(p)| + |f(t, p)| \leq C_{|max|}(p))$$

gilt. Demnach ist die Menge aller aktivierten Transitionen  $TA(\mathbf{z})$  des Petri-Netzes:

$$\begin{aligned} TA(\mathbf{z}) := \{t \in T \mid & (\forall p \in \bullet t : (\mathbf{z}(p) \setminus f(p, t) \supseteq C_{min}(p)) \wedge \\ & (|\mathbf{z}(p)| - |f(p, t)| \geq C_{|min|}(p)) \wedge \\ & (\mathbf{z}(p) \uplus f(t, p) \subseteq C_{max}(p)) \wedge \\ & (|\mathbf{z}(p)| + |f(t, p)| \leq C_{|max|}(p)))\}. \end{aligned}$$

Das **Feuern** einer seriell feuerebaren 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:

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

**Bemerkung 3.6:**

Die Teilmengen  $TA_{in}(p, \mathbf{z})$  und  $TA_{out}(p, \mathbf{z})$  ergeben sich auf kanonische Weise gemäß Def. 2.4.

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.

<sup>1</sup>Hierzu siehe Anhang.

**Definition 3.7:**

Es sei  $N = (P, T, F, B, f, A, K, k, C_{max}, C_{min}, C_{|max|}, C_{|min|}, \mathbf{z})$  ein mehrfach attribuiertes gefärbtes Petri-Netz mit Kapazitäten und  $\{t_1, \dots, t_r\}$  eine Menge von Transitionen.<sup>1</sup> 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. Mit

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

werde der **gemeinsame Nachbereich** der Transitionen  $t_1, \dots, t_r$  bezeichnet. Die Transitionenmenge  $\{t_1, \dots, t_r\}$  heißt **nebenläufig feuierbar** im Zustand  $\mathbf{z}$ , wenn alle Transitionen aus  $\{t_1, \dots, t_r\}$  aktiviert sind<sup>2</sup> und für alle  $p \in \bullet\{t_1, \dots, t_r\}$  gilt:

$$(\mathbf{z}(p) \setminus \biguplus_{t \in \{t_1, \dots, t_r\} \cap p\bullet} f(p, t) \geq C_{min}(p)) \wedge (|\mathbf{z}(p)| - \sum_{t \in \{t_1, \dots, t_r\} \cap p\bullet} |f(p, t)| \geq C_{|min|}(p))$$

sowie für alle  $p \in \{t_1, \dots, t_r\}\bullet$  gilt:

$$(\mathbf{z}(p) \uplus \biguplus_{t \in \{t_1, \dots, t_r\} \cap p\bullet} f(t, p) \leq C_{max}(p)) \wedge (|\mathbf{z}(p)| + \sum_{t \in \{t_1, \dots, t_r\} \cap p\bullet} |f(t, p)| \leq C_{|max|}(p)).$$

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

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

**Definition 3.8:**

Es sei  $N = (P, T, F, B, f, A, K, k, C_{max}, C_{min}, C_{|max|}, C_{|min|}, \mathbf{z})$  ein mehrfach attribuiertes gefärbtes Petri-Netz mit Kapazitäten.

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

$$(\mathbf{z}(p) \setminus \biguplus_{t \in TA_{out}(p)} f(p, t) \not\geq C_{min}(p)) \vee (|\mathbf{z}(p)| - \sum_{t \in TA_{out}(p)} |f(p, t)| < C_{|min|}(p)).$$

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

<sup>2</sup>Gemäß Def. 3.5

### 3 Kapazitierte gefärbte Petri-Netze

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

$$(\mathbf{z}(p) \uplus \bigoplus_{t \in TA_{in}(p)} f(t, p) \not\subseteq C_{max}(p)) \vee (|\mathbf{z}(p)| + \sum_{t \in TA_{in}(p)} |f(t, p)| > C_{|max|}(p)).$$

Man beachte, dass man eine Graduierung der Definition der Konflikte vornehmen könnte, indem dahingehend unterschieden wird, welche Teil-Bedingungen erfüllt sind oder nicht. Auf eine solche Verfeinerung des Konfliktes wird hier allerdings verzichtet.

#### Beispiel:

Im folgenden MCPN mit Kapazitäten aus Abbildung 3.11 soll veranschaulicht werden, welchen Einfluss Kapazitäten darauf haben, ob Transitionen aktiviert sind bzw. ob Plätze einen Konflikt haben.

