Q3. Modelltypen

Gliederung

1.Integration von Zeiten und Wahrscheinlichkeiten in diskrete Modelle

2.Zeitbehaftete Automatenmodelle

3.Stochastische und zeitbehaftete Petri-Netze

4.Weitere Modelltypen

5.Last-Maschine Modellierung

Literatur (primär)

- C. G. Cassandras, S. Lafortune
Introduction to Discrete Event Systems. Springer 2008
Kap. 5, 6
- B. Berard et al
Systems and Software Verification. Springer 1999
Kap. 5
- R. David, H. Alla
Discrete, Continuous, and Hybrid Petri Nets. Springer 2005
Kap. 5
- R. A. Sahner, K. S. Trivedi, A. Puliafito
Performance and Reliability Analysis of Computer Systems
Kluwer 1996
Kap. 7

Ziele:

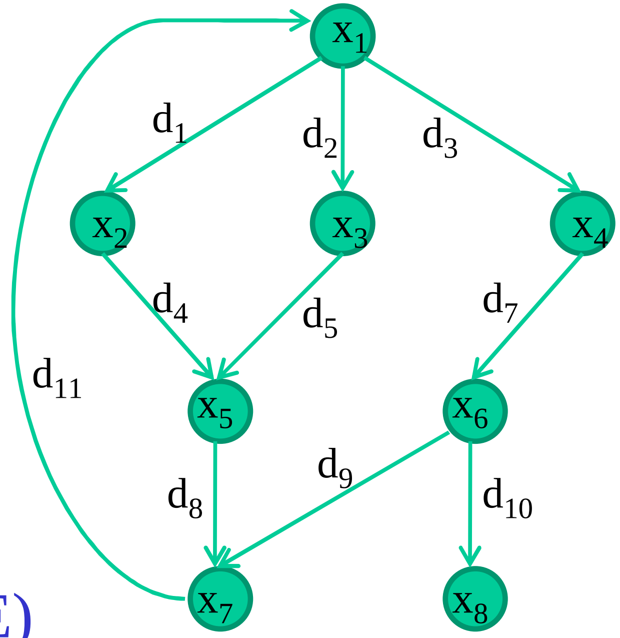
- Kennen lernen unterschiedlicher Methoden Zeit und Wahrscheinlichkeiten in funktionale Modelle zu integrieren
- Kennen lernen verschiedener stochastischer und zeitbehafteter Modelle
- Grenzen und Möglichkeiten von Modelltypen einordnen können
- Resultierende Prozessmodelle beschreiben können
- Möglichkeiten der strukturierten Modellierung erarbeiten

Q 3.1 Integration von Zeiten und Wahrscheinlichkeiten in diskrete Modelle

Systemverhalten beschrieben durch Zustands-/Transitionssystem

Systemdynamik entsteht durch Zustandswechsel

Endliche LTS mit Zyklen erlauben unendliches Verhalten



Dynamisches Verhalten (3 Möglichkeiten):

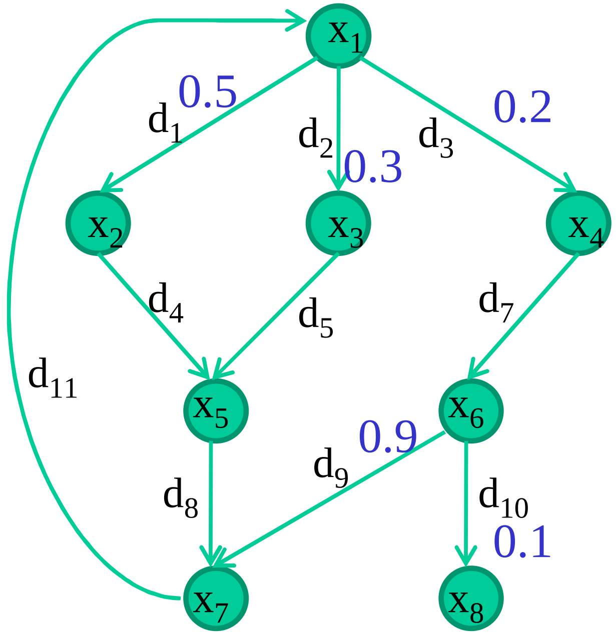
- Sequenz von Zuständen $(z_1,z_2,\ldots)$ $(z_i \in X)$
- Sequenz von Transitionen $(e_1,e_2,\ldots)$ $(e_i \in E)$
- Sequenz von Zustands-/Transitionspaaren $(z_1,e_1,z_2,e_2,\ldots.)$

$X=\{x_1, \ldots, x_8\}$
$E=\{d_1,\ldots, d_{11}\}$

Gewichtung von Transitionen:

➢Immer dann, wenn das System nicht deterministisch verhält, wird eine Wahrscheinlichkeitsverteilung über alle möglichen Alternativen definiert (globale Definition)

Beispiele:



•$\text{Prob}(d_1,d_4,d_8) = 0.5$

•$\text{Prob}(d_3,d_7,d_{10}) = 0.02$

•$\text{Prob}(d_1,d_5,d_8) = 0.0$

Ähnlich für Zustände und Zustand/Transition

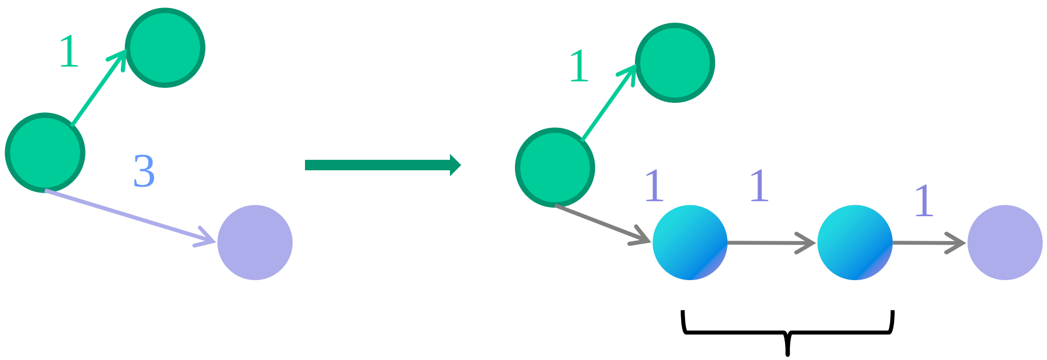
Weitere Aussagen sind möglich:
z.B. die Summe der Pfadwahrscheinlichkeiten aller Pfade der Länge $\leq k$, die in x_8 enden, konvergiert gegen 1 für $k \rightarrow \infty$
(Weiteres dazu im Kapitel Q4 Zeitdiskrete Markov-Prozesse)

Einführung von Zeit:

- ➢Einfachste Variante: System verweilt genau eine Zeiteinheit im Zustand und führt anschließend eine Transition aus

⇒ Pfadlänge bestimmt die Zeitdauer

- ➢ Bei unterschiedlichen Zeitdauern, Einführung von Zwischenzuständen



u.U. sehr viele neue Zwischenzustände

Zustandsfarbe Grün oder blau??

Allgemeine Form:

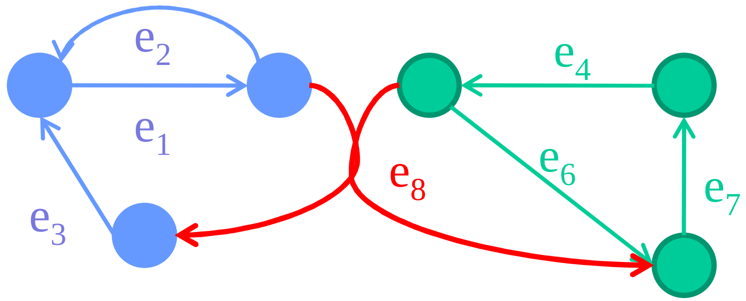
➢ $(z_1,t_1,z_2,t_2,z_3,..)$ oder $(t_1,e_1,t_2,e_2,\ldots)$ mit $t_i \geq 0$ Zeiten

implizite Annahme

- ➢ System verbleibt für eine Zeit im Zustand
- ➢ Wenn die Zeit abgelaufen ist, so wird ein Ereignis (zeitlos, atomar) ausgeführt
- ➢ Für Zustand-/Transitionssequenzen gilt dann $(z_1,t_1, e_1,z_2,t_2, e_2,z_3,..)$
- ➢ Damit sind die blau/günen Zustände auf der vorherigen Folie …

Es gibt andere Interpretationen in der Literatur, diese führen allerdings zu nicht klar definierten Zwischenzuständen

Verteilte Zustandsbeschreibung (in Petri-Netzen, Automatennetzen,…)



Kommunikation hier synchron

$e_1e_2e_1 \quad e_3e_1$

e_8

$e_4e_6e_7e_4 \quad e_7e_4$

Beliebige Mischung der blauen und grünen Sequenz bis zum roten synchronisierenden Ereignis möglich

$t_1e_1t_2e_2t_3e_1t_4 \quad t_5e_3t_6e_1$

e_8

$t_7e_4t_8e_6t_9e_7t_{10}e_4t_{11} \quad t_{12}e_7t_{13}e_4$

Relationen zwischen den Zeiten:

– $t_1 \leq t_2 \leq t_3 \leq t_4 \leq t_5 \leq t_6$

– $t_7 \leq t_8 \leq t_9 \leq t_{10} \leq t_{11} \leq t_{12} \leq t_{13}$

– $t_1+t_2+t_3+t_4 = t_7+t_8+t_9+t_{10}+t_{11}$

Einführung von absoluten Zeiten definiert eine Ordnung unter allen Ereignissen (außer bei gleichzeitigen Ereignissen)

Q 3.2 Zeitbehaftete Automatenmodelle

Vielzahl von unterschiedlichen Varianten existiert in der Literatur!

