Zeitbewertete Netzmodelle

Dr. Bernd Baumgarten
Fraunhofer SIT, Darmstadt
bernd.baumgarten@sit.fraunhofer.de
http://private.sit.fraunhofer.de/~baumgart/

Kolloquiumsvortrag
am Institut für Informatik
Brandenburgische TU Cottbus
14.12.2004
(Überarbeitete Version vom 22.12.04)

Ich danke Frau Prof. Dr. Monika Heiner und ihren Mitarbeitern für ihre Verbesserungsvorschläge.

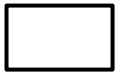
Übersicht

Petri-Netze, Stellen-Transitions-Systeme

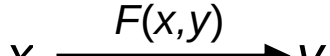
Ein **Stellen-Transitions-** (bzw. **ST-**) **System**, ist ein Tupel $Sys = (P, T, F, M_0)$ mit

- ***P*** endliche Menge von **Stellen**,

- ***T*** endliche Menge von **Transitionen**, disjunkt zu P,



- ***F***: $(P \times T) \cup (T \times P) \to \mathbb{N}_0$ ist die **Fluss-Funktion**,



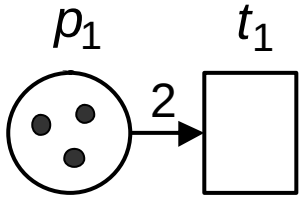
(**Kanten**, **Kantengewichte**)

Zeichenkonventionen:

,

- ***M₀*** : $P \to \mathbb{N}_0$ ist die **Anfangsmarkierung**.

Beispiel:



$P = \{p_1\}, T = \{t_1\}$
$F = \{ ((p_1, t_1), 2), ((t_1, p_1), 0) \}$
$M_0 = \{ (p_1, 3) \}$

Schaltungen – ein Beispiel

Was ist modelliert?

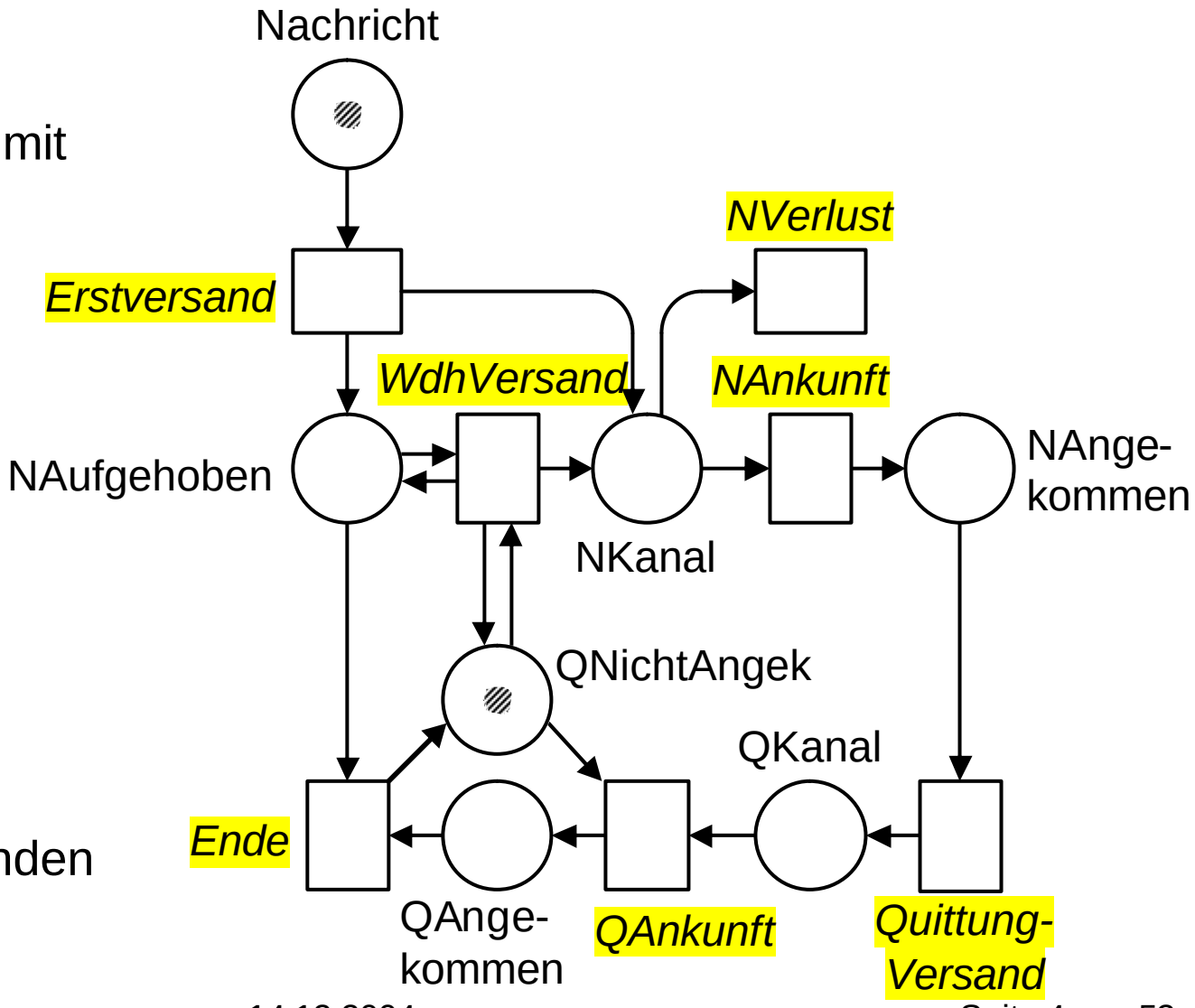
Versand **einer** Nachricht mit

- Nachrichtenkanal mit Verlust
- Quittierung ohne Verlust
- Wiederholung bei ausbleibender Quittung

Aufgabe:

Zeitwissen ➔

Unnötiges Warten, Wiederversenden **vermeiden**!



Ziele der Zeitbewertung

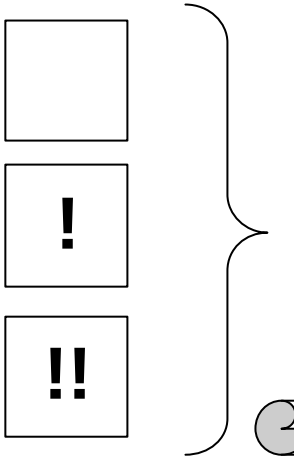
Zeitliche Anforderungen an die Transitionsschaltungen
in Stellen-Transitions-Systemen spezifizieren!

1. Grenzen bezüglich der früheren Schaltzeitpunkte **anderer Transitionen**,
2. Grenzen bezüglich der früheren Schaltzeitpunkte **derselben Transition**,
3. Grenzen bezüglich der der **wahren Uhrzeit bzw. absoluten Zeit**,
4. oder auch: **schnellstmöglich, unverzüglich**,
5. Grenzen auch für **Differenzen** zwischen abgelaufenen Zeiten.

Transitionstypen

Es zeigt sich, dass man drei Arten von Transitionen gebrauchen kann:

- KANN-Transitionen
- MUSS-Transitionen
- ASAP-Transitionen

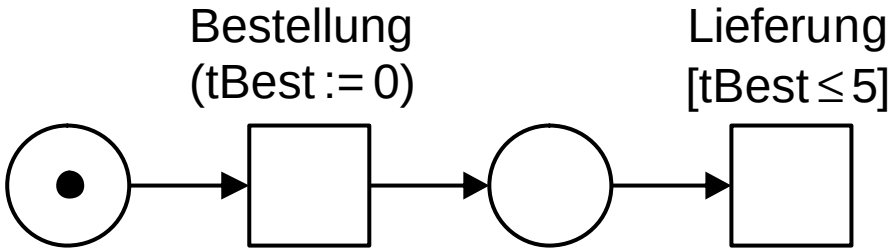


Alle ohne Zeitverbrauch! **Zeit** vergeht nur **zwischen** Transitions-schaltungen!

Wir führen in intuitiv verständlichen Beispielen **parallel**

- sowohl die verschiedenen **Zeitbezüge**
- als auch die **Transitionstypen** ein.

KANN Transitionen +
Grenzen bezüglich der früheren Schaltzeitpunkte anderer Transitionen



Lieferfrist (und autom. Storno bei Überschreitung)

Informelle Beschreibung:
Lieferung nur bis zu 5 Zeiteinheiten nach Bestellung – ansonsten gar nicht.

Lieferung ist eine **optionale**, eine **KANN-Transition**.

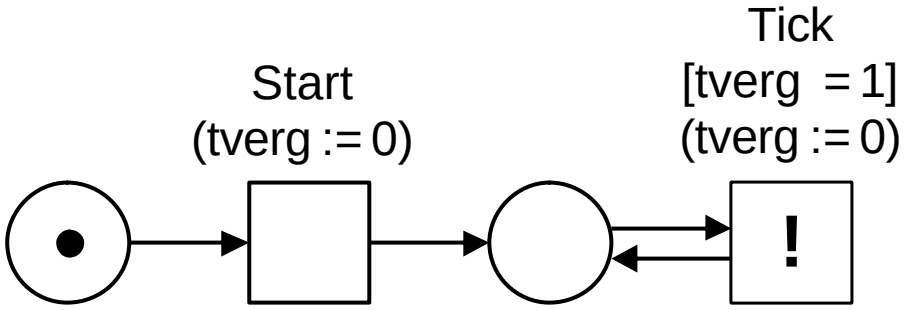
Anmerkungen:

tx	=	Zeit auf der Stoppuhr (dem Timer) tx
( )		umschließt Timerstarts.
[ ]		umschließt Zeitbedingungen.

Weitere Beispiele: Öffnungszeiten, gesetzliche Kann-Fristen, usw.

MUSS-Transitionen +

Grenzen bezüglich der früheren Schaltzeitpunkte derselben Transition

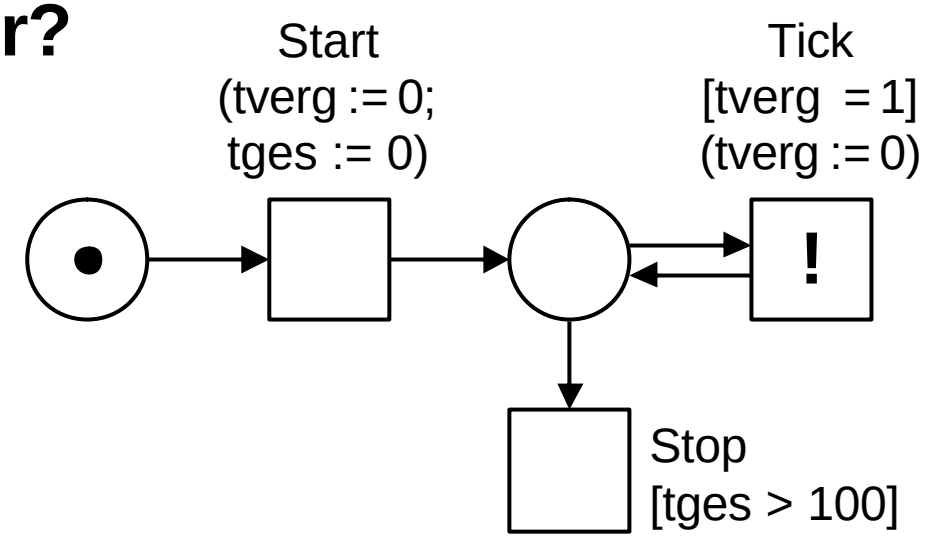


Informelle Beschreibung:
Sie tickt garantiert nach jeder Zeiteinheit, erstmals eine Zeiteinheit nach ihrem Start. Tick ist eine **notwendige**, eine **MUSS-Transition.**

Anmerkungen: **!** MUSS-Transition
Sie muss im angegebenen Zeitraum schalten, wenn sie bei dessen Ablauf aktiviert sind.

