Petri-Netze

Nico Bühler

Albert-Ludwigs-Universität Freiburg

14. Januar 2016

Inhaltsverzeichnis

Petri Netze

- 1962 von Carl Adam Petri entwickelt.

Petri Netze

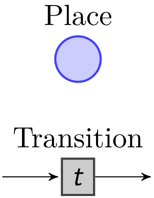
- 1962 von Carl Adam Petri entwickelt.
- Modulation und Simulation.

Petri Netze

- 1962 von Carl Adam Petri entwickelt.
- Modulation und Simulation.

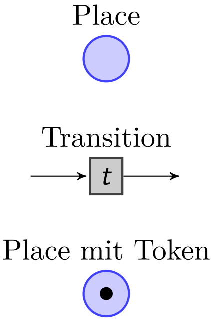
Petri Netze

- 1962 von Carl Adam Petri entwickelt.
- Modulation und Simulation.

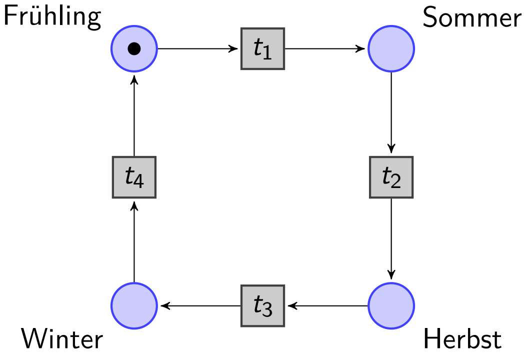


Petri Netze

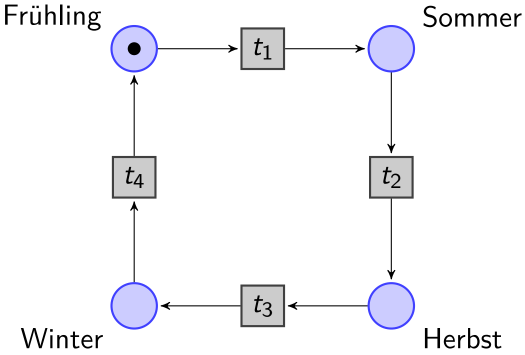
- 1962 von Carl Adam Petri entwickelt.
- Modulation und Simulation.



Vier Jahreszeiten (wie es sein sollte...)

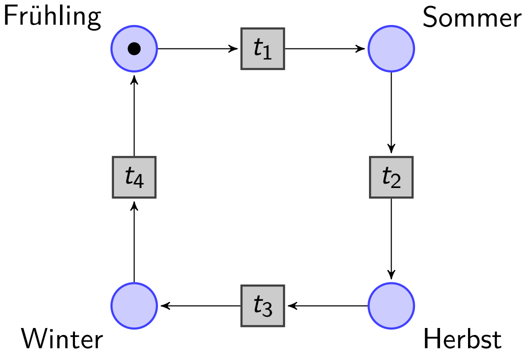


Vier Jahreszeiten (wie es sein sollte...)



t_3 feuert

Vier Jahreszeiten (wie es sein sollte...)

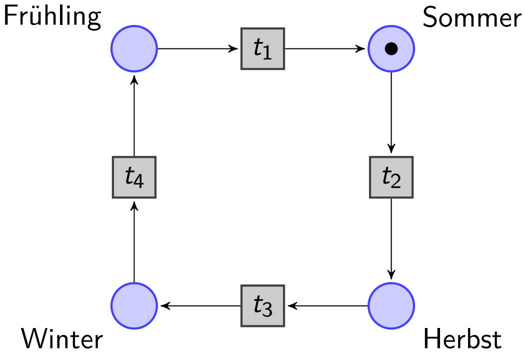


t_3 feuert

$\downarrow$

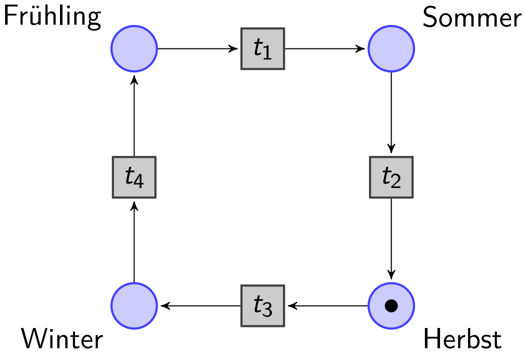
nichts passiert

Vier Jahreszeiten (wie es sein sollte...)



t_1 feuert

Vier Jahreszeiten (wie es sein sollte...)

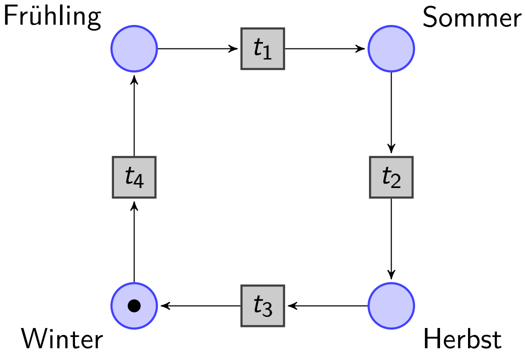


t_1 feuert

$\downarrow$

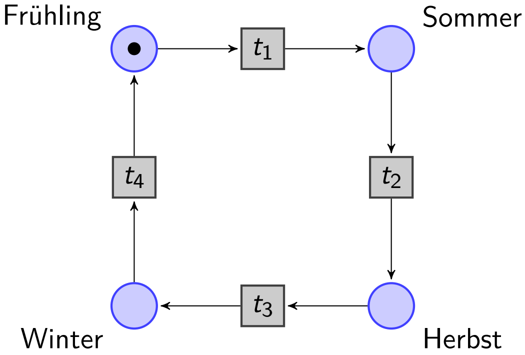
t_2 feuert

Vier Jahreszeiten (wie es sein sollte...)



