Einführung in Computational Engineering

Grundlagen der Modellierung und Simulation

2. Vorlesung: Diskrete Modellierung und Simulation

21. Oktober 2013

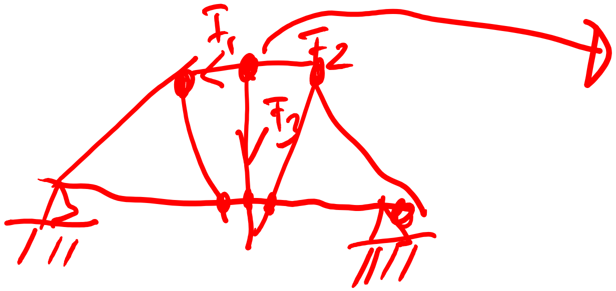
Prof. Dr. Jan Peters

produziert vom

Meisenantworten

- Keine englischen Wörter? Sorry!
- Flackerndes Licht? Email an Wolfgang Heenes!
- Mehr Zeit für Moodle Fragen? OK!
- Skript vor der Vorlesung? Wir geben uns Mühe!
- Moodle Fragen zur Klausurübung online stellen? OK!
- Reingesagte Kommentare aus der 1. Reihe unfair…OK!
- Folien vor Volesung? Wir geben uns Mühe!
- *Pünktlich sein*: Übungen und Vorlesung bleiben zeitlich austauschbar!
- Mehr Fragen? Wir geben uns Mühe!
- Bessere Handschrift? Gebe mir Mühe!
- Andere Süßigkeiten? Habe ich versucht, wollte bisher keiner nehmen…

Recap: Statisches vs Dynamisches Model



$$\left.\begin{aligned} \sum F_x &= F_1 - F_2 = 0 \\ \sum F_y &= F_3 + m_K g = 0 \end{aligned}\right\} \text{statisches Modell!}$$

$$\sum F_x = m \ddot{x}_K \neq 0 \quad \Big\} \text{ dynamisches Modell}$$

MOODLE FRAGE

Bitte jetzt auf Moodle eine Frage beantworten!

Überblick der Vorlesungsinhalte

1. Einführung
2. Diskrete Modellierung und Simulation
3. Zeitkontinuierliche Modellierung und Simulation
4. Teilschritte einer Simulationsstudie
5. Interpretation und Validierung
6. Modulare und objektorientierte Modellierung und Simulation
7. Parameteridentifikation von Modellen

Grundlagen der Modellierung und Simulation

2. DISKRETE MODELLIERUNG UND SIMULATION

2.1 Diskrete Modelle

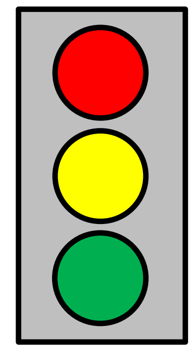
- Zeitdiskrete Modelle
  - Strukturiert mit diskreten Zeitschritten t_k für $k \in \mathbb{N}$

  - Falls Zeitachse nicht äquidistant, dann ereignisdiskrete Beschreibung bevorzugt

  - Häufiges Vorkommen in technischen und computergesteuerten Systemen (z.B. Regelungstechnik)

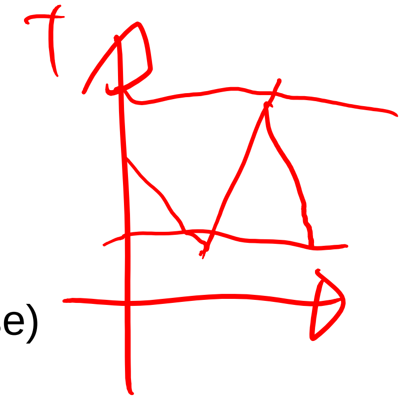
2.1 Diskrete Modelle

- Zeit- und Wertdiskrete Modelle
  - Zeitachse nicht notwendigerweise äquidistant
  - Gut in Tabellen darstellbar
  - Merkmal für qualitative Modelle
    - Abhängige Variable ebenfalls diskret / diskretisiert (wertdiskret)
  - Beispiele:
    - Ampelsignale / Ampel (siehe Modell)
    - Stockwerke bei einer Aufzugfahrt



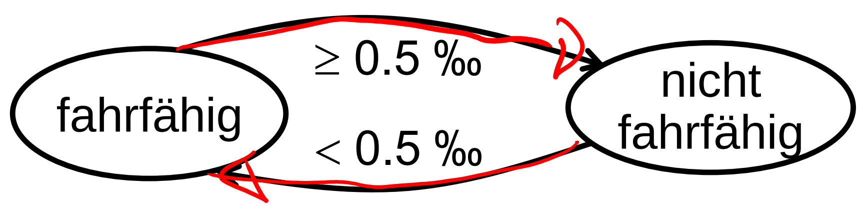
2.1 Diskrete Modelle

- Ereignisdiskrete Modelle
  - (Möglicherweise) nicht äquidistante zeitdiskrete Schritte
  - Zeit und Werte in realen Systemen oft kontinuierlich
    - Modellierung als diskretes Ereignis (qualitative Änderung von Interesse)



  - Mischung aus ereignisdiskreten und zeitkontinuierlichen Modellen möglich

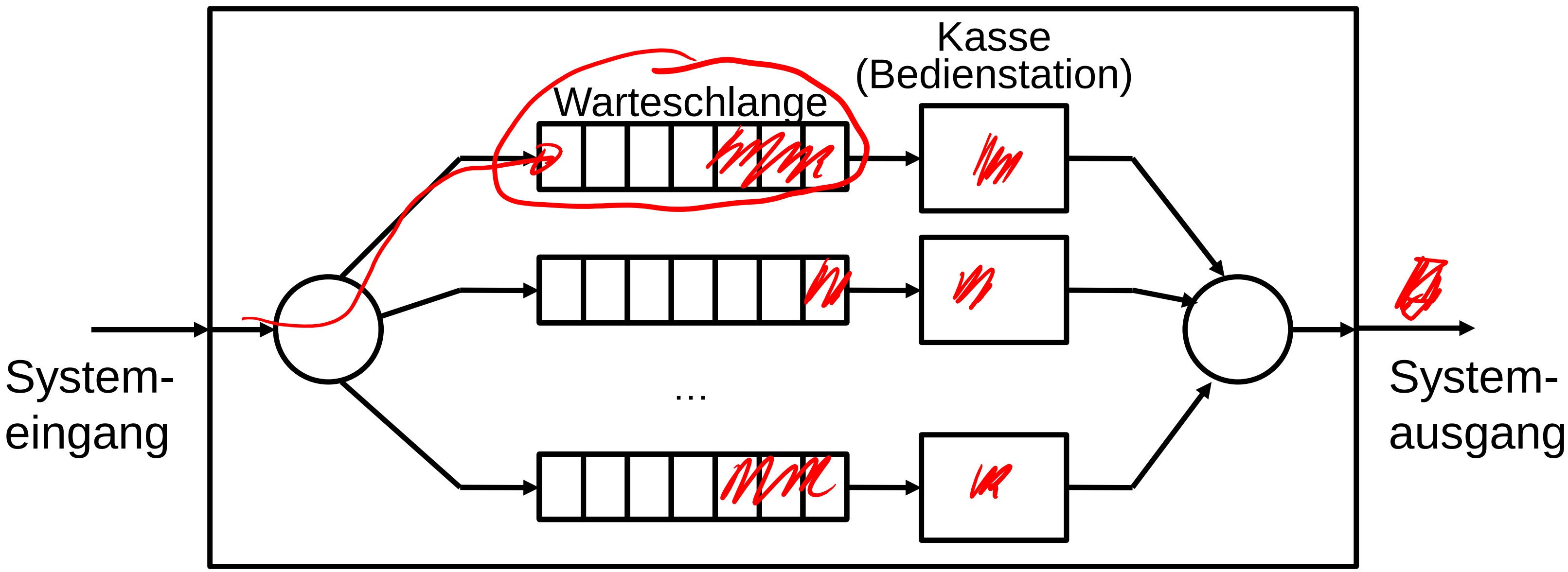
  - Beispiel:
    - Fahrtauglichkeit im Straßenverkehr nimmt kontinuierlich mit Blutalkoholgehalt ab



2.1 Diskrete Modelle: Beispiel

- Bestimmung der Attribute
  - Stochastisch (Kapitel 7) oder empirisch

- Systemmodell



2.1 Diskrete Modelle: Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mögliches Ziel
  - Berechnung der mittleren Wartezeit eines Kunden an einer Kasse im Supermarkt

- Unwichtig
  - Vorgang des Einkaufens

- Relevante Objektklassen
  - Kunde, Kasse und Kassierer/in

2.1 Diskrete Modelle: Beispiel

- Attribute der Objekte…
  - … für Kunden
    - Wareneinkaufsliste (Anzahl der Artikel und deren Volumenklasse)
    - Eintrittszeitpunkt in das System
    - Austrittszeitpunkt aus dem System
  - … für Kassen
    - Zustand (geöffnet / geschlossen)
  - … für Kassierer/innen
    - Bedienung (routiniert / langsam)

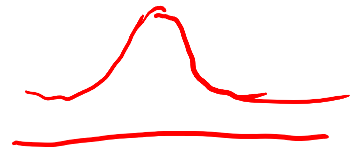
2.1 Diskrete Modelle: Beispiel

- Mögliche, relevante und noch unberücksichtigte Einflüsse
  - Wechseln Kunden häufig die Warteschlange, wenn es an anderer Stelle doch schneller geht? Strategie?
  - Verlassen Kunden häufig die Warteschlange, da Waren vergessen wurden?
  - Stehen Männer/Frauen bevorzugt bei attraktiveren KassiererInnen an, auch wenn dort die Warteschlange etwas länger ist?
  - Kennen manche Kunden / Kundinnen die routinierten Kassierer(innen) und stellen sich dort an, auch wenn die Warteschlange „etwas länger“ ist?

