Petri-Netze

Petri-Netze sind ein formaler Kalkül zur Modellierung von

- Abläufen mit nebenläufigen Prozessen, die auf gemeinsame Ressourcen zugreifen
- kausalen Beziehungen

Mit Petri-Netzen werden beispielsweise

- reale oder abstrakte Automaten und Maschinen
- kommunizierende Prozesse (in Rechnern)
- Verhalten von Software- und Hardware-Komponenten
- Geschäftsabläufe
- Spiele nach gewissen Regeln

beschrieben und modelliert.

Petri-Netze können durch (gerichtete, markierte) bipartite Graphen dargestellt werden.

Petri-Netze - formale Definition

Definition 1

Ein Petri-Netz ist ein Tripel $P = (S, T, F)$, wobei

1. S eine endliche Menge von Stellen ist
2. T eine endliche Menge von Transitionen ist
3. $F \subseteq (S \times T) \cup (T \times S)$ eine Relation ist.

- Stellen $s \in S$ repräsentieren Bedingungen oder Zustände eines Systems.
- Transitionen $t \in T$ repräsentieren Zustandsübergänge oder Aktivitäten eines Systems.

Graphische Darstellung von Petri-Netzen

Petri-Netze und gerichtete bipartite Graphen

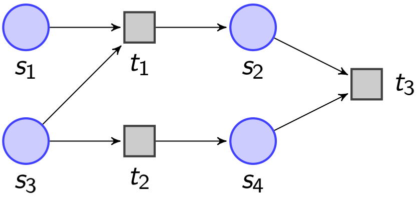
Ein Petri-Netz $P = (S, T, F)$ kann folgendermaßen als gerichteter bipartiter Graph dargestellt werden.

1. Stellen $s \in S$ sind Knoten des Graphen, dargestellt durch Kreise,
2. Transitionen $t \in T$ sind Knoten des Graphen, dargestellt durch Rechtecke,
3. Kanten $f \in F, f = (s, t)$ oder $f = (t, s)$ mit $s \in S, t \in T$, entsprechen gerichteten Kanten des Graphen.

Graphische Darstellung - Beispiele

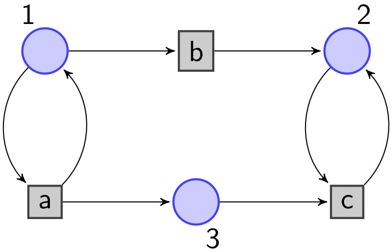
Petri-Netz P_1 definiert durch

$$\begin{aligned} S_1 &= \{s_1, s_2, s_3, s_4\} \\ T_1 &= \{t_1, t_2, t_3\} \\ F_1 &= \{(s_1, t_1), (s_3, t_1), (s_2, t_3), \\ &\quad (s_3, t_2), (s_4, t_3), (t_1, s_2), \\ &\quad (t_2, s_4)\} \end{aligned}$$



Petri-Netz P_2 definiert durch

$$\begin{aligned} S_2 &= \{1, 2, 3\} \\ T_2 &= \{a, b, c\} \\ F_2 &= \{(1, a), (1, b), (2, c), (3, c), \\ &\quad (a, 1), (a, 3), (b, 2), (c, 2)\} \end{aligned}$$



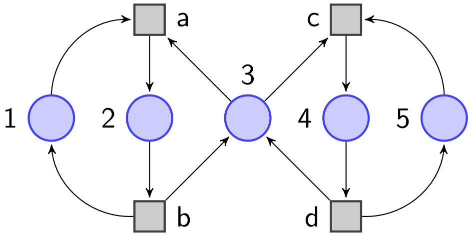
Graphische Darstellung - Beispiele

Petri-Netz P_3 definiert durch

$$S_3 = \{1, 2, 3, 4, 5\}$$

$$T_3 = \{a, b, c, d\}$$

$$F_3 = \{(1, a), (2, b), (3, a), (3, c), (4, d), (5, c) (a, 2), (b, 3), (b, 1), (c, 4), (d, 3), (d, 5)\}$$



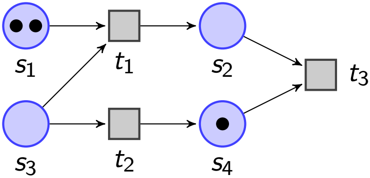
Markierte Petri-Netze

Definition 2

Sei $P = (S, T, F)$ ein Petri-Netz. Eine Funktion $M : S \to \mathbb{N}_0$ heißt Markierung des Petri-Netzes P. Das Petri-Netz P zusammen mit der Funktion M heißt markiertes Petri-Netz oder einfach Petri-Netz.

