Eigenschaften mobiler und eingebetteter Systeme:

Modellierung

EMES

Dr.-Ing. Matthias Werner
Dipl.-Inf. Jan Richling
Wintersemester 2001/2002

Motivation

- Problem: Vorhersage von Systemverhalten
- Betrachtung des *Real World*^{TM}-Systems: zu komplex
- Nutzung von Modellen zur Erlangung
  - qualitativer
  - quantitativer

  Aussagen

Modell

Ein *Modell* ist eine vereinfachte (abstrahierte) Abbildung eines Teils der wirklichen Welt (System).

- Ein Modell beschreibt einige Aspekte des Systems.
  - Welche? ⇒ Die, die der Modellierer will.
  - Wie? ⇒ Wie der Modellierer es für günstig hält.

- Ziele eines Modells:
  - System verstehen
  - Eigenschaften des Systems zu erkennen
  - Die Kommunikation über das System erleichtern

Arten von Modellen

- **Funktionsmodelle** geben abstrakte Sichten auf das System. Es werden drei Hauptklassen unterschieden:
  - objekt-orientierte Modelle
  - prozeß-orientierte Modelle
  - zustands-orientierte Modelle
- **Analytische Modelle** bieten einen Formalismus, um Ableitungen zu einen gegebenen Problem zu finden. Dabei können zwei Arten von Aussagen gefunden werden:
  - Quantitative Aussagen
  - Qualitative Aussagen
- **Simulationsmodelle** werden eingesetzt, wo analytische Modelle versagen. Ihr Aussagewert beschränkt sich zumeist auf einen Parametersatz

Vielfach können die gleichen Beschreibungstechniken für verschiedene Modellarten genutzt werden.

Petri-Netze (PN)

- C.A. Petri, Anfang der 60er Jahre
- Eine der ersten formalen Modellierungsmethoden
- Orientiert sich an Problembeschreibung, nicht am Rechner
- Ursprünglich vor allen zur Modellierung *asynchroner* Systeme genutzt
- Viele Erweiterungen
- Hier: Darstellung von Zeitabhängigkeiten

Wiederholung: Klassische Petri-Netze (I)

Struktur

- Ein Petri-Netz ist ein *gerichteter Graph*
- Die Knoten des Graphen sind *Plätze* (passiv) und *Transitionen* (aktiv)
- Die gerichteten Kanten verbinden *immer* unterschiedliche Arten von Knoten.
- Plätze, von denen gerichtete Kanten *zu* einer Transition gehen, heißen *Vorbedingung* dieser Transition. Wenn die Kante *von* einer Transition zu einem Platz geht, spricht man von einer *Nachbedingung*

Wiederholung: Klassische Petri-Netze (II)

Dynamik

- Ein Platz kann mit einer oder mehreren Marken (*Token*) belegt sein
- Wenn alle Vorbedingungen einer Transition mit mindestens einer Marke belegt sind, ist diese Transtition *(feuer)bereit* (aktiviert, ready)
- Eine feuerbereite Transition *kann* (muß aber nicht) *feuern* (schalten)
- Beim Feuern wird aus allen Vorbedingungen eine Marke entfernt und allen Nachbedingungen eine Marke hinzugefügt; Feuern ist atomar
- Die Gesamtheit aller Marken zu einem Zeitpunkt nennt man eine *Markierung* des PN; es kann zu Beginn eine *Startmarkierung* (initiale Markierung) gegeben sein