Die hier vorgestellten Modelle und Notationen orientieren sich am Buch von Casandras, Lafortune Kap. 5.2, 5.6, 6

Definition (Timed Automaton, Zeitautomat)

Ein Zeitautomat ist ein 6-Tupel $G_z = (X, E, f, \Gamma, x_0, \mathbf{V})$ mit

- X ist eine abzählbare Zustandsmenge
- E ist eine abzählbare Ereignismenge
- $f: X \times E \rightarrow X$ ist die (partielle) Zustandsübergangsfunktion
- $\Gamma: X \rightarrow 2^E$ ist die Ereignisaktivitätsfunktion

  d.h. $e \in \Gamma(x) \Leftrightarrow f(x,e)$ ist definiert
- x_0 ist der initiale Zustand
- $\mathbf{V} = \{\mathbf{v}_e \mid e \in E\}$ ist eine Weckerstruktur mit
  - $\mathbf{v}_e = \{v_{e,1}, v_{e,2}, \ldots\}$ wobei $v_{e,k} \in \mathbb{R}_+$, $k=1,2,3,\ldots$

Um das Verhalten des Automaten zu definieren, benötigen wir zusätzlich:

- $N_e \in \mathbb{N}$ Zähler für die Anzahl der Aktivierungen von Ereignis e
- $y_e \in \mathbb{R}_+$ Zeit bis zum Ablauf des Weckers für Ereignis e

Definition des Verhaltens:

Initialisierung

- $N_e = 1$ falls $e \in \Gamma(x_0)$ und 0 sonst
- $y_e = v_{e,1}$ falls $e \in \Gamma(x_0)$ und $\perp$ sonst
($\perp$ undefiniert mit $\min(x,\perp) = x$ für $x \in \mathbb{R}$)

Dynamik ausgehend vom Zustand (x, **V**, **N**, t) mit $\mathbf{N} = \{N_e \mid e \in E\}$, t Systemzeit

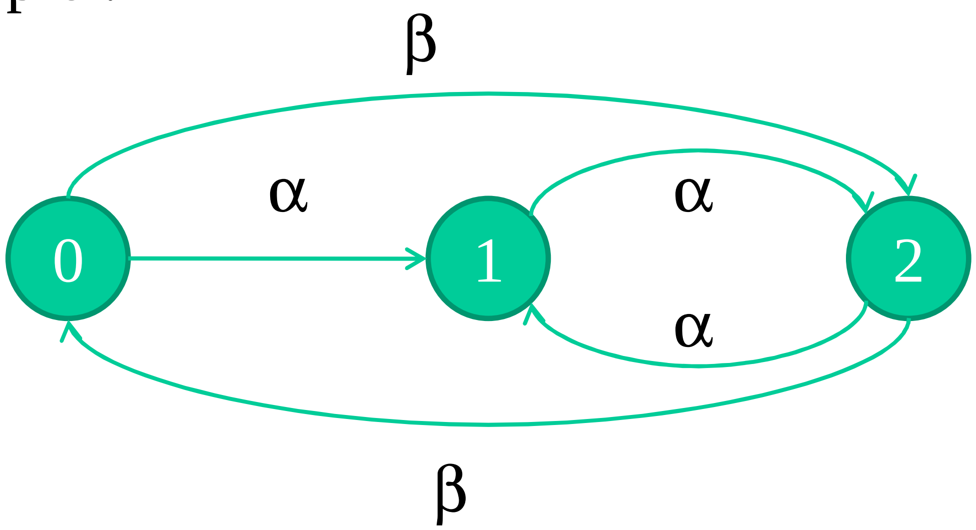
- $y' = \min_{e \in \Gamma(x)} (y_e)$ und $e' = \text{argmin}_{e \in \Gamma(x)} (y_e)$
- $t' = t + y'$ und $x' = f(x, e')$
- $N_e' = N_e + 1$ für $e \in \Gamma(x') \wedge (e = e' \vee e \notin \Gamma(x))$ und N_e sonst
- $y_e' = \begin{cases} v_{e,N_e+1} & \text{falls } e' \in \Gamma(x') \text{ und } (e = e' \text{ oder } e \notin \Gamma(x)), \\ y_e - y_{e'} & \text{falls } e \neq e' \text{ und } e \in \Gamma(x) \cap \Gamma(x'), \\ v_{e,N_e} & \text{falls } e \notin \Gamma(x) \text{ und } e \in \Gamma(x') \\ \perp & \text{falls } e \notin \Gamma(x') \end{cases}$

Neuer Zustand
(x‘, **V‘**, **N‘**, t‘)

Bemerkungen :

- Automatenmodell beschreibt ein globales Verhaltensmodell
- Verhalten der Form (z_0, t_0, e_0, z_1, t_1, e_1, ….)
mit $z_i \in X$, $t_i \in \mathbb{R}+$ ($t_i \leq t_{i+1}$), $e_i \in E$
- Verhalten bei gleichzeitigen Ereignissen (argmin nicht eindeutig) undefiniert (Verhaltensdefinition notwendig!)
- **V** wird als gegeben vorausgesetzt (Cassandras/Lafortune folgend), andere Möglichkeiten existieren, z.B.
    - **V** entsteht aus einer Messung
    - Dauer jeder Operation ist konstant
    - Zusätzliche Entscheidung, ob verstrichene Zeit der i-ten Aktivierung bei der i+1-ten Aktivierung berücksichtigt wird oder nicht (Restzeit verfällt)
- Theoretisch unendliche Abläufe, können begrenzt werden durch
    - Zustand ohne aktive Ereignisse ($\Gamma(x)=\emptyset$)
    - Vorgabe einer Endzeit oder maximalen Ereigniszahl

Beispiel:



X={0,1,2}

E={α, β}

$\mathbf{v}_{\alpha}$ = {1.0, 1.5, 1.5, …}

$\mathbf{v}_{\beta}$ = {2.0, 0.5, 1.5, …}

Ablauf: ….

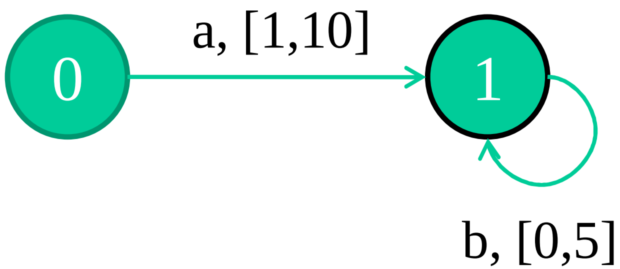
Mögliche Erweiterungen:

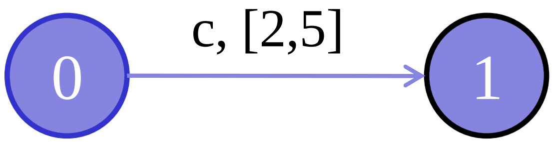
Stochastisches Verhalten

- Bezüglich der Zeit
  - $v_{e,k}$ beschrieben durch Zufallsvariable mit Vfkt. abhängig von
    - Ereignis e
    - $v_{e,1}, \ldots, v_{e,k-1}$ der vorherigen Zeiten
    - dem Zustand x
    - allen Ereigniszeiten bis zum aktuellen Zeitpunkt
    - …
- Bezüglich der Zustandswechsel
  - Übergangswahrscheinlichkeiten p(x,x',e) mit
    - $p(x,x',e) = 0$ falls $e \notin \Gamma(x)$
    - $\Sigma_{x' \in X} p(x,x',e) = 1.0$ für $e \in \Gamma(x)$
- Kombinationen aus beiden Varianten

Automat beschreibt einen stochastischen Prozess
(Klassifizierung später in Q4 und Q5)

Zeitbehaftete Automaten mit Intervallen und Transitionsbedingungen





- Zeitintervall beginnt bei Betreten des Ausgangszustandes
- Transition schaltet innerhalb des Intervalls (früheste, späteste Feuerungszeit)
- Vorgehen mit lokalen Uhren ist nicht kompositionell z.B. Komposition der beiden Automaten und die Transition schaltet im ersten (grünen) Automaten liefert Zustand (1,0) Wann soll/darf dann die Transition im blauen Automaten schalten?

- Erweitertes Automatenmodell (nach Alur TCS 1994) hier nur in Ansätzen erläutert

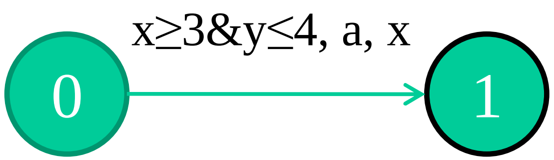
Definition einer Menge Wecker C (im Englischen als *clocks* bezeichnet)

Transitionen als Tripel (B, e, R) mit

- B Boolescher Ausdruck über die Wecker
- e Ereignis
- R $\subseteq$ C Menge der Wecker, die zurück gesetzt werden

- kann ebenfalls vorkommen und wird als leere Menge definiert

Beispiele mit C = {x, y}



$$(0,x=5,y=3) \xrightarrow{a} (1,x=0,\ y=3)$$

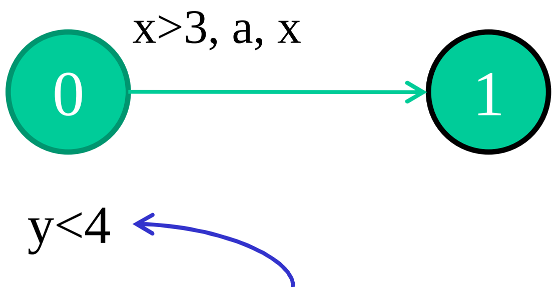
$$(0,x=2,y=3) \rightarrow (0,x=3,\ y=4) \xrightarrow{a} (1,x=0,\ y=4)$$

Zwei Arten von Transitionen: Zeitfortschritt und Zustandswechsel!

Transitionen müssen nicht zwangsläufig eintreten $\Rightarrow$ Gefahr von Deadlocks!

Beispiel $(0,x=3,y=4) \rightarrow (0,x=4,\ y=5)$ Deadlock!

Zusätzliches Element: Invarianten in Zuständen



Zustand muss verlassen werden, bevor $y \geq 4$

Vorsicht, inkonsistente Spezifikationen sind immer noch möglich

z.B. Zustand 0 wird betreten und $y>4$ oder
Zustand 0 wird betreten und $y-x \geq 1$

dies wird als inkorrekte Spezifikation angesehen

Invarianten garantieren Fortschritt, wenn die Bedingungen in den Transitionen so definiert sind, dass sie den Invarianten nicht widersprechen

Definition (*Timed Automaton with Guards*, zeitbehafteter Automat)

Ein zeitbehafteter Automat ist ein 6-Tupel $G_t = (X, E, C, Tra, Inv, x_0)$ mit

- X ist eine abzählbare Zustandsmenge
- E ist eine abzählbare Ereignismenge
- C ist eine Menge von (globalen) Weckern
- $Tra \subseteq X \times \mathcal{C}(C) \times E \times 2^C \times X$ ist die Transitionsmenge
- $Inv: X \rightarrow \mathcal{C}(C)$ ist die Menge der Zustandsinvarianten
- $x_0 \in X$ initialer Zustand

mit

– $\mathcal{C}(C)$ ist die Menge der *guards* (deutscher Begriff fehlt) der Form $c<r$, $c\leq r$, $c>r$ oder $c\geq r$ mit $c\in C$ und $r\in \mathbb{R}_+$

falls $g, g' \in \mathcal{C}(C)$, dann ist auch $g \wedge g' \in \mathcal{C}(C)$

wir nehmen an, dass Wecker durch ihren Wert ≥ 0 beschrieben sind

Bemerkungen:

- Zustände der Automaten werden im Englischen auch als *locations* im Gegensatz zu Zuständen (Engl. *states*) in Transitionssystemen bezeichnet
- Die Zeit schreitet auf allen Weckern mit Rate 1 fort
- Zu Beginn stehen alle Wecker auf 0 und müssen explizit im Modell initialisiert werden
- Guards und Invarianten müssen nicht alle Wecker berücksichtigen
- Automat ist deterministisch, falls in jedem Zustand für jeden Buchstaben des Alphabets nur maximal eine Transition schalten kann, ansonsten ist der Automat nichtdeterministisch (Auflösung des Nichtdeterminismus falls nötig über Gewichte/Wahrscheinlichkeiten)
- Übliche Annahme: Tra und Inv sind so definiert, dass in jedem Zustand eine Transition schalten kann und der Automat nicht beliebig lange im Zustand verweilen kann (außer evtl. in Endzuständen)
- Falls für all $x \in X$ Inv(x) = true und in Tra guard = true, dann verhält sich der Automat wie ein zeitloser Automat