Petri-Netz P_1 definiert durch

$$\begin{aligned} S_1 &= \{s_1, s_2, s_3, s_4\} \\ T_1 &= \{t_1, t_2, t_3\} \\ F_1 &= \{(s_1, t_1), (s_3, t_1), (s_2, t_3), (s_3, t_2), \\ &\qquad (s_4, t_3), (t_1, s_2), (t_2, s_4)\} \end{aligned}$$



Markierung $M_1 : S_1 \to \mathbb{N}_0$ definiert durch

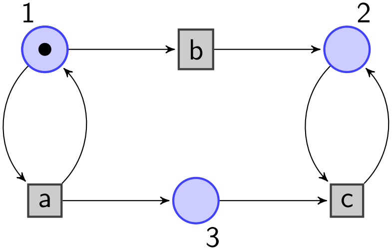
$$\begin{aligned} M_1(s_1) = 2, M_1(s_4) = 1 \\ M_1(s_2) = 0, M_1(s_3) = 0 \end{aligned}$$

- Markierungen dargestellt durch Punkte in Stellen.
- Anzahl Punkte entspricht Funktionswert.

Markierte Petri-Netze - Beispiele

Petri-Netz P_2 definiert durch

$$S_2 = \{1, 2, 3\}$$
$$T_2 = \{a, b, c\}$$
$$F_2 = \{(1, a), (1, b), (2, c), (3, c)(a, 1), (a, 3), (b, 2), (c, 2)\}$$



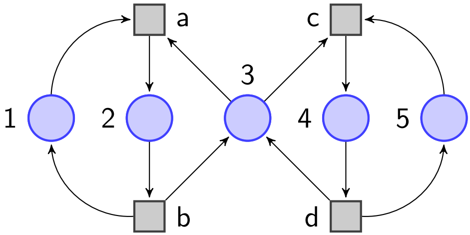
Markierung $M_2 : S_2 \to \mathbb{N}_0$ definiert durch

$$M_2(1) = 1, M_2(2) = 0$$
$$M_2(3) = 0$$

Markierte Petri-Netze - Beispiele

Petri-Netz P_3 definiert durch

$$
\begin{aligned}
S_3 &= \{1, 2, 3, 4, 5\} \\
T_3 &= \{a, b, c, d\} \\
F_3 &= \{(1, a), (2, b), (3, a), \\
&\quad (3, c), (4, d), (5, c) \\
&\quad (a, 2), (b, 3), (b, 1), \\
&\quad (c, 4), (d, 3), (d, 5)\}
\end{aligned}
$$



Markierung $M_3 : S_3 \to \mathbb{N}_0$ definiert durch

$$
\begin{aligned}
&M_3(1) = 1, M_3(3) = 1 \\
&M_3(2) = M_3(4) = M_3(5) = 0
\end{aligned}
$$

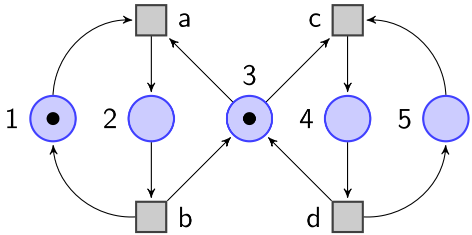
- Markierungen können durch Folgen der Länge $|S|$ dargestellt werden.
- i-tes Element der Folge ist Wert von M an i-ter Stelle.
- Markierung M_3 von P_3 wird durch die Folge $(1, 0, 1, 0, 0)$ dargestellt.

Vor- und Nachbereiche in Petri-Netzen

Definition 3

Sei $P = (S, T, F)$ ein Petri-Netz. Für $t \in T$ definieren wir

$$\text{Vorbereich}(t) := \{s \in S \mid (s, t) \in F\}$$
$$\text{Nachbereich}(t) := \{s \in S \mid (t, s) \in F\}.$$



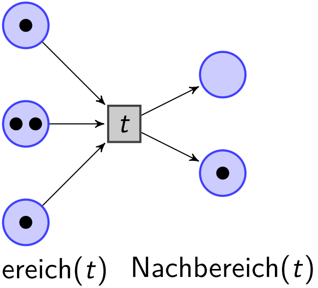
$$\text{Vorbereich}(a) = \{1, 3\}$$
$$\text{Nachbereich}(a) = \{2\}$$
$$\text{Vorbereich}(b) = \{2\}$$
$$\text{Nachbereich}(b) = \{1, 3\}$$
$$\text{Vorbereich}(c) = \{3, 5\}$$
$$\text{Nachbereich}(c) = \{4\}$$

Vor- und Nachbereiche in Petri-Netzen

Definition 3

Sei $P = (S, T, F)$ ein Petri-Netz. Für $t \in T$ definieren wir

$$\text{Vorbereich}(t) := \{s \in S \mid (s, t) \in F\}$$
$$\text{Nachbereich}(t) := \{s \in S \mid (t, s) \in F\}.$$



Schaltungen in Petri-Netzen

- **Schaltungen** eines Petri-Netzes überführen eine Markierung M des Petri-Netzes in eine Markierung M'.
- Schaltungen werden durch Transitionen des Petri-Netzes ausgeführt.
- Zu jedem Zeitpunkt wird nur eine Schaltung ausgeführt.
- Hintereinanderausführung mehrerer Schaltungen beschreibt die Dynamik oder Entwicklung des modellierten Systems.
- Dieses gilt auch für die Modellierung paralleler und nebenläufiger Systeme