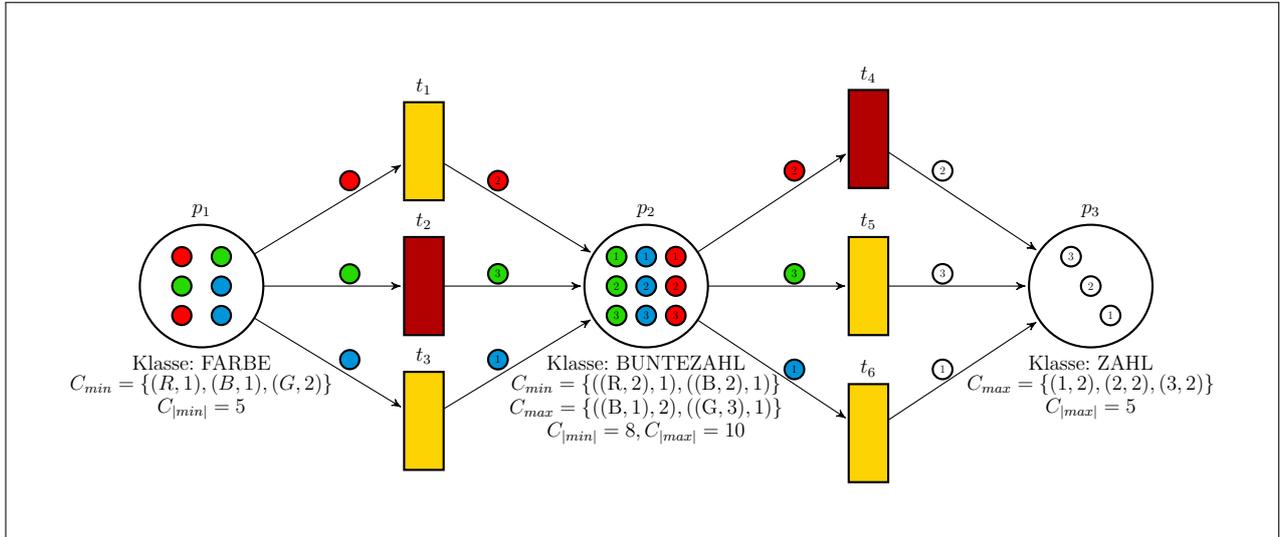


Abbildung 3.11: Aktiviertheit und Konflikte eines MCPN mit Kapazitäten

Die Transitionen  $t_1, t_3, t_5$  und  $t_6$  sind aktiviert, die Transition  $t_2$  und  $t_4$  allerdings nicht. Die Transition  $t_2$  ist auf Grund der Minimal-Kapazität von  $p_1$  und der Maximal-Kapazität von  $p_2$  nicht aktiviert, da

$$\begin{aligned} \mathbf{z}(p_1) \setminus f(p_1, t_2) &\equiv \{(R, 2), (B, 2), (G, 2)\} \setminus \{(G, 1)\} \equiv \{(R, 2), (B, 2), (G, 1)\} \\ &\not\subseteq \{(R, 1), (B, 1), (G, 2)\} \equiv C_{min}(p_1) \end{aligned}$$

und

$$\mathbf{z}(p_2) \uplus f(t_2, p_2) \equiv \left\{ \begin{array}{l} ((R, 1), 1), ((R, 2), 1), ((R, 3), 1), \\ ((B, 1), 1), ((B, 2), 1), ((B, 3), 1), \\ ((G, 1), 1), ((G, 2), 1), ((G, 3), 1) \end{array} \right\} \uplus \{((G, 3), 1)\}$$

$$\begin{aligned} &\equiv \left\{ \begin{array}{l} ((R, 1), 1), ((R, 2), 1), ((R, 3), 1), \\ ((B, 1), 1), ((B, 2), 1), ((B, 3), 1), \\ ((G, 1), 1), ((G, 2), 1), ((G, 3), 2) \end{array} \right\} \not\subseteq \left\{ \begin{array}{l} ((R, 1), \infty), ((R, 2), \infty), ((R, 3), \infty), \\ ((B, 1), 2), ((B, 2), \infty), ((B, 3), \infty), \\ ((G, 1), \infty), ((G, 2), \infty), ((G, 3), 1) \end{array} \right\} \\ &\equiv C_{max}(p_2) \end{aligned}$$

