Petri-Netze

Teil 2

Chandran Goodchild

Department of Automata Theory
University of Freiburg

Pro Seminar, 2017

Struktur

1. Tim Maffenbeier hat die Semantik von Petri-Netzen eingeführt.
2. Fortsetzung
   - Warum haben Petri-Netze die bereits vorgestellte "Mächtigkeit" bzw. wie fügen sich Petri Netze in der Chomsky Hierarchie ein.
   - Definitionen.
   - Eigenschaften und Anwendungen von Petri-Netzen.
   - Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen.

Outline

1. Tim Maffenbeier hat die Semantik von Petri-Netzen eingeführt.

2. Fortsetzung
    - Warum haben Petri-Netze die bereits vorgestellte "Mächtigkeit" bzw. wie fügen sich Petri Netze in der Chomsky Hierarchie ein.
    - Definitionen.
    - Eigenschaften und Anwendungen von Petri-Netzen.
    - Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen.

Warum haben Petri-Netze die vorgestellte ''Mächtigkeit''?

Chomsky-Hierarchie

- Warum sind Petri-Netze mächtiger als Reguläre Sprachen (REG) aber nicht mächtiger als Kontext Freie Sprachen (CFL)?

Warum haben Petri-Netze die vorgestellte "Mächtigkeit"?

Chomsky-Hierarchie

- Warum sind Petri-Netze mächtiger als Reguläre Sprachen (REG) aber nicht mächtiger als Kontext Freie Sprachen (CFL)?
- Im Fall von Regulären Sprachen ist immer nur das letzte Zeichen nicht Terminal - die nächste Eingabe hängt also von dem aktuellen Zustand ab. Dies lässt sich wie folgt in einem Petri-Netz modellieren:

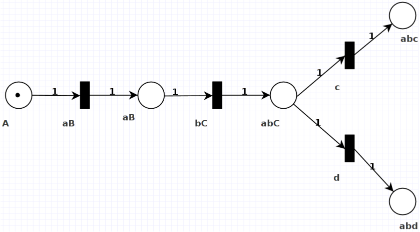


Abbildung: Wobei $A \rightarrow aB$, $B \rightarrow bC$, und $C \rightarrow d \mid c$.

- In Kontext Freie Sprachen benötigen wir für jedes Nichtterminalsymbol einen zentralen Zustand:

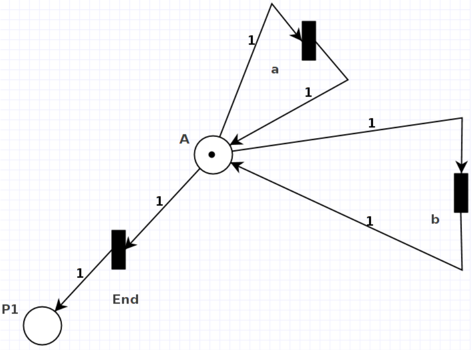


Abbildung: Wobei $A \rightarrow a \mid b$.

Warum haben Petri-Netze die vorgestellte "Mächtigkeit"?

Wir können also manche CFL Wörter darstellen, welche können wir nicht darstellen?

- Das Wort aAb können wir mit Petri-Netzen beispielsweise **nicht darstellen**, weil die Transitionen im Petri-Netz zufällig gefeuert werden und wir nicht garantieren können, dass nach der nicht terminalen Transition die terminale Transition b gefeuert wird. Umgekehrt betrachtet kann das Nichtterminalsymbol A nicht verändert werden, wenn das Terminalsymbol b bereits festgelegt ist.

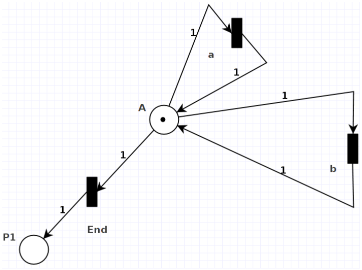


Abbildung: Wobei $A \to a \mid b$.

Outline

1. Tim Maffenbeier hat die Semantik von Petri-Netzen eingeführt.

2. Fortsetzung
   - Warum haben Petri-Netze die bereits vorgestellte "Mächtigkeit" bzw. wie fügen sich Petri Netze in der Chomsky Hierarchie ein.
   - Definitionen.
   - Eigenschaften und Anwendungen von Petri-Netzen.
   - Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen.