Dynamisches Verhalten:

Zustand des Automaten $(x, \mathbf{c}(t))$ mit $x \in X$, $c(t) \in \mathbb{R}_+^{|C|}$ Einstellung der Wecker und t der aktuellen Zeit

Zwei Arten von Transitionen:

1.Zeitschritt:

die Zeit ändert sich von t_1 auf t_2 ($d=t_2-t_1>0$), ohne dass ein Ereignis eintritt
Zustand ändert sich von $(x, \mathbf{c}_1(t_1))$ auf $(x, \mathbf{c}_2(t_2))$ mit $\mathbf{c}_2(t_2) = \mathbf{c}_1(t_1)+d$
Schritt ist möglich, falls $Inv(x, \mathbf{c}(t), t) = true$ für alle $t \in [t_1,t_2]$
Darstellung: $(x, \mathbf{c}_1(t_1)) \rightarrow^d (x, \mathbf{c}_2(t_2))$

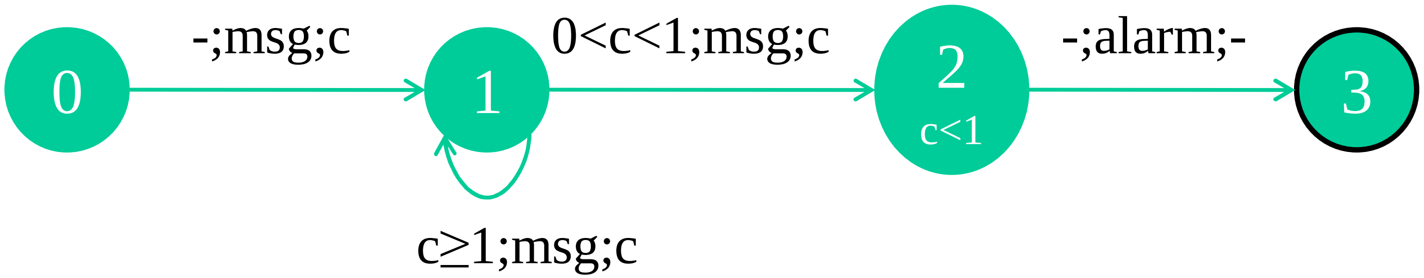
2.Ereignisschritt:

Transition $(x_{in}, guard, e, res, x_{out})$ schaltet zum Zeitpunkt t_e und ändert den Zustand von $(x_{in}, \mathbf{c}(t_e))$ nach $(x_{out}, \mathbf{c}(t_e^+))$ wobei

- $guard(\mathbf{c}(t_e)) = true$
- $\mathbf{c}_i(t_e^+) = \mathbf{c}_i(t_e)$ falls $i \notin res$ und $\mathbf{c}_i(t_e^+) = 0$ falls $i \in res$

Darstellung: $(x_{in}, \mathbf{c}(t_e)) \rightarrow^e (x_{out}, \mathbf{c}(t_e^+))$

Beispiel:



Ablauf: ….

Parallele Komposition:

- Aufbau von Automatennetzen durch Synchronisation Transitionen (z.B. zur Modellierung paralleler Prozesse)
  - Einige Transitionen gekoppelt, d.h. müssen synchron in den gekoppelten Automaten ausgeführt werden
  - Andere Transitionen bleiben lokal
- Gekoppelter Automat ist wieder ein zeitbehafteter Automat
globaler Zustand entsteht aus der Vereinigung der Zustände der komponierten Automaten
- Alle Wecker werden zu globalen Variablen
- Guards von gekoppelten Automaten entstehen aus der Konjunktion der Guards gekoppelter Transitionen
- Menge der Wecker, die zurückgesetzt werden, entspricht der Vereinigung der Mengen in den Transitionen
- Zustandsinvarianten entstehen aus der Konjunktion der Zustandsinvarianten in den einzelnen Automaten

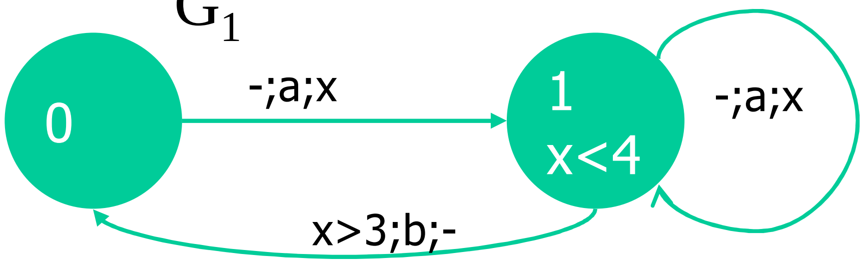
Definition (Parallele Komposition zeitbehafteter Automaten)

Seien $G_1 = (X_1, E_1, C_1, Tra_1, Inv_1, x_{01})$ und $G_2 = (X_2, E_2, C_2, Tra_2, Inv_2, x_{02})$ zweitbehaftete Automaten, dann ist die parallel Komposition $G_1 \| G_2$ ein zeitbehafteter Automat $G_{12} = (X_{12}, E_{12}, C_{12}, Tra_{12}, Inv_{12}, x_{012})$ mit

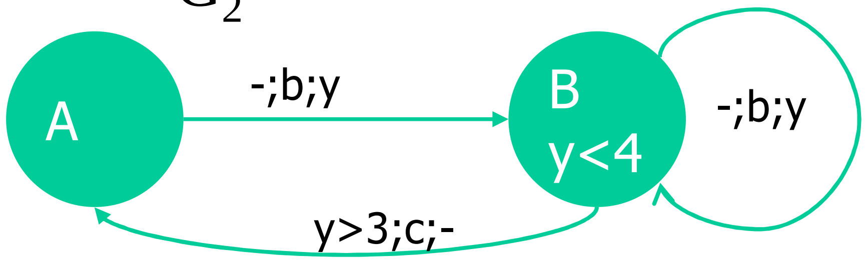
- $X_{12} = X_1 \times X_2$
- $E_{12} = E_1 \cup E_2$
- $C_{12} = C_1 \cup C_2$
- $x_{012} = (x_{01}, x_{02})$
- $Inv_{12} : X_{12} \rightarrow \mathcal{C}(C_{12})$ mit $\mathcal{C}(C_{12}) = \mathcal{C}(C_1) \wedge \mathcal{C}(C_2)$,
so dass $Inv_{12}(x_1, x_2) = Inv_1(x_1) \wedge Inv_2(x_2)$
- $Tra_{12} \subseteq X_{12} \times \mathcal{C}(C_{12}) \times E_{12} \times 2^{C12} \times X_{12}$ Transitionsmenge mit
  - Für alle $e \in E_1 \cap E_2$ und $(x_i, guard_i, e, res_i, y_i) \in Tra_i$ $(i=1,2)$
    $((x_1,x_2), guard_1 \wedge guard_2, e, res_1 \cup res_2, (y_1, y_2)) \in Tra_{12}$
  - Für alle $e \in E_1 \setminus E_2$ und $(x_1, guard_1, e, res_1, y_1) \in Tra_1$
    $((x_1,x_2), guard_1, e, res_1, (y_1, x_2)) \in Tra_{12}$
  - Für alle $e \in E_2 \setminus E_1$ und $(x_2, guard_2, e, res_2, y_2) \in Tra_2$
    $((x_1,x_2), guard_2, e, res_2, (x_1, y_2)) \in Tra_{12}$

Beispiel:

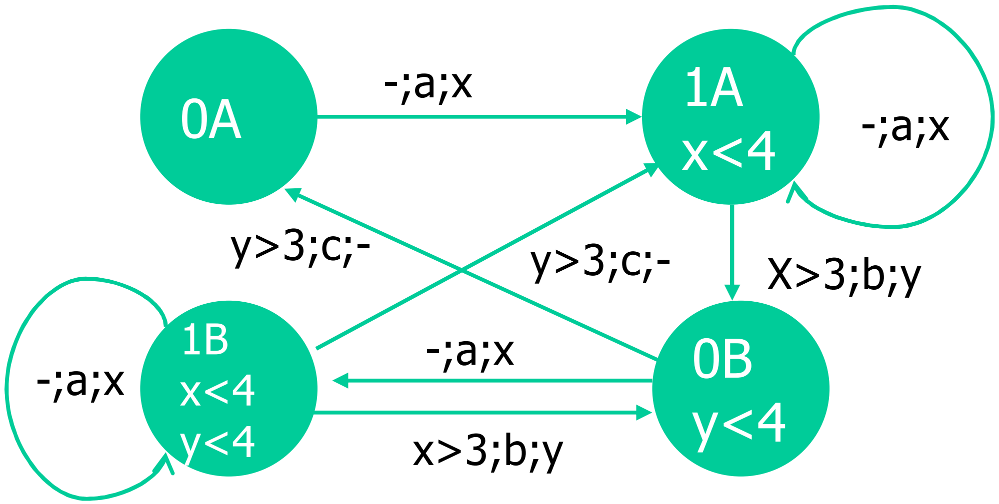
G_1



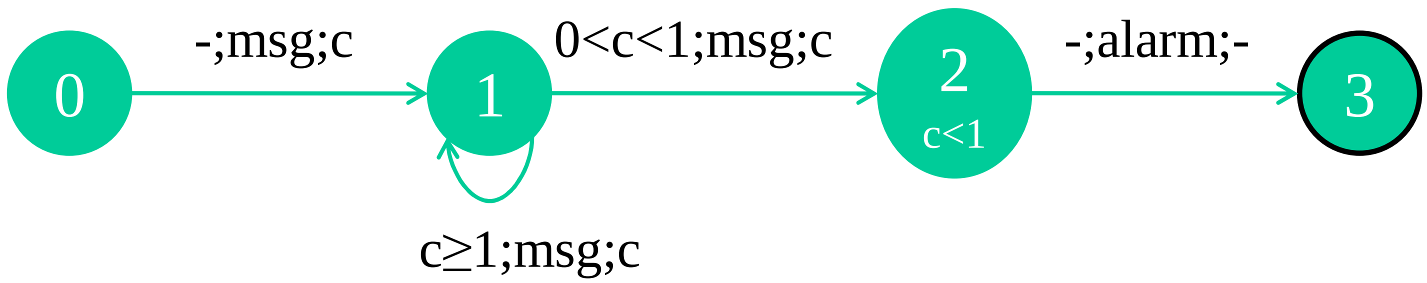
G_2



G_{12}



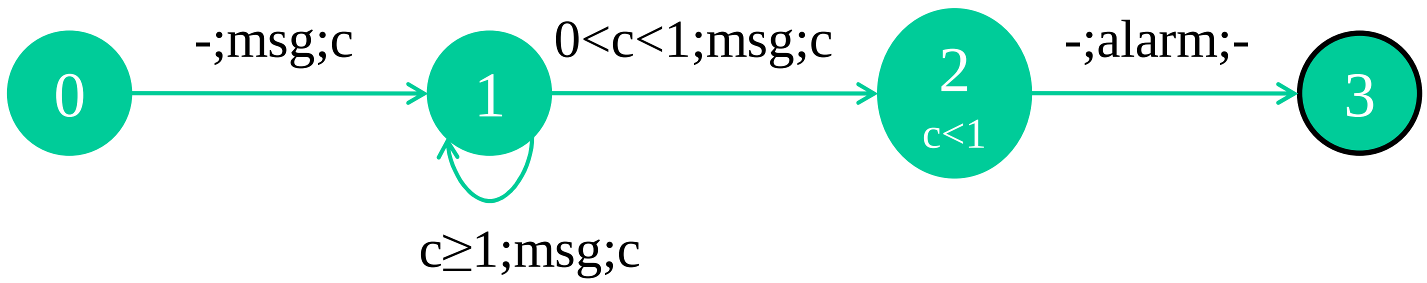
Beispiel für das dynamische Verhalten



Verhalten des Automaten (verbale Beschreibung):