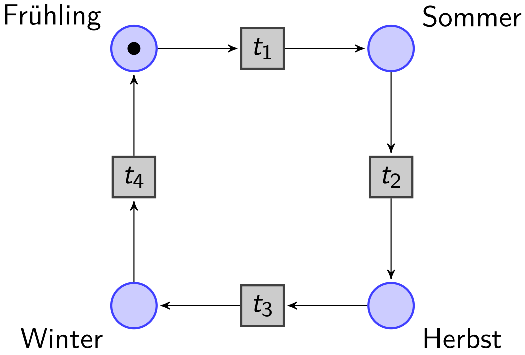
t_1 feuert
↓
t_2 feuert
↓
t_3 feuert

Vier Jahreszeiten (wie es sein sollte...)



t_1 feuert
↓
t_2 feuert
↓
t_3 feuert
↓
t_4 feuert

Vier Jahreszeiten (wie es sein sollte...)



$$
\begin{array}{c}
(1,0,0,0) \\
\downarrow \\
(0,1,0,0) \\
\downarrow \\
(0,0,1,0) \\
\downarrow \\
(0,0,0,1)
\end{array}
$$

Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

- P: Places, nichtleere Menge.

Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

- P: Places, nichtleere Menge.
- T: Transitions, nichtleere Menge.

Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

- P: Places, nichtleere Menge.
- T: Transitions, nichtleere Menge.
- F: Flow Relations, $F \subseteq (P \times T) \cup (T \times P)$.

Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

- P: Places, nichtleere Menge.
- T: Transitions, nichtleere Menge.
- F: Flow Relations, $F \subseteq (P \times T) \cup (T \times P)$.

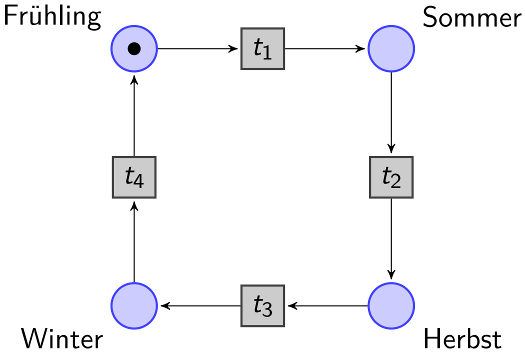
Definition Petri-Netz

Definition

$N = (P, T, F)$ ist ein Petri-Netz, mit:

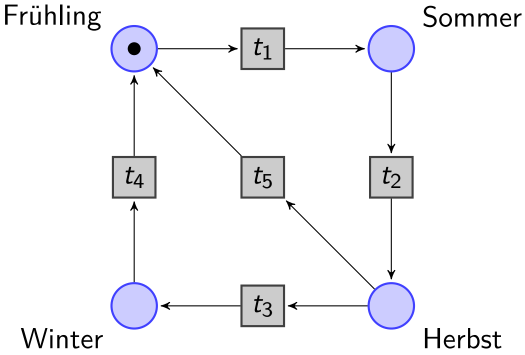
- P: Places, nichtleere Menge.
- T: Transitions, nichtleere Menge.
- F: Flow Relations, $F \subseteq (P \times T) \cup (T \times P)$.

Vier Jahreszeiten (wie es sein sollte...)

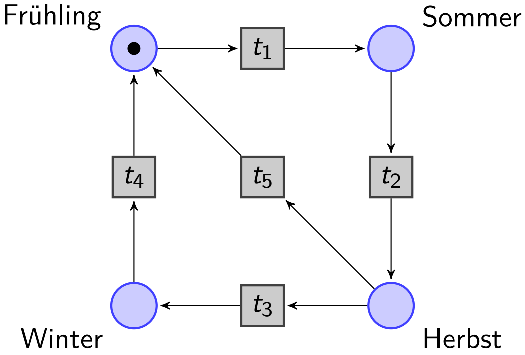


Wo ist der Fehler?

Vier Jahreszeiten (wie es tatsächlich ist...)



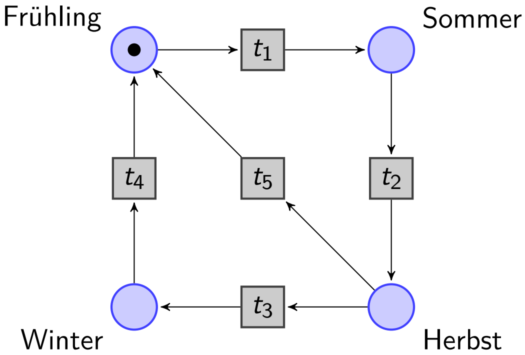
Vier Jahreszeiten (wie es tatsächlich ist...)

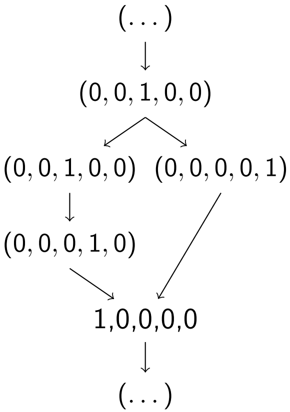


Petri-Netze

... sind nicht-deterministisch.

Vier Jahreszeiten (wie es tatsächlich ist...)





Zugbeispiel Erklärung

Zugbeispiel Erklärung

- 2 Züge, 2 Bahnhöfen, 1 Bahngleis.
- Gleis darf nur von einem Zug befahren werden.

Grundidee:

- Mehrere Tokens (hier Züge) in einem Petri-Netz.
- Jeder Place darf höchstens einen Token beinhalten.

Zugbeispiel Erklärung

- 2 Züge, 2 Bahnhöfen, 1 Bahngleis.
- Gleis darf nur von einem Zug befahren werden.

Grundidee:

- Mehrere Tokens (hier Züge) in einem Petri-Netz.
- Jeder Place darf höchstens einen Token beinhalten.

$\Longrightarrow$ Wechselseitiger Ausschluss.

Zugbeispiel Erklärung

- 2 Züge, 2 Bahnhöfen, 1 Bahngleis.
- Gleis darf nur von einem Zug befahren werden.

Grundidee:

- Mehrere Tokens (hier Züge) in einem Petri-Netz.
- Jeder Place darf höchstens einen Token beinhalten.

$\Longrightarrow$ Wechselseitiger Ausschluss.

Petri-Netze

... sind nicht-deterministisch.