Definitionen

Definition

Ein **Erreichbarkeitsgraph** ist ein Graph $G = (V, E)$, mit einer Menge von Knoten V und Kanten E, wobei jeder Knoten einen erreichbaren Zustand in einem gegebenen System darstellt. Jeder Zustand wird durch Kanten mit allen direkt erreichbaren Folgezuständen verbunden.

Definitionen

Definition

Ein **Erreichbarkeitsgraph** ist ein Graph $G = (V, E)$, mit einer Menge von Knoten V und Kanten E, wobei jeder Knoten einen erreichbaren Zustand in einem gegebenen System darstellt. Jeder Zustand wird durch Kanten mit allen direkt erreichbaren Folgezuständen verbunden.

Definition

Ein Petri-Netz ist **begrenzt** genau dann wenn, der Erreichbarkeitsgraph endlich ist. [1] [2]

Definitionen

Definition

Ein Petri-Netz ist **einfach** genau dann wenn folgendes gilt:
$[x, y \in (C \cup E) \wedge ({}^{\bullet}x = {}^{\bullet}y) \wedge (x^{\bullet} = y^{\bullet})] \rightarrow x = y$

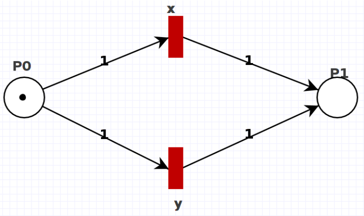


Abbildung: Ein Beispiel eines nicht einfachen Petri-Netzes. [4]

Outline

1. Tim Maffenbeier hat die Semantik von Petri-Netzen eingeführt.

2. Fortsetzung
   - Warum haben Petri-Netze die bereits vorgestellte ”Mächtigkeit” bzw. wie fügen sich Petri Netze in der Chomsky Hierarchie ein.
   - Definitionen.
   - Eigenschaften und Anwendungen von Petri-Netzen.
   - Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen.

Eigenschaften von Petri-Netzen

- Sie können verwendet werden um verteilte Systeme zu synchronisieren. [3]

Eigenschaften von Petri-Netzen

- Sie können verwendet werden um verteilte Systeme zu synchronisieren. [3]
- Mit Petri-Netzen können Prozesse ”ge-scheduled” werden. [4]

Eigenschaften von Petri-Netzen

- Sie können verwendet werden um verteilte Systeme zu synchronisieren. [3]
- Mit Petri-Netzen können Prozesse "ge-scheduled" werden. [4]
- Und sie können verwendet werden um deadlocks zu erkennen, bzw. den wechselseitigen Ausschluss nachzuweisen. [4]

Verteilte Netze synchronisieren

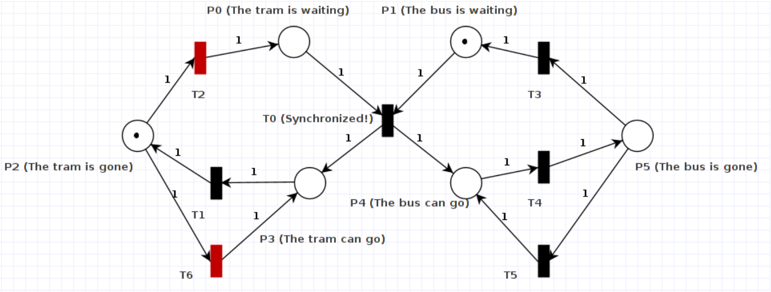


Abbildung: Ein einfaches Beispiel der Synchronisierung zweier Prozesse in einem verteilten System. Hier muss der Bus auf die Straßenbahn warten bevor er weiterfahren darf - analog müsste die Straßenbahn auch auf den Bus warten. Die rot markierten Transitionen sind zum feuern bereit, weil die Gewichte bzw. Bedingungen ihrer einkommenden Kanten erfüllt sind.