Nur ein zeitliches MUSS ist ein wirksames MUSS.

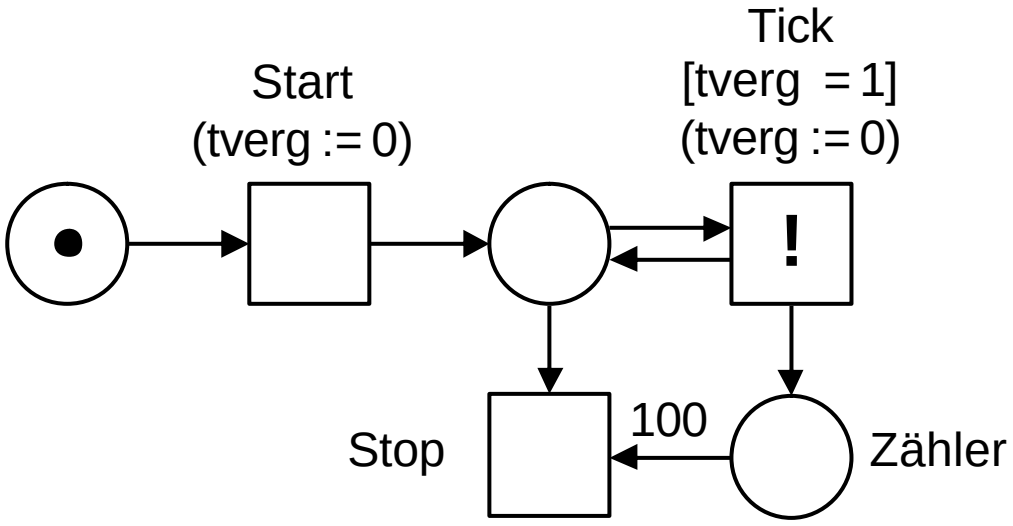
Zeiten ←→ Zähler?



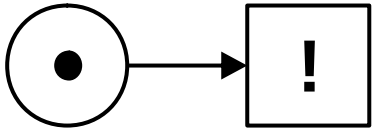
Erweiterte Standuhr:

darf stehbleiben,

muss aber mindestens 100 mal ticken.



Absolute Zeit +
Grenzen bezüglich der Weltzeit bzw. absoluten Zeit



StarteFeuerwerk
[tAbs=1.1.2005,0:00h]

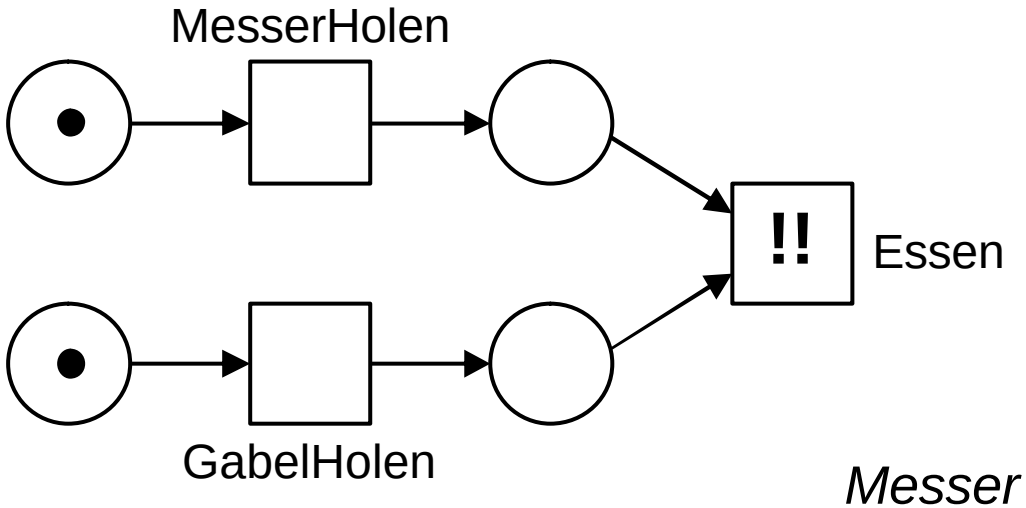
Neujahrsfeuerwerk

Informelle Beschreibung:
Das Neujahrsfeuerwerk soll am 1.1.2005 um Mitternacht beginnen.

Anmerkungen: **tAbs** = absolute Weltzeit (z.B. GMT)
StarteFeuerwerk: MUSS-Transition.

ASAP-Transitionen

schnellstmöglich, unverzüglich, **A**s **S**oon **A**s **P**ossible

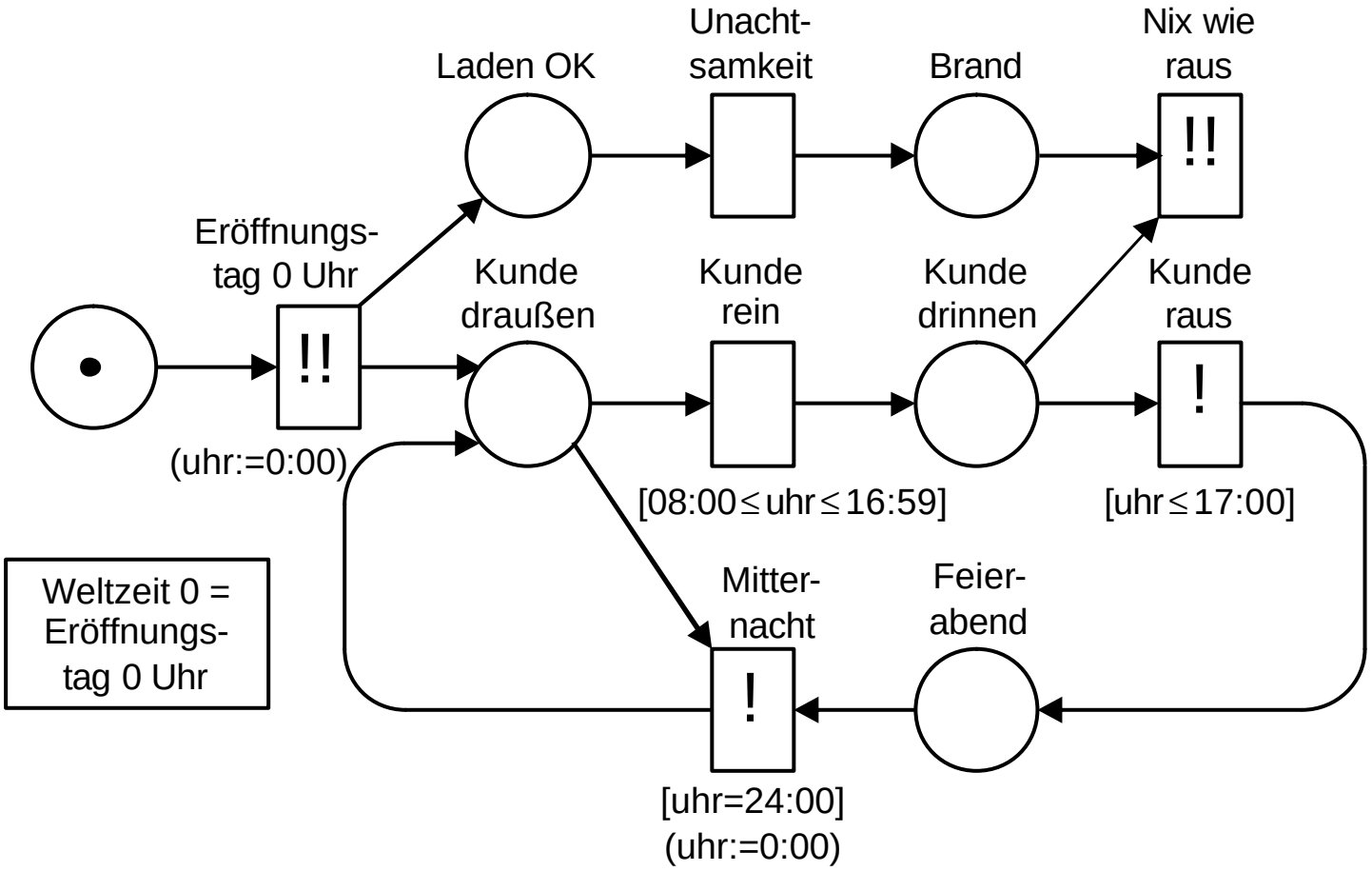


Messer und Gabel (und Hunger)

Informelle Beschreibung:
Essen geschieht, sobald Messer und Gabel geholt sind.

Anmerkungen: **!!** **ASAP-Transition**
Sie schaltet, sobald sie „**markenaktiviert**“ ist **und** ihre **Zeitbedingungen** (hier hat sie keine) erfüllt sind.