2.1 Diskrete Modelle: Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Modellparameter
  - Ankunftsverhalten von Kunden



  - Abfertigungs- / Bedienungsverhalten der Kassierer/innen
  - Bedienstrategien (z.B. FIFO, Prioritäten, …)
  - Anzahl der besetzten Kassen

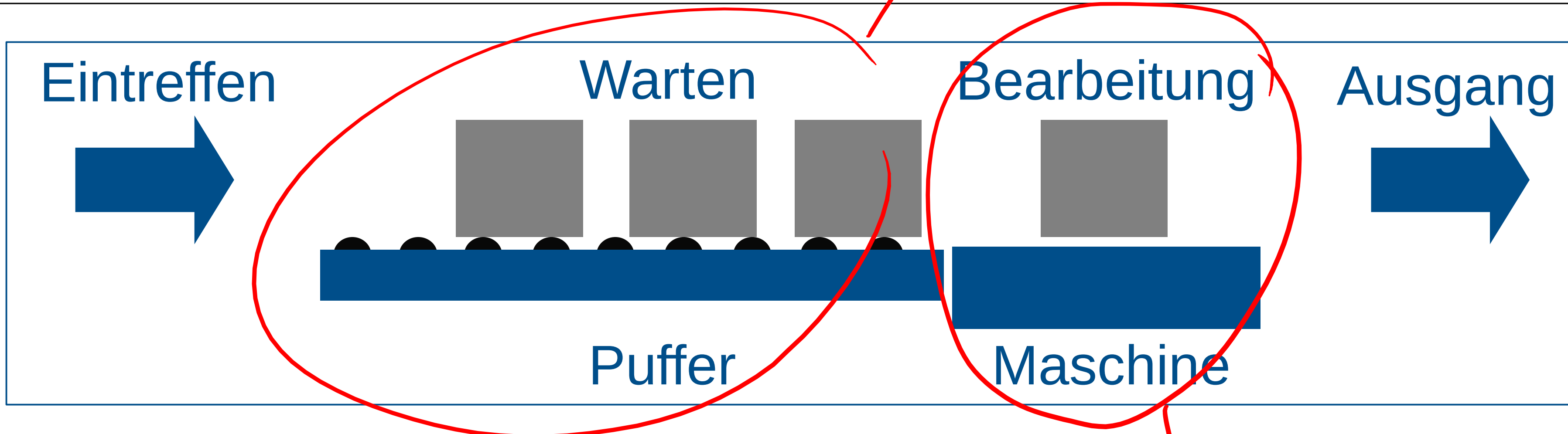
2.1 Diskrete Modelle: Beispiel

- Mögliche Bewertungsgrößen
  - Verweilzeit der Kunden (Mittelwert, Varianz, …)
  - Wartezeit der Kunden an verschiedenen Kassen

  - Durchsatz des Systems „Supermarkt" (Kunden pro Zeiteinheit)
  - Auslastung eines/r Kassierer/in
  - Füllung des Systems (Anzahl von Kunden) oder von Systemkomponenten (Kassierer/in)

2.1 Beispiel: Stückprozesse

Fließband!



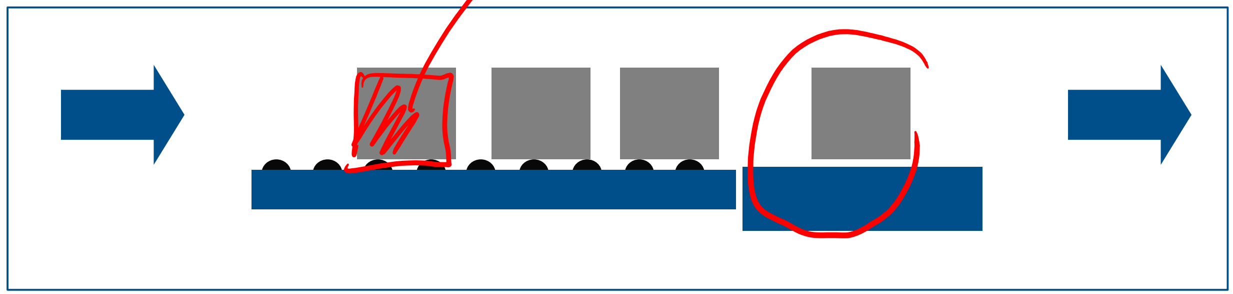
= Kassierer

- Typische Fragen:
  - Wie groß muss der Puffer sein?
  - Wie ist die Maschinenauslastung?
  - Wie lange hält sich ein Werkstück im System auf?

2.1 Beispiel: Stückprozesse

$\dot{x} = f(x)$

- Benötigte Information:
  - Eintreffzeiten
  - Bearbeitungszeiten

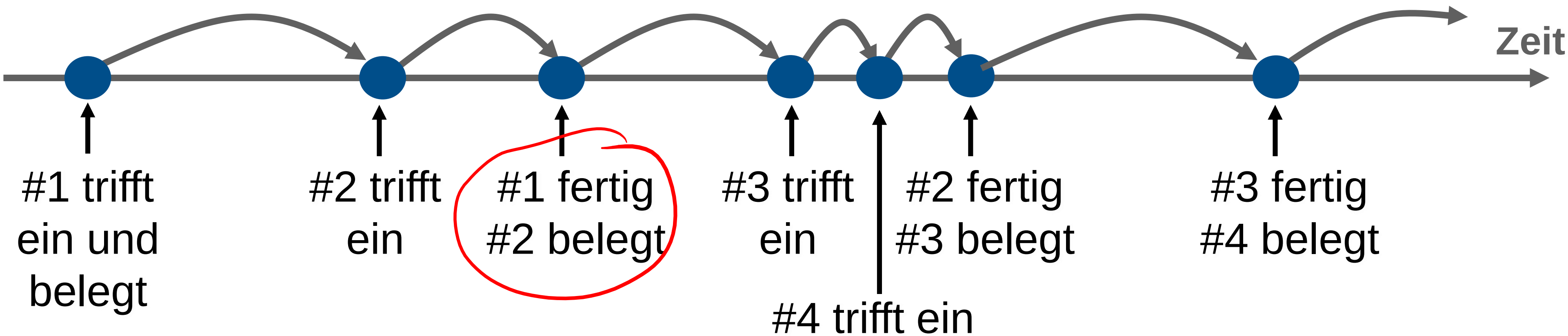


- Unwichtig:
  - Bewegungsdynamik der Stücke
  - Genauer Ablauf der Bearbeitung

2.1 Beispiel: Stückprozesse

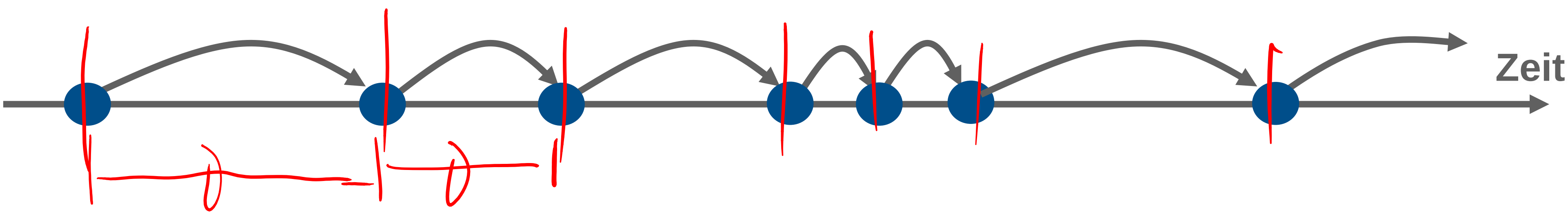
- Zeitdiskretisierung: Zeitskala wird auf die Zeitpunkte reduziert, an denen etwas passiert (Ereignisse, events)
  - Eintreffen eines Stückes
  - Belegen der Maschine
  - Verlassen der Maschine

(E, Z)



2.1 Beispiel: Stückprozesse

- Ereignisdiskrete Simulation
  - Simulationsuhr schreitet von Ereignis zu Ereignis



2.2 Ereignisdiskrete Modelle: Terminologie

TECHNISCHE UNIVERSITÄT DARMSTADT

▪ System	Ansammlung von Objekten (zeitabhängige Interaktion miteinander nach spezifizierten Regeln)
▪ Modell	Abstrakte, logische und mathematische Darstellung der Objekte und Wechselbeziehungen in einem System
▪ Entität	Objekt in einem System, das innerhalb des Modells explizit dargestellt wird
▪ Attribut	Variable, die den Zustand einer Entität beschreibt
▪ Ereignis	Beschreibung der Aktualisierung des Modellzustands

2.2 Ereignisdiskrete Modelle: Terminologie

TECHNISCHE UNIVERSITÄT DARMSTADT

$L = \{(T_i, E_i)\}$

- Ereignisliste — Liste mit Zeitpunkt und Typ der geplanten Ereignisse (nach Ereigniszeitpunkten aufsteigend geordnet)
- Aktivität — Zeitlich erstreckter Vorgang zwischen den initiierenden und abschließenden Ereignissen einer Operation, die den Zustand einer Entität transformiert
- Simulationsuhr — Variable (gibt aktuellen Stand der Simulationszeit wieder)
- Zeitführungsroutine — Prozedur zur Auswahl des nächsten Ereignisses aus der Ereignisliste und Vorstellen der Simulationsuhr auf den nächsten Ereigniszeitpunkt

2.2 Ereignisdiskrete Modelle: Terminologie

75% ± 25%

- Ergebnisroutine — Prozedur zur Berechnung der statistischen Schätzwerte der Ergebnisvariablen anhand statistischer Zähler und zur Ausgabe des Ergebnisprotokolls am Ende des Simulationslaufs

- Steuerprogramm — Programmteil, der wiederholt die Zeitführungsroutine zur Bestimmung des nächsten Ereignistyps aufruft und die zugehörige Ereignisroutine aufruft, bis Simulationslauf endet

MOODLE FRAGEN

Bitte auf Moodle jetzt Fragen beantworten!