Scheduling

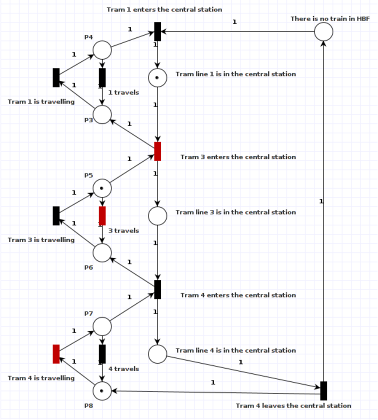


Abbildung: Ein einfaches Beispiel eines "milner schedules" in einem verteilten System. Hier erreichen drei Straßenbahnen den Hauptbahnhof immer in der gleichen Reihenfolge, egal wie lange sie für ihre Reise benötigen. Die rot markierten Transitionen gelten als bereit.

Deadlocks

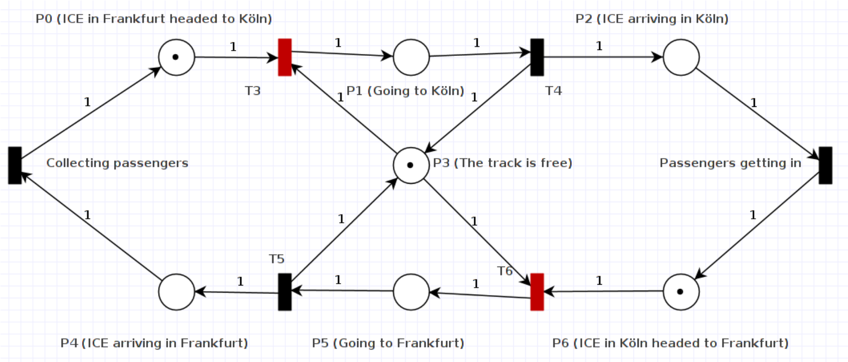


Abbildung: Ein einfaches Beispiel der Nebenlaufigkeit in einem Petri-Netz. Dieses Netz stellt zwei Züge dar die ein einziges Gleis zwischen Köln und Frankfurt teilen müssen. Die roten Transitionen gelten weiterhin als bereit. [4]

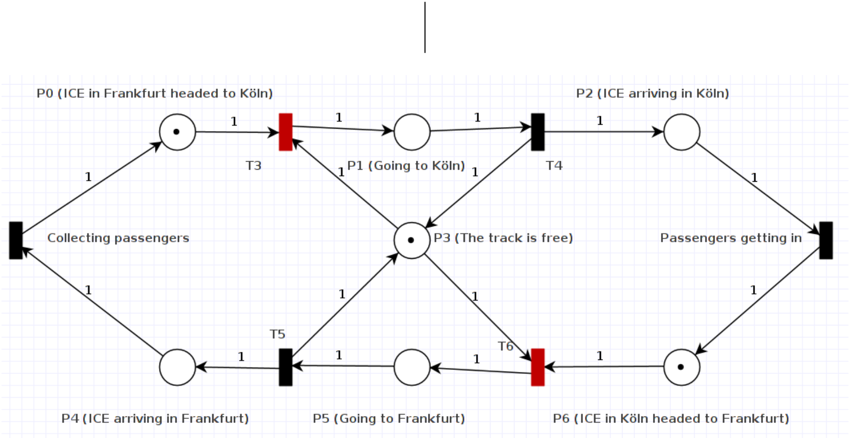
Outline

1. Tim Maffenbeier hat die Semantik von Petri-Netzen eingeführt.

2. Fortsetzung
   - Warum haben Petri-Netze die bereits vorgestellte "Mächtigkeit" bzw. wie fügen sich Petri Netze in der Chomsky Hierarchie ein.
   - Definitionen.
   - Eigenschaften und Anwendungen von Petri-Netzen.
   - Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen.

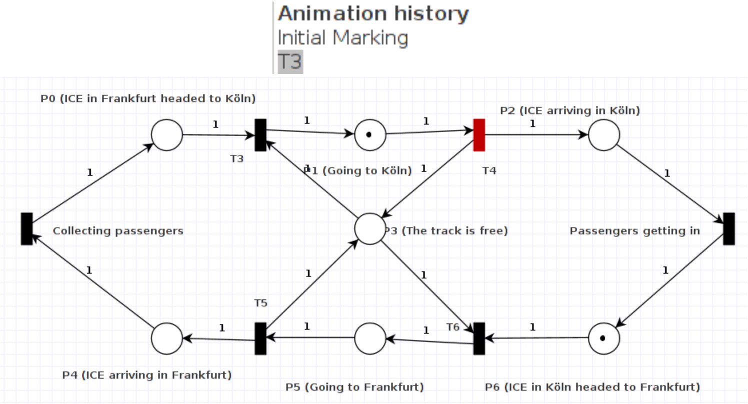
Vom Erreichbarkeitsgraph zur Definition von Petri-Netzen