ist. Die Transition  $t_4$  ist auf Grund der Minimal-Kapazität von  $p_2$  nicht aktiviert, da

$$\begin{aligned} \mathbf{z}(p_2) \setminus f(p_2, t_4) &\equiv \left\{ \begin{array}{l} ((R, 1), 1), ((R, 2), 1), ((R, 3), 1), \\ ((B, 1), 1), ((B, 2), 1), ((B, 3), 1), \\ ((G, 1), 1), ((G, 2), 1), ((G, 3), 1) \end{array} \right\} \setminus \{((R, 2), 1)\} \\ &\equiv \left\{ \begin{array}{l} ((R, 1), 1), ((R, 2), 0), ((R, 3), 1), \\ ((B, 1), 1), ((B, 2), 1), ((B, 3), 1), \\ ((G, 1), 1), ((G, 2), 1), ((G, 3), 1) \end{array} \right\} \not\supseteq \left\{ \begin{array}{l} ((R, 1), 0), ((R, 2), 1), ((R, 3), 0), \\ ((B, 1), 0), ((B, 2), 1), ((B, 3), 0), \\ ((G, 1), 0), ((G, 2), 0), ((G, 3), 0) \end{array} \right\} \\ &\equiv C_{min}(p_2) \end{aligned}$$

ist. Der Platz  $p_1$  induziert eine Output-Konfliktsituation, da

$$|\mathbf{z}(p_1)| - \sum_{t \in TA_{out}(p_1)} |f(p_1, t)| = 6 - (1 + 1) = 4 < 5 = C_{|min|}(p_1)$$

ist. Der Platz  $p_2$  induziert sowohl eine Input- als auch eine Output-Konfliktsituation, da

$$|\mathbf{z}(p_2)| + \sum_{t \in TA_{in}(p_2)} |f(t, p_2)| = 9 + (1 + 1) = 11 > 10 = C_{|max|}(p_2)$$

und

$$|\mathbf{z}(p_2)| - \sum_{t \in TA_{out}(p_2)} |f(p_2, t)| = 9 - (1 + 1) = 7 < 8 = C_{|min|}(p_2)$$

ist. Der Platz  $p_3$  induziert keine Konfliktsituation.

## 3.2 Konfliktlösung bei gefärbten Petri-Netzen mit Kapazitäten

Bei der Konfliktlösung in gefärbten Petri-Netzen mit Kapazitäten können dieselben Bewertungsfunktionen wie in Abschnitt 2.2 verwendet werden, des Weiteren auch die vier Konfliktlösungskonzepte aus Abschnitt 2.5, 2.6 und 2.7. Die Algorithmen aus Abschnitt 2.3 müssen allerdings auf den Multimengen-Formalismus bzw. die neue Gesamtkapazitäten-Restriktion angepasst werden. Da diese Änderungen hauptsächlich an den Zulässigkeitsbedingungen vorgenommen werden müssen, wird hier exemplarisch nur der Algorithmus für Prioritäten verallgemeinert.<sup>1</sup>

<sup>1</sup>In [BKK<sup>+</sup>14, Kap. 5] kann die Verallgemeinerung der Algorithmen für Wahrscheinlichkeiten und Nutzen für den Multimengen-Formalismus nachvollzogen werden.

### Prioritäten

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

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

bzw.

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

#### Algorithmus 3.9: Lokale Freigabe mittels Prioritätenverfahren bei Output-Konflikt in gefärbten Petri-Netzen

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

Für  $i = 1, \dots, \kappa_{out}(p)$  führe aus:

Falls gilt:  $|\mathbf{z}(p)| - |f(p, t_i)| - \sum_{t \in TE_{out}(p)} |f(p, t)| \geq C_{|min|}(p)$ , führe aus:

Falls gilt:  $\mathbf{z}(p) \setminus f(p, t_i) \setminus \bigcup_{t \in TE_{out}(p)} f(p, t) \supseteq C_{min}(p)$ ,

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

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

#### Algorithmus 3.10: Lokale Freigabe mittels Prioritätenverfahren bei Input-Konflikt in gefärbten Petri-Netzen

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

Für  $i = 1, \dots, \kappa_{in}(p)$  führe aus:

Falls gilt:  $|\mathbf{z}(p)| + |f(t_i, p)| + \sum_{t \in TE_{in}(p)} |f(t, p)| \leq C_{|max|}(p)$ , führe aus:

Falls gilt:  $\mathbf{z}(p) \uplus f(t_i, p) \uplus + \uplus_{t \in TE_{in}(p)} f(t, p) \subseteq C_{max}(p)$ ,  
 setze:  $TE_{in}(p) := TE_{in}(p) \cup \{t_i\}$ .