2.2.1 DES: Anwendung

- DES (= <u>d</u>iscrete-<u>e</u>vent <u>s</u>imulation)
  - Eine der am besten etablierten Simulationsmethoden
  - Findet häufig Anwendung bei der Simulation von Warteschlangen

2.2.1 DES: Anwendung

- Produktionssysteme
  - Fertigung
  - Montage
  - Logistik

Anwd

- Verkehrssysteme
  - Verkehrsplanung
  - Rangierbahnhöfe
  - Umschlagterminals
  - Speditionen

- Wirtschaftssysteme
  - Geschäftsprozesse
  - Krankenhausorganisation

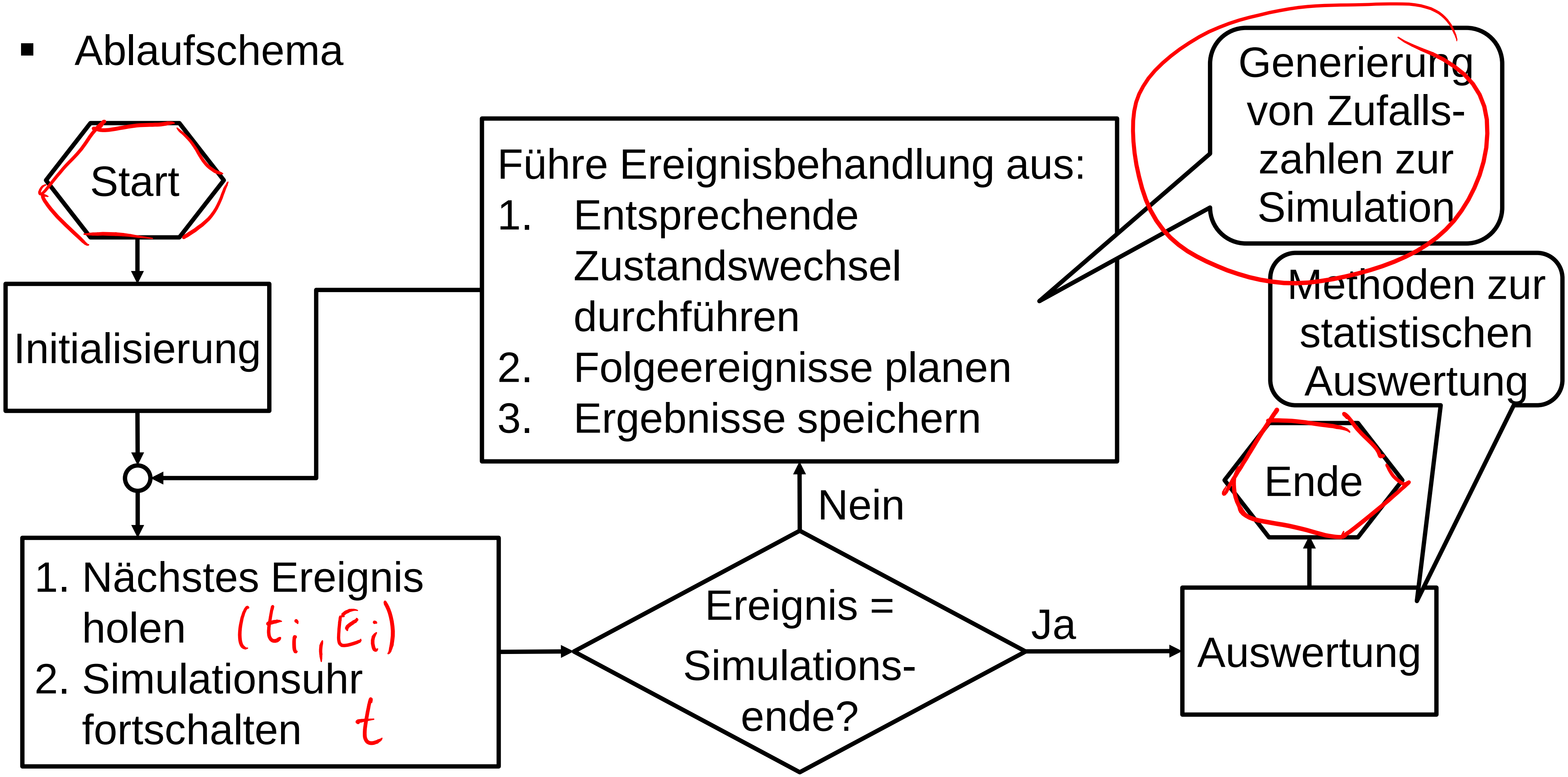
- Informationstechnik
  - Mobilfunknetze
  - Satellitenauslastung
  - Rechnernetze
  - Parallelrechner

2.2.1 DES: Zielsetzung

- Kapazitätsauslegung
  - Ressourcen (Maschinen, Roboter, Prozessoren, …)
- Systemauslastung
  - Produktionsplanung
  - Ablauforganisation
- Logistik
  - Nachschubstrategien
  - Lagerhaltungsstrategien
- Störanalyse
  - Notfallstrategien
  - Redundante Auslegung

2.2.1 DES: Funktionsweise

- Ablaufschema



2.2.1 DES: Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

Initialisierung

$L := \{(t_1, E_1)\};\quad t_0 := 0;\quad N := 0;$

$S := idle;$

Ereignisschleife

$while\ t < t_{max}$

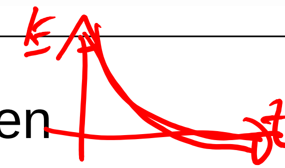
$t := t_1;\ \ E := E_1;\ L := L \setminus \{(t_1, E_1)\};$

$switch(E):$

$case\ A:\ \ ArrivalRoutine;$

$case\ D:\ \ DepartureRoutine;$

$sort(L);$



- Zufallszahlengeneratoren
  - Ankunftszwischenzeiten, z.B. $exp(\lambda)$
  - Serverbelegungszeiten, z.B. $norm(\mu, \sigma)$
- Ereignisse
  - A (= Arrival / Ankunft)
  - D (= Departure / Abreise)
- Systemspezifische Variablen
  - S (= Zustand des Servers)
  - L (= Ereignisliste)

2.2.1 DES: Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

function ArrivalRoutine;
begin

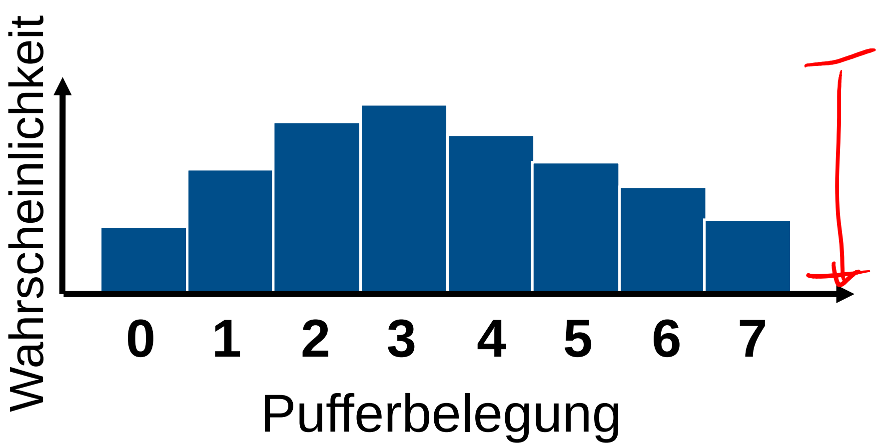
aktuelle t   Zufall

$L := L \cup \{(t + \exp(\lambda), A)\};$
if $S = busy$
then $N := N + 1$
else begin
$S := busy;$
$L := L \cup \{(t + norm(\mu, \sigma), D)\};$
end;
end;