$$M_0 = (1, 0, 0, 1, 0, 0, 1)$$



$$M_0 = (1, 0, 0, 1, 0, 0, 1)$$

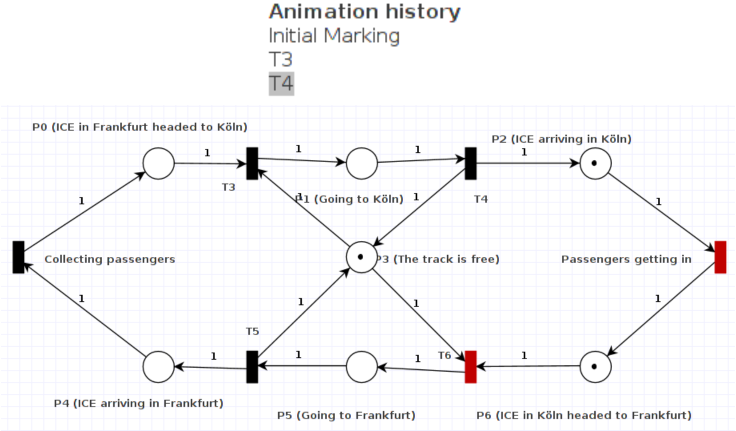
$$M_1 = (0, 1, 0, 0, 0, 0, 1)$$



Vom Erreichbarkeitsgraph zur Definition von Petri-Netze.

$$M_1 = (0, 1, 0, 0, 0, 0, 1)$$

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



Vom Erreichbarkeitsgraph zur Definition von Petri-Netze.

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

$$|$$

$$M_3 = (0, 0, 1, 0, 0, 1, 0)$$

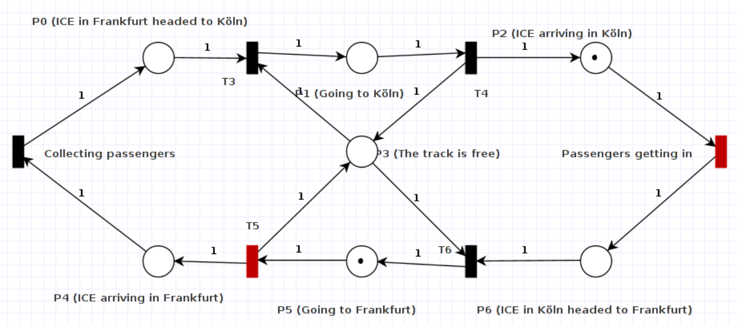
Animation history

Initial Marking

T3

T4

T6

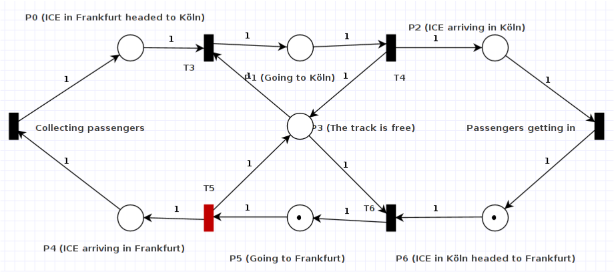


Vom Erreichbarkeitsgraph zur Definition von Petri-Netze.