Interpretation der PN-Elemente

Transitionen (Potentielle) Ereignisse im System

Plätze (Potentielle) Eigenschaften eines Systems; mathematisch: Predikate

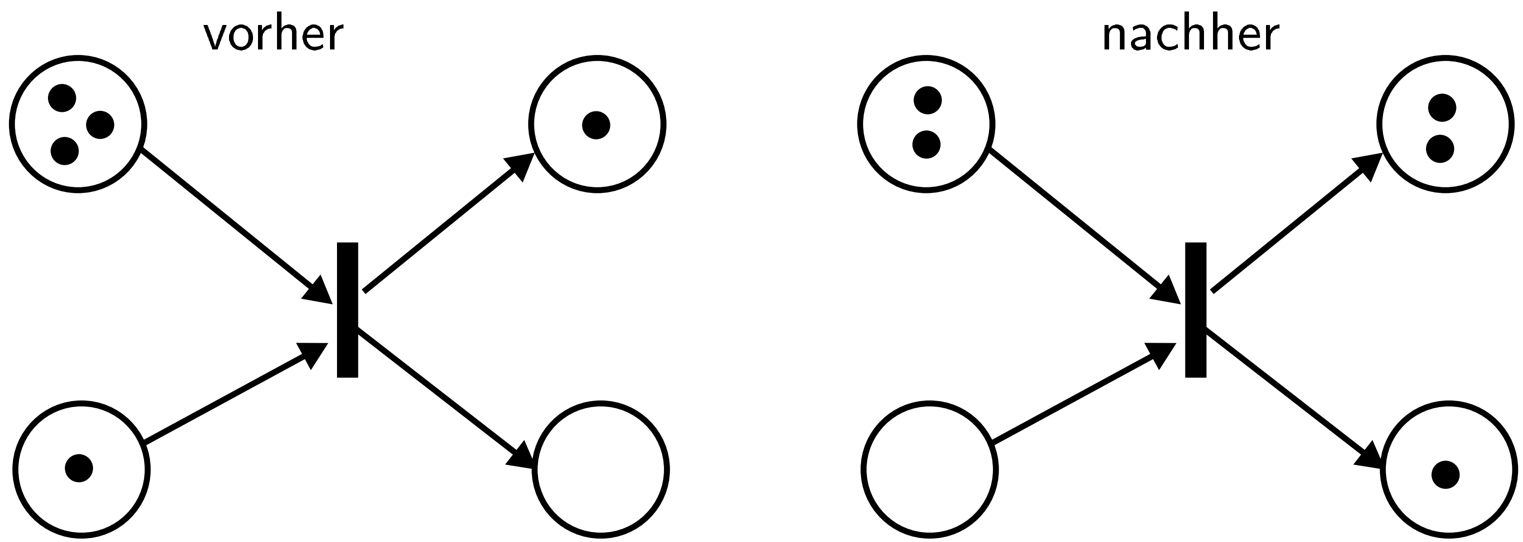
Markierung Systemzustand

Feuern Der Übergang des Systems von einen in einen anderen Zustand

Eingehende Kanten Abhängigkeit eines Ereignisses

Ausgehende Kanten Folge eines Ereignisses

PN – Beispiel



Eigenschaften von PN

Erreichbarkeit Eine Markierung M_n ist *erreichbar* von einer initialen Markierung M_0, wenn entweder $n = 0$ oder eine Feuersequenz $s =< T_1, T_2, ..., T_n >$ existiert, die zur Markierung M_n führt.

Lebendigkeit Ein PN mit einer Markierung M ist *lebendig*, wenn jede Transition des Netzes durch eine Sequenz von endlich vielen Schaltvorgängen Aktiviert werden kann.

Sicherheit Ein PN heißt mit einer Markierung M *sicher*, wenn es keine Schaltsequenz gibt, die zu einer Markierung M' führt, bei der ein Platz mit mehr als einer Marke belegt ist.

Zeitbehaftete PN

- „Klassische“ PN erlauben nur die Definition einer Halbordnung von Ereignissen
  $\Rightarrow$ meist nicht hinreichend für in EMES interessante Systeme
- Möglichkeiten Zeit einzubringen:
  - Zeitbehaftete Plätze
  - Zeitbehaftete Marken
  - Zeitbehaftete Kanten
- Fokus auf letzteren Ansatz
- Zeitabhängige PNs sind meist schwieriger zu analysieren als „normale“ PN
- Möglichkeit der Simulation (Vorsicht, Zustandsraumexplosion!)

Time PN

- Time PN (auch Intervall-PN, IPN) weisen jeder Transition ein Zeitintervall $[t_1, t_2]$ zu.

- Wird eine Transition zum Zeitpunkt t aktiv, so kann sie irgendwann nach $t + t_1$ und vor $t + t_2$ feuern, falls sie dann noch aktiv ist.

- Bleibt eine Transition aktiv, so muß sie spätestens zum Zeitpunkt $t + t_2$ feuern.

- Feuern erfolgt zeitlos.

- Ein „klassisches“ PN entspricht einem IPN, bei dem jeder Transition $[0, \infty]$ zugewiesen wird.

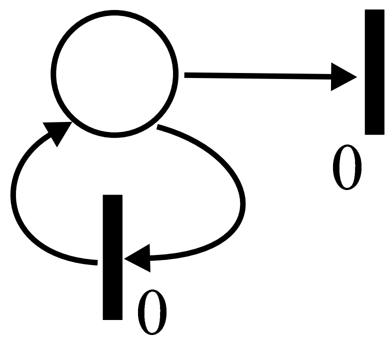
Timed PN

- Timed PN (auch Duration-PN, DPN) weisen jeder Transition eine Zeitdauer d zu.