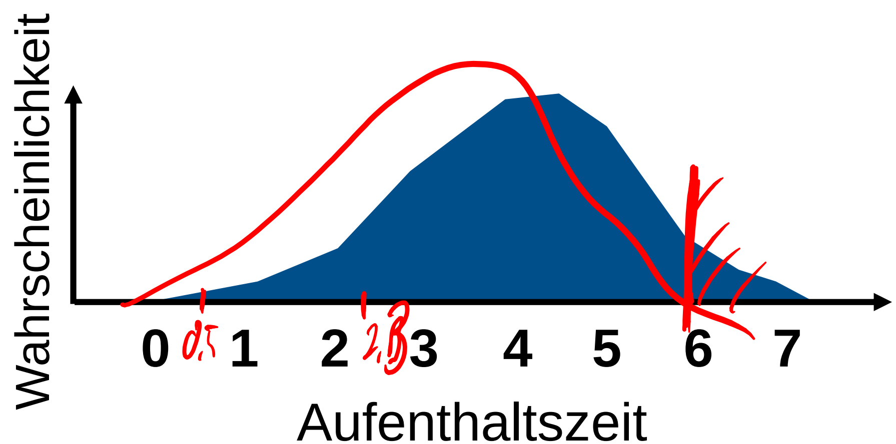
function DepartureRoutine;
begin
if $N = 0$
then $S := idle;$
else begin
$N := N - 1;$
$L := L \cup \{(t + norm(\mu, \sigma), D)\};$
end;
end;

2.2.1 DES: Statistische Verteilungen

- Diskrete Verteilung
  - Bezogen auf Ressourcen (z.B. Pufferbelegung, Maschinenbelegung)

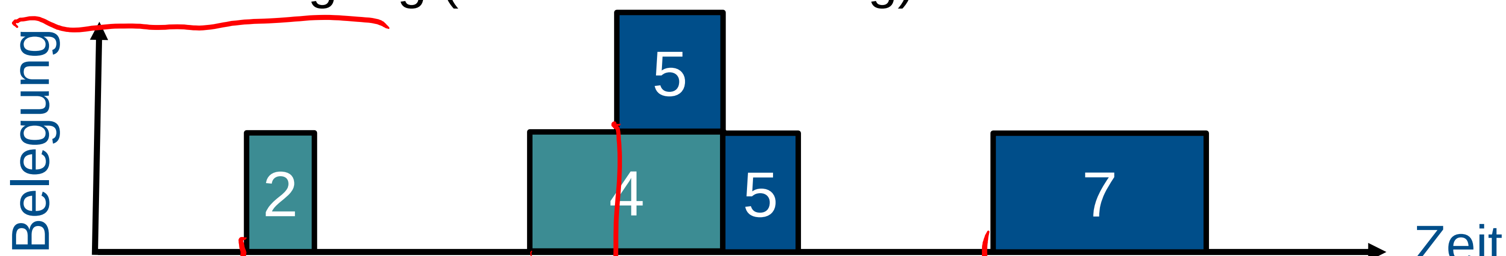


- Kontinuierliche Verteilung
  - Bezogen auf Werkstücke (z.B. Verweilzeit im Puffer oder im System)

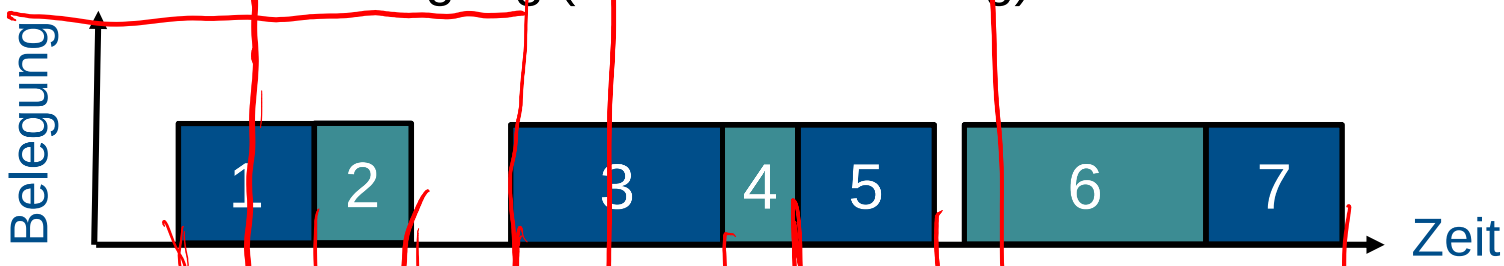


2.2.1 DES: Beispiel

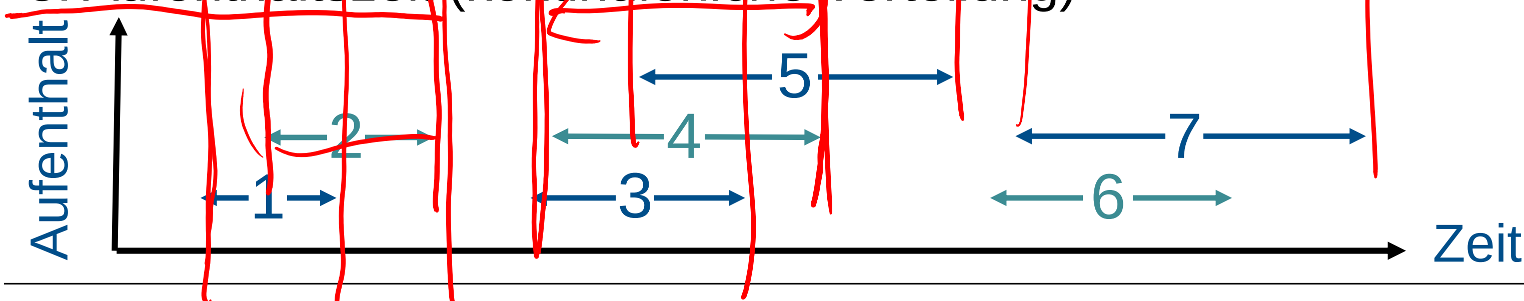
1. Pufferbelegung (diskrete Verteilung)



2. Maschinenbelegung (diskrete Verteilung)



3. Aufenthaltszeit (kontinuierliche Verteilung)



TH3

2.2.2 Schätzung der Momente einer Verteilung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Diskrete Verteilungen
  - P_i ist die Anzahl der Entitäten im Puffer bis i
  - Summations-Variablen
    - $S_n = \sum_{i=1}^{n} P_i$
    - $T_n = \sum_{i=1}^{n} {P_i}^2$
  - Schätzung der Momente
    - $E(P) = \frac{S_n}{n}$ (1.Moment) = Mittelwert!
    - $Var(P) = \frac{T_n}{n} - \left(\frac{S_n}{n}\right)^2$ — $Var[X] = E[X^2] - E[X]^2$
    - $\frac{T_n}{n}$ (2. Moment)

2.2.2 Schätzung der Momente einer Verteilung

- Kontinuierliche Verteilungen
  - $A(t)$ ist die Aufenthalts- / Verweilzeit
  - Summations-Variablen
    - $S(t) = \int_0^t A(\tau)d\tau$
    - $T(t) = \int_0^t A^2(\tau)d\tau$
  - Schätzung der Momente
    - $E(A) = \frac{S(t)}{t}$
    - $Var(A) = \frac{T(t)}{t} - \left(\frac{S(t)}{t}\right)^2$

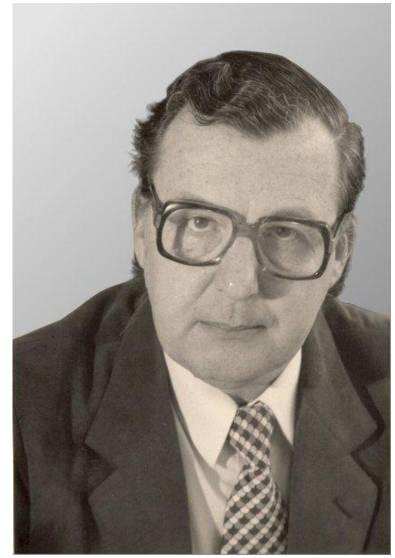
Literatur zu Kapitel 2.2 DES (Auswahl)

TECHNISCHE UNIVERSITÄT DARMSTADT

- B. Page: *Diskrete Simulation* (Springer, 1991)
- H.-J. Siegert: *Simulation zeitdiskreter Systeme* (Oldenbourg, 1991)
- A.M. Law, W.D. Kelton: *Simulation Modeling and Analysis* (McGraw-Hill, 3rd. ed., 2000)
- U. Kiencke: *Ereignisdiskrete Systeme* (Oldenbourg, 1997)
- G.S. Fishman: *Discrete-Event Simulation* (Springer, 2001)
- J. Banks, J.S. Carson II: *Discrete-Event System Simulation* (Prentice Hall, 1984)
- C. G. Cassandras, S. Lafortune: Introduction to Discrete Event Systems (Kluwer, 1999)

2.3 Petrinetze

- Carl Adam Petri
  - Am 12. Juli 1926 geboren und am 2. Juli 2010 gestorben
  - Erfinder der Petrinetze
  - Dissertation (eingereicht: 1961, verteidigt: 1962) an der TU Darmstadt über Petrinetze abgelegt



(Quelle Bild: http://www2.informatik.hu-berlin.de/top/lehre/petriweb/petri.jpg)