Zugbeispiel Erklärung

- 2 Züge, 2 Bahnhöfen, 1 Bahngleis.
- Gleis darf nur von einem Zug befahren werden.

Grundidee:

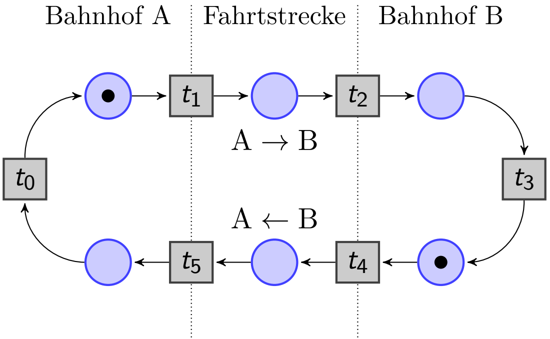
- Mehrere Tokens (hier Züge) in einem Petri-Netz.
- Jeder Place darf höchstens einen Token beinhalten.

$\Longrightarrow$ Wechselseitiger Ausschluss.

Petri-Netze

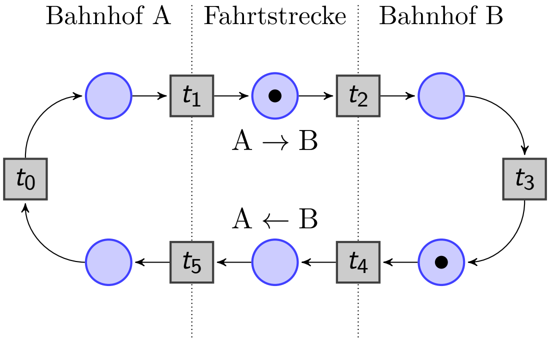
... sind nicht-deterministisch.
... sind nebenläufig.

Zugbeispiel Versuch 1

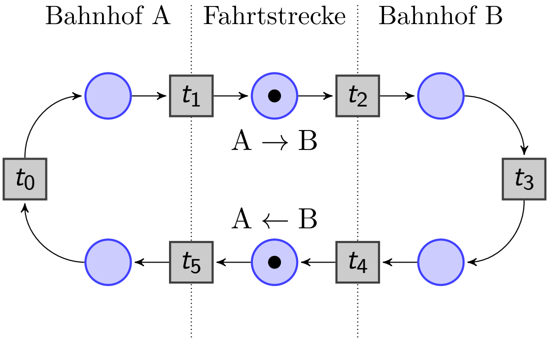


Ist die Anfangsbedingung gegeben?

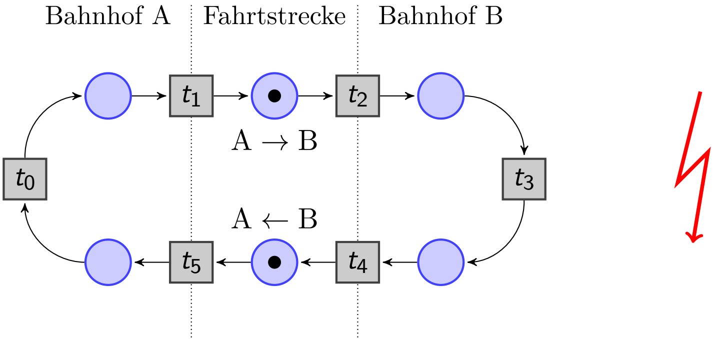
Zugbeispiel Versuch 1



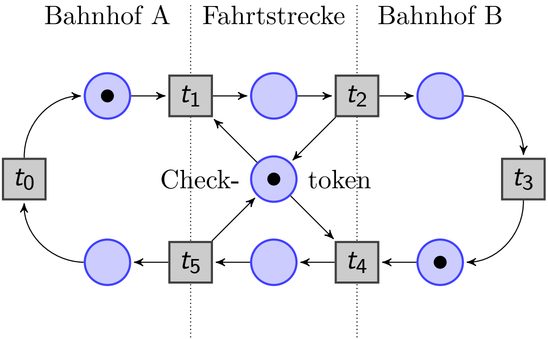
Zugbeispiel Versuch 1



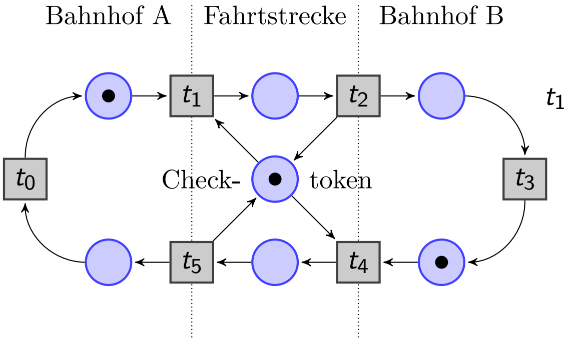
Zugbeispiel Versuch 1



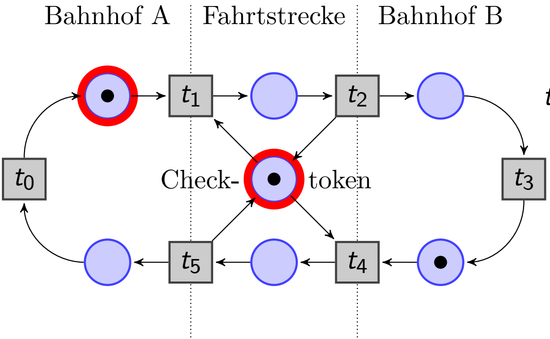
Zugbeispiel, Versuch 2



Zugbeispiel, Versuch 2



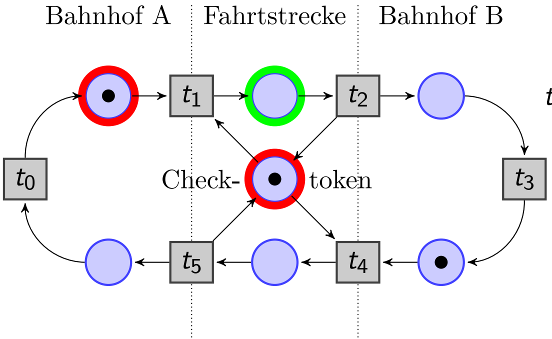
Zugbeispiel, Versuch 2



t_1 hat:

- 2 Vorbedingungen

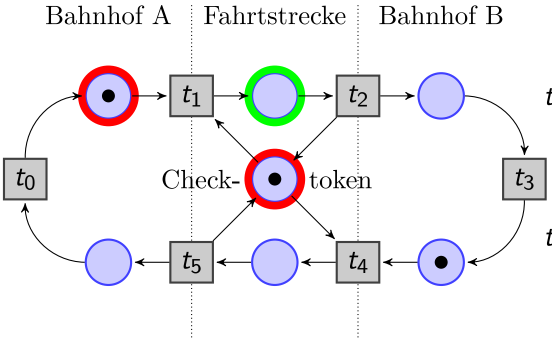
Zugbeispiel, Versuch 2



t_1 hat:

- 2 Vorbedingungen
- 1 Nachbedingung

Zugbeispiel, Versuch 2

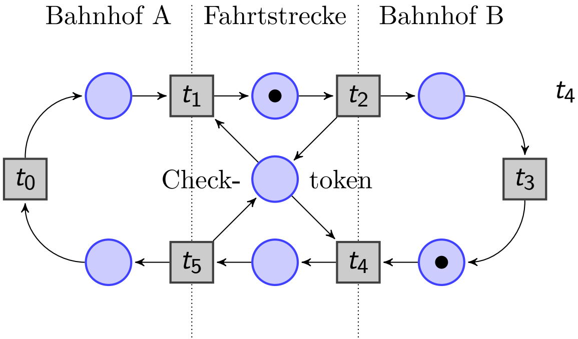


t_1 hat:

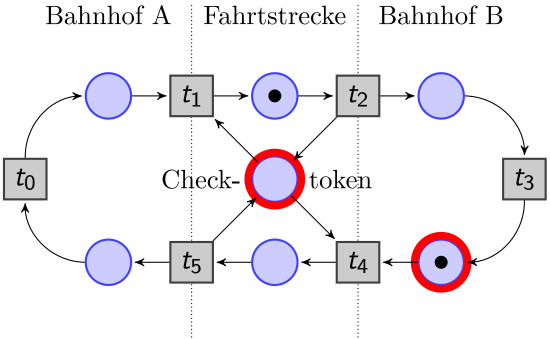
- 2 Vorbedingungen
- 1 Nachbedingung

t_1 kann feuern.

Zugbeispiel, Versuch 2



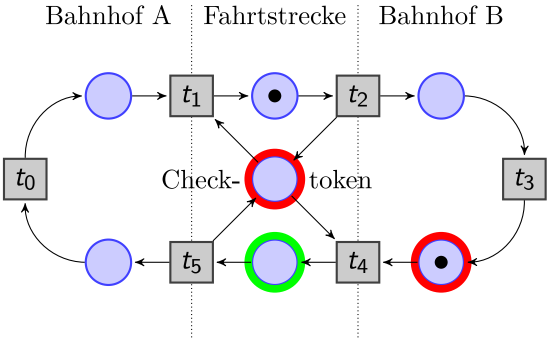
Zugbeispiel, Versuch 2



t_4 hat:

- 2 Vorbedingungen

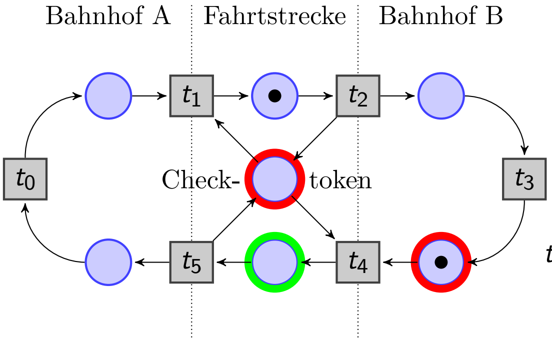
Zugbeispiel, Versuch 2



t_4 hat:

- 2 Vorbedingungen
- 1 Nachbedingung

Zugbeispiel, Versuch 2

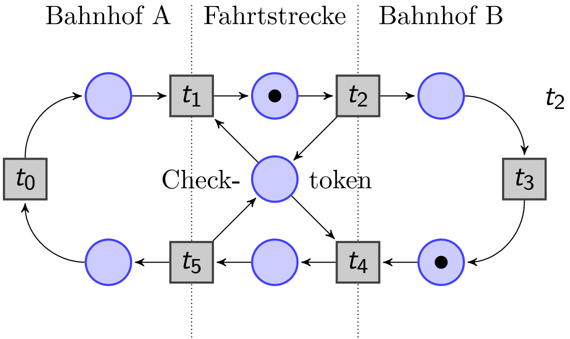


t_4 hat:

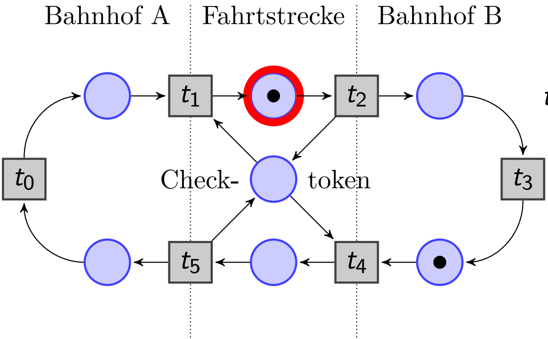
- 2 Vorbedingungen
- 1 Nachbedingung

t_4 kann nicht feuern.