$$M_3 = (0, 0, 1, 0, 0, 1, 0)$$

$$M_4 = (0, 0, 0, 0, 0, 1, 1)$$

Animation history
Initial Marking
T3
T4
T6
Passengers getting in



$$M_3 = (0,0,1,0,0,1,0)$$

$$M_4 = (0,0,0,0,0,1,1)$$

$$M_5 = (0,0,0,1,1,0,1)$$

Animation history

Initial Marking

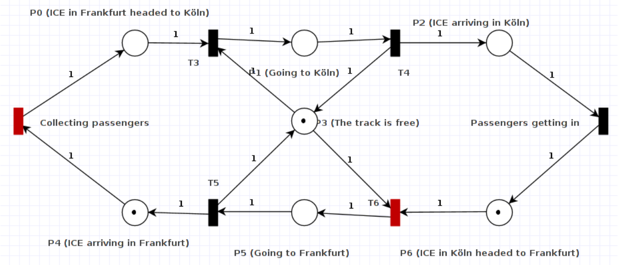
T3

T4

T6

Passengers getting in

T5



$$M_4 = (0,0,0,0,0,1,1)$$

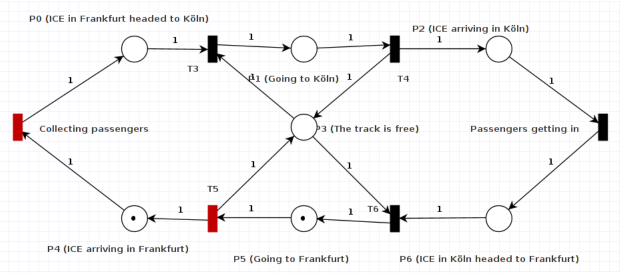
$$|$$

$$M_5 = (0,0,0,1,1,0,1)$$

$$|$$

$$M_6 = (0,0,0,0,1,1,0)$$

Animation history
Initial Marking
T3
T4
T6
Passengers getting in
T5
T6



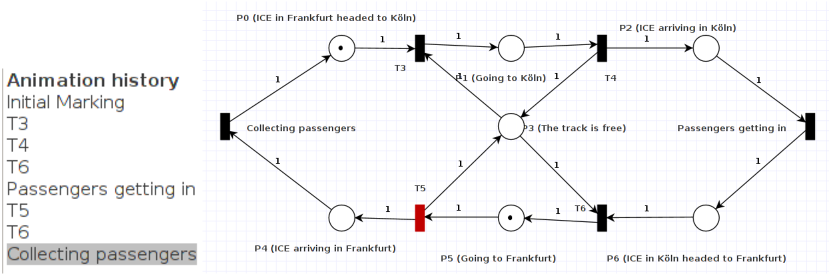
$$M_5 = (0,0,0,1,1,0,1)$$

$$|$$

$$M_6 = (0,0,0,0,1,1,0)$$

$$|$$

$$M_7 = (1,0,0,0,0,1,0)$$



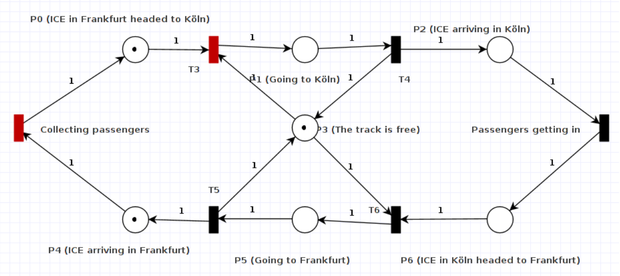
Vom Erreichbarkeitsgraph zur Definition von Petri-Netze.

$$M_6 = (0, 0, 0, 0, 1, 1, 0)$$

$$M_7 = (1, 0, 0, 0, 0, 1, 0)$$

$$M_8 = (1, 0, 0, 1, 1, 0, 0)$$

Animation history
Initial Marking
T3
T4
T6
Passengers getting in
T5
T6
Collecting passengers
T5



Vom Erreichbarkeitsgraph zur Definition von Petri-Netze.