2.3 Petrinetze

- Abstraktes und formales Modell eines Informationsflusses

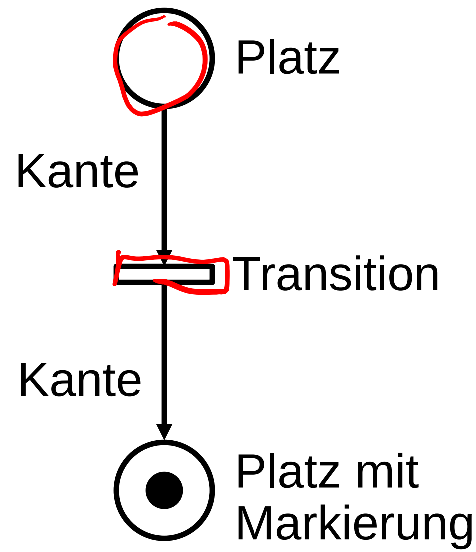
- Geeignet zur Modellierung von
  - Ereignisdiskreten Systemen mit nebenläufigen Eintritten von Ereignissen und gleichzeitigen Beschränkungen an Art, Häufigkeit und Vorgänger dieser Ereignisse

- Anwendung zur Beschreibung paralleler Ausführung von Jobs in Rechenanlagen

2.3 Petrinetze

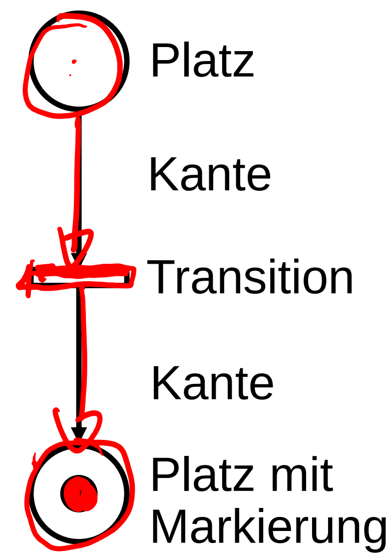
- Bipartiter Graph
  - Graph mit Knoten aus zwei disjunkten Teilmengen
    und keine Kanten innerhalb der jeweiligen Teilmengen

- Petrinetz $PN = (P, T, A, K, M_0)$
  - gerichteter, bipartiterGraph
  - Zwei Arten von Knoten (Plätze, Transitionen)
  - Kanten nur zwischen zwei unterschiedlichen Knoten



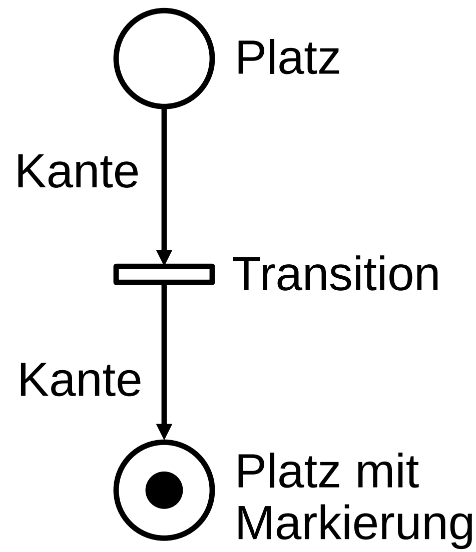
2.3 Petrinetze

- Plätze (places)
  - Beschreiben (mögliche) Zustände (Darstellung als Kreis)
  - Endliche Menge von Plätzen $P = \{p_1, p_2, \dots, p_n\}$
- Transitionen
  - Beschreiben (mögliche) Ereignisse (Darstellung als Rechteck)
  - Endliche Menge von Transitionen $T = \{t_1, t_2, \dots, t_m\}$
- Gerichtete Kanten (directed edges)
  - Verbinden Plätze und Transitionen
  - Endliche Menge gerichteter Kanten $A = \{a_1, a_2, \dots, a_l\}$



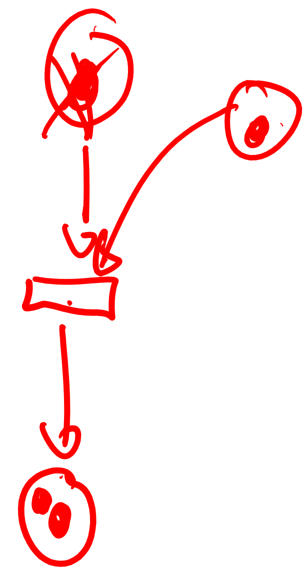
2.3 Petrinetze

- Markierungen (tokens)
  - Aktueller Zustand eines Petrinetzes
  - Gespeichert in Plätzen
    (Darstellung als schwarze Punkte im Kreis)
  - Zustand des Petriznetzes $M = \{m_1, m_2, \ldots, m_n\}$
    - $m_i \in M$ ist die Anzahl der Marken in Platz $p_i \in P$
  - Anfangsmarkierung $M_0 = \{m_1', m_2', \ldots, m_n'\}$
  - Maximale Kapazität der Plätze $K = \{k_1, k_2, \ldots, k_n\}$



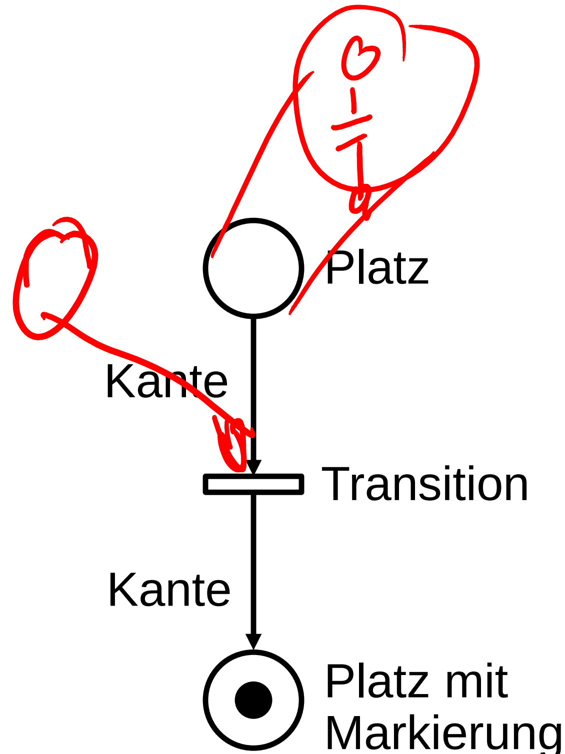
2.3 Petrinetze

- Dynamische Eigenschaften eines Petrinetzes resultierend aus seiner Ausführung
  - Markierungen werden durch Schalten von Transitionen erzeugt, gelöscht oder verschoben (firing of transitions)
  - Transition ist schaltbereit (bzw. aktiviert), wenn alle ihre Eingangsplätze markiert sind (bzw. mindestens eine Marke enthalten)
  - Schaltbereite Transitionen können schalten („feuern“), wobei eine Marke aus jedem Eingangsplatz weggenommen und in jeden Ausgangsplatz eine Marke hinzugefügt wird



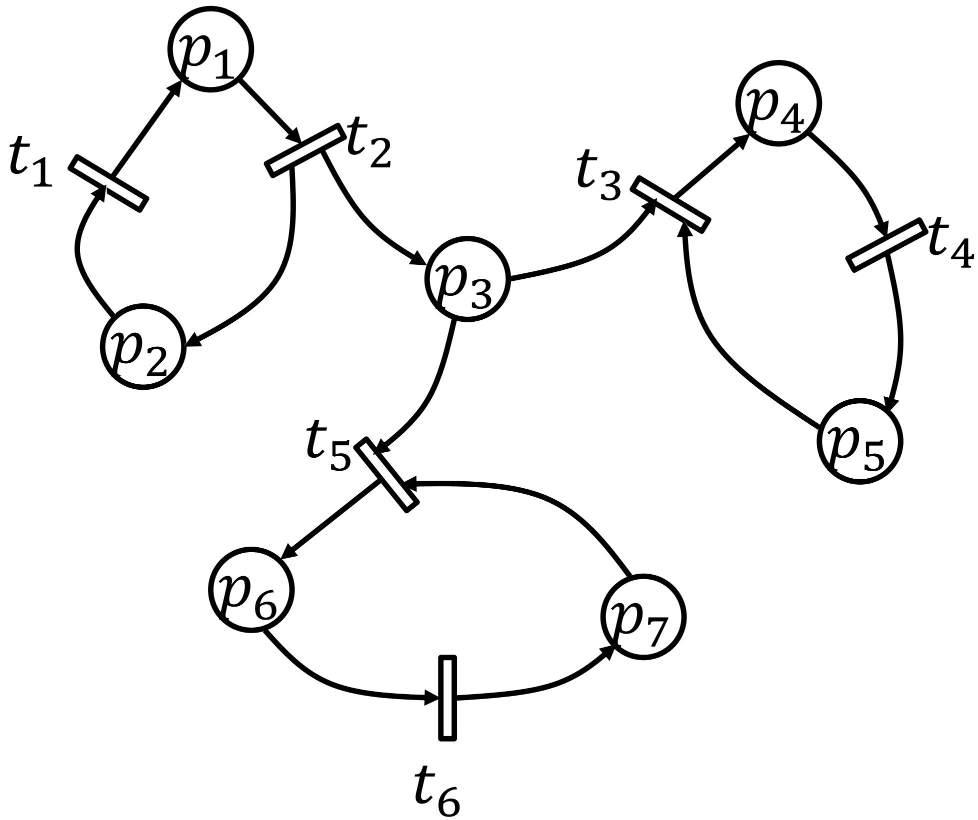
2.3 Petrinetze: Typen & Hierarchien

- Zeitdiskrete Modelle
  - Petrinetz beschreibt logisch, was in welcher Reihenfolge passiert
- Zeitkontinuierliche Modelle
  - Vorhersage im Petrinetz, wann etwas passiert
- Plätze oder Transitionen eines Petrinetzes können wiederrum aus Sub-Petrinetzen bestehen
  - Transitionen als Subnetze müssen mit Transitionen beginnen und enden
  - Plätze als Subnetze müssen mit Plätzen beginnen und enden