Zugbeispiel, Versuch 2



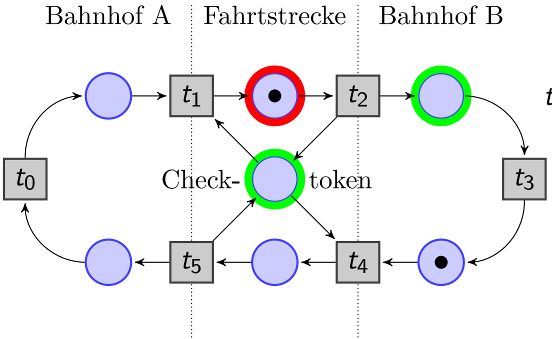
Zugbeispiel, Versuch 2



t_2 hat:

- 1 Vorbedingungen

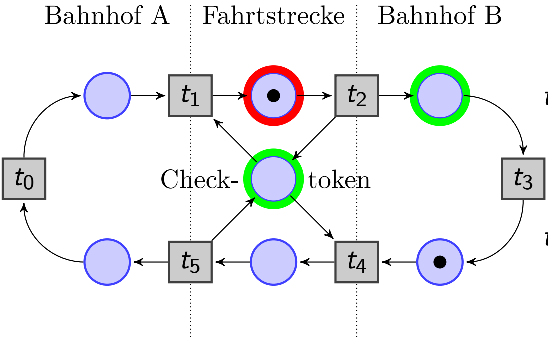
Zugbeispiel, Versuch 2



t_2 hat:

- 1 Vorbedingungen
- 2 Nachbedingung

Zugbeispiel, Versuch 2



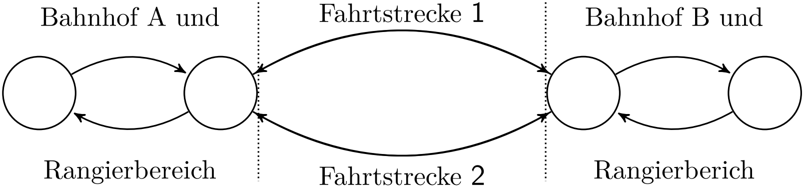
t_2 hat:

- 1 Vorbedingungen
- 2 Nachbedingung

t_2 kann feuern.

Zugbeispiel, Fortsetzung

Mehrere Gleise, mehrere Züge?



Kapazität und Gewichtung

Kapazität C

Gewichtung W

Kapazität und Gewichtung

Kapazität C

Gibt die maximale Anzahl an Tokens pro Place an.

Gewichtung W

Kapazität und Gewichtung

Kapazität C

Gibt die maximale Anzahl an Tokens pro Place an.

Gewichtung W

Gewichtung der Kante

- vor einer Transition gibt die Anzahl der benötigten Tokens zum Feuern dieser an (Vorbedingung).

Kapazität und Gewichtung

Kapazität C

Gibt die maximale Anzahl an Tokens pro Place an.

Gewichtung W

Gewichtung der Kante

- vor einer Transition gibt die Anzahl der benötigten Tokens zum Feuern dieser an (Vorbedingung).
- nach einer Transition gibt die Anzahl der von ihr ausgehenden Tokens an (Nachbedingung).

Kapazität und Gewichtung

Kapazität C

Gibt die maximale Anzahl an Tokens pro Place an.

Gewichtung W

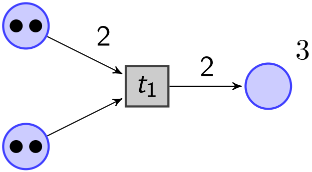
Gewichtung der Kante

- vor einer Transition gibt die Anzahl der benötigten Tokens zum Feuern dieser an (Vorbedingung).
- nach einer Transition gibt die Anzahl der von ihr ausgehenden Tokens an (Nachbedingung).

Wenn nicht angegeben: $C = 1$ und $W = 1$.

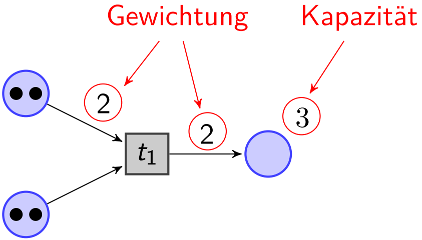
Kapazität und Gewichtung, Beispiel

C unbegrenzt und $W = 1$ wenn nicht anders angegeben.



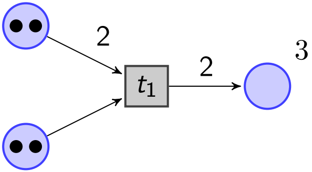
Kapazität und Gewichtung, Beispiel

C unbegrenzt und $W = 1$ wenn nicht anders angegeben.



Kapazität und Gewichtung, Beispiel

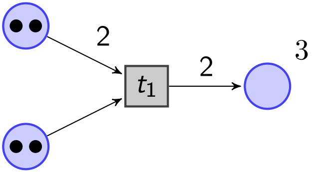
C unbegrenzt und $W = 1$ wenn nicht anders angegeben.



Sind Vor- und Nachbedingung erfüllbar?

Kapazität und Gewichtung, Beispiel

C unbegrenzt und $W = 1$ wenn nicht anders angegeben.

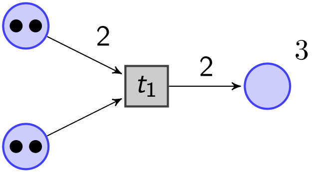


Sind Vor- und Nachbedingung erfüllbar?

- Vorbedingung ✓

Kapazität und Gewichtung, Beispiel

C unbegrenzt und $W = 1$ wenn nicht anders angegeben.

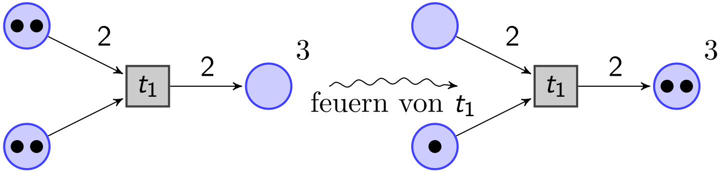


Sind Vor- und Nachbedingung erfüllbar?

- Vorbedingung ✓
- Nachbedingung ✓

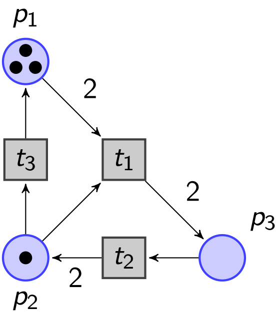
Kapazität und Gewichtung, Beispiel

C unbegrenzt und $W = 1$ wenn nicht anders angegeben.