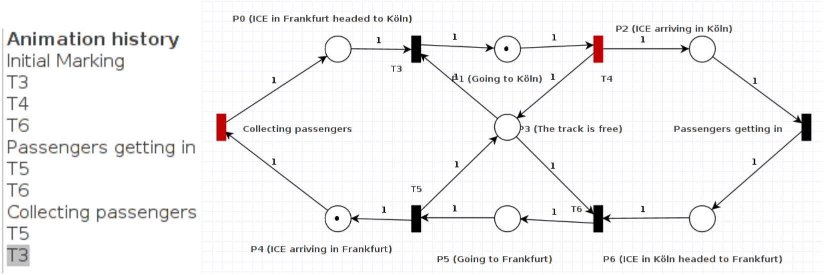
$$M_7 = (1, 0, 0, 0, 0, 1, 0)$$

$$|$$

$$M_8 = (1, 0, 0, 1, 1, 0, 0)$$

$$|$$

$$M_9 = (0, 1, 0, 0, 1, 0, 0)$$



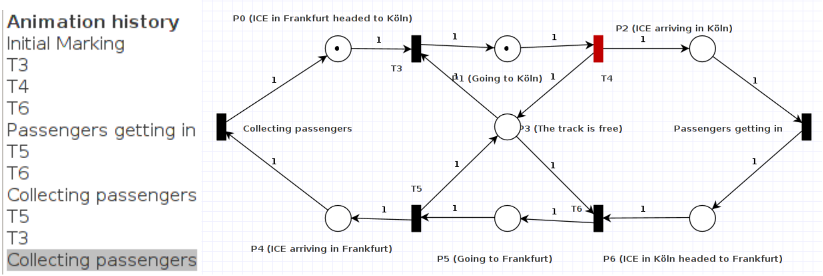
$$M_8 = (1, 0, 0, 1, 1, 0, 0)$$

$$|$$

$$M_9 = (0, 1, 0, 0, 1, 0, 0)$$

$$|$$

$$M_{10} = (1, 1, 0, 0, 0, 0, 0)$$



$$M_9 = (0, 1, 0, 0, 1, 0, 0)$$

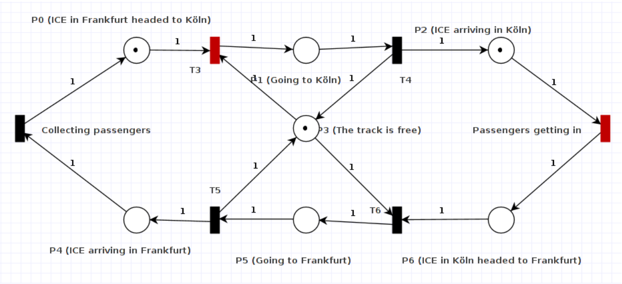
$$|$$

$$M_{10} = (1, 1, 0, 0, 0, 0, 0)$$

$$|$$

$$M_{11} = (1, 0, 1, 1, 0, 0, 0)$$

Animation history
Initial Marking
T3
T4
T6
Passengers getting in
T5
T6
Collecting passengers
T5
T3
Collecting passengers
T4

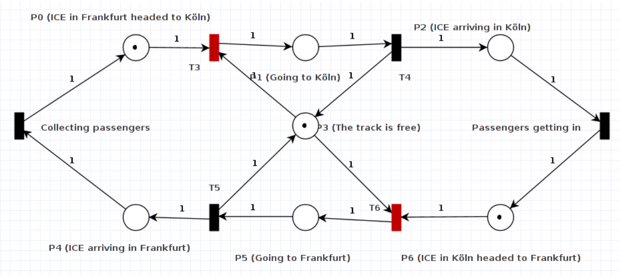


$$M_{10} = (1, 1, 0, 0, 0, 0, 0)$$

$$M_{11} = (1, 0, 1, 1, 0, 0, 0)$$

$$M_{12} = (1, 0, 0, 1, 0, 0, 1) = M_0!$$

Animation history
Initial Marking
T3
T4
T6
Passengers getting in
T5
T6
Collecting passengers
T5
T3
Collecting passengers
T4
Passengers getting in



Was haben wir gerade gemacht?

- Elf unterschiedliche Zustände wurden erreicht.

Was haben wir gerade gemacht?

- Elf unterschiedliche Zustände wurden erreicht.
- Ein Zyklus ist entstanden.

Was haben wir gerade gemacht?

- Elf unterschiedliche Zustände wurden erreicht.
- Ein Zyklus ist entstanden.
- Wegen solchen Zyklen ist der Erreichbarkeitsgraph beschränkt.