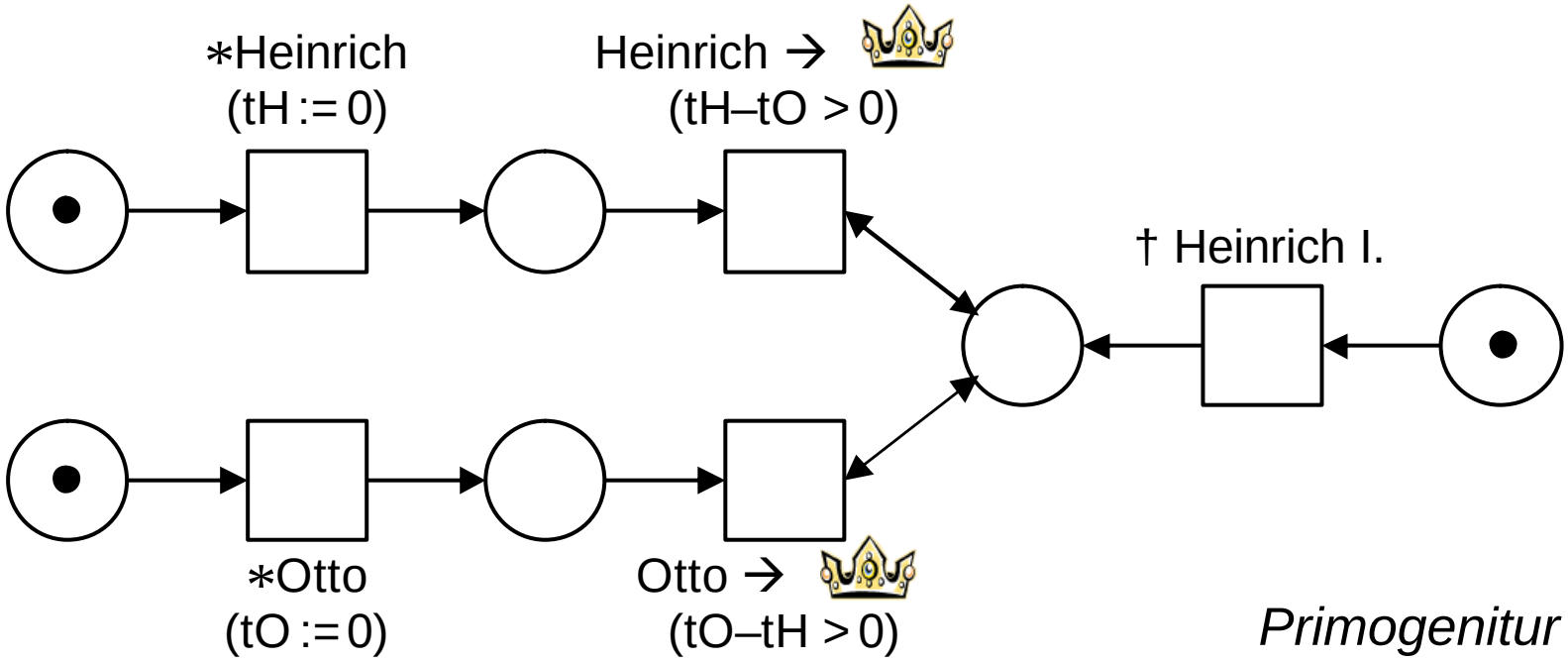
KANN + MUSS + ASAP – ein Beispiel



„Öffnungszeiten tägl. 8-17 Uhr“ (legales Kundenverhalten)

Timerdifferenzen

Grenzen auch für **Differenzen** zwischen abgelaufenen Zeiten



Informelle Beschreibung:
Des Königs Heinrich I. („der Vogler“) erstgeborener Sohn war nach des Vaters Tod erster Thronanwärter. Aus Prinz Otto wurde später Otto der Große, aus Prinz Heinrich wurde Heinrich der Zänker.

Diskrete Systeme

Empirismus
(Pragmatismus, Positivismus,
Materialismus, Behaviourismus,
Realismus, ...)

→

Ein System „ist“
was man davon
– direkt oder indirekt –
beobachten
kann – und will.

Anwendungs-Beispiel:

Gibt es diskrete Systeme?

Gibt es kontinuierliche Systeme?

→

Systeme
(diskret oder kontinuierlich)
können wir per Entschluss
+ Abstraktionsfähigkeit
als **diskret beobacht- und**
beeinflussbar ansehen.

philosophisch-physikalische Diskussionen!

Diskret beobachtete Systeme

DORTS = **D**iscretely **O**bserved **R**eal **T**ime **S**ystem *sys*

natürliche DORTS-Semantik: (praktisch identisch!)

”

Menge *Obs*(*sys*) möglicher **Beobachtungen** (= Beob.-protokolle)

$$obs = \big((a, t_0), (v_1, t_1), (v_2, t_2), \ldots, (v_n, t_n), (w, t_{n+1})\big)$$

$n = 0, 1, \ldots$ **endliche** Folge

$v_k \in Events$ im Kontext **relevante sichtbare atomare Ereignisse** (Datenobjekte) –

vom Beobachter, vom System oder gemeinsam initiiert

$t_k \in \mathbb{R}$, $t_k \le t_{k+1}$ **Momente** der atomaren Ereignisse in abs. Zeit

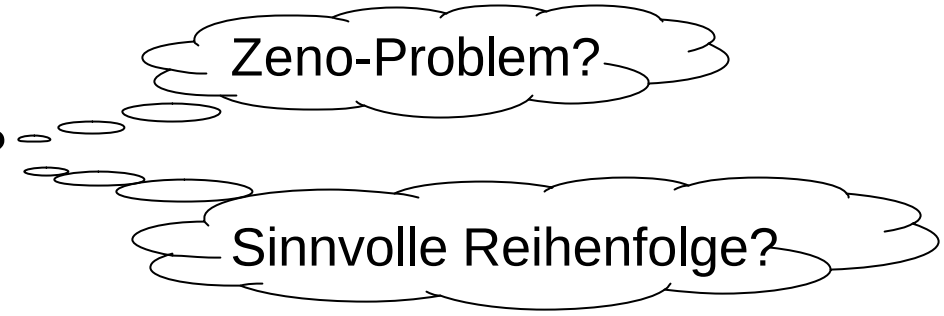
a, w **Beginn** bzw. **Ende** der Beobachtung (**optional**)

Implementation = sys_1
Spezifikation = sys_2

Konformität: $Obs(sys_1) \subseteq Obs(sys_2)$

DORTS – Frage und Vereinfachungen

- Warum nur $t_k \leq t_{k+1}$ und nicht $t_k < t_{k+1}$?



- **Vereinfachung** #1: $t_k \in \mathbb{N}$ (Zeiteinheit = Tick–Tick)

  Vereinfachung #2: $t_k < a,\ a \in \mathbb{R} \rightarrow t_k \leq b,\ b \in \mathbb{N}$

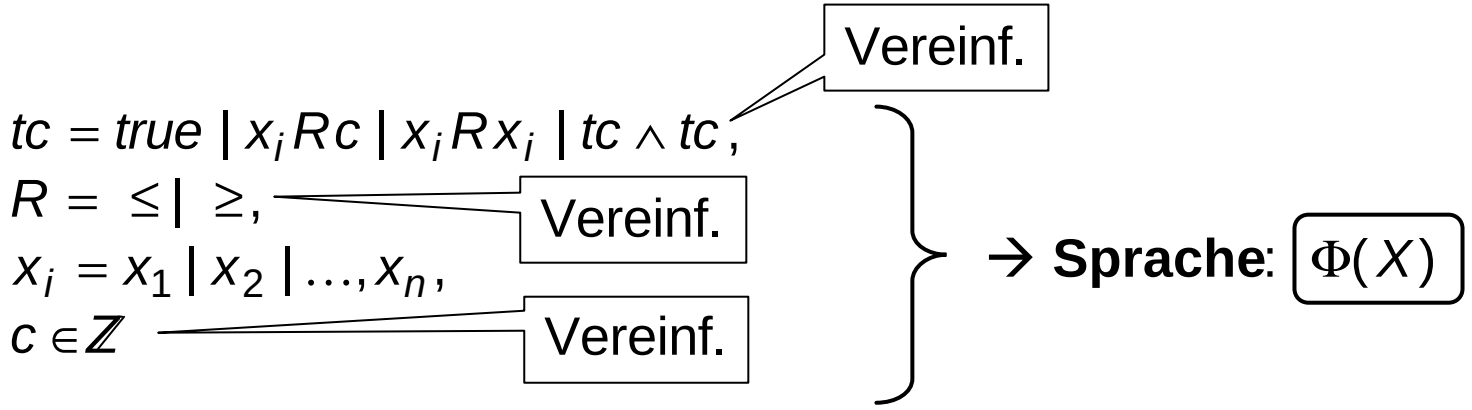
  Vereinfachung #3: $a, w \in Events \rightarrow obs = ((v_1, t_1), (v_2, t_2), \ldots, (v_n, t_n))$

explizit hineinmodellieren!

Timerbedingungen

$X = \{x_1, x_2, \ldots, x_n\}$ endliche Menge von “Timer-(Uhren-)Namen”.

Timerbedingung-Grammatik:



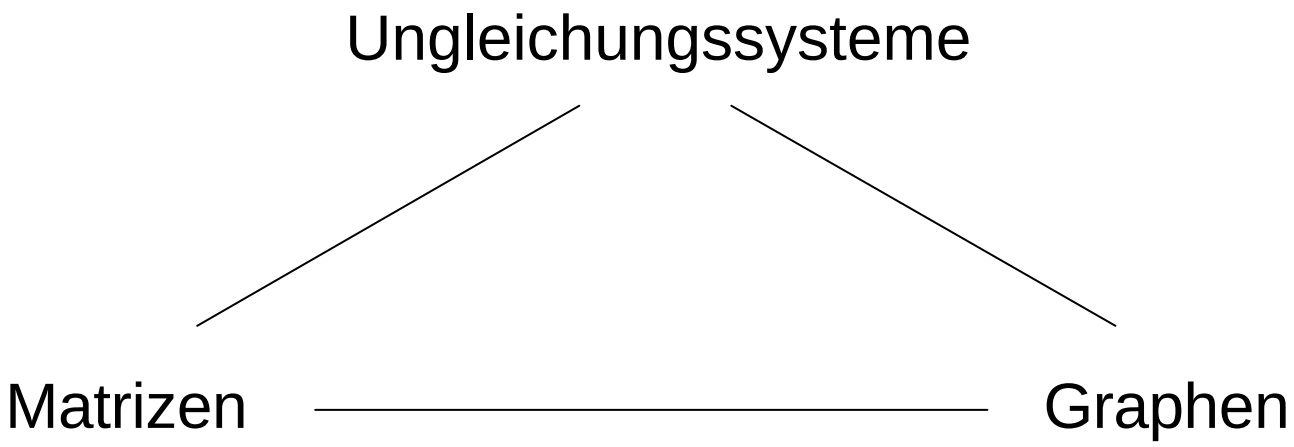
Technischer Einschub:
Umgang mit Timerbedingungen

Fragen zu Timerbedingungen

Ziel: Wann sind 2 tc's **äquivalent**?
Wann ist eine tc **unerfüllbar**?

Weg: Mit abgespeckter linearer Optimierung??

Wegen spezieller Form der Timerbedingungen ...



Vom Ungleichungssystem zur Matrix

***tc* sei Timerbedingung** für $X = \{x_1, \ldots, x_n\}$.

1. Ergänze: $0 \le x_k - x_k \le 0 \qquad (1 \le k \le n)$, **2.** Forme um:

$a \le x_m - x_l \le b, \quad l > m$

$-\infty \le x_i - x_k \le \infty \quad (1 \le i,k \le n).$ ➔ $-b \le x_l - x_m \le -a$

2. Bilde { Maximum aller Untergrenzen / Minimum aller Obergrenzen } von { x_k / $x_i - x_k$,$1 \le k \le i \le n$ }

3. ➔ Intervallform:

$$\forall 1 \le k \le n: \quad -D_{0k} \le x_k \le D_{k0}$$
$$\forall 1 \le k \le i \le n: \quad -D_{ki} \le x_i - x_k \le D_{ik}$$

Normal-form(en)

bzw. ... mit $\boxed{x_0 \equiv 0}$

➔ **Obergrenzenform**,: $\boxed{\forall 1 \le k,i \le n : x_i - x_k \le D_{ik}}$

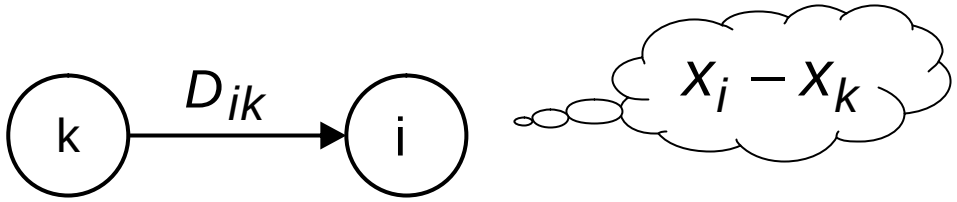
bzw. ...