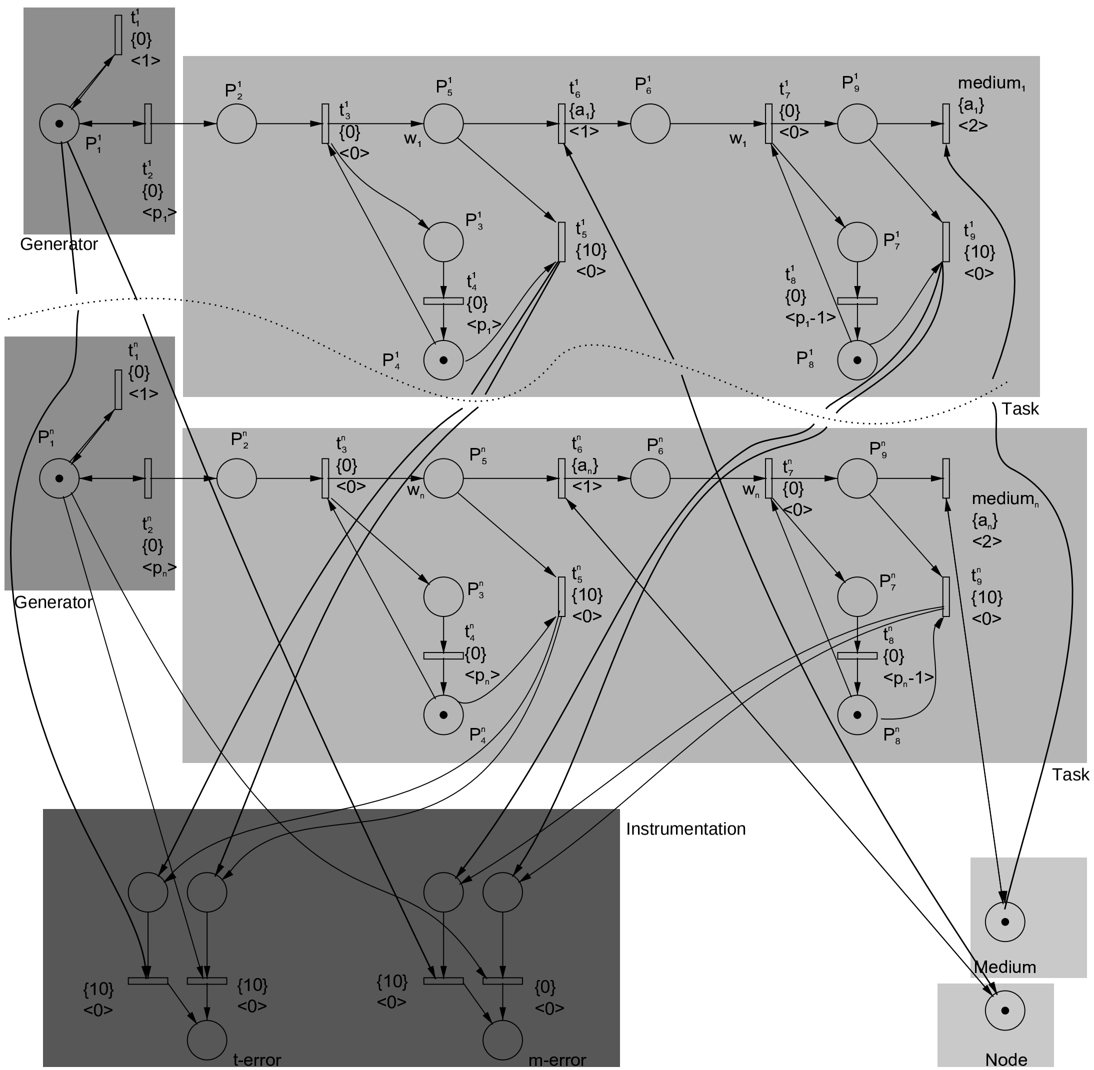
- Eine aktivierte Transition feuert sofort.

- Das Feuern ist nach d Zeiteinheiten beendet.

- Ein „klassisches“ PN kann durch durch ein DPN durch zusätzliche Schleifen dargestellt werden.