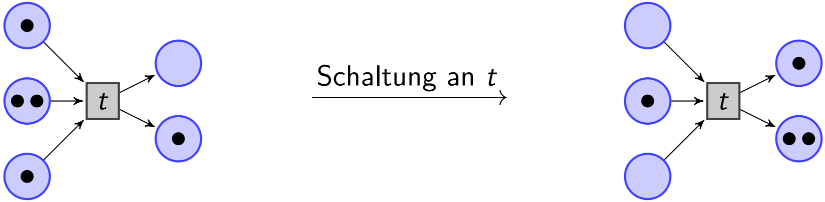
Schaltungen in Petri-Netzen

Definition 4

Sei $P = (S, T, F)$ ein Petri-Netz, $M : S \to \mathbb{N}_0$ eine Markierung von P und $t \in T$. Die Transition t kann schalten, wenn für alle $s \in \text{Vorbereich}(t)$ gilt $M(s) \geq 1$. Die Schaltung von t führt zur Markierung M' definiert durch

$$\begin{aligned} M'(v) &= M(v) - 1, && \text{für alle } v \in \text{Vorbereich}(t) \setminus \text{Nachbereich}(t) \\ M'(v) &= M(v) + 1, && \text{für alle } v \in \text{Nachbereich}(t) \setminus \text{Vorbereich}(t) \\ M'(v) &= M(v), && \text{für alle anderen Stellen } v \in S. \end{aligned}$$

M' heißt Nachfolgemarkierung von M bei Transition t.



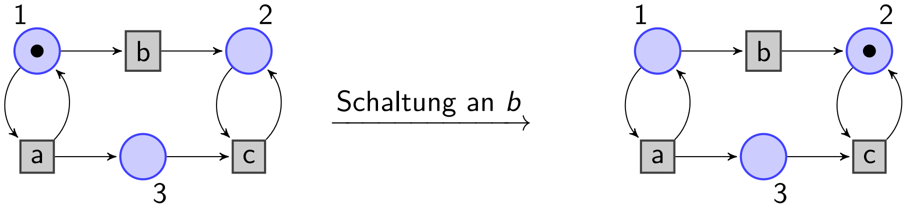
Schaltungen in Petri-Netzen

Definition 4

Sei $P = (S, T, F)$ ein Petri-Netz, $M : S \to \mathbb{N}_0$ eine Markierung von P und $t \in T$. Die Transition t kann schalten, wenn für alle $s \in \text{Vorbereich}(t)$ gilt $M(s) \geq 1$. Die Schaltung von t führt zur Markierung M' definiert durch

$$
\begin{aligned}
M'(v) &= M(v) - 1, && \text{für alle } v \in \text{Vorbereich}(t) \setminus \text{Nachbereich}(t)\\
M'(v) &= M(v) + 1, && \text{für alle } v \in \text{Nachbereich}(t) \setminus \text{Vorbereich}(t)\\
M'(v) &= M(v), && \text{für alle anderen Stellen } v \in S.
\end{aligned}
$$

M' heißt Nachfolgemarkierung von M bei Transition t.



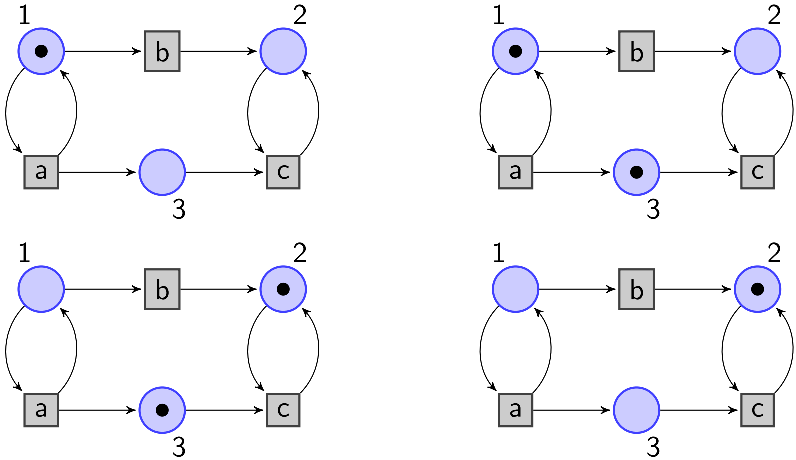
Folgen von Schaltungen

- Schaltungen können sukzessive ausgeführt werden $\rightarrow$ Folgen von Schaltungen
- trotzdem können Petri-Netze zur Modellierung paralleler und nebenläufiger Systeme eingesetzt werden

Folgen von Schaltungen oder **Schaltfolgen** können dargestellt werden als

- Folge von Markierungen
- Folgen der geschalteten Transitionen

Folgen von Schaltungen



Folge von Markierungen:

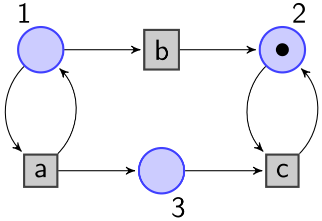
$$M_0 = (1,0,0) \qquad M_1 = (1,0,1)$$
$$M_2 = (0,1,1) \qquad M_3 = (0,1,0)$$

Folge von Transitionen

$$a, b, c$$

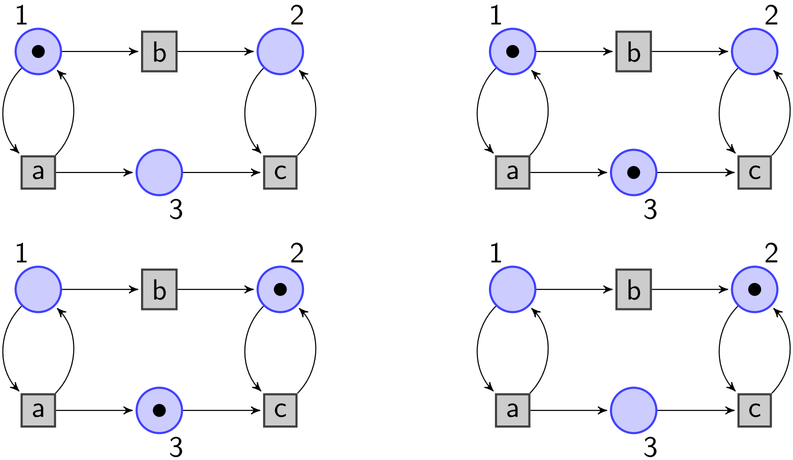
Blockierte Petri-Netze und erreichbare Markierungen

- Sei $P = (S, T, F)$ ein Petri-Netz und $M : S \to \mathbb{N}_0$ eine Markierung.
- P mit Markierung M heißt blockiert, wenn keine Transition $t \in T$ bei Markierung M schalten kann.
- $M' : S \to \mathbb{N}_0$ Markierung von P.
- Existieren Markierungen $M_0 = M, M_1, \ldots, M_n = M'$, wobei M_{i+1} Nachfolgemarkierung von M_i ist, so sagen wir, dass M' von M erreichbar ist.



Petri-Netz ist blockiert

Erreichbare Markierungen

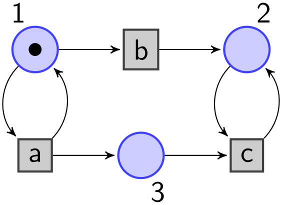


Folge von Markierungen:

$$M_0 = (1,0,0) \qquad M_1 = (1,0,1)$$
$$M_2 = (0,1,1) \qquad M_3 = (0,1,0)$$

M_1, M_2, M_3 sind von M_0 erreichbar.

Konflikte zwischen Transitionen



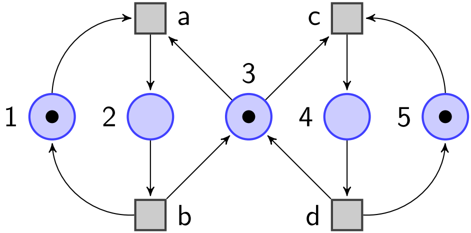
- Transitionen a und b können nicht gleichzeitig schalten,
- obwohl $M(v) \geq 1$ für alle $v \in \text{Vorbereich}(a) \cup \text{Vorbereich}(b)$.

Definition 5

Sei $P = (S, T, F)$ ein Petri-Netz. Transitionen $t, t' \in T$ stehen in Konflikt, wenn

$$\text{Vorbereich}(t) \cap \text{Vorbereich}(t') \neq \emptyset.$$

Konflikte zwischen Transitionen



- Transitionen a und c stehen in Konflikt.
- Transitionen b und d stehen nicht in Konflikt.

Konflikte und Schaltfolgen

Sei $P = (S, T, F)$ ein Petri-Netz mit Markierung $M : S \to \mathbb{N}_0$ und seien $t, t' \in T$. Stehen t, t' nicht in Konflikt und können sowohl t als auch t' schalten, dann kann t' in der Nachfolgemarkierung M' von M bei Transition t schalten.

Petri-Netze und Sprachen

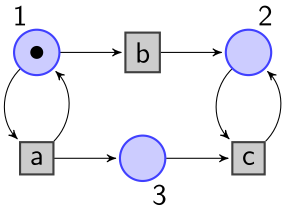
Definition 6

Sei $P = (S, T, N)$ ein Petri-Netz mit (Anfangs-) Markierung M_0. Wir nennen

$$L(P) := \{w \in T^* \mid w \text{ ist eine mögliche Folge von Schaltungen von } P \text{ mit Markierung } M_0\}$$

die durch P definierte Sprache.