4. ➔ **Difference Bound Matrix**, **DBM**, für *tc*

$$D(tc) = (D_{ik})\ i,k = 0, ..., n$$

Von der Matrix zum Kostengraphen

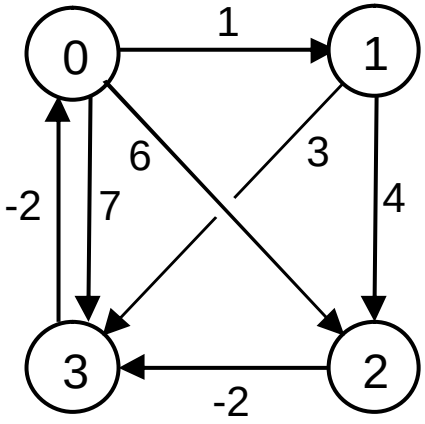
Schema:



als Default weglassbar:

Beispiel:

$$\begin{aligned} x_1 &\le 1 \\ x_2 &\le 6 \\ 2 &\le x_3 \le 7 \\ x_3 - x_1 &\le 3 \\ x_2 - x_1 &\le 4 \\ 2 &\le x_2 - x_3 \end{aligned} \quad \Rightarrow \quad \begin{aligned} x_0 - x_0 &\le 0 \\ x_0 - x_1 &\le \infty \\ &\vdots \\ x_3 - x_1 &\le 3 \\ x_3 - x_2 &\le -2 \\ x_3 - x_3 &\le 0 \end{aligned} \quad \Leftrightarrow \quad \begin{pmatrix} 0 & \infty & \infty & -2 \\ 1 & 0 & \infty & \infty \\ 6 & 4 & 0 & \infty \\ 7 & 3 & -2 & 0 \end{pmatrix} \quad \Leftrightarrow$$



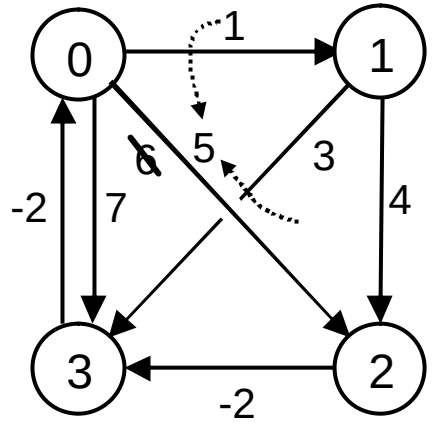
Kanonische Darstellungen von Timerbedingungen

Manche Ungleichungen implizieren andere: $(x_2 - x_1 \leq 4) \wedge (x_1 \leq 1) \Rightarrow (x_2 \leq 5)$

➔ In Timerbedingung: evtl. verschärfe Ungleichungen (gleiche Systemlösungen!)

In DBM: $D_{20} \leq D_{21} + D_{10}$

Im Graphen: „billigere“ Wege.



Schärfste Ungleichungen in *tc*

➔➔ $\forall 0 \leq i,j,k \leq n : D_{ik} \leq D_{ij} + D_{jk}$ in $D(tc)$

➔➔ **Minimalkostengraph**: $G(tc)$

} „**kanonisch**“ „**K**(…)“

tc_1, tc_2 sind **lösungsäquivalent** $\Leftrightarrow$ $K(D(tc_1)) = K(D(tc_2))$

Timerbedingung *tc* ist **lösbar**

$\Leftrightarrow$ $K(D(tc_1))$ enthält kein $-\infty$

$\Leftrightarrow$ $G(tc)$ enthält keinen „Gewinnzyklus“

Floyd-Warshall-Algorithmus

berechnet ➔

Minimalkostengraphen zu gegebenem Digraphen mit reellwertigen Kantenkosten,

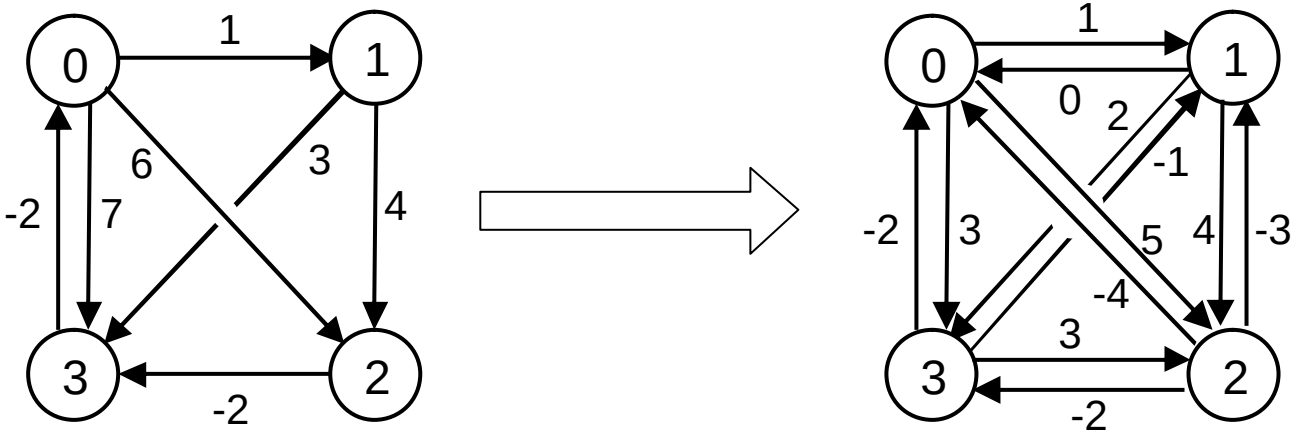
also auch

kanonische DBM und

kanonische Normalform einer Timerbedingung.

```
FOR j:=1,…,n
  FOR i:=1,…,n
    FOR k:=1,…,n
      D_ik^(j) := min(D_ik^(j-1), D_ij^(j-1) + D_jk^(j-1))
```

Im Beispiel:



Timernetze (Baumgarten 1990 +2005?)

Ein **Timernetz** ist ein Tupel $N = (P, T, F, M_0, Type, X, C, St)$ mit

- ST-System (P, T, F, M_0);
- $Type : T \rightarrow$ {KANN, MUSS, ASAP}; mit nichts,! oder !! drin
- $X = \{x_1, x_2, \ldots, x_n\}$ ist eine Menge $\neq \emptyset$ von Timer(Uhren)namen; $x_1 \approx$ **absolute Zeit**
- $C : T \rightarrow \Phi(X)$; [*Timerbedingung*]
- $St : T \rightarrow$ $(\{x_2, \ldots, x_n\})$ (*Timerstarts*)

Praktische Anwendung → **globale Parameter**

Zeiteinheit,
z.B., msec, Sekunden, Tage;
Bezugszeitpunkt von x_1,
z.B. $x_1 = 0 \approx$ 1. Jan. 2000, 0:00 h MEZ,
früh genug dass $x_1 \geq 0$.

Zur natürlichen Semantik von Timernetzen

Uhrenstand (timer reading): $tr : X \to \mathbb{N}_0$

abgeleitete Uhrenstände: $tr_{Y:=0}(x_k) := \text{IF } x_k \in Y \text{ THEN 0 ELSE } tr(x_k)$

$[tr + x](x_k) := tr(x_k) + x$

Zeitmarkierung (timed marking): Paar (M, tr)

Markierung M + Timer reading *tr.*

Menge: $TM(sys)$.

Zeitaktivierung (temporal activation): t ist zeitaktiviert unter (M, tr):

t ist (marken)aktiviert unter M +

tr erfüllt Timerbedingung $C(t)$.

Zeitschaltung (temporal occurence): Paar (t, x)

Transition t + Weltzeitwert $x \in \mathbb{N}_0$.

Pause (passage of time): Paar (d, x)

Symbol $d \notin T$ + Weltzeitwert $x \in \mathbb{N}_0$.

Chroniken und Zeitschaltfolgen

Chroniken (**histories**) und deren **erreichte Zeitmarkierungen**

Induktionsbasis

e ist eine Chronik von N. Sie führt zu $TM(e) = (M_0, X := 0)$.

Induktionsschritt

Sei h Chronik und $TM(h) = (M, tr)$.

Fehlererkennung?

Schaltung: Ist t zeitaktiviert unter (M, tr)

$\Rightarrow$ $h \circ (t, tr(x_1))$ ist Chronik und $TM(h \circ (t, tr(x_1))) := (Mt, tr_{St(t):=0})$

Pause: Wird zwischen $tr(x_1)$ und $x > tr(x_1)$ keine MUSS- oder ASAP-Transition durch Zeitablauf zeitlich deaktiviert

$\Rightarrow$ $h \circ (d, x)$ ist Chronik und

$TM(h \circ (d, x)) := (Mt, tr + x - tr(x_1))$.

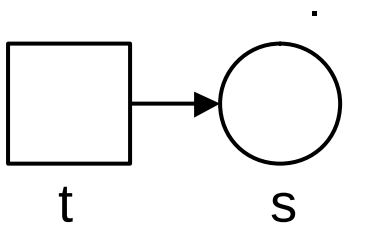
Deadline-Paradoxon!

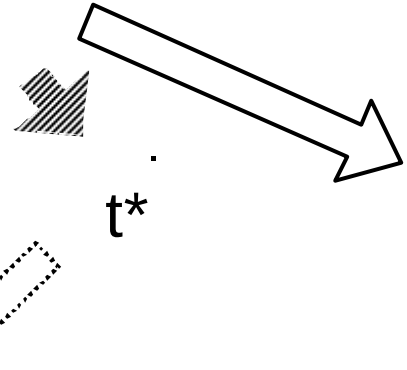
Zeitschaltfolge (Beobachtung, observation) von N:
eine Chronik von N, aus der
alle Pausen weggestrichen worden sind.

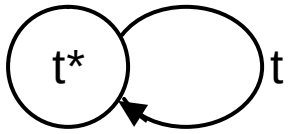
auch für etikettierte Timernetze?

„Akzeptoren“ für zeitfreie Beobachtungen

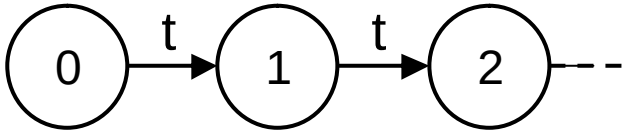
Ziel:
möglichst einfaches (und leicht herstellbares) **Datenobjekt**,
anhand dessen wir **ohne Markenspiel** mechanisch **abprüfen** können,
ob eine gegebene Folge von Transitionen
eine **Schaltfolge** unseres Systems ist.