Der komplette Erreichbarkeitsgraph - Teil 1

- $M_0 = (1,0,0,1,0,0,1) = M_{22}$
  - $M_1 = (0,1,0,0,0,0,1)$
    - $M_2 = (0,0,1,1,0,0,1)$
      - $M_3 = (0,0,0,1,0,0,2)$
        - $M_4 = (0,0,0,0,0,1,1)$
          - $M_5 = (0,0,0,1,1,0,1)$
            - $M_6 = (0,0,0,0,1,1,0)$
              - $M_7 = (1,0,0,0,0,1,0)$
                - $M_8 = (1,0,0,1,1,0,0) = M_{25}$
              - $M_9 = (0,0,0,1,2,0,0)$
                - $M_{10} = (1,0,0,1,1,0,0) = M_{25}$
            - $M_{11} = (1,0,0,1,0,0,1) = M_0$
      - $M_{12} = (0,0,1,0,0,1,0)$
        - $M_{13} = (0,0,0,0,0,1,1) = M_4$
        - $M_{14} = (0,0,1,1,1,0,0)$
          - $M_{15} = (0,0,0,1,1,0,1) = M_5$
            - $M_{16} = (1,0,1,1,0,0,0) = M_{28}$
              - $M_{17} = (0,1,1,0,0,0,0)$
                - $M_{18} = (0,1,0,0,0,0,1) = M_1$
                - $M_{19} = (0,0,2,1,0,0,0)$
                  - $M_{20} = (0,0,1,1,0,0,1) = M_2$
              - $M_{21} = (1,0,0,1,0,0,1) = M_0$
  - Tree 2

Der komplette Erreichbarkeitsgraph - Teil 2

$M_{22} = (1,0,0,1,0,0,1) = M_0$

$M_{23} = (1,0,0,0,0,1,0)$ Tree 1

$M_{24} = (0,0,1,1,0,0,1)$

$M_{25} = (1,0,0,1,1,0,0) = M_8$

$M_{26} = (2,0,0,1,0,0,0)$ $M_{29} = (0,1,0,0,1,0,0)$

$M_{27} = (1,1,0,0,0,0,0)$ $M_{30} = (0,0,1,1,1,0,0) = M_{14}$ $M_{31} = (1,1,0,0,0,0,0)$

$M_{28} = (1,0,1,1,0,0,0) = M_{16}$ $M_{32} = (1,0,1,1,0,0,0) = M_{28}$

Was hat das mit der Definition von Petri-Netzen zu tun?

- Jede Transition verursacht eine Änderung des aktuellen Zustands.

Was hat das mit der Definition von Petri-Netzen zu tun?

- Jede Transition verursacht eine Änderung des aktuellen Zustands.
- Diese Zustandsänderung lässt sich mit einer ”Änderungsmatrix” darstellen.

Was hat das mit der Definition von Petri-Netzen zu tun?

- Jede Transition verursacht eine Änderung des aktuellen Zustands.
- Diese Zustandsänderung lässt sich mit einer "Änderungsmatrix" darstellen.
- Dazu betrachten wir ein Beispiel mit wenigen Zuständen und Transitionen:

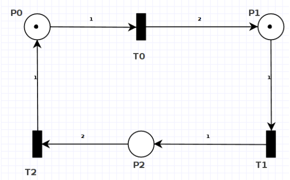


Abbildung: Ein Petri-Netz mit weniger Transitionen und Zuständen. Beachten Sie die gewichtung an den Kanten.

Die Eingangsmatrix aller Transitionen

$$\begin{array}{c|ccc} & T_0 & T_1 & T_2 \\ \hline P_0 & 1 & 0 & 0 \\ P_1 & 0 & 1 & 0 \\ P_2 & 0 & 0 & 2 \end{array}$$

Abbildung: In der **input matrix,** I wird die Gewichtung der eingehenden Kanten von jeder Transition dargestellt. 5

Die Ausgangsmatrix aller Transitionen

$$
\left(
\begin{array}{c|ccc}
 & T_0 & T_1 & T_2 \\
\hline
P_0 & 0 & 0 & 1 \\
P_1 & 2 & 0 & 0 \\
P_2 & 0 & 1 & 0
\end{array}
\right)
$$

Abbildung: In der **Ausgangsmatrix,** O wird die Gewichtung der ausgehenden Kanten von jeder Transition dargestellt.5

Die Ausgangsmatrix aller Transitionen

$$\left(\begin{array}{c|ccc} & T_0 & T_1 & T_2 \\ \hline P_0 & 0 & 0 & 1 \\ P_1 & 2 & 0 & 0 \\ P_2 & 0 & 1 & 0 \end{array}\right)$$

Abbildung: In der **Ausgangsmatrix,** O wird die Gewichtung der ausgehenden Kanten von jeder Transition dargestellt.5

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.
- T die Menge der "tokens" darstellt.