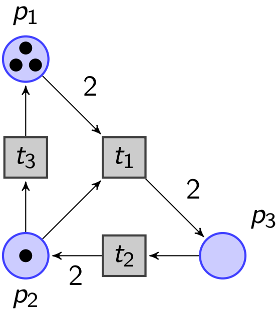
Berechnung der Folgezustände

Gegeben ist folgendes PN mit C unbegrenzt:



Berechnung der Folgezustände

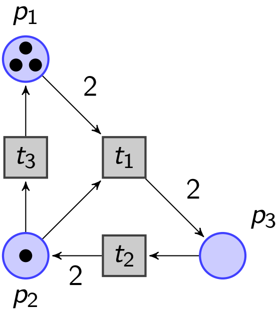
Gegeben ist folgendes PN mit C unbegrenzt:



Wenn t_1 feuert, dann...

Berechnung der Folgezustände

Gegeben ist folgendes PN mit C unbegrenzt:

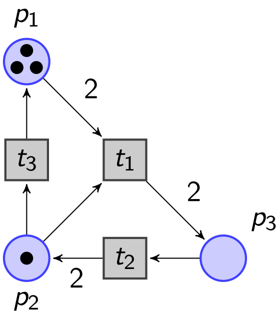


Wenn t_1 feuert, dann...

- werden p_1 zwei Token entzogen,

Berechnung der Folgezustände

Gegeben ist folgendes PN mit C unbegrenzt:

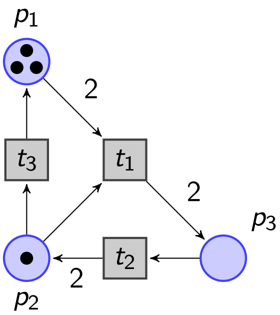


Wenn t_1 feuert, dann...

- werden p_1 zwei Token entzogen,
- wird p_2 ein Token entzogen,

Berechnung der Folgezustände

Gegeben ist folgendes PN mit C unbegrenzt:

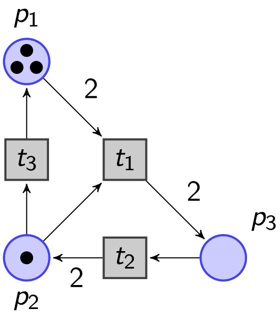


Wenn t_1 feuert, dann...

- werden p_1 zwei Token entzogen,
- wird p_2 ein Token entzogen,
- bekommt p_3 zwei Token.

Berechnung der Folgezustände

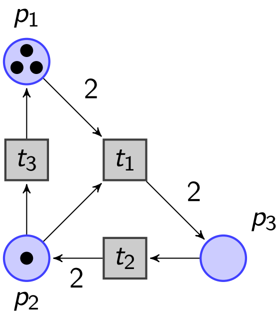
Gegeben ist folgende Matrix mit C unbegrenzt:



$$\underline{t}_1 = \begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix}$$

Berechnung der Folgezustände

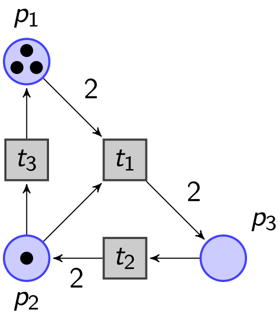
Gegeben ist folgende Matrix mit C unbegrenzt:



$$\underline{t}_1 = \begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix} \qquad \underline{t}_2 = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}$$

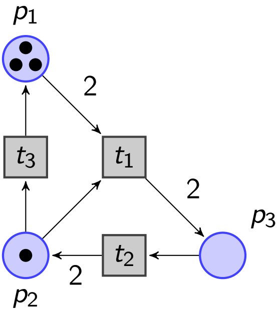
Berechnung der Folgezustände

Gegeben ist folgende Matrix mit C unbegrenzt:



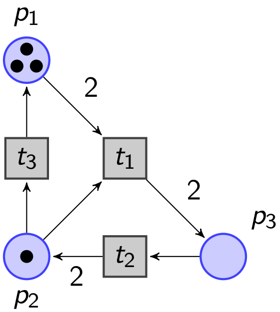
$$\underline{t}_1 = \begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix} \qquad \underline{t}_2 = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix} \qquad \underline{t}_3 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$$

Berechnung der Folgezustände



- Gegeben Startzustand M und Folgezustand M'.

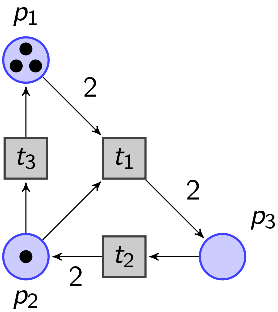
Berechnung der Folgezustände



- Gegeben Startzustand M und Folgezustand M'.
  Beispiel:

$$M = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} \qquad t_1 = \begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix}$$

Berechnung der Folgezustände



- Gegeben Startzustand M und Folgezustand M'.
  Beispiel:

$$M = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} \qquad t_1 = \begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix}$$

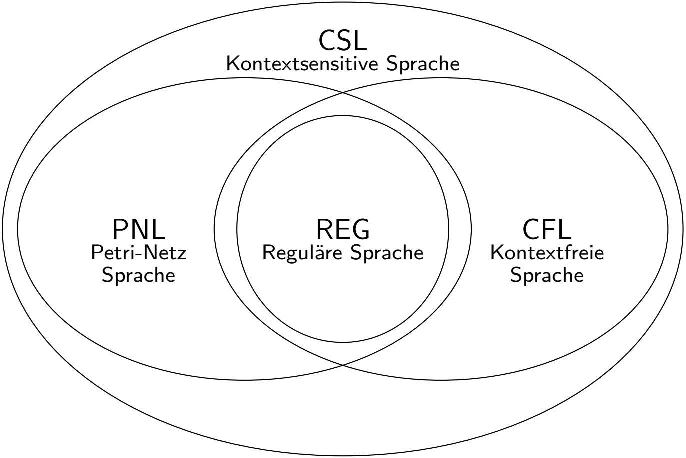
$$M' = M + t_1 = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

Petri-Netze als Sprachakzeptoren

- $L(N)$ ist die Sprache, die vom Petri-Netz N akzeptiert wird.