Erreichbarkeitsgraph des ST-Systems
Knoten: erreichbare Markierung, mindestens M_0
Kante: $M[t\rangle$ → (Markg. M, Transition t, Markg. Mt)
ist Akzeptor der Schaltfolgensprache



Kanonischer Akzeptor der Schaltfolgensprache $Occ(N)$
Knoten: erreichbare Restsprachen, Anfangsknoten $Occ(N)$
Kante: (Restsprache L, Transition t, Sprache $t^{-1}(L)$)

Akzeptoren für Zeitschaltfolgen

1. Methode: das **Timernetz** selbst
   Folge „nachspielen“
   → etwas **umständlich**: Markenspiel + Zeitprüfungen

2. Methode **Zeit-Erreichbarkeitsgraph**
   Knoten: Zeitmarkierung
   Kanten: Zeitschaltung
   \+ kein „Markenspiel“ mehr
   – **sehr viele Knoten und Kanten**, falls
   keine Obergrenze/ dichter Zahlenbereich sogar ∞ viele

3. Methode **Zonen-Erreichbarkeitsgraph** 
   Knoten: Markierung + Wissen über Uhrenstand(sbereich)
   Kanten: Transition (+ Timerbedingung + Timerstarts)
   \+ kein „Markenspiel mehr“
   \+ rein zeitlich unerreichbare **Markierungen fallen weg**
   – noch mit Zeitprüfungen (linearer Aufwand)
   – ggf. noch gleiche Markierung „mit **verschiedenem** Wissen“

Der Zonen-Erreichbarkeitsgraph

vorläufig **nur KANN**-Trans.!

Prinzip: Transitionen führen von Zone zu Zone;
erreichte Zone = erreichte Markierung
+ Wissen über Uhrenstände (aus Vorgeschichte)

Typ: Knoten: **Zone** := **(Markierung,Timerbed. in kanon. Form)**
Kantenanschrift: Transition

$$\begin{pmatrix} 0 & \infty & \dots & \infty \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

stets!!

induktiv:

- Anfangsknoten: $(M_0, K(tc_0))$
- (M,tc) Knoten UND $M[t\rangle$ UND $tc \wedge TC(t)$ **lösbar** $\Rightarrow$

a) $(Mt, KTC(tc,t))$ Knoten und

b) $(M,tc) \xrightarrow{(t,KTC(tc,t),St(t))} (Mt,KTC(tc,t))$ Kante. $KTC(tc,t))?$

Dynamik der Timerbedingungen: *KTC*(*tc*,*t*)

Wie ändert sich unser Uhrenwissen bei einer Schaltung von *t*?

Zunächst: bisheriges Wissen + „*TC*(*t*) erfüllt“

- neu gestartete Timer „übernehmen die Rolle von 0“ im Vergleich zu weiterlaufenden
- je zwei neu gestartete Timer laufen gleich
- alle Timer wieder einzeln unbeschränkt, $x_i \leq \infty$
- in kanonische Form bringen ➔ *KTC*(*tc*,*t*)

ZEG ist *t*-deterministischer **Timed Automaton** (evtl. ∞)

→ Helm-Tiger-Effekt ☺

Timed Automata (Alur, Dill 1990-1994)

Ein **Zeitautomat** (**timed automaton**) ist ein 6-Tupel $A = (L, L_0, \Sigma, X, I, E)$, wobei

- L endliche Menge von **Lokationen**;
- $L_0 \subseteq L$ **Anfangslokationen**;
- Σ endliche Menge von **Etiketten**;
- X endliche Menge von **Uhren**;
- $I : L \to \Phi(X)$ **Invarianten** der Lokationen; (Timerbedingungen)
- $E \subseteq L \times \Sigma \times \Phi(X) \times \mathbf{P}(X) \times L$ **Schaltungen** von A.

Zeit vergeht beim Aufenthalt in Lokationen.

Schaltung (s, a, j, l, s')

- **überführt** den Automaten **schlagartig** von einer Lokation zur nächsten, deren **Invariante** true sein und bleiben muss (sonst (nicht hin) – bzw. weg);
- geht nur wenn j mit den aktuellen Uhrenständen (anfangs 0) ***true*** ergibt;
- setzt die Uhren in l auf 0

➔ **natürliche Semantik** von Beobachtungsfolgen

(Σ-deterministischer) TA ist **brauchbarer Akzeptor** für *Obs.*

Zeitinvarianz in DORTS und Timernetzen

Definition **Zeitverschiebung**: $obs + d := ((v_1, t_1 + d), \ldots, (v_n, t_n + d))$

(Zeit(translations)) **Invarianz**: $Obs(sys) + \mathbb{R}_{\geq 0}$ bzw. $\mathbb{N}_0 \subseteq Obs(sys)$

Hinreichend bei Timernetzen: t_1 (Weltzeit) wird nicht benutzt

➔ Im ZEG werden **alle t_1-Komponenten weggelassen**.

Rücknahme der Vereinfachungen

- Timerbedingungen mit <, >, =, $\vee$ oder $\neg$
- reelle oder rationale Zahlen als Zeitwerte
- ZEG bei Timernet mit MUSS-Transitionen?
- ZEG bei Timernet mit ASAP-Transitionen?

Erweiterungen

- Etikettierte Timernetze

Erweiterte Timerbedingungen

Bei < und > brauchen wir (in dichten Zahlenbereichen)

1-Bit-**Anzeige** in DBM, ob $< \text{ oder } \leq$,

z.B. Zahlen mit Superskript ‚<': $x \leq 10^{<} :\Leftrightarrow x < 10$.

Rechenregeln: $\text{Min}(a, a^{<}) := a^{<},\ a + b^{<} := (a+b)^{<},\ \ldots$

Bei $\vee$ oder $\neg$:

TC-Lösungsmengen **nicht mehr** einfache **konvexe** Mengen

TCs **nicht mehr** als **DBM** zu charakterisieren

→ Berechnen schwieriger – aber nicht unmöglich (**BDD**-Methoden).

ZEG wird analog definiert,
ist immer noch (entsprechend erweiterter) Timed automaton.

Erweiterte Zahlenbereiche für Zeiten

Bei rationalen oder reellen Zahlen für Zeitangaben
alles analog, aber ...

- in Timerbedingung einer **ASAP**-Transition nur
  „**inclusive Untergrenze**“ $a \leq x_i$ zulässig;
  bei $a < x_i$ evtl. kein frühester Zeitpunkt!

- kein Ersatz mehr durch ausmodellierte Zählermechanismen.

ZEG bei Anwesenheit von MUSS-Transitionen

(M,tc) ➔ $(Mt,KTC(tc,t))$ – weiterrechnen!

erweiterte Timerbedingungen

$$tc = true \mid x_i R c \mid x_i R x_i \mid (tc \wedge tc) \mid (tc \vee tc) \mid \neg tc \quad (\neg \text{ bzw. } \vee \text{ genügt})$$

In $(Mt,KTC(tc,t))$ seien z.B. m **MUSS-Transitionen markenaktiviert**.

2^m **Kombinationen** → unter erweiterter Timerbedingung

$$tc \wedge tc_{a_1} \wedge \ldots \wedge tc_{a_k} \wedge \neg tc_{b_1} \wedge \ldots \wedge tc_{b_{(m-k)}}$$

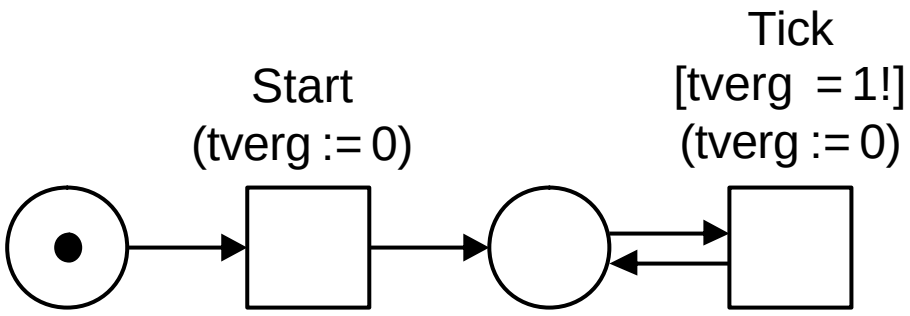
→ spezifische schärfere **Beschränkung der Timerwerte** nach oben

Disjunktion über alle diese Kombinationen, spez. Kanonisierung ➔ $KTC'(tc,t)$

Viele Timer ➔ Komplexitätsproblem

Ähnliche Erweiterungen
bei Anwesenheit von
ASAP bzw. (MUSS und ASAP)

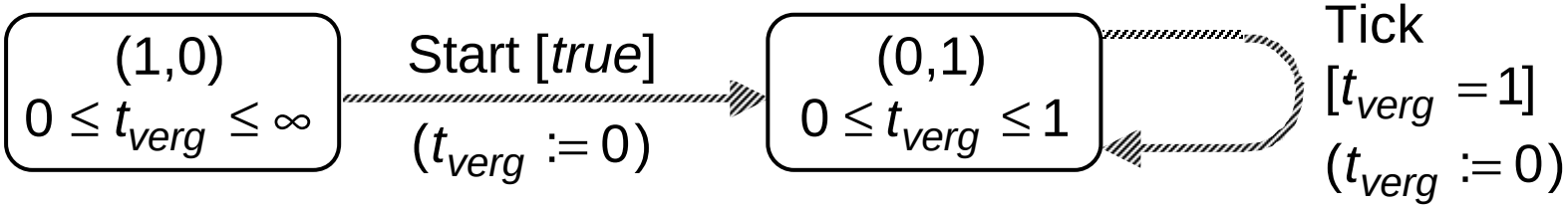
Klingt schrecklich, muss aber nicht sein ...



Standuhr

invariant!