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

**Beispiel:**

In Abbildung 3.12 ist ein MCPN mit Kapazitäten und normierter Prioritätsfunktion dargestellt.

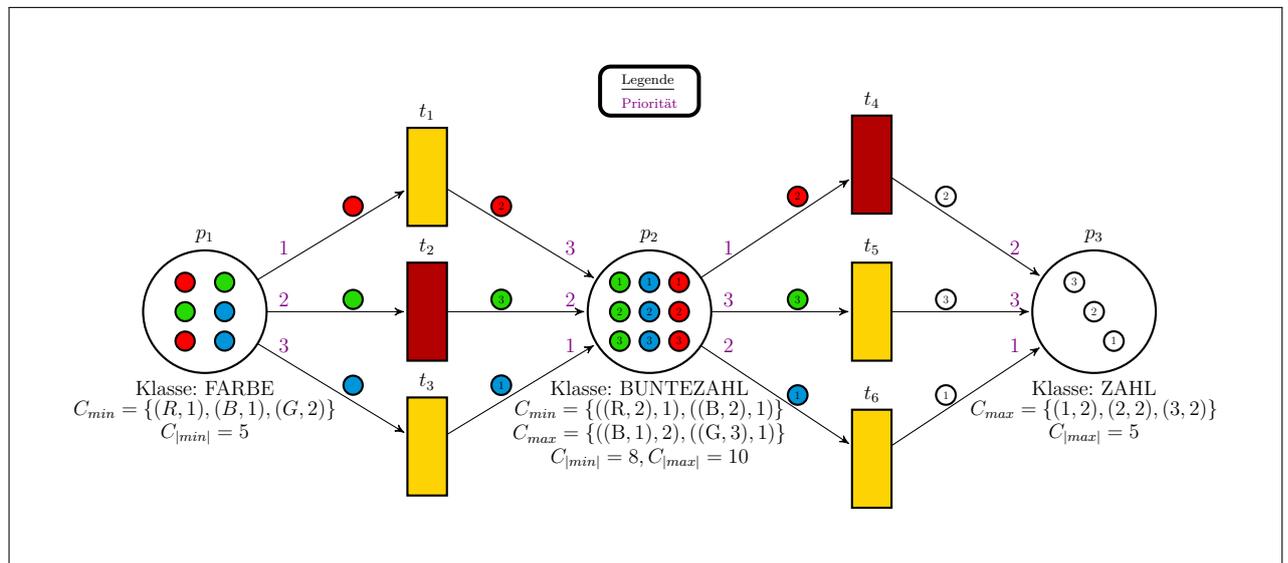


Abbildung 3.12: MCPN mit Kapazitäten und Prioritäten

Für das MCPN mit Kapazitäten und Prioritäten aus Abb. 3.12 ergibt sich:

$$\begin{aligned} TE_{out}(p_1) &= \{t_1\}, \\ TE_{out}(p_2) &= \{t_6\}, \\ TE_{out}(p_3) &= \emptyset \end{aligned}$$

bzw.

$$\begin{aligned} TE_{in}(p_1) &= \emptyset, \\ TE_{in}(p_2) &= \{t_3\}, \\ TE_{in}(p_3) &= \{t_6\}. \end{aligned}$$

In diesem Artikel wurden bis jetzt nur CPN mit einer *konstanten* Pfeilgewichtung behandelt. Die allgemeinere und in der Praxis häufig benutzte Form von CPN mit *variabler* Pfeilgewichtung wird unter anderem in [BKK<sup>+</sup>14, Kapitel 6] beschrieben. Auch auf solche CPN mit Variablen lassen sich Kapazitäten definieren, in äquivalenter Weise wie in diesem Artikel dargestellt; auf eine explizite Ausformulierung sei daher hier verzichtet.