2.3 Petrinetze: Beispiel

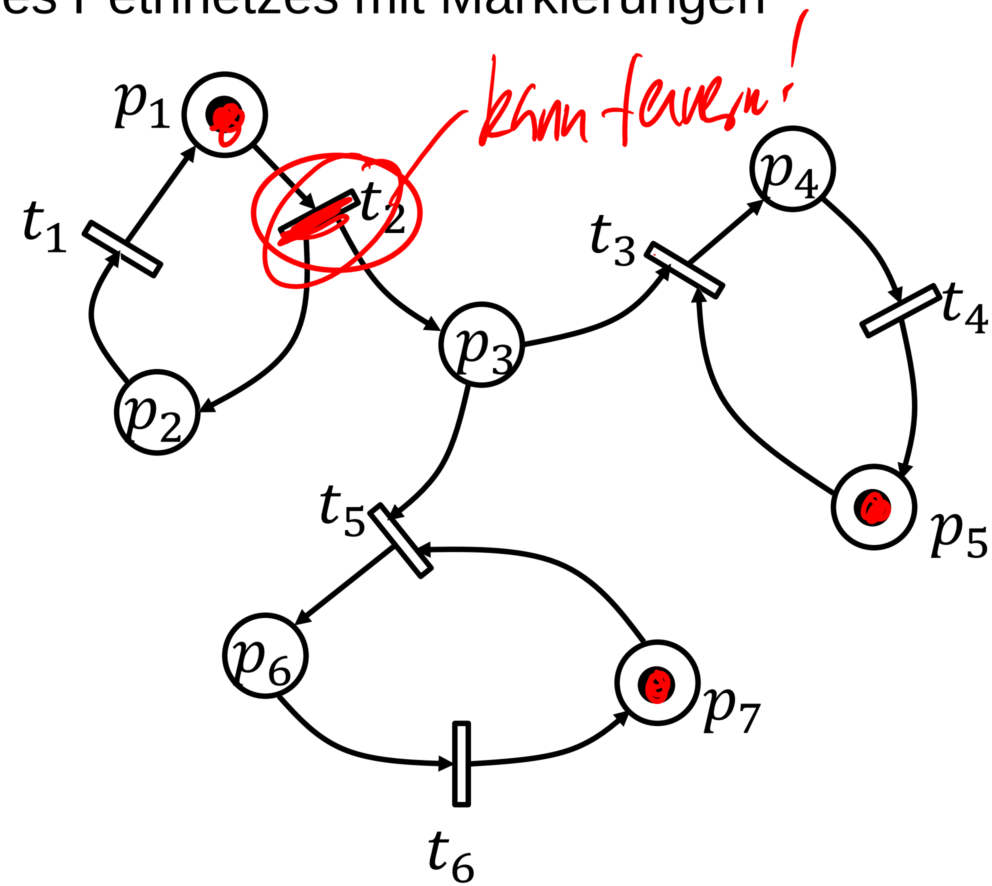
- Darstellung eines einfachen Petrinetzes als Graph