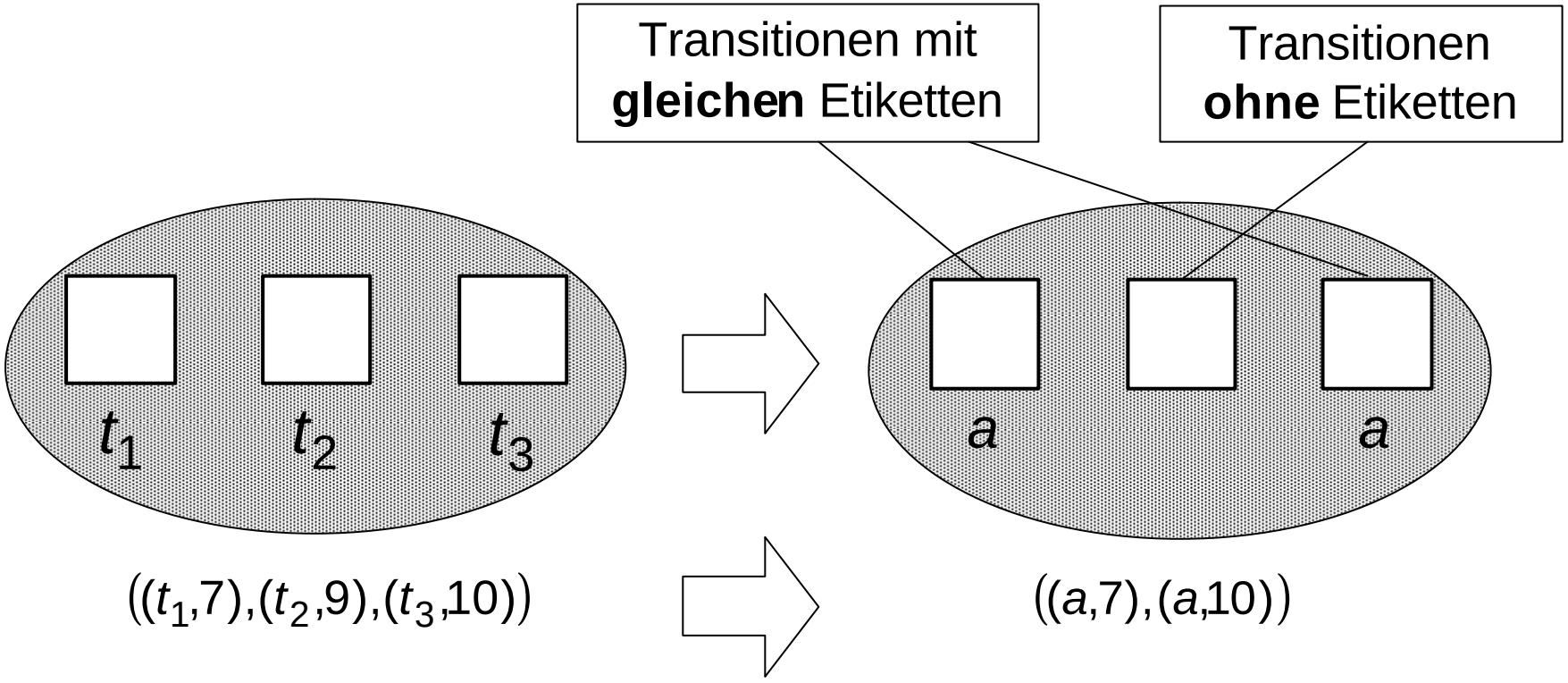
Anfangszeitzone: $0 \leq$ tabs = tverg $\leq \infty$



ZEG(*Standuhr*)
in benutzerfreundlicher Schreibweise

Etikettierte Timernetze

Etikettierung: $h : T \rightarrow A$, partiell, A Alphabet



Informeller Vergleich mit anderen Modellwelten

d.h. mit

- speziellen Timernetzen (→ **Orthogonalität**)
- Time-Netzen,
- Timed-Netzen.

Time Nets (Merlin,Farber 1976. Auch: Waiting time nets)

6-Tupel (P,T,F,M_0,E,L), wobei
(P,T,F,M_0) ST-System und

$E : T \to \mathbb{N}_0$	**früheste Schaltzeit** der Transition (E..arliest)
$L : T \to \mathbb{N}_0 \cup \{\infty\}$	**späteste Schaltzeit** der Transition (L atest)

E = **aktivierte Wartezeit**:
t muss ununterbrochen mindestens $E(t)$ Zeiteinheiten aktiviert sein, bevor es schaltet.

Preemption:
Zwischenzeitige Deaktivierung durch Transition $u \neq t$ ist möglich.

L = **längste aktivierte Zeit**:
t kann nicht ununterbrochen mehr als $L(t)$ Zeiteinheiten aktiviert sein, ohne zu schalten (**Deadline**)

Zunächst: Anfangsmarkierung, und “Systemuhr” startet bei 0.

natürliche Semantik: (Transition,Schaltzeitpunkt) (...) (...) . . .

Timed Nets (Ramchandani 1974. Auch: Reaction time nets)

5-Tupel (P,T,F,M_0,D), wobei (P,T,F,M_0) ST-System und

(P,T,F,M_0) ST-System und

$D : T \rightarrow \mathbb{R}_{\geq 0}$ **Schaltdauer** der Transition

Bei Aktivierung (und Gewinn des **Markenwettbewerbs**):

Reservation:
Einzug der Inputmarken, ***busy*-Zustand**;

nach $D(t)$ Zeiteinheiten
Ausgabe der Outputmarken, Ende *busy*-Zustand;

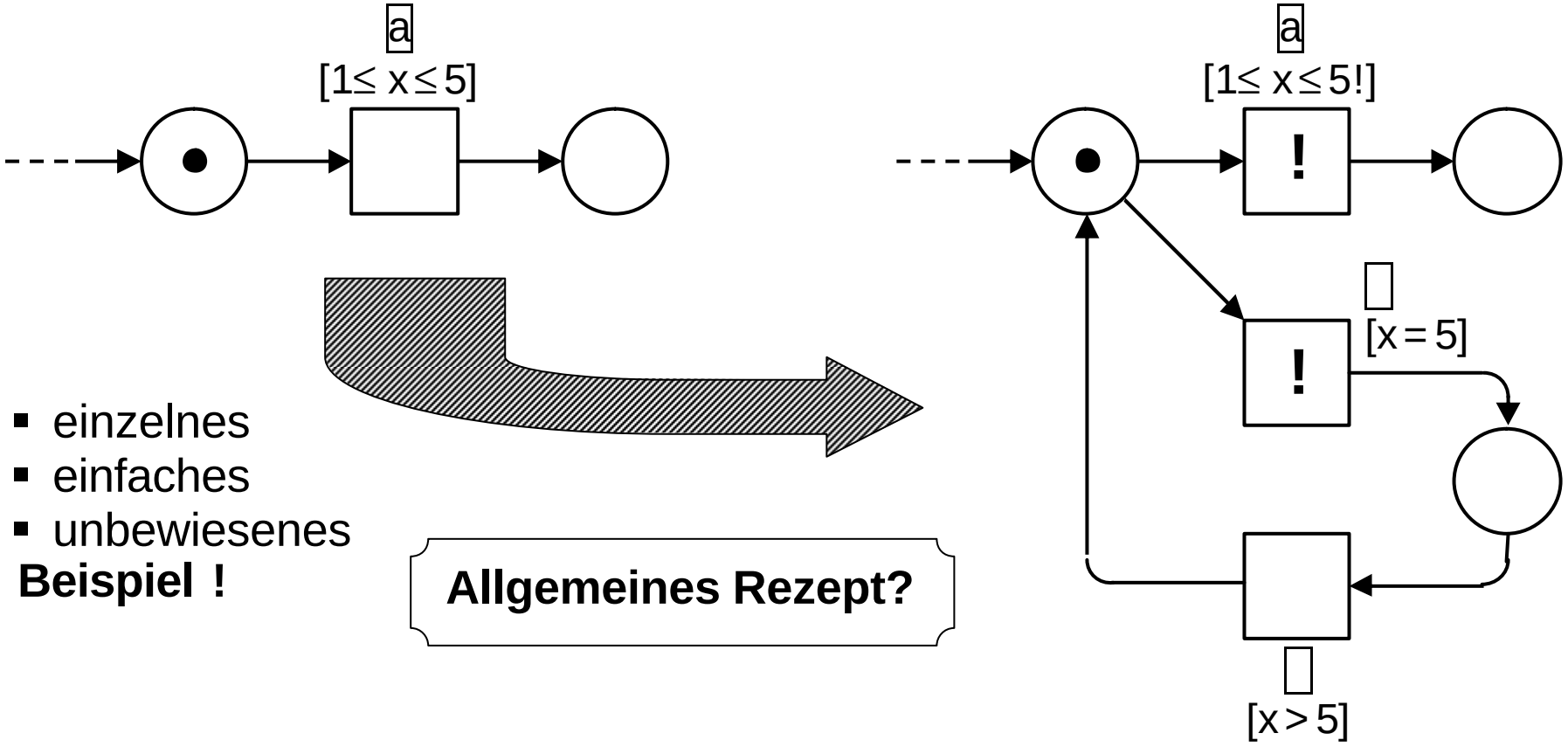
(Konfliktregelung, Mehrfachschaltbarkeit)

natürliche Semantik?	(Transitionsanfang,Schaltzeitpunkt) (...) (...) . . .
oder	(Transitionsende,Schaltzeitpunkt) (...) (...) . . .
oder gar	(Transitionsanfang/-ende,Schaltzeitpunkt) (...) (...) . . .

Simulationen: z.B. KANN durch MUSS (in Timernetzen)

Ähnlich:
- MUSS durch KANN und ASAP;
- ASAP durch MUSS + weitere Uhr

in **etikettierten** Timernetzen:

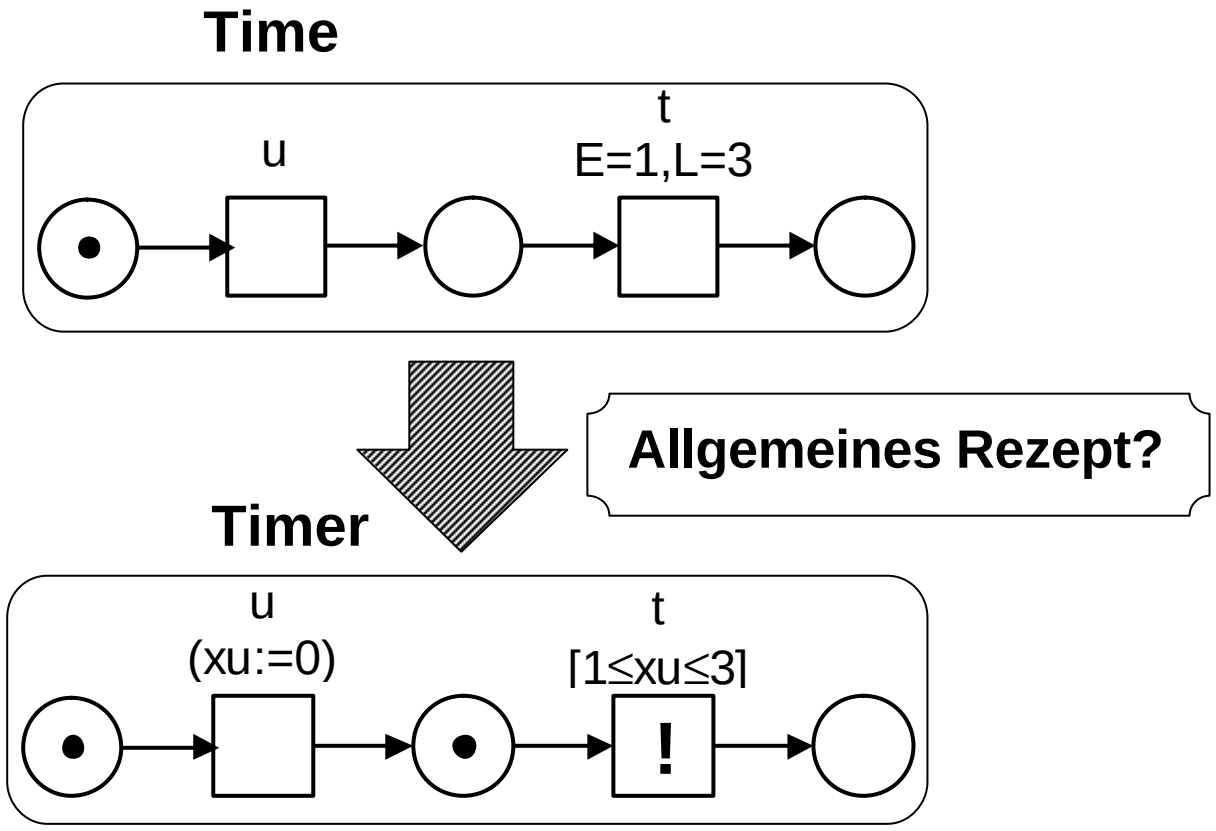


- einzelnes
- einfaches
- unbewiesenes

Beispiel !