Petri-Netz P_2



$$L(P_2) = \{a^n b c^m \mid m \leq n, \; n, m \in \mathbb{N}_0\}$$

Petri-Netze und Sprachen

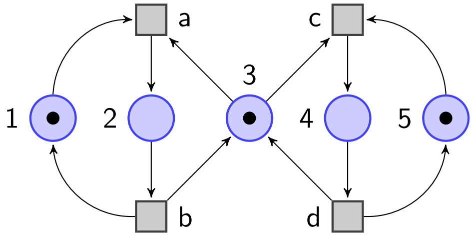
Definition 6

Sei $P = (S, T, N)$ ein Petri-Netz mit (Anfangs-) Markierung M_0. Wir nennen

$$L(P) := \{w \in T^* \mid w \text{ ist eine mögliche Folge von Schaltungen von } P \text{ mit Markierung } M_0\}$$

die durch P definierte Sprache.

Petri-Netz P_3



$$L(P_3) = L((ab \mid cd)^*)$$

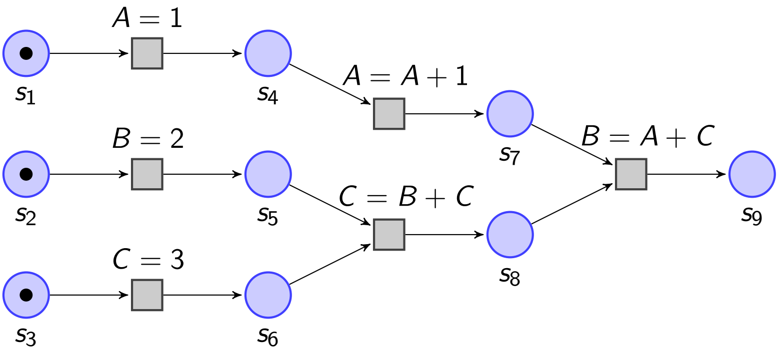
Modellierung mit Petri-Netzen - Programmabläufe

- Transitionen entsprechen Programminstruktionen
- Stellen entsprechen Bedingungen für die Ausführung einer Instruktion

$$
\begin{aligned}
A &= 1 \\
B &= 2 \\
C &= 3 \\
A &= A + 1 \\
C &= B + C \\
B &= A + C
\end{aligned}
$$

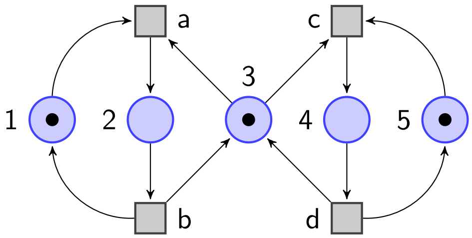
Modellierung mit Petri-Netzen - Programmabläufe

- Transitionen entsprechen Programminstruktionen
- Stellen entsprechen Bedingungen für die Ausführung einer Instruktion



Modellierung mit Petri-Netzen - zyklische Prozesse

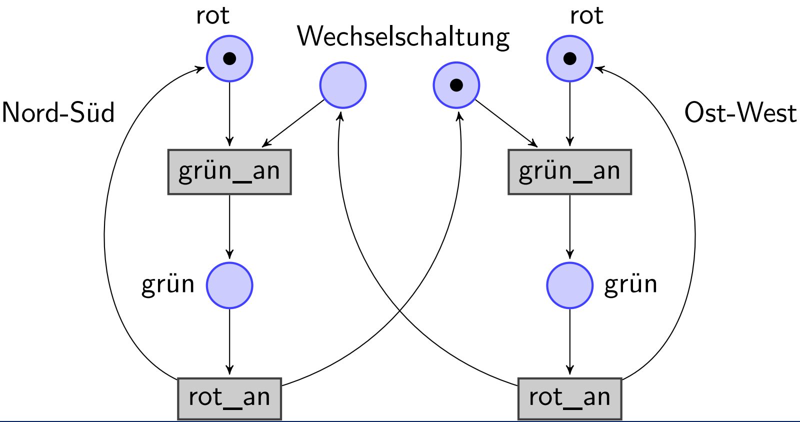
- zwei Prozesse P_1, P_2 sollen immer wieder und in beliebiger Reihenfolge ausgeführt werden
- die Prozesse greifen jedoch auf eine gemeinsame Ressource zu
- daher kann zu jedem Zeitpunkt immer nur einer der Prozesse ausgeführt werden



- Transitionen a, b modellieren Prozess P_1
- Transitionen c, d modellieren Prozess P_2