7 Plätze
6 Transitionen

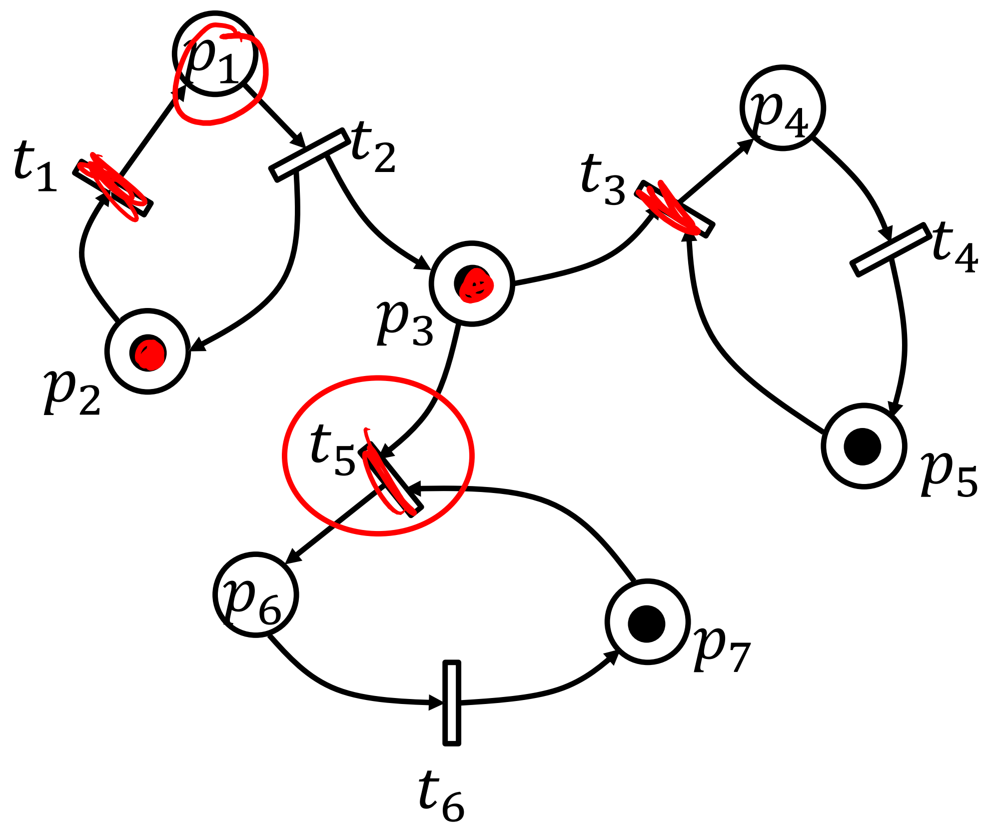
2.3 Petrinetze: Beispiel

- Darstellung eines Petrinetzes mit Markierungen



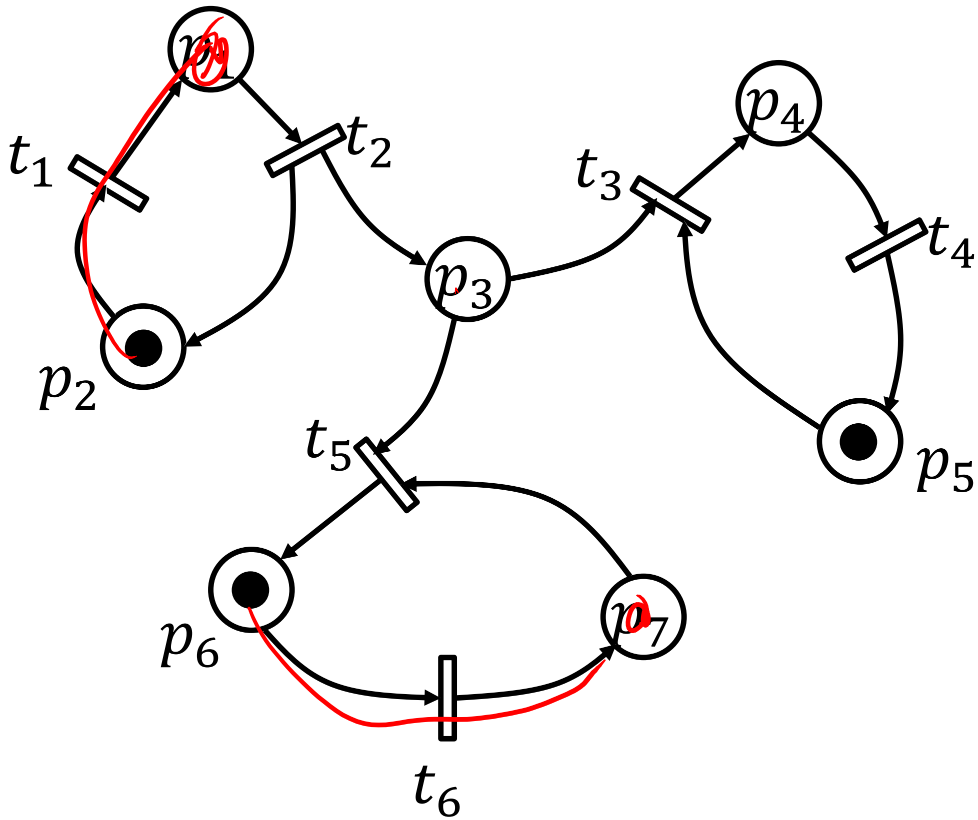
2.3 Petrinetze: Beispiel

- Petrinetz nachdem t_2 geschaltet wurde

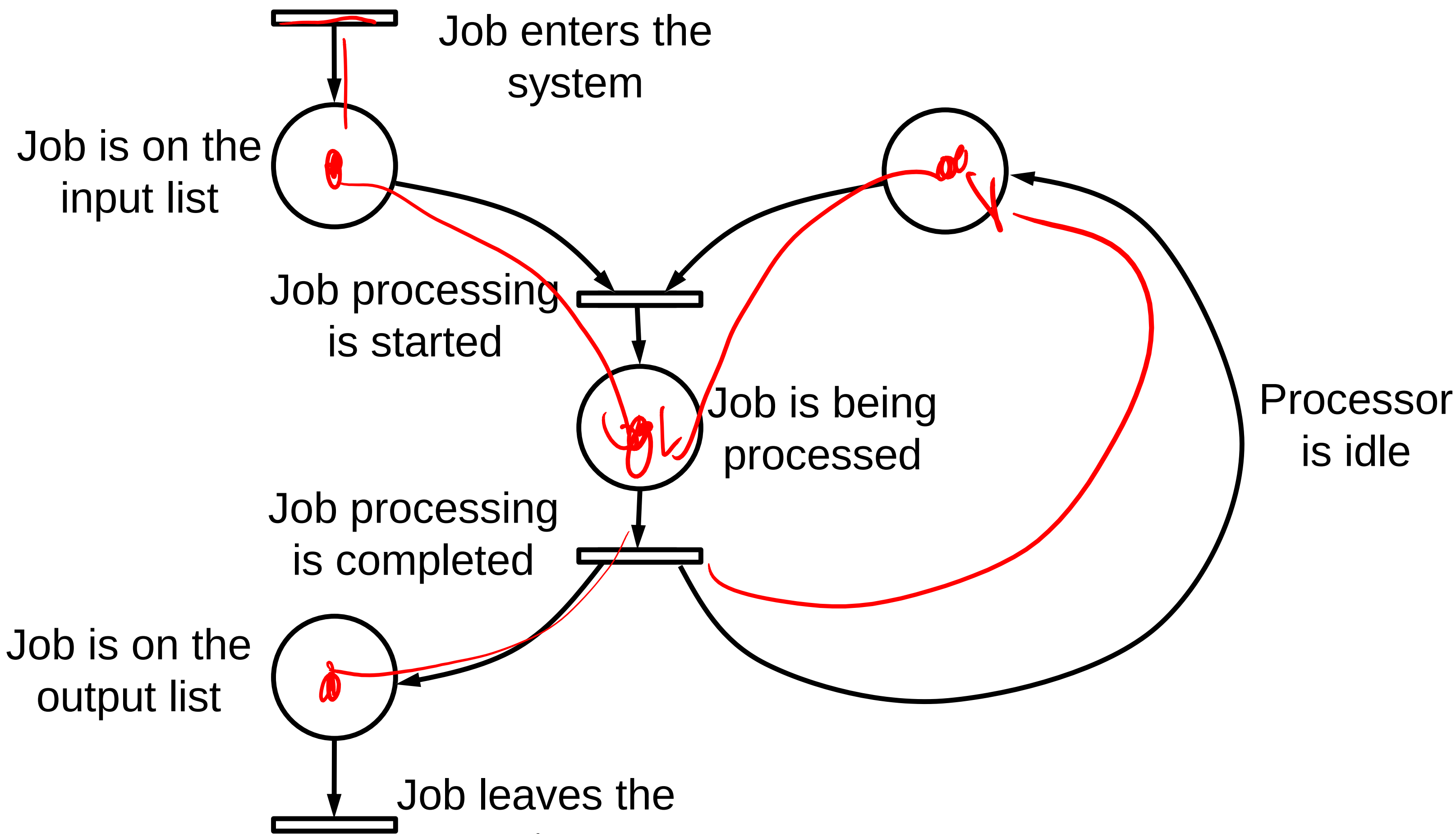


2.3 Petrinetze: Beispiel

- Petrinetz nachdem t_5 geschaltet wurde



Beispiel: Modell einer einfachen Rechenanlage



MOODLE FRAGE

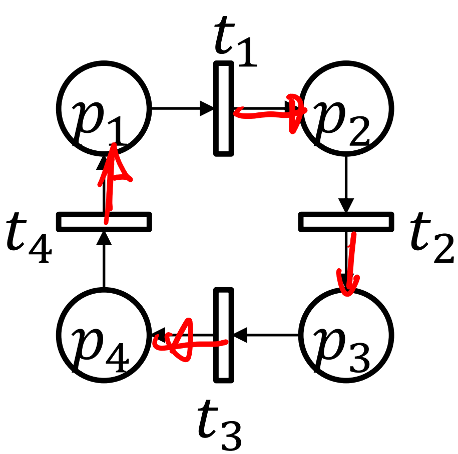
Bitte jetzt auf Moodle mal Fragen beantworten!

2.3 Petrinetze: zeitdiskrete mathematische Darstellung

- Inzidenzenmatrix W
  - Beschreibt die Beziehung zwischen den Knoten und Kanten des Petrinetzes
  - Inzidenzenmatrixdefiniert als $W = W^{+} - W^{-}$

2.3 Petrinetze: zeitdiskrete mathematische Darstellung

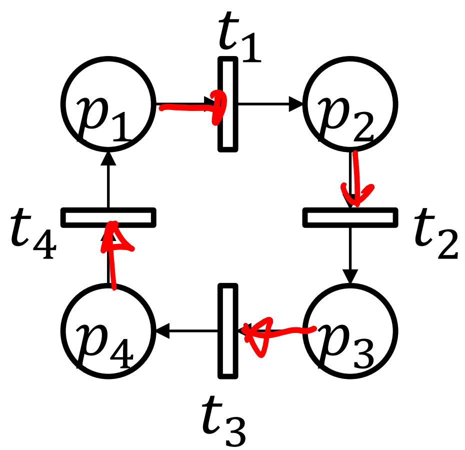
- Matrix W^+ („post-weights“) besteht aus Verbindungen $t_i \rightarrow p_j$
- „Welche Plätze erhalten eine Markierung“



$$W^+ = \begin{array}{c c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

2.3 Petrinetze: zeitdiskrete mathematische Darstellung

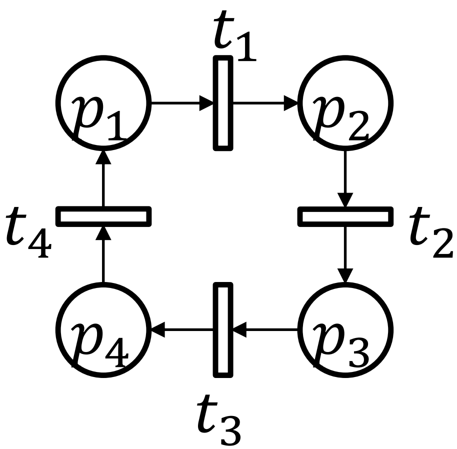
- Matrix W^- („pre-weights") besteht aus Verbindungen $p_i \rightarrow t_j$
- „Aus welchen Plätzen Markierungen entnommen werden"



$$W^- = \begin{array}{cc} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

2.3 Petrinetze: zeitdiskrete mathematische Darstellung

- Inzidenzenmatrix $W = \underbrace{W^+}_{\text{post}} - \underbrace{W^-}_{\text{pre}}$

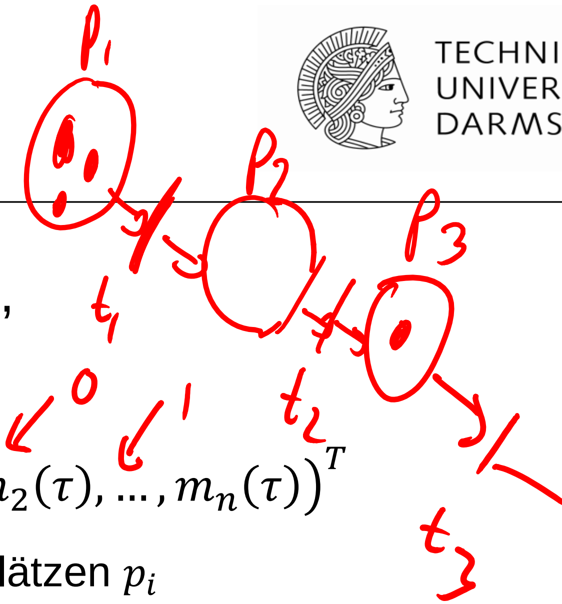


$$W^+ = \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{array}$$

$$W^- = \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \end{array}$$

$$W = W^+ - W^- = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

2.3 Petrinetze: zeitdiskrete Simulation



- Benötigte Informationen: Position der Markierungen, Kapazitäten und Schaltbedingungen
  - Markierungsvektor (Zustandsvektor) $m(\tau) = \big(m_1(\tau), m_2(\tau), \dots, m_n(\tau)\big)^T$
    - Markierungen m_i zu diskreten Zeitpunkten τ bei n-vielen Plätzen p_i
  - Kapazitäten $k = (k_1, k_2, \dots, k_n)^T$ mit $m_i(t) \leq k_i$, $i = 1,2, \dots, n$
  - Transitionen $t = (t_1, t_2, \dots, t_m)^T$

2.3 Petrinetze: zeitdiskrete Simulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bedingungen zur Aktivierung einer Transition t_j zum Zeitpunkt τ
  - Alle „Vor-Plätze“ (VP) müssen hinreichend viele Markierungen besitzen

$$\forall p_i \in VP : \left(m_i(\tau - 1) \geq w_{ij}^{-}\right) \wedge \left(w_{ij}^{-} \neq 0\right)$$

  - Kapazität in allen „Nach-Plätzen“ (NP) darf die maximale Kapazität der Plätze nach Schalten der Transition nicht überschreiten

$$\forall p_i \in NP : \; m_i(\tau - 1) \leq k_i + w_{ij}^{-} - w_{ij}^{+}$$

Checken vorm Feuern

2.3 Petrinetze: zeitdiskrete Simulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Notwendigen Bedingungen:

  Verifikation der Aktivierungsbedingungen

  - Schaltfunktion $u_j = \begin{cases} 1 & \forall\, t_j \text{ aktiviert} \\ 0 & \text{sonst} \end{cases}$
  - Schaltvektor $u = \big(u_1(\tau), u_2(\tau), \dots, u_m(\tau)\big)^T$

- Nach der Transition gilt $m_i(\tau) = m_i(\tau - 1) + w_{ij} \leq k_i$

Schaltvektor!