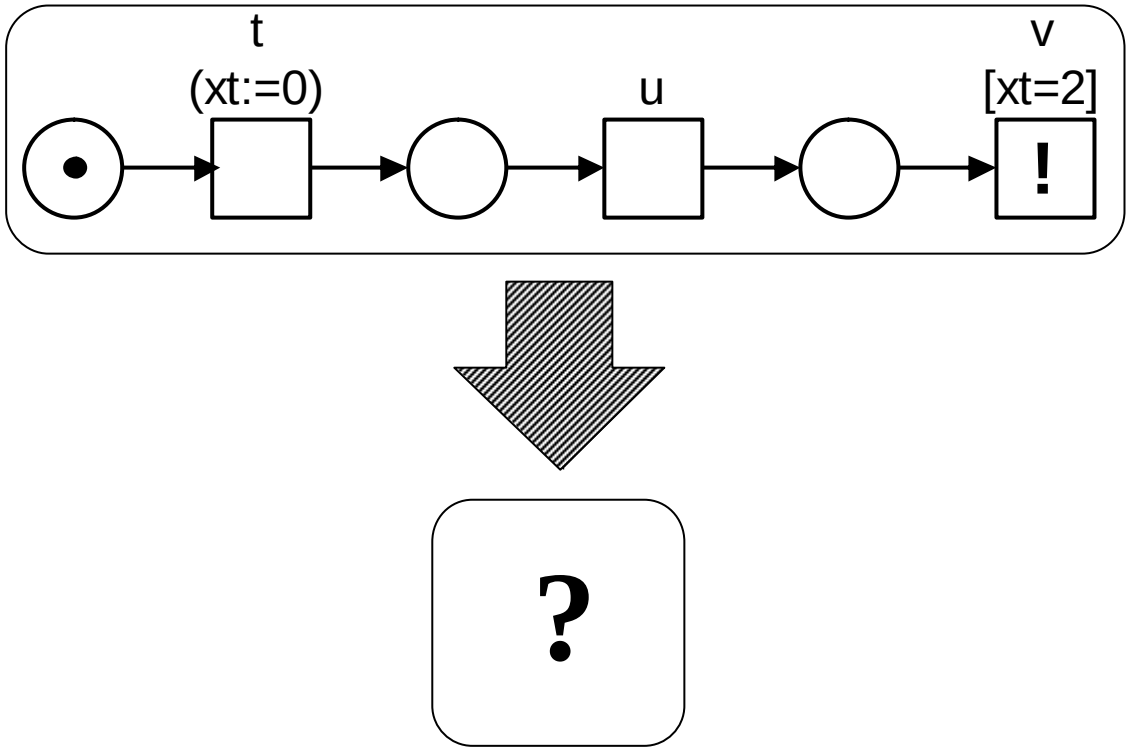
Allgemeines Rezept?

Simulationen: z.B. Time Nets → Timernetze



Simulationen: z.B. Timernetze → Time Nets

Timer



Time

Meine Fragen

Bestimmung bzw. Vergleich der

- **natürlichen Semantiken** (Beobachtungsfolgensprachen),
- **Ausdrucksmächtigkeiten**
  ($\subseteq$- bzw. $=$-Beziehungen zwischen den Klassen der Beobachtungsfolgensprachen)
- **Analysemöglichkeiten** und deren **Komplexität**

bzw. Möglichkeiten **wechselseitiger Simulation**

von
- ST-Systemen mit Zeitzählern,
- Time-Netzen,
- Timed-Netzen,
- Timer-Netzen (gewöhnlich/etikettiert) und
- Timed Automaten

Übergang von „zeitlichen Anforderungen“ zu Timer-Netzen
Requirements ➔ Models, Praxisnähe ...

Hinweise

S.4 Schaltungen – ein Beispiel

Einige Schaltfolgen

a) Volle Runde ohne Verlust:
Erstversand – NAnkunft – QuittungVersand – QAnkunft – Ende

b) Behebung eines Verlusts:
Erstversand – Nverlust – WdhVersand – Nankunft

Unerwünschte Abläufe

c) Überflüssige Wiederholungen, z.B.:
Erstversand – NAnkunft – WdhVersand – WdhVersand

d) Langes Warten (nicht in Schaltfolge alleine ausdrückbar):
Erstversand – Nverlust – <langes Warten des N-Senders> – WdhVersand – Nankunft

(c) und (d) wären vermeidbar, wenn Zeitwissen vorhanden ist. Wenn z.B. sicher ist, dass

- die Nachricht innerhalb eines Tages ankommt oder verlorengeht,
- die Nachricht nach Ankunft innerhalb eines Tages quittiert wird und
- die Quittung Nachricht innerhalb eines Tages ankommt,

wird der Sender eher jeweils nach 3-4 Tagen ohne Quittungsempfang die Nachricht wiederholen als in kürzeren oder längeren Intervallen.

Solche Aspekte darzustellen erfordert Modelle mit integrierter qualitativer Zeit.

S.7 KANN-Transititionen etc.

Der Kunde **darf** die Bank nach 8 Uhr betreten; er **muss** aber nicht.
Wir **können** eine Kapitallebensversicherung bis 31.12.04 noch abschließen, **müssen** aber nicht.

S.8 MUSS-Transitionen etc.

Gesetzgeber und Rechtsprechung trugen früh der Tatsache Rechnung, dass **quantitative Zeitgrenzen** notwendig sind, aber **gerne vergessen werden**: Ein Vertragspartner darf dem anderen i.d.R. noch nachträglich eine Frist setzen, wenn dies beim Vertragsbschluss vergessen wurde. Sonst könnte er sich erst im Moment seines Todes (oder dem des anderen) zu Recht beklagen, dass die Vertragspflicht nicht erfüllt wurde – und dann ist es bekanntlich zu spät (St. Nimmerleins-Effekt).
Weite Bereiche der Forschung in formalen Beschreibungsmitteln haben sich trotzdem lange dagegen gesträubt, quantitative Zeit zu berücksichtigen. Allerdings enthielten ihre formalen Semantiken (→ must and may) durchweg „Beobachtungen" die entweder erst im Unendlichen (also nie) gemacht werden konnten oder ganz konkrete Zeitbedingungen voraussetzten: Wenn meine Standuhr sekündlich tickt, weiß ich aufgrund technisch-zeitlicher Vorkenntnisse bereits nach zwei Sekunden ohne Tick, dass sie steht, also ohne Reparatur, Rütteln oder Aufziehen voraussichtlich (→ Uri Geller) nie mehr reagiert.

S.10 Absolute Zeit etc.

Ich könnte das Feuerwerk per Brief bestellen, aber nicht ohne Bezug auf absolute Uhr bzw. Kalender.
Es genügt z.B. nicht zu schreiben „Ausführung genau x Tage und y Stunden nach Absenden des Briefes" – Woher weiß der Feuerwerker wann er abgeschickt wurde?
Es genügt auch nicht zu schreiben „Ausführung genau x Tage und y Stunden nach Empfang des Briefes" – Woher weiß ich wann er empfangen wird?
Ich könnte schreiben „genau z Stunden nach dem bekannten Flugzeugabsturz von abc" – aber nur da ich mich vorher über dessen Uhrzeit informieren konnte (und der Feuerwerker hoffentlich auch).

S.12 KANN+MUSS+ASAP

<u>Technische Aspekte</u>
Zeit in Minuten
anwendungsorientierte andere Zeitschreibweise

Anwendungsaspekte
Besucher darf zur Öffnungszeit rein, muss aber nicht.
Wenn er drinnen ist, dann muss er pünktlich raus, bei Feuer aber sofort.
Systemstart (mit uhr:=0) entspricht
0:00 Uhr des Geschäftseröffnungstages (Lokalzeit)

S.14 Diskrete Systeme

... direkt oder indirekt

Wir lassen (nach Können und Wille, vgl. unten) beliebige Beobachtungsmethoden, -umgebungen und Hilfsmittel zu. Beispielsweise existieren Viren, obwohl ich noch keine direkt gesehen habe.

... beobachten kann

In den Malen, in denen wir uns begegnet sind, hätte es jedes zweite Mal an meiner Stelle auch ein verheimlichter Zwillingsbruder von mir sein können. Wir können das nicht (z.B. durch Engagement einer Detektei) bei allen Personen unserer Umgebung überprüfen lassen.
Und wenn wir es tatsächlich wollten, wären wir wahrscheinlich paranoid. Paranoide Standpunkte zeichnen sich oft dadurch aus, dass sie die unbeobachtbaren Bereiche des Alltags mit bedrohlichen Vorgängen füllen (Komplotte gegen den/ heimliche Beobachtung oder heimliche Schmähungen des Betroffenen), an die der „Normale“ nicht denkt. Logisch gesehen sind sie meist **nicht zu widerlegen**.

... beobachten will

Wenn unsere IT-Abteilung eine PC-Beschaffung ausschreibt, dann ist sicher nicht von der Farbe auf der Innenseite ihrer Gehäuse die Rede. Diese ist zwar mit geringem Aufwand beobachtbar, aber sie interessiert uns absolut nicht.

... diskret beobachtbar

Wir können beispielsweise einen kontinuierlichen Vorgang wie den Flug eines geworfenen Steins im Dunkeln mit einem Stroboskop beobachten. Jegliche raumzeitlichen Bahnunterschiede sind prinzipiell beobachtbar, aber evtl.

nur mit Glück, denn wir würden z.B. gar nicht nicht mitbekommen, wenn der Stein zwischen zwei Blitzen eine rasche Schleife flöge.
Er tut es nicht nach den Naturgesetzen. Aber wie sicher sind die Naturgesetze? Wenn wir oft benug geworfene Steine beobachten, glauben wir, dass sie immer so fliegen, wie wir das gewohnt sind. Dazu neigen wir wahrscheinlich aus Darwinistischen Gründen: Menschen, die Gestzmäßigkeiten erkennen, überleben im Schnitt länger. Aber niemand ist gewissermaßen **absolut** sicher, dass unbeobachtete geworfene Steine nicht blitzschnell ein paar fröhliche Schleifen fliegen. Keine Angst: der Autor glaubt's auch nicht; aber er bedauert, dass man es **nicht beweisen** kann. Man vergleiche diese Überlegungen mit dem philosophisch motivierten und bisher millionenfach bestätigte „Naturgesetz", dass losgelassene Gegenstände bis zum 1.1.2500 nach unten fallen – und ab dann nach oben!