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.
- T die Menge der "tokens" darstellt.
- I die eben vorgestellte Eingangsmatrix darstellt.

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.
- T die Menge der "tokens" darstellt.
- I die eben vorgestellte Eingangsmatrix darstellt.
- O die eben vorgestellte Ausgangsmatrix darstellt.

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.
- T die Menge der "tokens" darstellt.
- I die eben vorgestellte Eingangsmatrix darstellt.
- O die eben vorgestellte Ausgangsmatrix darstellt.
- Und M_0 den Anfangszustand darstellt.

Die Definition

- Nun haben wir alles was wir für die Definition benötigen.

Definition

Petri-Netze sind definiert als Tupel $P_n : (P, T, I, O, M_0)$, wobei: [4]

- P die Menge der "places" darstellt.
- T die Menge der "tokens" darstellt.
- I die eben vorgestellte Eingangsmatrix darstellt.
- O die eben vorgestellte Ausgangsmatrix darstellt.
- Und M_0 den Anfangszustand darstellt.

Bemerkung:

Mit Hilfe der Eingangs und Ausgangsmatrix kann eine Übergangsmatrix $T = O - I$ erstellt werden, womit der neue Zustand aus dem aktuellen Zustand sich berechnen lässt. Falls eine bestimmte Transition zu einer negativen Anzahl an tokens führt kann diese Transition nicht ausgeführt werden.

Zusammenfassung

- Petri-Netze sind ein tolles Werkzeug um Systeme zu modellieren. Insbesondere:
  - Um Netze zu synchronisieren.

Zusammenfassung

- Petri-Netze sind ein tolles Werkzeug um Systeme zu modellieren. Insbesondere:
    - Um Netze zu synchronisieren.
    - Um scheduling Probleme zu lösen.

Zusammenfassung

- Petri-Netze sind ein tolles Werkzeug um Systeme zu modellieren. Insbesondere:
  - Um Netze zu synchronisieren.
  - Um scheduling Probleme zu lösen.
  - Um deadlocks zu verhindern bzw. erkennen.

Zusammenfassung

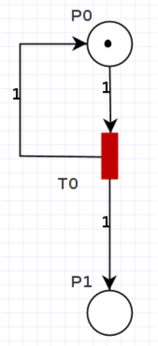
- Petri-Netze sind ein tolles Werkzeug um Systeme zu modellieren. Insbesondere:
  - Um Netze zu synchronisieren.
  - Um scheduling Probleme zu lösen.
  - Um deadlocks zu verhindern bzw. erkennen.
- Alle Zustände lassen sich in einem Erreichbarkeitsbaum darstellen.

Zusammenfassung

- Petri-Netze sind ein tolles Werkzeug um Systeme zu modellieren. Insbesondere:
  - Um Netze zu synchronisieren.
  - Um scheduling Probleme zu lösen.
  - Um deadlocks zu verhindern bzw. erkennen.
- Alle Zustände lassen sich in einem Erreichbarkeitsbaum darstellen.
- Mit Hilfe der Eingangs und Ausgangsmatrizen lassen sich Petri-Netze definieren: $P_n : (P, T, I, O, M_0)$ [4]

Ausblick

Wie sieht der Erreichbarkeitsgraph dieses Netzes aus? [5]



Prof. Dr. Wolfgang Thomas
Applied Automata Theory.
RWTH Achen
http://drona.csa.iisc.ernet.in/

Finkel, A. Leroux, J.
Softw Syst Model (2015) 14: 719. doi:10.1007/s10270-014-0426-0.
http://link.springer.com/article/10.1007

Dr. Chris Ling
The Petri Net Method.
School of Computer Science Software Engineering, Monash University
Slides at: www.utdallas.edu/~gupta/courses/semath/petri.ppt

Prof. Dr. Kristof Van Laerhoven
Introduction to Embedded Systems.
Technical Faculty, University of Freiburg
Slides at: https://es.informatik.uni-freiburg.de/teaching/embedded-systems

Prof. Dr. Ir. Wil van der Aalst
State space Analysis: Properties, Reachability Graph and Coverability Graph.
Technische Universiteit Eindhoven
Slides at: http://cpntools.org/_media/book/covgraph.pdf