Petri-Netze als Sprachakzeptoren

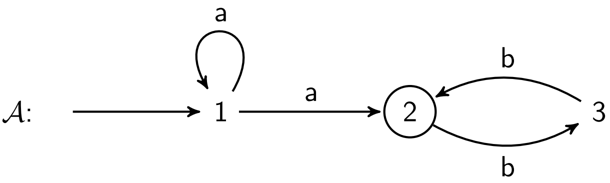
- $L(N)$ ist die Sprache, die vom Petri-Netz N akzeptiert wird.



Petri-Netze als Sprachakzeptoren

Beispiel:

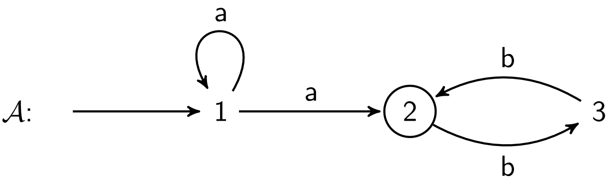
Entwickeln eines Petri-Netzes, aus NFA $\mathcal{A}$:

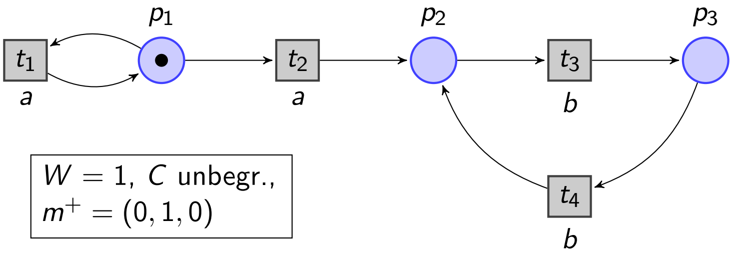


Petri-Netze als Sprachakzeptoren

Beispiel:

Entwickeln eines Petri-Netzes, aus NFA $\mathcal{A}$:





$W = 1$, C unbegr.,
$m^+ = (0, 1, 0)$

Petri-Netze als Sprachakzeptoren

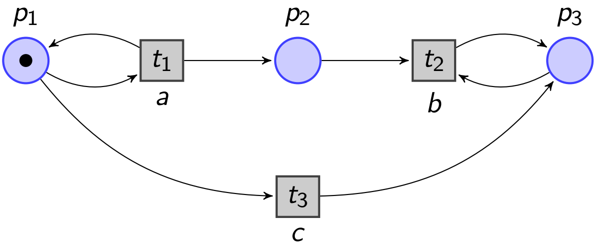
Beispiel:
PNL: $L = \{a^n c b^n \mid n \geq 0\}$,

Petri-Netze als Sprachakzeptoren

Beispiel:
PNL: $L = \{a^n c b^n \mid n \geq 0\}$,
Startmarkierung: $m^- = (1, 0, 0)$,
Endmarkierung: $m^+ = (0, 0, 1)$

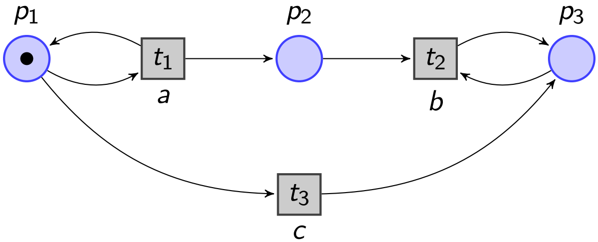
Petri-Netze als Sprachakzeptoren

Beispiel:
PNL: $L = \{a^n c b^n \mid n \geq 0\}$,
Startmarkierung: $m^- = (1, 0, 0)$,
Endmarkierung: $m^+ = (0, 0, 1)$