Modellierung mit Petri-Netzen - Ampelschaltung

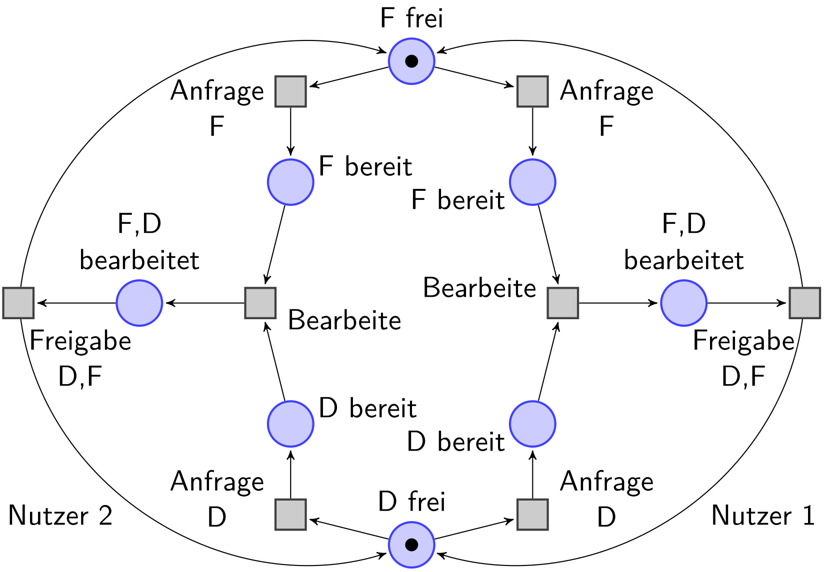
- Ampelschaltung an der Kreuzung zweier Straßen (Ost-West, Nord-Süd)
- zu jedem Zeitpunkt hat eine Straße „grün“, die andere „rot “
- die Grün-Phasen und Rot-Phasen der beiden Straßen wechseln sich ab
- ignorieren zur Vereinfachung die Gelb-Phase



Modellierung mit Petri-Netzen - Geteilte Ressourcen

- zwei Nutzer teilen sich eine Festplatte (F) und einen Drucker (D)
- zu jedem Zeitpunkt kann nur einer der Nutzer beide Ressourcen nutzen, um eine Datei zu drucken

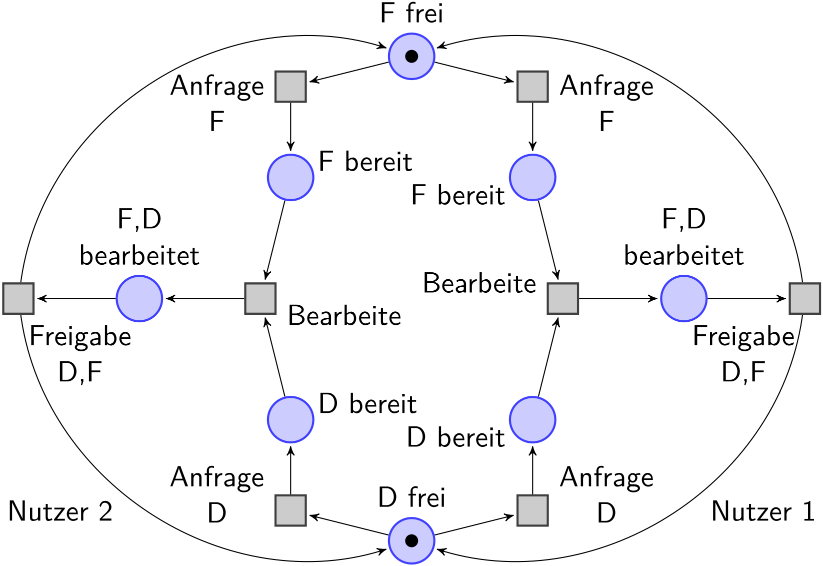
Modellierung mit Petri-Netzen - Geteilte Ressourcen



Lebendige Petri-Netze

- Sei $P = (S, T, F)$ ein Petri-Netz und $M : S \to \mathbb{N}_0$ eine Markierung.
- P mit Markierung M heißt blockiert, wenn keine Transition $t \in T$ bei Markierung M schalten kann.

Modellierung mit Petri-Netzen - Geteilte Ressourcen



Nutzer 2 fragt F, Nutzer 1 D an

Lebendige Petri-Netze

- Sei $P = (S, T, F)$ ein Petri-Netz und $M : S \to \mathbb{N}_0$ eine Markierung.
- P mit Markierung M heißt blockiert, wenn keine Transition $t \in T$ bei Markierung M schalten kann.
- $M' : S \to \mathbb{N}_0$ Markierung von P.
- Existieren Markierungen $M_0 = M, M_1, \ldots, M_n = M'$, wobei M_{i+1} Nachfolgemarkierung von M_i ist, so sagen wir, dass M' von M erreichbar ist.