- Die Ankunft der ersten Nachricht führt zu einem Übergang von 0 nach 1
- Nachrichten, die im Zustand 1 im Abstand von mindestens 1 Zeiteinheit ankommen, führen dazu, dass der Automat im Zustand bleibt und Wecker c zurückgesetzt wird
- Die erste Nachrichten, die im Zustand 1 im Abstand von weniger als 1 Zeiteinheit ankommt, führt zu einem Übergang in Zustand 2 und ein Zurücksetzen von c
- Im Zustand 2 wird innerhalb von einer Zeiteinheit *alarm* ausgegeben und der Automat geht in Zustand 3 über

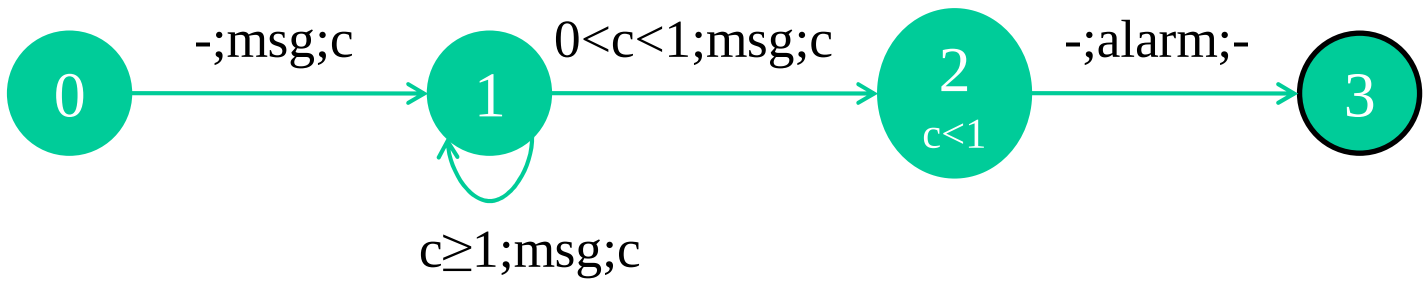
Formale Verhaltensbeschreibung!?



Weglassen der Zeit ??
Verbindung von *alarm* und kurz hintereinander gesendeten Nachrichten geht verloren!!

Einfügen von Zwischenzuständen, um zu zählen ??
Zeit ist reellwertig, dadurch kein Zeittakt vorhanden!
Zustandsraum $X \times \mathbb{R}_+$ (überabzählbar)

Man kann aber unterscheiden zwischen Zeiten, die das zukünftige Verhalten des Automaten beeinflussen und solchen, die dies nicht tun!



Weiterverfolgen der Idee, Zeitintervalle zu definieren, die Verhalten beeinflussen!
Interessante Intervalle werden durch *guard* und *Inv* definiert und als **Regionen** bezeichnet!

Zustandsweise Betrachtung:

- Zustand 0: keine Zeitbeschränkungen
- Zustand 1: c ist bei Betreten des Zustand immer gleich 0

Unterscheidung der Zwischenankunftszeiten 0, 0<1, ≥1

- Zustand 2: c ist bei Betreten des Zustand immer gleich 0

Unterscheidung der Zeit <1, ≥1

- Zustand 3: keine Zeitbeschränkungen

- 4 Regionen: $R_1 = 0$, $R_2 = (0,1)$, $R_3 = 1$ und $R_4 = (1,\infty)$

Idee zur Analyse: Erstellung eines zeitlosen Automaten mit

- Zustandsbeschreibung aus Zustand x und Region R_i

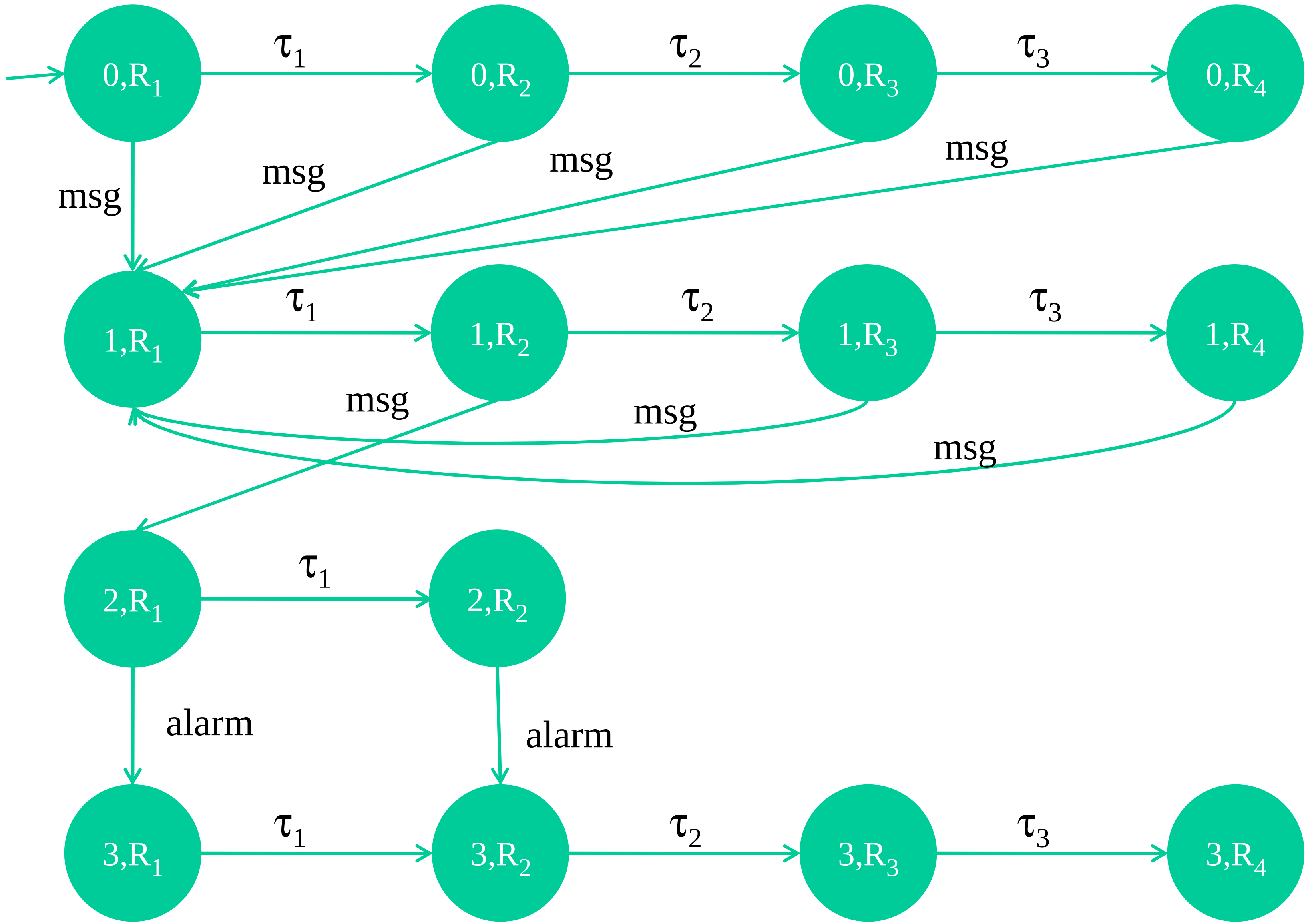
Interpretation: Automat befindet sich während Zeit t aus Region R_i im Zustand x

- Zwei Arten von Transitionen
  - Zeitschritt: Übergang aus Zustand und Region in eine neue Region
  - Ereignisschritt: Übergang aus Zustand und Region durch Transition *tran* in neuen Zustand und neue Region
    (durch Zurücksetzen von Weckern)

Analyse des resultierenden zeitlosen Automaten erlaubt Aussagen über das Verhalten des zeitbehafteten Automaten!

Vorstellung des Ansatzes hier am Beispiel

Später etwas mehr dazu


0,R1
0,R2
0,R3
0,R4
τ1
τ2
τ3
msg
msg
msg
msg
1,R1
1,R2
1,R3
1,R4
τ1
τ2
τ3
msg
msg
msg
2,R1
2,R2
τ1
alarm
alarm
3,R1
3,R2
3,R3
3,R4
τ1
τ2
τ3

Q3.4 Zeitbehaftete und stochastische Petri-Netze

Erweiterungen von Petri-Netzen im Zeit noch zahlreicher als im Automatenbereich

Zeit kann verbunden werden mit

- Transitionen
  - Schaltdauern, Zeit bis zum Schalten
- Stellen
  - Token werden auf Stellen festgehalten
- Tokens
  - Token bringt eine Zeit mit
- Kanten
  - Zeit vergeht beim Lauf entlang einer Kante

Zeit kann deterministisch, als Intervall oder stochastisch definiert werden

Wir betrachten hier kurz:

1. Netze mit deterministischen Zeitdauern der Transitionen
basierend auf Ramchandani 1974

2. Netze mit Zeitintervallen an den Transitionen
basierend auf Merlin 1974

3. Netze mit exponentiellen Zeitdauern der Transitionen
basierend auf Molly 1982

Literatur:

- P. Starke. Analyse von Petri-Netz-Modellen. Teubner 1990, Kap. 18, 19
- C. G. Cassandras, S. Lafortune. Introduction to Discrete Event Systems. Springer 2008. Kap. 5.3
- R. David, H. Alla. Discrete, Continuous and Hybrid Petri Nets. Springer 2005. Kap. 3.4
- B. Berthomieu, M. Diaz. Modeling and Verification of Time Dependent Systems Using Time Petri Nets. IEEE Trans. Softw. Eng. 17 (3), 1991.