## 4 Ausblick

In diesem Artikel werden Formalismen und Lösungsstrategien in Konfliktsituationen für Petri-Netze aus [BKK<sup>+</sup>14], allerdings dort noch ohne Betrachtung von Kapazitätsbeschränkungen, in konsequenter Weise auf kapazitierte Petri-Netze übertragen und weiterentwickelt. Dabei ergeben sich insbesondere bei gefärbten Petri-Netzen mit Kapazitäten interessante Fallunterscheidungen und Wahlmöglichkeiten, die das Untersuchungsgebiet der Petri-Netze auf das bislang in der einschlägigen Literatur nur spärlich behandelte Teilgebiet der Petri-Netze mit Kapazitäten signifikant erweitern.

Bei den gefärbten Petri-Netzen mit Kapazitäten wird der allgemeinere Fall der Gewichtung durch Variablen bislang noch ausgespart; er ist allerdings Gegenstand der weitergehenden Untersuchung. Zudem stehen die folgenden Fragestellungen im Blickfeld von zukünftigen Forschungsvorhaben:

Bei einer Differenzierung zwischen zulässigen und unzulässigen Zuständen von kapazitierten Petri-Netzen in Bezug auf Einhaltung der Kapazitätsbeschränkungen (vgl. Bemerkung 2.2 und Bemerkung 3.2) könnte es von großem Interesse sein, ein Petri-Netz aus einem anfänglich oder zwischenzeitlich unzulässigen Zustand auf effiziente Weise in einen zulässigen Zustand zu überführen. Hierbei kann sich die Effizienz an einer aufwands- oder kostenminimalen Rechnung (Feuerreihenfolge) ausrichten (falls existent), wobei sich die Frage nach effizienten Strategien zur Erlangung der Zulässigkeit der Zustände von Petri-Netzen mit Kapazitäten anknüpft.

Bei den kooperativen Konfliktlösungen (Abschn. 2.7) wird bereits die Frage nach Strategien mit dem Ziel eines maximalen Gesamtnutzens unter Beibehaltung der Zulässigkeit aller Zustände eines kapazitierten Petri-Netzes gestellt. Hierbei könnte auch die Einführung von „harten“ und „weichen“ Kapazitätsschranken von großem Interesse für die Modellierung praxisrelevanter Gegebenheiten sein, wobei harte Schranken immer und bedingungslos, weiche Schranken hingegen nur „möglichst“ eingehalten werden müssen, letztere also ggf. unter An-

rechnung von Strafkosten auch verletzt werden können. Auch hier stellt sich wieder die Frage nach effizienten Strategien mit dem Ziel nutzenmaximaler Rechnungen in Petri-Netzen mit Kapazitäten.

Ergänzend könnten für die gefärbten Petri-Netze mit Kapazitäten (Kap. 3) weitere/andere Kapazitäts-Restriktionen (wie z.B. in [CH93]) in die Entwicklung allgemeinerer Konfliktlösungskonzepte einbezogen werden.

# Literaturverzeichnis

- [BKK<sup>+</sup>14] BACHMANN, B., KLEINE-DÖPKE, T., KRUSE, H.-J., OCHEL, L. und PROSS, S. : Petri-Netz-Formalismen und Lösungsansätze für allgemeine Konfliktsituationen bei Feuerprozessen in Petri-Netz-Modellen. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Band 2, 2014.
- [CH93] CHRISTENSEN, S. und HANSEN, N.D.: Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. Lecture Notes in Computer Science, Volume 691, 186-205, Berlin Heidelberg, 1993.
- [Die10] DIESTEL, R.: Graphentheorie. Stuttgart, 4. Auflage, 2010.
- [Ger04] GERBER, S.: Petri-Netze 2. Institut für Informatik, Universität Leipzig, Sommersemester 2004.
- [HI09] HOLLER, M.J. und ILLING, G.: Einführung in die Spieltheorie. Berlin, 7. Auflage, 2009.
- [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.
- [Mon01] MONRO, G.P.: The Concept of Multiset. In: Proceedings of the Workshop on Multiset Processing, 347-358, London 2001.
- [Pro13] PROSS, S.: Hybrid Modeling and Optimization of Biological Processes. Forschungsreihe des Fachbereichs Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, Band 1, 2013.

- [Pro14] PROSS, S.: Diskrete Modellierung und Optimierung praxisrelevanter Prozesse mit Petri-Netzen. AMMO - Berichte aus Forschung und Technologietransfer, Fachhochschule Bielefeld, Heft Nr. 4, 2014.
- [PS07] PELEG, B. und SUDHÖLTER, P.: Introduction to the theory of cooperative games. Heidelberg, 2. Auflage, 2007.
- [Syr87] SYROPOULOS, A.: Mathematics of Multisets. Zeitschrift f. math. Logik und Grundlagen d. Math., 33:171–178, 1987.
- [Vol96] VOLKMANN, L.: Fundamente der Graphentheorie. Wien, 1996.
- [Wie05] WIESE, H.: Kooperative Spieltheorie. München, 2005.
- [Win15] WINTER, S.: Grundzüge der Spieltheorie. Berlin Heidelberg, 2015.