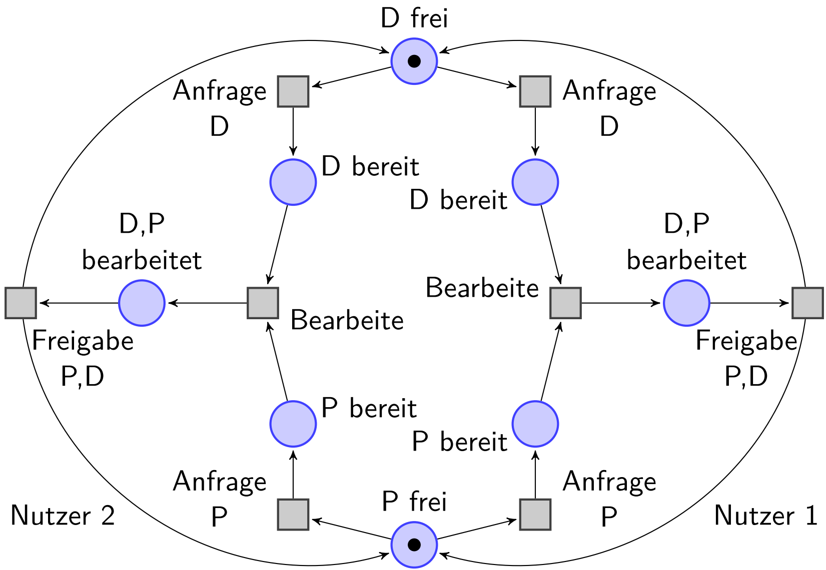
Definition 7

Sei $P = (S, T, F)$ ein Petri-Netz mit Anfangsmarkierung M_0. Eine Transition $t \in T$ heißt lebendig, wenn es zu jeder Markierung M', die von M_0 erreichbar ist, eine Markierung M'' gibt, die von M' erreichbar ist und in der t schalten kann.

Definition 8

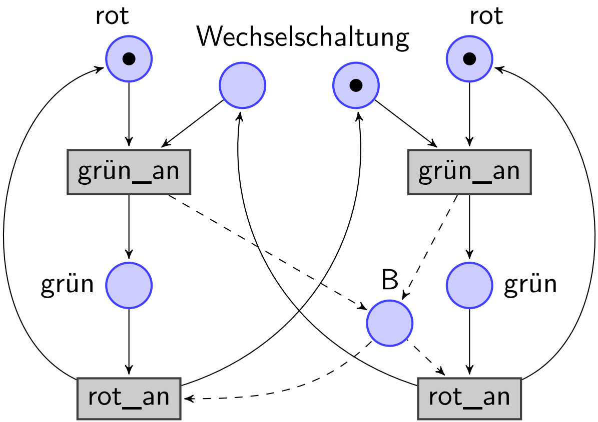
Ein Petri-Netz $P = (S, T, F)$ mit Anfangsmarkierung M_0 heißt lebendig, wenn alle Transitionen $t \in T$ lebendig sind.

Modellierung mit Petri-Netzen - Geteilte Ressourcen



Petri-Netz ist nicht lebendig.

Erweiterte Ampelschaltung

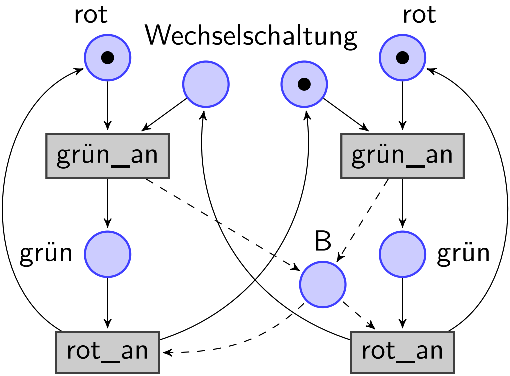


- B zählt Anzahl der Ampeln mit „grün".
- Es sollte nie $M(B) \geq 2$ gelten.

Sichere Petri-Netze

Definition 9

Sei $P = (S, T, F)$ ein Petri-Netz mit Anfangsmarkierung M_0. Dann heißt P sicher oder binär, wenn für alle von M_0 erreichbaren Markierungen M und alle Stellen $s \in S$ gilt $M(s) \leq 1$.



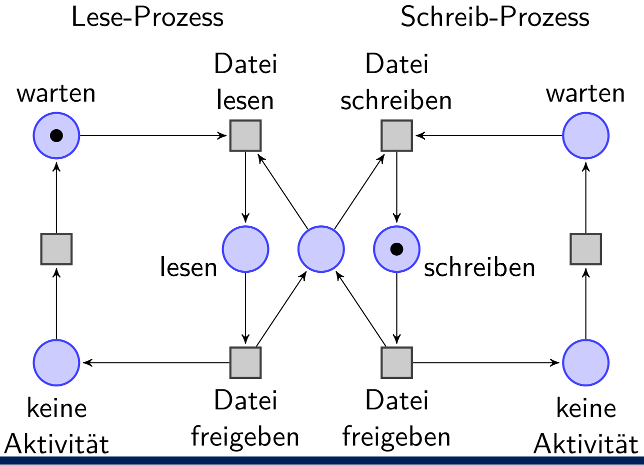
- sicheres Petri-Netz

⇒ es gilt immer

$$M(B) \leq 1$$

Modellierung mit Petri-Netzen - Lese-Schreib-Prozesse

- Ein Lese-Prozess und ein Schreib-Prozess greifen auf dieselben Dateien zu
- zu jedem Zeitpunkt soll nur einer der Prozesse auf eine Datei zugreifen können



Erweiterungen von Petri-Netzen - Kantengewichte

- bislang nur die Grundvariante von Petri-Netzen kennengelernt
- es gibt viele Erweiterungen, die die Ausdrucksstärke von Petri-Netzen erhöhen
- betrachten zwei Varante
  1. Petri-Netze mit Kantengewichten
  2. Petri-Netze mit Kapazitäten

Definition 10

Ein Petri-Netz mit Kantengewichten ist ein Petri-Netz $P = (S, T, F)$ mit einer Funktion $g : F \to \mathbb{N}$.