Deterministische Schaltdauern für Transitionen

Sei $N = (S,T,F,W,m_0)$ ein Petri-Netz und

D eine Abbildung, die jeder Transition eine positive rationale Zahl als Schaltdauer zuweist

Einschränkungen:

- Schaltdauer 0 wird ausgeschlossen
- Transitionen können nicht mehrere Schaltvorgänge simultan ausführen
- o.B.d.A. sei D(t) ganzzahlig (kann durch Normierung der Zeitskala bei rationalen Zeiten immer erreicht werden)

$\Rightarrow$ Wir betrachten das Netz zu Zeitpunkten 0, 1, 2, …

Netzverhalten wird beschrieben durch die Schaltregeln

- D(t) soll angeben, wie lange das Schalten dauert
- Transition konsumiert Marken zu Beginn des Schaltens
- und gibt diese nach Abschluss des Schaltvorgangs wieder frei

(widerspricht unserer bisherigen Sicht der Trennung von Transition und Zustand!! $\Rightarrow$ spätere Transformation stellt diese Sicht wieder her)

Zustand des Netzes ist ein Tupel [m, u], wobei

- m: $S \rightarrow \mathbb{N}$ (Markierung der Stellen, wie üblich)
- u: $T \rightarrow \mathbb{N}$ (ordnet jeder Transition eine natürliche Zahl $u(t)<D(t)$ zu)

Transitionszustand mit

- $u(t) = 0$ Transition ist nicht aktiv
- $u(t) > 0$ Transition ist aktiv, *im Inneren* befinden sich Marken, Schaltvorgang hat zum Zeitpunkt τ-u(t) begonnen τ ist die aktuelle Zeit

Widerspricht eigentlich unserer Sicht !!

Die Zustandsbeschreibung enthält die aktuelle Zeit nicht!!

Dynamik des Netzes durch einfaches Inkrementieren der Zeit

Zu einem neuen Zeitpunkt

i.Schalten Transitionen und geben Marken frei

ii.Inkrementieren aktive Transitionen ihre Aktivierungszeiten

iii.Werden Transitionen aktiv und konsumieren Marken

(in dieser Reihenfolge)

Sei $V \subseteq T$ eine Menge von Transitionen, die zum Zeitpunkt τ mit dem Schalten neu beginnen und [m, u] sei der Zustand zum Zeitpunkt τ, ferner sei [m‘, u‘] der Zustand zum Zeitpunkt $\tau+1$, wobei

$$m'(s) = m(s) - \sum_{t \in V} W(s,t) + \sum_{t \in V, d(t)=1} W(t,s) + \sum_{t \in T, u(t)=d(t)-1} W(t,s)$$

$$u'(t) = \begin{cases} 1 & \text{falls } t \in V \wedge d(t) > 1 \\ u(t)+1 & \text{falls } t \notin V \wedge 0 < u(t) < d(t)-1 \\ 0 & \text{sonst} \end{cases}$$

Notation:

- [m, u] [V> [m‘, u‘]
- [m, u] [*> [m‘, u‘] ⇔ $[m_1, u_1]\ [V_1> \ldots [V_{X-1}>\ [m_X, u_X]$
  mit $[m, u] = [m_1, u_1]$ und $[m', u'] = [m_X, u_X]$

Auswahl der Schaltmenge V im Zustand [m, u] !?
Zu beachten ist

1. $t \in V \Rightarrow u(t) = 0$ (Transitionen können nicht nebenläufig schalten)
2. $\forall s \in S: \Sigma_{t \in V} W(s,t) \leq m(s)$ (alle Transitionen aus V müssen Schalten können)

Darüber hinaus soll V maximal sein, d.h.

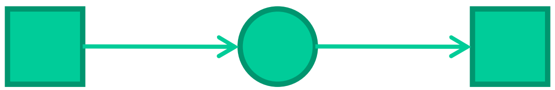
3. Keine Menge $W \subseteq T$ und $V \subset W$ erfüllt die Bedingungen 1. und 2.

Auswahl der Menge V erfolgt indeterministisch!

Initiale Markierung sei $[m_0, 0]$

1. Erreichbarkeitsmenge EM enthält alle Zustände $[m, u]$ mit $[m_0, 0][*>[m, u]$
2. Erreichbarkeitsgraph enthalt für jedes $[m, u] \in EM$ einen Knoten und eine mit V bewertete Kante, falls $[m, u][V>[m‘, u‘]$

Beispiel:



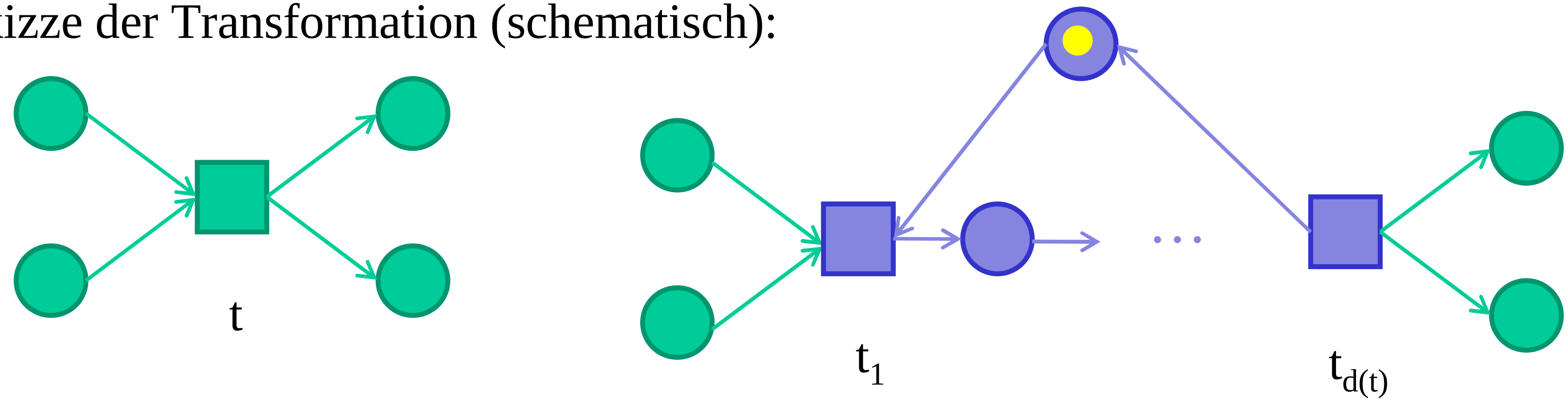
$d(t_1)=3$ $d(t_2)=2$

Erreichbarkeitsgraph:

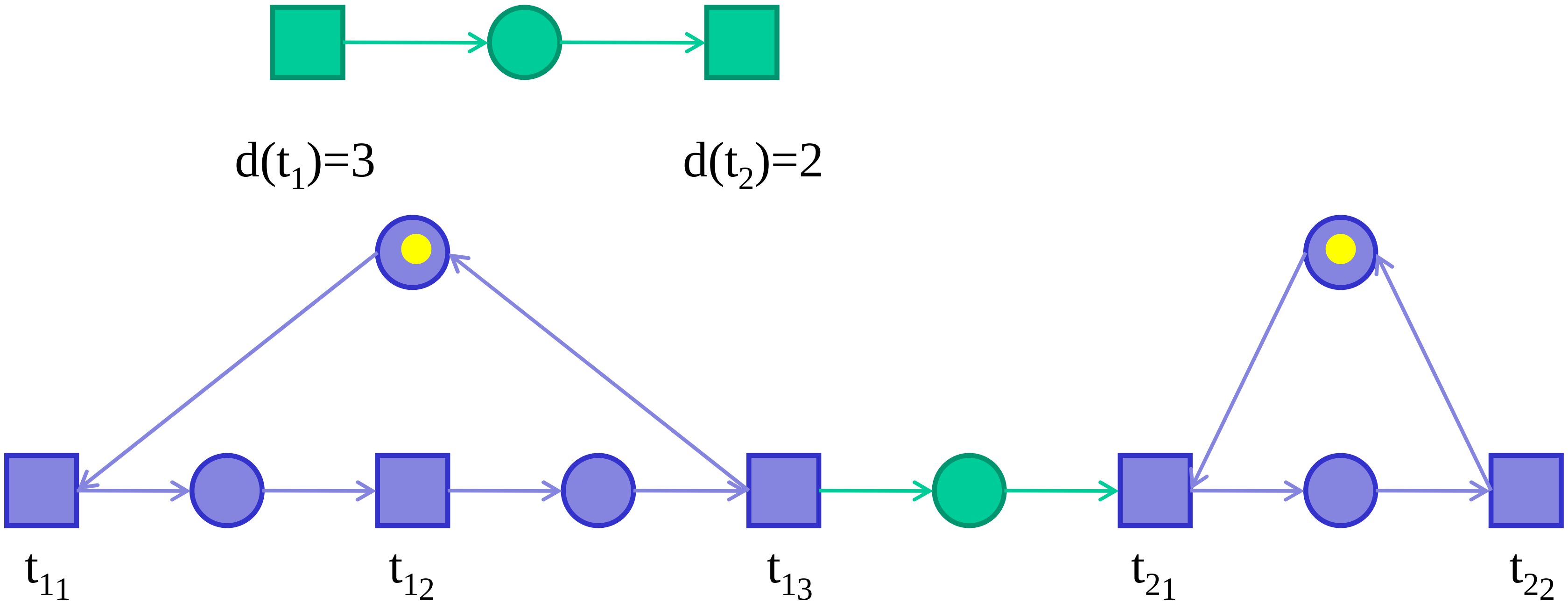
Folgerungen:

- Falls $d(t) = 1$ für alle $t \in T$, dann gilt für alle $[m, u] \in EM$ $u=0$
- Damit ist der Erreichbarkeitsgraph des zeitbehafteten Netzes und eines zeitlosen Netzes mit Maximumsstrategie identisch (Maximumsstrategie := es schaltet immer eine maximale Menge aktivierter Transitionen nebenläufig)
- Netze mit Schaltdauer 1 können wie zeitlose Petri-Netze unter einer Schrittsemantik analysiert werden
- Jedes Netz mit endlichen Schaltzeiten kann in ein äquivalentes Netz mit Schaltzeiten 1 transformiert werden
  - Die Transformation kann automatisch erfolgen

Skizze der Transformation (schematisch):



Am Beispiel:



Netze mit Zeitintervallen

Sei $N = (S,T,F,W,m_0)$ ein Petri-Netz,

eft (earliest firing time) und lft (latest firing time) zwei Abbildungen von T nach $\mathbb{N}$, so dass $eft(t) \leq lft(t)$

(wie schon bei konstanten Zeitdauern, können rationale Werte durch Änderung der Zeitskala in natürliche Zahlen definiert werden)

Eine Transition t, die zum Zeitpunkt τ aktiviert wurde,

- kann frühestens zum Zeitpunkt τ + eft(t) schalten
- muss spätestens zum Zeitpunkt τ + lft(t) schalten
- kann im Intervall [τ+eft(t), τ+lft(t)] zu einem beliebigem Zeitpunkt schalten
- muss sofort schalten oder deaktiviert werden, wenn eft(t)=lft(t)

Zustand des Netzes, wie vorher [m, u]