S.15 **DORTS**

... streng genommen: **SDORTS = sequential DORTS**, d.h. **sequentiell** diskret beobachtete Realzeitsysteme

Allgemeiner könnte man zulassen, dass verschiedene Teilsysteme zeitlich kreiert (abgespalten), wieder gelöscht und während ihrer Lebenszeit sequentiell beobachtet werden können. Dann erhalten wir in jeder Beobachtung pro sequentiellem Teilsystem eine sequentielle DORTS-Beobachtung (+ in welchem Event welches Systems das Teilsystem entstand bzw. sich auflöste). Aber wir könnten nicht mehr sicher die Reihenfolge zwischen Ereignissen in verschiedenen Teilsystemen feststellen, vor allem wenn diese den gleichen Zeitstempel tragen. Anders ausgedrückt: unsere Beobachtungen sind dann keine **Folgen** sondern **Teilordnungen**.

Diese allgemeinen – ggf. **nebenläufigen** – DORTS werden **hier** aber **nicht weiter untersucht**.

Wieso **endliche** Folge?
Weil der Beobachter nur sinnvolle Ereignisse notiert, die im Vergleich zum vorigen Zustand „mindestens ein Bit verändert haben". Dazu ist eine Mindestaufwand und eine Mindestzeit nowendig.
(Gegenbeispiel: Achilles und die Schildkröte, das Zenon-Paradoxon, wobei eine endlicher Zeitraum in unendlich viele Zeiträume zwischen letztlich ereignislosen Zeitpunkten zerlegt wird)

Man kann aus physikalischen und biologischen Gründen weder ein Beobachtungsprotokoll mit unendlich vielen Einzeleinträgen produzieren, noch eine Beobachtung über einen unendlichen Zeitraum hinweg durchführen.

Warum **Beobachtungsbeginn** und **–ende?**

Mittels des **Beobachtungsendes** kann ich (z.B.) Fristüberschreitungen belegen. Ein Beispiel:
Annahme: Der Getränkeautomat soll innerhalb 5 Sekunden nach Geldeinwurf und Wahl das Getränk herausgeben.
OK: Münzeinwurf 12:00:00, Colaknopf gedrückt 12:00:05, Coladose herausgegeben 12:00:08
nicht OK: Münzeinwurf 12:00:00, Colaknopf gedrückt 12:00:05, wütend weggegangen (=Ende der Beobachtung) 12:01:30

Mittels des **Beobachtungsbeginns** kann ich (z.B.) belegen, dass das System Zeit hatte, in den Grundzustand zurückzukehren. Ein Beispiel.
Die B-Cola-Konkurrenz späht den regelmäßigen Test der A-Cola-Automaten aus. Während der A-Cola-Tester sich dem A-Cola-Automat nähert, wirft der Konkurrent heimlich eine Münze ein, wählt Cola und verschwindet.
Naives Beobachtungsprotokoll: „Als erstes gab der Automat eine Coladose heraus, ohne dass ich Geld einwarf.“ Naive Reaktion: Der Automat wird wegen des vermeintlichen Defektes ausgetauscht.
Besser: „Beobachtungsbeginn 12:00:07, Coladose herausgegeben 12:00:08, Kommentar: möglicher aber nicht bewiesener Automatenfehler (denkbar: vorheriger Münzeinwurf vor Beobachtungsbeginn) und Testfortsetzung wie geplant aber vorsichtshalber Menge der relevanten Beobachtungsereignisse um Inventur (Vergleich der Einnahmen und Ausgaben) erweitert,“ usw.

S.16 **DORTS-Frage**

Warum nicht streng wachsende Zeitpunkte?
Weil mehrere Ereignisse stattfinden können, bevor die Uhr eine Einheit weiterrückt.
Trotzdem kann eine **Reihenfolge** der Ereignisse zum gleichen Uhrenstand sinnvoll sein. Wenn wir beispielsweise einen Schnellkopierer (5 Kopien/sec) mit einer Uhr mit Sekundenzeiger testen, können wir trotzdem sowohl die Leistungsangabe überprüfen als auch nachsehen ob die 5 Kopien des gleichen Uhrzeigerstandes in der korrekten Reihenfolge herauskamen.

Zenon-Problem (vgl. oben)
Es ist i.a. unerwünscht, dass in einem beschränkten Zeitraum beliebig bzw. unendlich viele Ereignisse stattfinden. Wir ersparen uns die Definition von Nicht-Zenon-Netzen und überlassen es dem Modellierer (bzw. bürden es ihm auf), den zeitverbrauch geeignet zu modellieren, dass keine Zenoneffekte auftreten.

S.16 **DORTS-Vereinfachungen**

Ganzzahlige Zeitangaben
Zeiten werden i.a. als Dezimalzahlen bis zu n Stellen hinter dem Komma benötigt und abgelesen, bei festem n. Wir wählen z.B. den kleinsten ablesbaren Zeitabstand 10^{-n} als Zeiteinheit.

Nichtnegative Zeitangaben
Alle Uhren außer ggf. der absoluten laufen ab 0 vorwärts. Als Nullmoment der Absolutzeit wählen wir einen Zeitpunkt vor allen interessierenden Ereignissen, vgl. S.23

< bzw. > brauchen wir nicht
So können wir z.B. $x<p$ durch $x \leq 3$ ersetzen.

a bzw. w brauchen wir nicht in ihrer Sonderrolle.
Wenn wir wir es dem Modellierer aufbürden, geeignete Transitionen für Beobachtungsbeginn und -ende hinein zu modellieren, dann reichen normal modellierte Ereignisse als Folgenglieder.

S.23 **Timernetze**

... wurden 1990 im Buch „Petri-Netze“ des Autors informell und noch nicht mit allen hier vorgestellten Merkmalen eingeführt. Grund war der Eindruck, dass Timernetze praktisch verständlicher und problemnäher als die damals bekannten zeitbewerteten Netzmodelle (vgl. S.38 ff) sein könnten. Aus Zeitmangel hätte der Autor nichts dagegen gehabt, wenn andere daraufhin die Timernetze formalisiert und analysiert hätten. Dies fand aber nicht statt.

Interessanterweise verallgemeinern die Timernetze die Timed Automata (vgl. S.30) etwa so, wie ST-Systeme die Automaten verallgemeinern. Beide Modelle wurden indessen etwa zeitgleich und unabhängig voneinander

vorgestellt. Allerdings wurden Timed Automata sofort formal eingeführt und erforscht, und mittlerweile existiert darüber eine reiche Literatur.

S.25 Chroniken und Zeitschaltfolgen

Das Problem der ungestarteten Uhren

Zur Vereinfachung der Timernetz-Definition werden beim Systemstart fiktiv alle Uhren gleichzeitig bei 0 gestartet. Dies hat den Nachteil, dass man hinterher schlecht erkennen kann, ob man eine Uhr abliest, die man bewusst zum richtigen Moment gestartet hat oder eine, die man eigentlich vergessen hat zu starten – sie läuft ja von alleine. Ein die Timernetze unterstützendes Tool kann aber diesbezüglich leicht Buch führen und jedes Ablesen einer nur implizit gestarteten Uhr mit einer Warnung versehen.
Über Jahre waren die Timernetze so definiert, dass ungestartete Uhren einen erkennbaren Sonderwert hatten und die sie ablesende Timerbedingung den Wert „false“ erhielt. Aber auch diese Variante macht nur zusammen mit einer Warnung Sinn, da sonst nicht erkennbar ist, ob die Transition wegen des Ablesens einer nicht ablesbaren Uhr oder lediglich wegen Nichterfüllung der Zeitbedingung nicht schaltete. Insofern war diese Lösung nicht wirklich besser als die nun gewählte definitorisch einfachere.

Deadline-Paradoxon

Es ist typisch, dass hier eine ungestörte Pause, ein reines Zeitvergehen, nur möglich ist, wenn zu erfüllende Zeitbeschränkungen dies nicht verhindern. Wer verspricht, bis Feierabend alles zu erledigen, verspricht (zumindest rein logisch), dass bei unerledigten Resten eine Sekunde vor Feierabend die Welt stehenbleibt!

S.26 Akzeptoren

Occ(N) ist die Menge der Schaltfolgen von N. $t^{-1}(L)$ ist die Menge aller Wörter w mit $tw \in L$

S.29 Dynamik der Timerbedingungen

Beim Helmtiger-Effekt führt man die Beantwortung einer neuen (und nur schwer verständlichen) Frage auf die existierende (aber keineswegs einfache) Antwort einer anderen Frage zurück.
Der Helmtiger-Witz ist ein harmloser Kinderscherz: „Weißt Du wie man einen Helmtiger fängt?“– Das Kind antwortet aus doppeltem Grund „Nein“: es weiß ja noch nicht einmal, was ein Helmtiger sein soll, geschweige denn wie man ihn fängt! Dann erzählt man eine langwierige Geschichte (unterschiedliche Versionen), an deren Ende der Helmtiger angeblich wütend seinen Helm auf den Boden wirft – „... und dann kann man ihn fangen wie jeden anderen Tiger auch.“
Über Timed Automata exisitert eine reiche Literatur. Normale Tiger wurden bereits in großer Zahl gefangen.

S.33 Erweiterte Timerbedingungen

BDDs sind binary decision diagrams. Sie entsprechen aussagenlogischen Termen mit lediglich einem Konnektor IF-THEN-ELSE, den Konstanten *true* und *false* und Aussagevariablen. Sie lassen sich auf eine (graphisch gut verständliche) kanonische Form reduzieren, bei der Äquivalenz und Identität zusammenfallen. BDDs sind so etwas wie die moderne Allzweckwaffe der Aussagenlogik.

S.34 Erweiterte Zahlenbereiche

Wenn eine ASAP-Transition zur Zeit 3,5 markenaktiviert wird, aber ihre Zeitbedingung erst die Schaltung zur Zeit >4 erlaubt, müsste sie zum ersten Zeitpunkt >4 schalten – bei rationalen oder reellen Zeitwerten eine Unmöglichkeit!

S.34 **ZEG bei MUSS-Transitionen**

Ähnliche Effekte durch ASAP:
Beim Parsen ist (mittels Buchführung) sicherzustellen, dass ASAP-Transitionen

- ebenfalls die Lebenszeit ihrer aktivierenden Zeitmarkierungen beschränken und
- zum gleichen Zeitpunkt schalten müssen
  - wie die Transtition, die sie sofort aktivierte, oder, wenn sie nicht gleich zeitaktiviert war
  - in dem die kleinstmögliche Pause endete, die sie aktivierte.

S.41-43 **Simulationen**

Dies sind nur ad hoc erfundene informelle Beispiele für intuitive Übersetzungen bzw. einen schwerer übersetzbaren Fall
Sie sind nicht als korrekte Übersetzung mit gleicher natürlicher Semantik mathematisch bewiesen.
Erst recht ist hier noch kein Übersetzungsalgorithmus angegeben.
Und im letzten Beispiel ist die Unübersetzbarkeit nicht behauptet, geschweige denn bewiesen.
Das Ganze ist hier nur „angedacht“.

Im Gegensatz zu 1990 (vgl. Notizen zu S.23) wird das Thema aber nun weiter verfolgt. ☺