# 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  $M$  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 zwei Multimengen:

$$M_1 = \{(\text{ROT}, 3), (\text{BLAU}, 2), (\text{GRÜN}, 1)\},$$

$$M_2 = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 1)\}.$$

$M_2$  ist eine (echte) Teilmenge von  $M_1$  ( $M_2 \subseteq M_1$ ):

$$h_2(\text{ROT}) = 2 \leq 3 = h_1(\text{ROT}),$$

$$h_2(\text{BLAU}) = 1 \leq 2 = h_1(\text{BLAU}),$$

$$h_2(\text{GRÜN}) = 1 \leq 1 = h_1(\text{GRÜN}).$$

**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:

$$M_1 = \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\},$$

$$M_2 = \{(\text{ROT}, 2), (\text{BLAU}, 1), (\text{GRÜN}, 2)\}.$$

(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  $A = \{\text{ROT, BLAU, GRÜN}\}$  sei die Multimenge

$$M_1 = \{(\text{ROT}, 2), (\text{BLAU}, 2), (\text{GRÜN}, 1)\}$$

und über der Menge  $B = \{\text{ROT, PINK, SCHWARZ}\}$  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

$$\bigoplus_{i=1, \dots, N} M_i := M_1 \uplus M_2 \uplus M_3 \uplus \dots \uplus M_{N-1} \uplus M_N$$

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  $A$ .

## **Autoren und Kontakte**

### **Autoren**

Timo Kleine-Döpke, M. Sc.

Prof. Dr. rer. pol. Hermann-Josef Kruse

### **Kontaktdaten**

Fachhochschule Bielefeld  
Fachbereich Ingenieurwissenschaften und Mathematik  
FSP Angewandte Mathematische Modellierung und Optimierung

Am Stadtholz 24  
33609 Bielefeld

*Ansprechpartner:* Prof. Dr. Dr. Rainer Ueckerdt

*Tel.:* +49.521.106-7403

*Fax:* +49.521.106-7176

*E-Mail:* [ammo@fh-bielefeld.de](mailto:ammo@fh-bielefeld.de)

*Web:* [www.fh-bielefeld.de/ammo](http://www.fh-bielefeld.de/ammo)

*Ansprechpartner:* Prof. Dr. Hermann-Josef Kruse

*Tel.:* +49.521.106-7411

*Fax:* +49.521.106-7176

*E-Mail:* [hermann-josef.kruse@fh-bielefeld.de](mailto:hermann-josef.kruse@fh-bielefeld.de)

Schulstrasse 10  
33330 Gütersloh

*Ansprechpartner:* Timo Kleine-Döpke, M. Sc.

*Tel.:* +49.5241.21143-27

*Fax:* +49.5241.21143-20

*E-Mail:* [timo.kleine-doepke@fh-bielefeld.de](mailto:timo.kleine-doepke@fh-bielefeld.de)

### **Veröffentlichungsreihe:**

#### **AMMO – Berichte aus Forschung und Technologietransfer**

- Heft 1: Dezember 2013, AMMO-Team, *Informationen über den Forschungs- und Entwicklungsschwerpunkt Angewandte Mathematische Modellierung und Optimierung.*
- Heft 2: April 2014, R. Walden und V.-M. Roemer, *Methoden der quantitativen rechnergestützten CTG-Analyse.*
- Heft 3: Juli 2014, R. Ueckerdt, H.-W. Schmidt, M. Weber, E. Mindlina, *Entwicklung einer Dispatcherfunktion zur Überprüfung von Nominierungsmengen in der Betriebsführung von Erdgasspeichern.*
- Heft 4: September 2014, S. Proß, *Diskrete Modellierung und Optimierung praxisrelevanter Prozesse mit Petri-Netzen.*
- Heft 5: Oktober 2014, H.-J. Kruse, *Optimumgraphen*

ISSN 2198-4824

Herausgeber: Sprecher FSP AMMO  
Fachhochschule Bielefeld