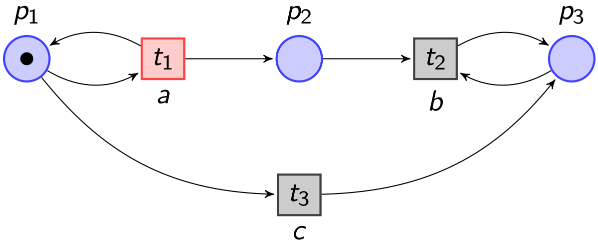


$\boxed{W = 1,\ C \text{ unbegr.}}$

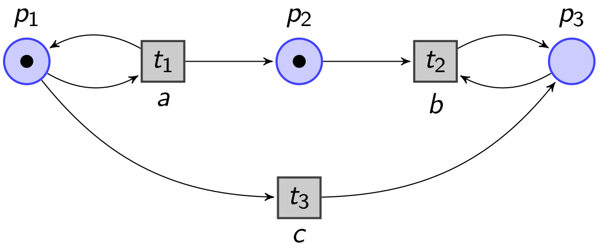
Petri-Netze als Sprachakzeptoren



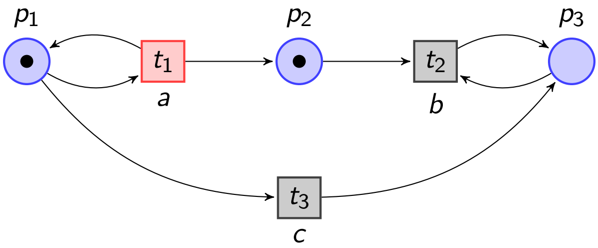
Petri-Netze als Sprachakzeptoren



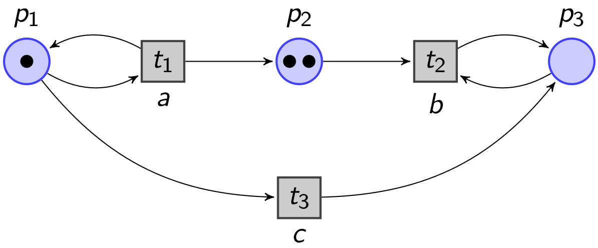
Petri-Netze als Sprachakzeptoren



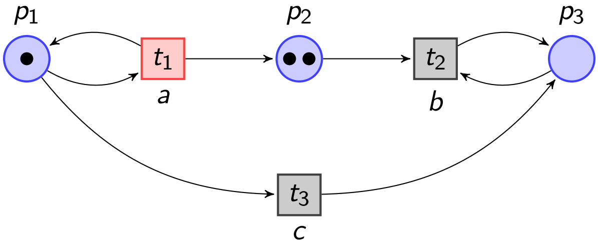
Petri-Netze als Sprachakzeptoren



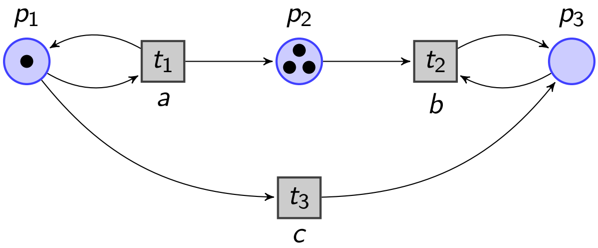
Petri-Netze als Sprachakzeptoren



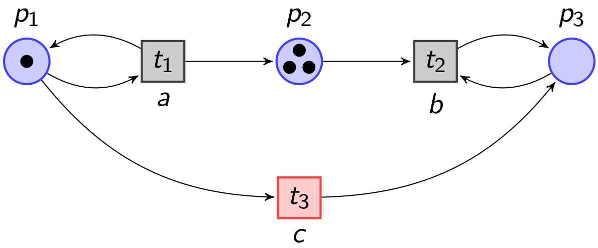
Petri-Netze als Sprachakzeptoren



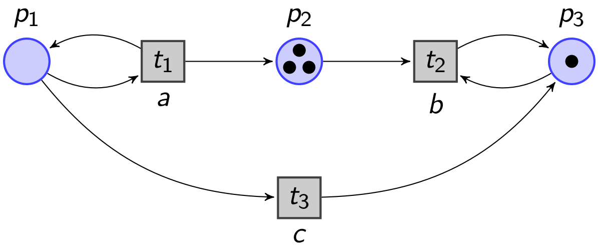
Petri-Netze als Sprachakzeptoren



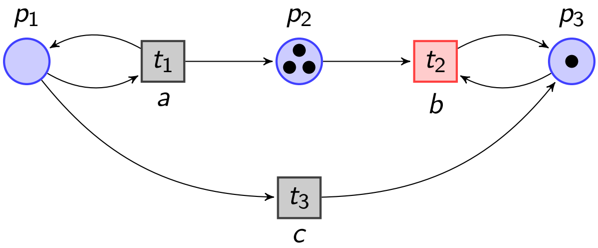
Petri-Netze als Sprachakzeptoren



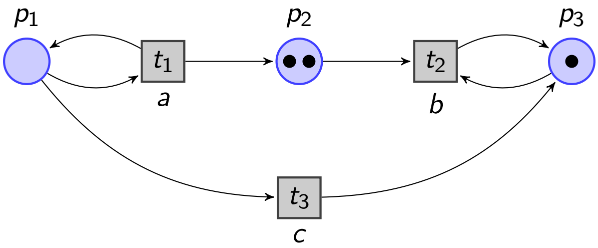
Petri-Netze als Sprachakzeptoren



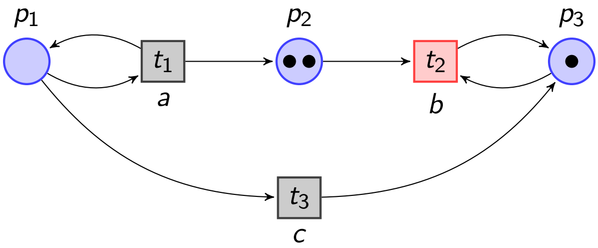
Petri-Netze als Sprachakzeptoren



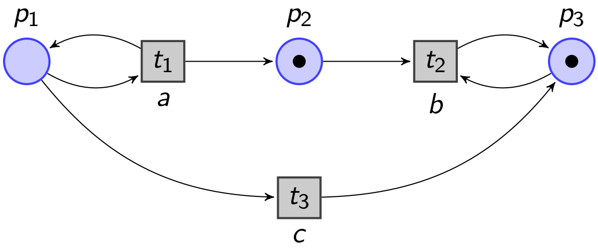
Petri-Netze als Sprachakzeptoren



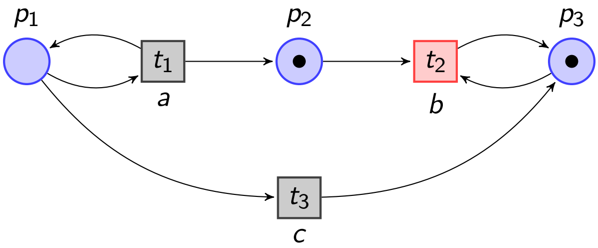
Petri-Netze als Sprachakzeptoren



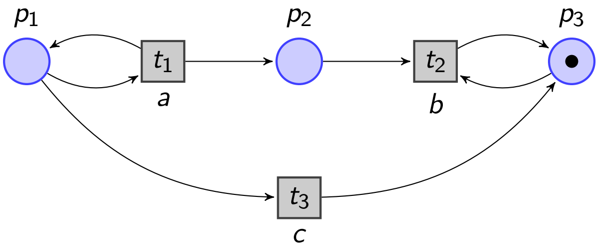
Petri-Netze als Sprachakzeptoren



Petri-Netze als Sprachakzeptoren



Petri-Netze als Sprachakzeptoren



Petri-Netze

Simulation und Modellbildung von

- nichtdeterministischen,
- dynamischen,
- nebenläufigen

Prozessen.

Quellen

- Lilius, Johan; Wojciech Penczek
  Applications and Theory of Petri Nets
  Springer Verlag, 2010
- Prof. Dr. Van Laerhoven, Kristof
  Skript zur Vorlesung „Einführung in Embedded Systems“
  Universität Freiburg, 2015
- Prof. Dr. Scherer, Raimar
  Skript zur Vorlesung „Bauinformatik Vertiefte Grundlagen“
  Technische Universität Dresden
- Prof. Dr. Thomas, Wolfgang
  Skript zur Vorlesung „Applied Automata Theory“
  RWTH Aachen, 2005