Schaltregeln:

- eine Transition t kann nur schalten, wenn für alle $s \in \text{Vorbereich}(t)$ gilt $M(s) \geq g((s, t))$
- Schalten einer Transition t entfernt aus jeder Stelle $s \in \text{Vorbereich}(t)$ genau $g((s, t))$ Marken und fügt jeder Stelle $s \in \text{Nachbereich}(t)$ genau $g((t, s))$ Marken hinzu

Erweiterungen von Petri-Netzen - Kapazitäten

- bislang nur die Grundvariante von Petri-Netzen kennengelernt
- es gibt viele Erweiterungen, die die Ausdrucksstärke von Petri-Netzen erhöhen
- betrachten zwei Varante
  1. Petri-Netze mit Kantengewichten
  2. Petri-Netze mit Kapazitäten

Definition 11

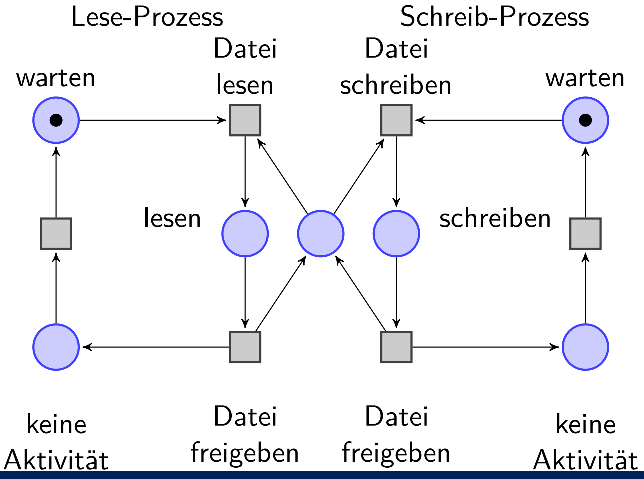
Ein Petri-Netz mit Kapazitäten ist ein Petri-Netz $P = (S, T, F)$ mit einer Funktion $k : S \to \mathbb{N}_0$.

Schaltregeln:

- eine Transition t kann nur schalten, wenn durch die Schaltung die Anzahl der Marken auf einer Stelle $s \in$ Nachbereich(t) nicht größer als $k(s)$ wird
- können Petri-M Netze mit Kantengewichte und Kapazitäten kombinieren

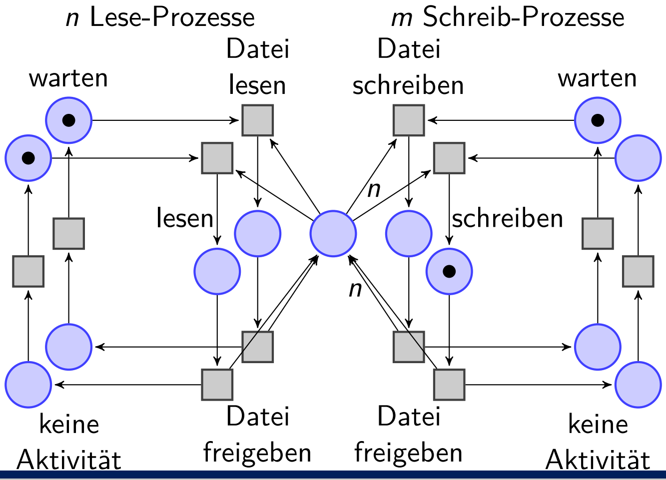
Modellierung mit Kantengewichten - Lese-Schreib-Prozesse

- n Lese-Prozesse und m Schreibprozesse auf einer Datei
- Lese-Prozesse können gleichzeitig zugreifen, Schreib-Prozesse können nicht mit anderen Prozessen gleichzeitig auf Dateien zugreifen



Modellierung mit Kantengewichten - Lese-Schreib-Prozesse

- n Lese-Prozesse und m Schreibprozesse auf einer Datei
- Lese-Prozesse können gleichzeitig zugreifen, Schreib-Prozesse können nicht mit anderen Prozessen gleichzeitig auf Dateien zugreifen



Modellierung mit Kantengewichten und Kapazitäten

- zyklischer Produzenten-Prozess legt Produkte in Puffer ab, jeweils zwei gleichzeitig (Kantengewicht)
- Pufferkapazität ist auf 6 begrenzt
- Konsument entnimmt Produkte einzeln