- m ist die Markierung

(im Gegensatz zum vorherigen Modell bleiben Marken auf den Stellen während der Schaltdauer, d.h. Zustand bleibt erhalten, Schalten atomarer Vorgang)

- $u(t) \in \mathbb{N} \cup \{\perp\}$

$u(t) = \perp$ bedeutet, dass t nicht aktiviert ist (Wecker abgestellt)

$u(t) \neq 0 \Rightarrow 0 \leq u(t) \leq lft(t)$

- initiale Markierung $[m_0, u_0]$ mit $u_0[t] = \perp$ falls t in m_0 nicht aktiviert und $u_0(t)=0$ falls t in m_0 aktiviert

- Es existieren potenziell unendlich viele Zustände [m, u], falls in m eine Transition t mit $eft(t)<lft(t)$ aktiviert ist

Zustandsänderungen durch Zeitschritt oder Schalten einer Transition (ähnlich zum Verhalten zeitbehafteter Automaten)

- t schaltbar in [m, u] $\Leftrightarrow \forall s \in S\ W(s,t) \leq M(s) \wedge u(t) \geq eft(t)$
- Schalten einer schaltbaren Transition t ändert die Markierung von [m, u] in [m‘, u‘] mit $m‘(s) = m(s) + W(t,s) – W(s,t)$ für alle $s \in S$ und $u‘(t‘) =$
  - 0 falls $W(\bullet t‘) \leq m‘ \wedge (t=t‘ \vee \neg(W(\bullet t‘) \leq m) \vee \bullet t \cap \bullet t‘ \neq \emptyset)$
  - $u(t‘)$ falls $W(\bullet t‘) \leq m‘ \wedge W(\bullet t‘) \leq m \wedge \bullet t \cap \bullet t‘ = \emptyset$
- Zeitschritt der Länge τ ist in der Markierung [m, u] erlaubt, falls $t \leq \min_t: u(t) \neq \perp \wedge (left(t) - u(t))$ und ändert die Markierung zu [m, u‘] mit $u‘(t) =$
  - $u(t) + t$ falls $u(t) \neq \perp$
  - $\perp$ falls $u(t) = \perp$

Dynamik durch

- Zeitschritte [m, u] [τ> [m, u‘]

Zeitschritt τ muss in [m, u] erlaubt sein

- Schalten einer Transition [m, u] [t> [m‘, u‘] (Transitionen schalten einzeln!)

Transition t muss in [m, u] schaltbar sein

Zustand [m‘, u‘] ist von [m, u] aus erreichbar, falls

- Zeitschritte $\tau_1, \tau_2, \ldots, \tau_X$,
- Transitionen $t_1, t_2, \ldots, t_{X-1}$,
- Zustände $[m_x, u_x]$ und $[m_x, u'_x]$ ($0 \leq x \leq X$) existieren,

so dass

- $[m_1, u_1] = [m, u]$ und $[m_X, u'_X] = [m', u']$,
- $[m_x, u_x]$ [τ_x> $[m_x, u'_x]$ und
- $[m_x, u'_x]$ [t_x> $[m_{x+1}, u_{x+1}]$

Die Schreibweise lautet [m, u] [*> [m‘, u‘]

Einbeziehung der Zeit führt zu unendlichen (sogar überabzählbaren) Erreichbarkeitsmengen ⇒
Aussagen über Netzeigenschaften auf Basis der Markierung m

Einige Eigenschaften

- Ein zeitbehaftetes Netz ist beschränkt, wenn nur endlich viele Markierungen m auftreten
- Ein Transition t heißt lebendig in einem Zustand [m, u], wenn von [m, u] ein Zustand [m‘, u‘] erreichbar ist, in dem t geschaltet werden kann
  andernfalls heißt die Transition tot in [m, u]
- Ein zeitbehaftetes Netz heißt lebendig, wenn kein Zustand [m, u] erreichbar ist, in dem eine Transition t tot ist

[m, u] im zeitbewerteten Netz erreichbar ⇒
m im zugehörigen zeitlosen Netz erreichbar
(Umkehrung gilt nicht!)

Daraus folgt: Wenn das zeitlosen Netz beschränkt ist, so ist auch das zeitbehaftete Netz beschränkt

Aussagen bzgl. der Lebendigkeit übertragen sich nicht!

Wir betrachten nur beschränkte Netze:
Betrachte Klassen von Zuständen, statt einzelner Zustände
[m, R] ist eine Zustandsklasse, wobei

- m eine Markierung ist und
- R eine Region, die durch Vektoren θ charakterisiert ist wobei
  - θ für jede in m aktivierte Transition einen Eintrag enthält
  - erlaubte Vektoren durch eine Ungleichungen $\alpha_i \leq \theta(i) \leq \beta_i$ und $\theta(i) - \theta(j) \leq \chi_{i,j}$ (t_i, t_j in m aktiviert) definiert sind

Transition t_i kann in [m, R] schalten, falls ein Vektor $\theta \in R$ existiert, so dass $\theta(i) \leq \theta(j)$ für all in m aktivierten Transitionen t_j

Ziel: Darstellung des Erreichbarkeitsgraphen/der Erreichbarkeitsmenge durch Zustandsklassen

Initiale Klasse $[m_0, R]$ wobei R charakterisiert ist durch

- Für alle in m_0 aktivierten Transitionen t_i sei $\alpha_i = eft(t_i)$ und $\beta_i = lft(t_i)$
- Für alle in m_0 aktivierten Paare von Transitionen t_i und t_j sei $\chi_{i,j} = lft(t_i)-eft(t_j)$ (redundant!)

Es bleibt nun zu zeigen, wie das Schalten einer Transition zu einem neuen Zustand führt

- Transition t_i schaltet im Zustand [m, R] und führt zu einem neuen Zustand [m‘, R‘]
  - $m'(s) = m(s) - W(s, t_i) + W(t_i,s)$ für alle $s \in S$
  - Durch das Schalten von t_i werden
    - Transitionen aus $T^- \subseteq T$ deaktiviert
    - Transitionen aus $T^+ \subseteq T$ neu aktiviert

Schalten findet zum Zeitpunkt $\theta(i)$ statt!

➢ $\theta(i)$ muss so gewählt werden, dass Werte $\theta(k)$ (t_k in m aktiviert $t_k \neq t_i$) existieren mit $\theta(i) \leq \theta(k)$ und $\theta(k)$ erfüllt die Ungleichungen für t_k

➢ Bestimmung von R‘ in 3 Schritten:

1. Entfernen von $\theta(i)$ aus den Ungleichungen
2. Entfernen von $\theta(k)$ für alle $t_k \in T^-$ aus den Ungleichungen
3. Hinzufügen von $\theta(k)$ für alle $t_k \in T^+$ zu den Ungleichungen

1. Entfernen von $\theta(i)$ aus den Ungleichungen

   für alle in m aktivierten Transitionen $t_j \neq t_i$ und $t_k \neq t_i$:

   - $\alpha_j = \max(0, -\chi_{i,j}, \alpha_j - \beta_i)$
   - $\beta_j = \min(\chi_{j,i}, \beta_j - \alpha_i)$
   - $\chi_{j,k} = \min(\chi_{j,k}, \beta_j - \alpha_k)$

2. Entfernen von $\theta(l)$ $t_l \in T^-$ aus den Ungleichungen für alle in m aktivierten Transitionen $t_j \neq t_l$ und $t_k \neq t_l$ (die noch nicht entfernt wurden)
   - $\alpha_j = \max(\alpha_l - \chi_{l,j}, \alpha_j)$
   - $\beta_j = \min(\beta_j + \chi_{j,l}, \beta_j)$
   - $\chi_{j,k} = \min(\chi_{j,k}, \chi_{j,l} + \chi_{l,k})$
3. Hinzufügen von $\theta(j)$ $t_j \in T^+$ mit neuen Ungleichungen für alle Transition $t_k \neq t_j$, die bereits aktiv sind
   - $\alpha_j = eft(t_j)$
   - $\beta_j = lft(t_j)$
   - $\chi_{j,k} = \beta_j - \alpha_k$ und $\chi_{k,j} = \beta_k - \alpha_j$

Abschließender Schritt: Berechnung einer kanonischen Darstellung

➢Finde die kleinsten Wert für α_j und die größten Werte für β_j und $\chi_{j,k}$, die sich aus den Ungleichungen ergeben

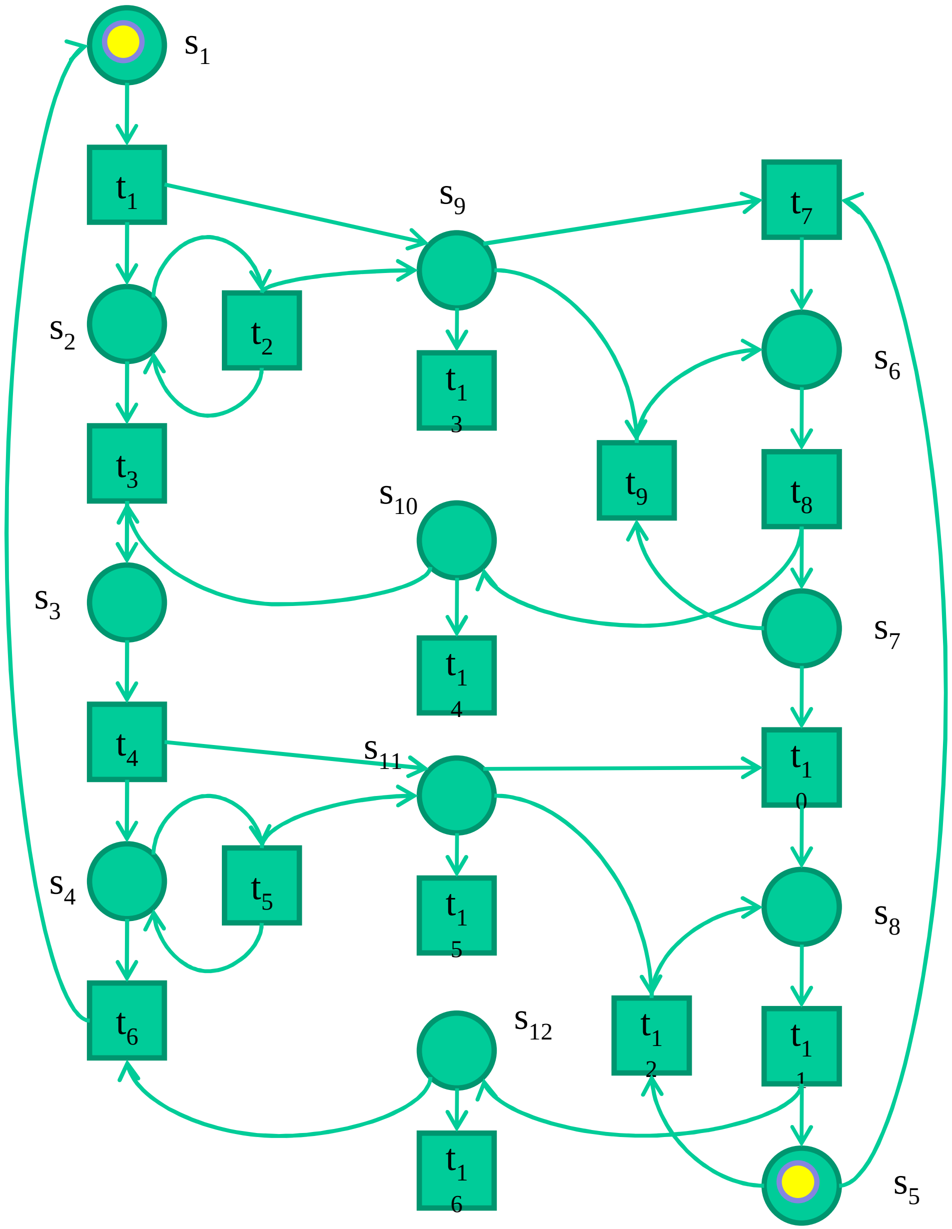
Einige Resultate:

➢Kanonische Darstellung ist eindeutig und erlaubt einen einfachen Vergleich, ob der Zustand bereits erzeugt wurde

➢Falls alle Werte für eft(t) und lft(t) rationale Zahlen sind, so ist die Erreichbarkeitsmenge des zeitbehafteten Netzes endlich, wenn die Erreichbarkeitsmenge des zugehörigen zeitlosen Netzes endlich ist

➢Erreichbarkeitsmenge können sehr groß werden, wenn viele überlappende Intervalle existieren

Beispiel: Alternating Bit Protokoll (abstrakt)



Trans	Intervall	Beschreibung
t_1	$[0,\infty)$	Sende Paket 0
t_2	$[5,6]$	Wiederhole Paket 0
t_3	$[0,1]$	Empfange Ack 0
t_4	$[0,\infty)$	Sende Paket 1
t_5	$[5,6]$	Wiederhole Paket 1
t_6	$[0,1]$	Empfange Ack 1
t_7	$[0,1]$	Empfange Paket 0
t_8	$[0,2]$	Sende Ack 0
t_9	$[0,1]$	Erneut Paket 0
t_{10}	$[0,1]$	Empfange Paket 1
t_{11}	$[0,2]$	Sende Ack 1
t_{12}	$[0,1]$	Erneut Paket 1
t_{13}	$[0,1]$	Verlust Paket 0
t_{14}	$[0,1]$	Verlust Ack 0
t_{15}	$[0,1]$	Verlust Paket 1
t_{16}	$[0,1]$	Verlust Ack 1

Stochastische Petri-Netze

Sei N = (S,T,F,W,m_0) ein Petri-Netz,

Λ eine Abbildung von T nach $\mathbb{R}_+$, die zu jeder Transition eine Schaltrate angibt

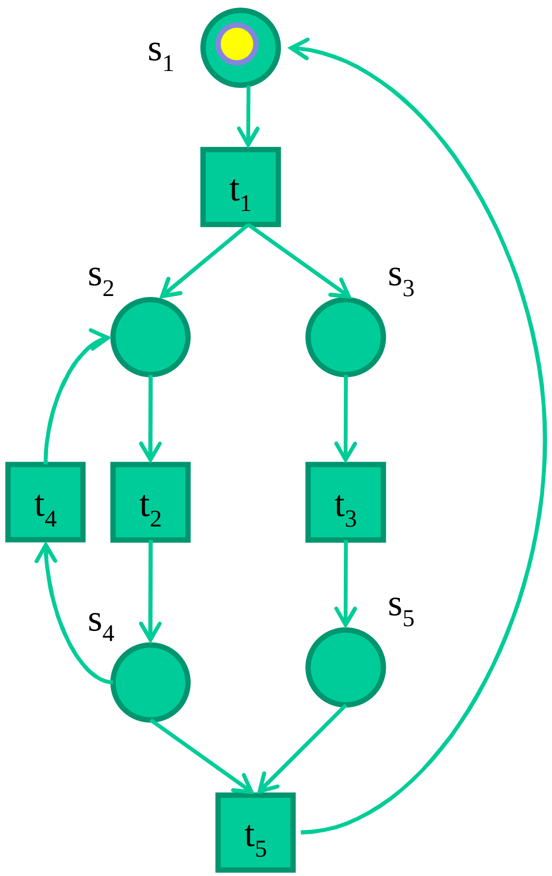
Wenn eine Transition aktiviert ist,

- so schaltet nach einer exponentiell verteilten Zeit mit Rate $\Lambda(t)$, sofern sie nicht vorher deaktiviert wird
- Es schaltet immer die Transition, mit der kürzesten Schaltzeit (Race-Condition)

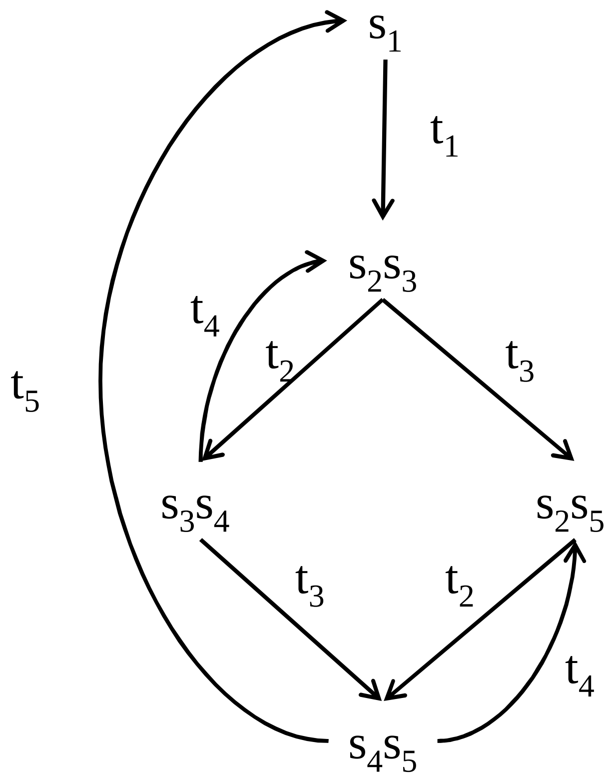
Implikationen aus der Nutzung von Exponentialverteilungen,

- Wenn zwei Transitionen nebenläufig aktiviert sind, so kann jede von beiden die kürzere Zeit haben ($\Rightarrow$ Erreichbarkeitsmenge des zeitlosen und stochastischen Netzes sind identisch)
- Es spielt keine Rolle, ob nach einem Schaltvorgang alle Transitionen eine neue Schaltzeit ermitteln oder ihre Restzeit beibehalten
- Das Netz beschreibt einen zeitkontinuierlichen Markov-Prozess (siehe Q5)

Beispiel



s_1
t_1
s_2
s_3
t_4
t_2
t_3
s_4
s_5
t_5



s_1
t_1
s_2s_3
t_4
t_2
t_3
t_5
s_3s_4
s_2s_5
t_3
t_2
t_4
s_4s_5

Q3.4 Weitere Modelltypen

Viele weitere Modelltypen existieren zur Beschreibung von zeitbehafteten Systemen, z.B.

- Simulationssprachen (siehe MAO)
- Warteschlangennetze (siehe Kap. Q 7/8)
- Zeitbehaftete Prozessalgebren (kurzes Beispiel folgt)
- …

übliche Semantik bewertetes Transitionssystem und/oder stochastischer Prozess

Prozessalgebren spielen ein große Rolle für zeitlose Modelle (CCS, CSP, …)

Erweiterungen von Prozessalgebren wie CCS um Zeitbegriff existieren, sind aber nicht trivial, da die Zeit formalisiert werden muss
Probleme bei parallelen Abläufen

Definition einer „vernünftigen“ Semantik erfordert einige Überlegungen

Viele Ansätze existieren, viele weisen Schwächen/Unzulänglichkeiten auf (insgesamt ein weites Feld)

Wir betrachten hier:

- Performance Evaluation Process Algebra (PEPA)
  Hillston 1996
  stochastische (exponentielle) Zeiten, zur Leistungsanalyse
  Probleme werden durch exponentielle Zeiten weitgehend umgangen
  (siehe Kap. Q5)

Syntax (Erweiterung von CCS um Transitionsraten)

P, Q Prozessterm und Act(P) eine Aktivitätsmenge, die P ausführen kann

$P :== (a, r).P \mid P + Q \mid P \parallel_L Q \mid P/L \mid A$

- *Präfix (a,r).P*: Führt die Aktion vom Typ a aus, die Ausführung beansprucht eine exponentiell verteilte Zeit mit Rate r
- *Choice P + Q*: Führt die Aktivitäten P und Q parallel aus, diejenige, die zuerst fertig wird, bestimmt den Fortgang (Race Condition)
- *Cooperation* $P \parallel_L Q$: Aktionen aus der Menge L werden synchron in P und Q ausgeführt, andere Aktionen lokal und parallel
- *Hiding P/L*: Die Aktionen aus L werden mit einem Label ε versehen, dass nicht sichtbar ist und nicht in Kooperationen verwendet werden kann
- Constant A = P: A verhält sich wie P
  (notwendig um unendliches Verhalten zu beschreiben)

Klammern werden benutzt um Prioritäten zu setzen

z.B. $P \parallel_L Q \parallel_K R$ legt nicht fest, wie sich R bzgl. Aktionen aus L verhält

$(P \parallel_L Q) \parallel_K R$ definiert dies eindeutig

Unterscheidung in Aktionen und Aktivitäten:

Beobachtung von Prozesstermen über die ausgeführten Aktionen
Aktivitäten werden aus einer Menge Act $\subseteq A \times \mathbb{R}+ \cup \{\top\}$ gewählt

- A ist ein endliches Alphabet, welches das spezielle Symbol ε enthält und Aktionstypen beschreibt
    - Aktionsmengen $L,K \subseteq A \setminus \{\varepsilon\}$
- Aktivitäten setzen sich aus einer Aktion und einer Rate bzw. dem Symbol $\top$ zusammen
    - Das Symbol $\top$ beschreibt eine passive Aktion (Erläuterung später)
    - Schreibweise: $\alpha = (a,r) \in$ Act

Act(P) ist eine Multimenge über Act

Für jedes $\alpha \in$ Act(P) gibt es einen Prozessterm Q, so dass in P Aktivität α ausgeführt wird und der Prozess sich wie Q verhält

Schreibweise: $P \xrightarrow{\alpha} Q$ (Analogie zu bewerteten Transitionssystemen!!)