- Berechnung des Markierungsvektors zum Zeitpunkt τ

$$m(\tau) = m(\tau - 1) + W\, u(\tau)$$

Neuer Zustand Alter Zustand Inzidenzmatrix

MOODLE FRAGE

Bitte jetzt auf Moodle mal Fragen beantworten!

2.3 Petrinetze: zeitdiskrete Simulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zusätzliche Anforderungen für ein Petrinetz
  - Häufigkeiten bestimmter Schaltvorgänge
  - Prioritäten von Transitionen
  - Sequenzen von Transitionen

- Lösungsmöglichkeiten
  - Über die konkrete Wahl der Aktivierungsfunktionen
  - Direkt mit in die Modellierung einfließen lassen

2.3 Petrinetze: charakteristische Eigenschaften

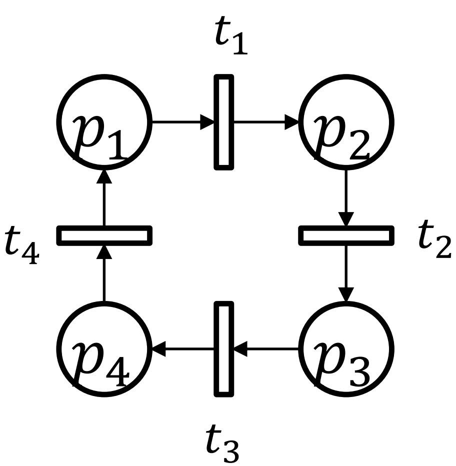
TECHNISCHE UNIVERSITÄT DARMSTADT

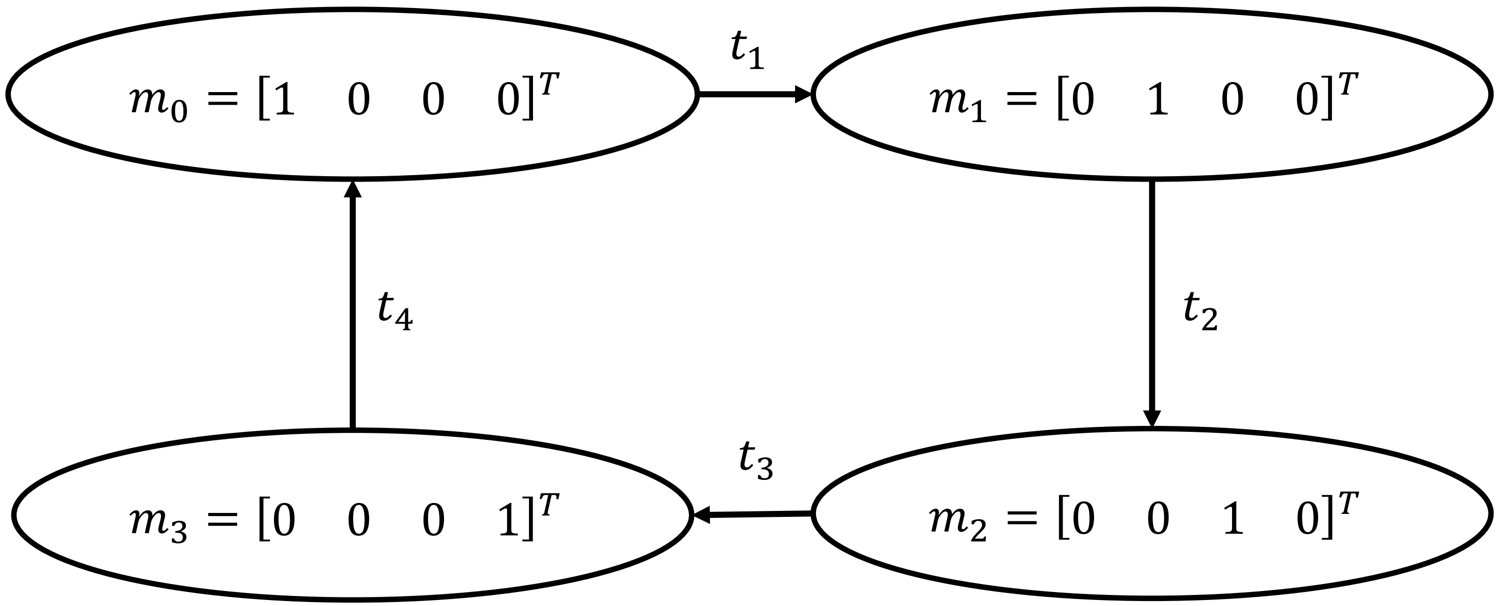
- Erreichbarkeit (reachability)
  - Ein Zustand heißt erreichbar von einem Anfangszustand, wenn eine Schaltsequenz von Anfangszustand zum Zustand existiert

- Erreichbarkeitsgraph (reachability graph)
  - Grundlage für Untersuchung, ob erwünschte Zustände nicht erreicht oder unerwünschte Zustände erreicht werden können und ob Vorgänger gefährlicher Zustände vermieden werden können

2.3 Petrinetze: charakteristische Eigenschaften

- Beispiel: Erreichbarkeitsgraph





2.3 Petrinetze: charakteristische Eigenschaften

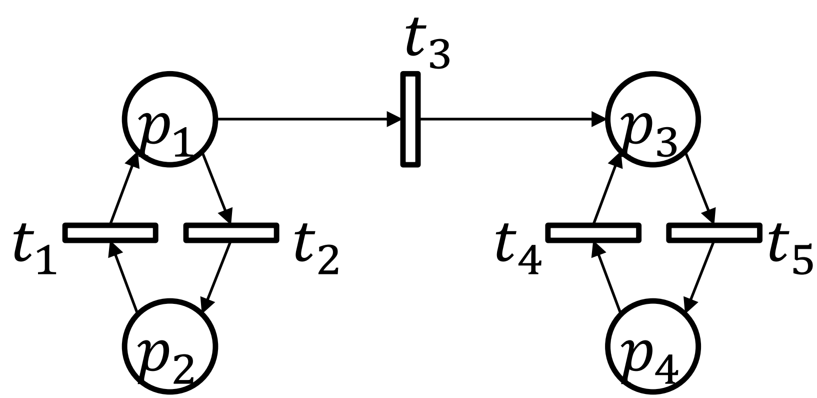
- Beschränktheit (boundedness)
  - Ein Petrinetz heißt <u>beschränkt</u>, wenn in keinem Platz p_i jemals mehr als eine gewisse Maximalzahl von k_i Markierungen vorhanden ist
  - Ist $k_i = 1$, dann ist das Petrinetz sicher

- Verklemmung (deadlock)
  - Eine Verklemmung ist ein Zustand im Petrinetz, in dem keine weitere Transition mehr möglich ist
  - Kann auftreten, wenn eine Transition aus dem Petrinetz herausführt

2.3 Petrinetze: charakteristische Eigenschaften

- Lebendigkeit (liveness)
  - Eine Transition heißt <u>tot</u>, wenn sie bei keiner Folgemarkierung aktivierbar ist
  - Ein Petrinetz heißt <u>nicht lebendig</u>, wenn es eine tote Transition enthält

- Beispiel: nicht lebendiges Petrinetz
  - t_1 und t_2 sind tote Transitionen, nachdem t_3 geschaltet wurde



2.3 Petrinetze: zeitkontinuierlich (Ausblick)

- Ermöglichen Darstellung des zustands- und zeitabhängigen Verhaltens in einem Modell

- Realisierungsmöglichkeit mit Zeitverzögerung $\hat{\tau}$ beim Schalten nach der Aktivierung

- Modellierung zufallsverteilter Zeitverzögerungen der Schaltregeln führt auf stochastische Petrinetze

2.3 Petrinetze: zeitkontinuierlich (Ausblick)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stochastisches Petrinetz $SPN = (P, T, A, K, M_0, R)$
  - Gerichteter, bipartiter Graph
  - $R = \{r_1, r_2, \dots, r_m\}$ ist die Menge der Schaltraten, wobei r_i den Mittelwert der zur Transition t_i gehörenden exponentiell verteilten Schaltrate darstellt
  - Schaltraten können im Allgemeinem von der jeweiligen Markierung des Petrinetzes abhängen

Literatur zu Kap. 2.3 Petrinetzen (Auswahl)

- B. Baumgarten: *Petri-Netze: Grundlagen und Anwendungen* (BI-Wiss.Verlag, 1990)
- H.-M. Hanisch: *Petri-Netze in der Verfahrenstechnik* (Oldenburg, 1992)
- U. Kiencke: *Ereignisdiskrete Systeme* (Oldenburg, 1997)
- M.A. Marsan, G. Balo, G. Conte: *Performance Models of Multiprocessor Systems* (MIT Press, 1986)
- W. Reisig: *Petrinetze – Eine Einführung* (Springer Verlag, 1982)
- W. Reisig: *Systementwurf mit Netzen* (Springer Verlag, 1985)
- Petri Nets World: http://www.informatik.uni-hamburg.de/TGI/PetriNets/