Für einen Prozessterm P ist A(P) die Menge der Aktionen, die P ausführen kann

Diese ist über die Terme definiert:

- $A((a,r).P) = \{a\}$
- $A(P+Q) = A(P) \cup A(Q)$
- $A(P \,\|_L\, Q) = \{A(P) \setminus L\} \cup \{A(Q\} \setminus L\} \cup \{A(P) \cap A(Q) \cap L\}$
- $A(P / L) = A(P) \setminus L$
- $A(Q = P) = A(P)$

Bestimmung der Gewichte $r_a(P)$:

- $r_a((b,r).P) = r$ falls $a=b$ und 0 sonst
- $r_a(P + Q) = r_a(P) + r_a(Q)$
- $r_a(P \,\|_L\, Q) = r_a(P) + r_a(Q)$ falls $a \notin L$ und $\min(r_a(P), r_a(Q))$ sonst
- $r_a(P \setminus L) = r_a(P)$ falls $a \notin L$ und 0 sonst

Es gilt:

- $r < w\top$ für $r \in \mathbb{R}$, $w \in \mathbb{N}$ (dadurch ist $\min(r, w\top) = r$)
- $w\top < v\top$ für $w<v$
- $w\top + v\top = (w+v)\top$
- $(w\top)/(v\top) = w/v$

Operationale Semantik für PEPA:

Interpretation, wenn die Bedingung (oberhalb) gilt, kann die Ableitung(unterhalb) ausgeführt werden

Präfix

$$\frac{}{(a,r).P \stackrel{(a,r)}{\longrightarrow} P}$$

Choice

$$\frac{P \stackrel{(a,r)}{\longrightarrow} P'}{P + Q \stackrel{(a,r)}{\longrightarrow} P'} \qquad \frac{Q \stackrel{(a,r)}{\longrightarrow} Q'}{P + Q \stackrel{(a,r)}{\longrightarrow} Q'}$$

Cooperation

$$\frac{P \stackrel{(a,r)}{\longrightarrow} P'}{P \|_L Q \stackrel{(a,r)}{\longrightarrow} P' \|_L Q} (a \notin L) \qquad \frac{Q \stackrel{(a,r)}{\longrightarrow} Q'}{P \|_L Q \stackrel{(a,r)}{\longrightarrow} P \|_L Q'} (a \notin L)$$

$$\frac{P \stackrel{(a,r_1)}{\longrightarrow} P' \wedge Q \stackrel{(a,r_2)}{\longrightarrow} Q'}{P \|_L Q \stackrel{(a,r_3)}{\longrightarrow} P' \|_L Q'} (a \in L) \quad \text{where } r_3 = \frac{r_1}{r_a(P)} \frac{r_2}{r_a(Q)} \min\left(r_a(P), r_a(Q)\right)$$

Constant (Q= P)

$$\frac{P \xrightarrow{(a,r)} P'}{Q \xrightarrow{(a,r)} P'}$$

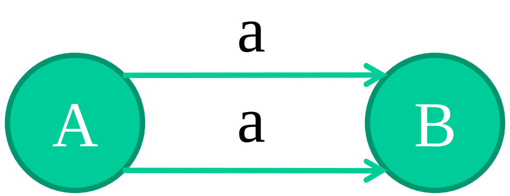
Hiding

$$\frac{P \xrightarrow{(a,r)} P'}{P/L \xrightarrow{(a,r)} P'/L}(a \notin L) \qquad \frac{P \xrightarrow{(a,r)} P'}{P/L \xrightarrow{(\epsilon,r)} P'/L}(a \in L)$$

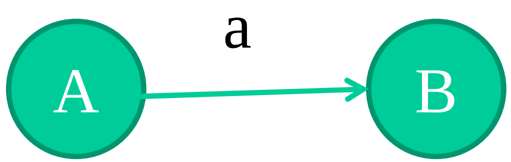
Regeln erlauben Aufbau eines bewerteten (endlichen) Transitionssystems

- durch rekursive Anwendung der Ableitungen
- Bewertung der Kanten mit Aktivitäten (Aktion + Rate)
- unendliche Transitionssysteme durch Erweiterung der Prozessalgebra (hier nicht vorgestellt)

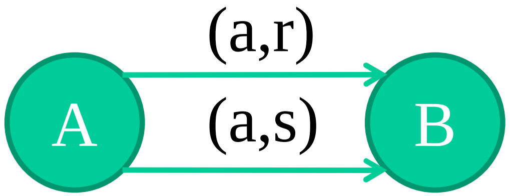
Für zeitlose Prozessalgebren



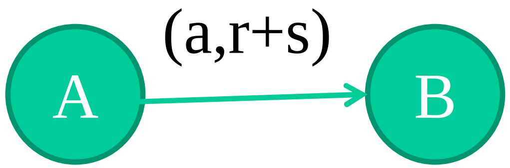
=



Für stochastische Prozessalgebren

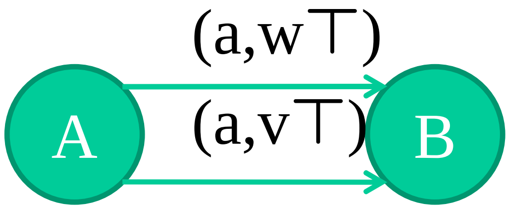


=

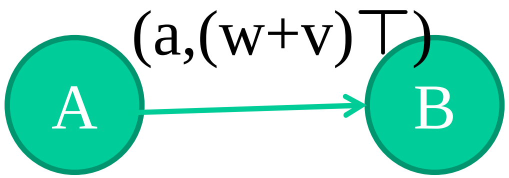


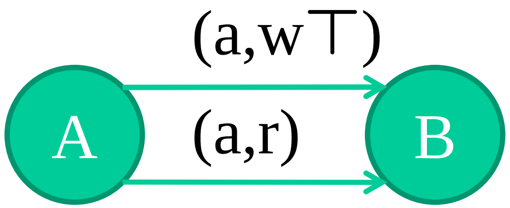
$r,s \in \mathbb{R}$

$w,v \in \mathbb{N}$



=





kann nicht zusammengefasst werden

Ein Beispiel:

Process = (use, r_1).(task, r_2).Process

Resource = (use,r_3).(update,r_4).Resource

System = Process $\|_{\{use\}}$ Resource

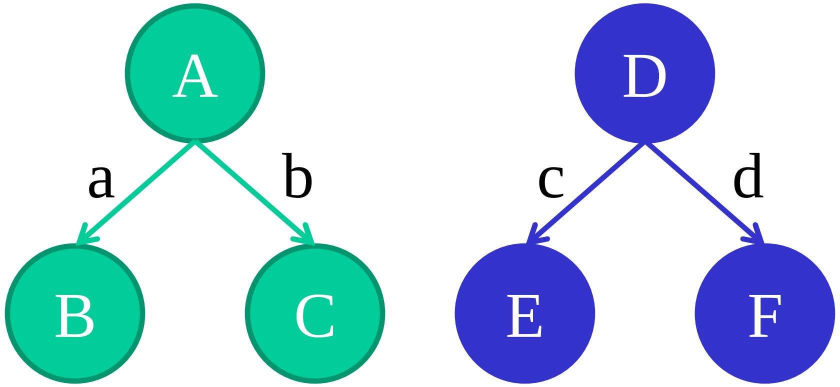
Resultierende bewertetes Transitionssystem:

Interpretation als Markov-Prozess später!

Prozessalgebra ⇒ Rechnen mit Prozessen (siehe CCS)

Stochastische Prozessalgebra ⇒ Rechnen mit stochastischen Prozessen (hier nur in Ansätzen erläutert)

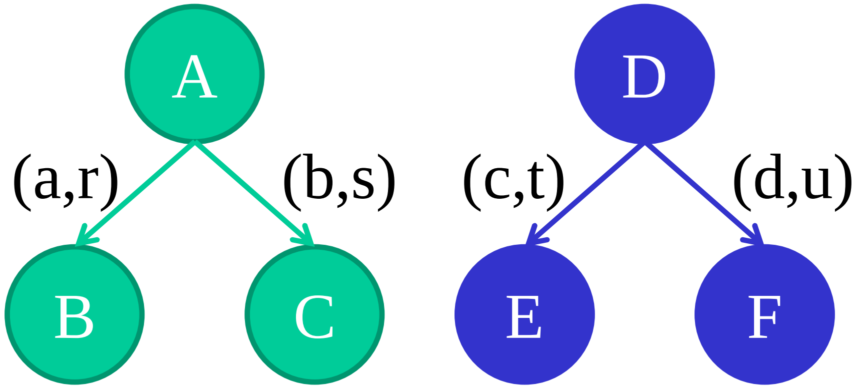
Zur Erinnerung: Bisimulation ≈



A≈D, falls

- a=c, b=d, B≈E und C≈F
- a=d, b=c, B≈F und C≈E

Stochastische Bisimulation ~



A ~ D, falls

- a=c, b=d, r=t, s=u, B~E und C~F
- a=d, b=c, r=u, s=t, B~F und C~E

„Rechenregeln“ für stochastische Prozessalgebren:

Sei $P_1 \sim P_2$, dann

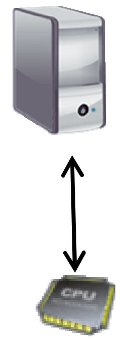
- $(a,r).\ P_1 \sim (a,r).\ P_2$
- $P_1 + Q \sim P_2 + Q$
- $Q + P_1 \sim Q + P_2$
- $P + Q \sim Q + P$
- $P_1 \|_L Q \sim P_2 \|_L Q$
- $Q \|_L P_1 \sim Q \|_L P_2$
- $P \|_L Q \sim Q \|_L P$
- $P_1/L \sim P_2/L$

Q3.5 Last-Maschine Modellierung

Übliche Sichtweise:	u.U. hierarchisch, d.h. Maschine
Last	generiert Last für weitere
Maschine	Maschinen

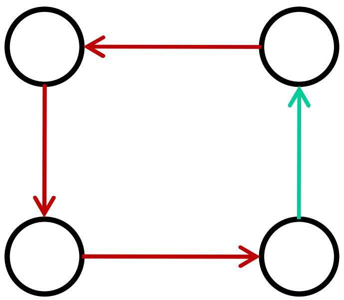
Typische Anforderungen:

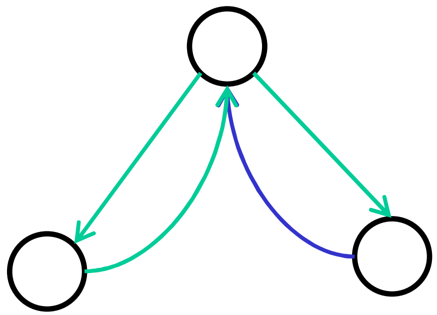
- ➢Hierarchische Beschreibung
- ➢Heterogene Beschreibung
- ➢Kompositionelle Beschreibung

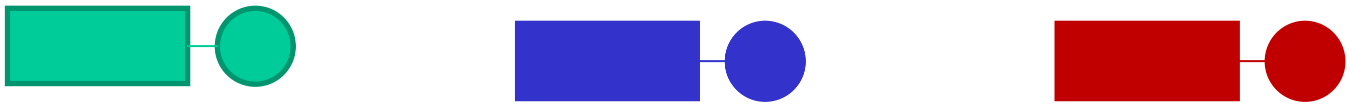


Sequentielle Prozesse nutzen Ressourcen $\Rightarrow$ Warteschlangennetze

Last: Automaten/Prozessketten/Petri-Netze







Ressourcen: Stationen in Warteschlangennetzen

Prozessketten (evtl. Abbildung auf SPN oder Warteschlangennetze)