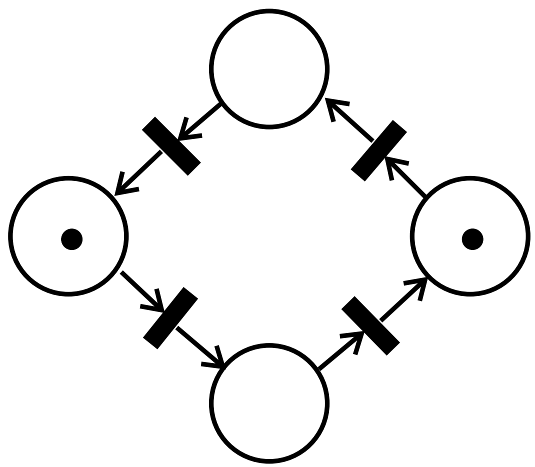


Beispiel: MSS mit DPN

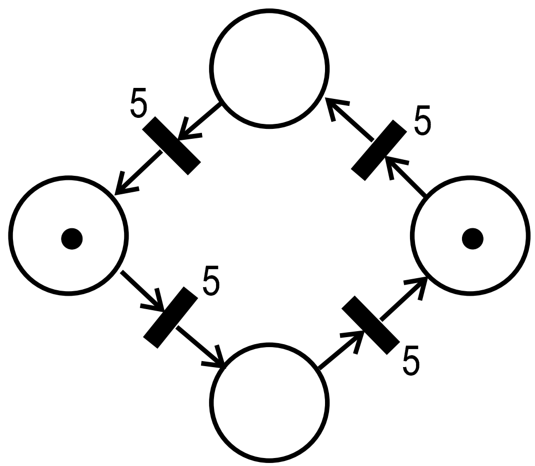


Zeitbehaftete vs. Zeitlose PN

- Zeiten schränken den Zustandraum eines PNs ein
- Garantien (Invarianten) im zeitlosen PN halten auch im zeitbehafteten, *aber nicht umgekehrt*
- Beispiel: Sicherheit in einem PN und DPN:



unsicher



sicher

Statistisches Zeitverhalten

- Häufig sind nicht exakte Zeiten bekannt
- Dafür bekannt: statistische Verteilungen
- Beispiele:
  - Lastverhalten
  - Ausfallverhalten
- Bekannte Ansätze:
  - Warteschlangen
  - Markov-Ketten
  - Stochastische Petri-Netze

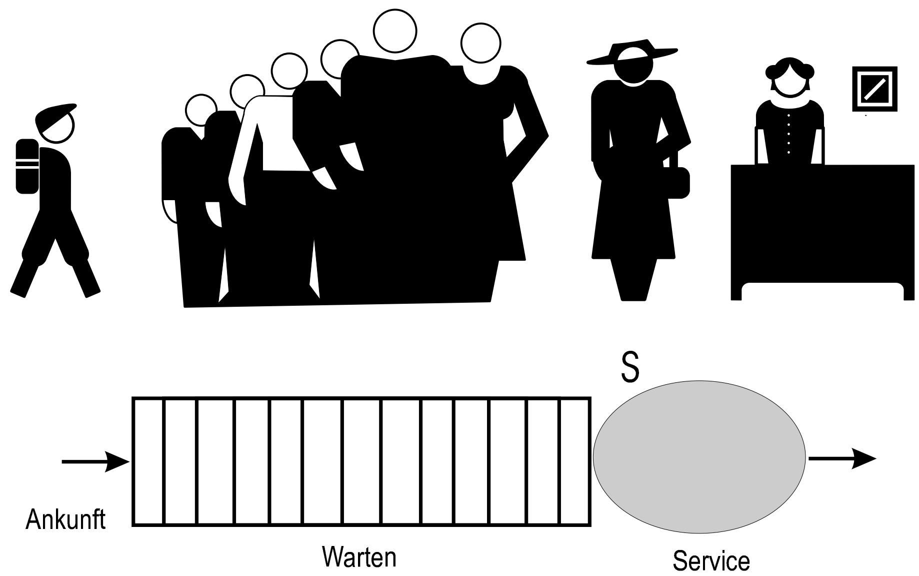
Warteschlangen

- Warteschlangen-Theorie (Queuing-Theory) wird vor allen zur Leistungs- und Zeitbewertungen in Client-Server-Systemen eingesetzt

- Grundidee: Bankschalter
  - Kunden, die unregelmäßig ankommen
  - Warteschlange vor den Schaltern
  - Ein oder mehrere Schalter
  - Die genaue Ankunftszeit der einzelnen Kunden ist nicht bekannt
  - Die Verweildauer am Schalter ist nicht bekannt

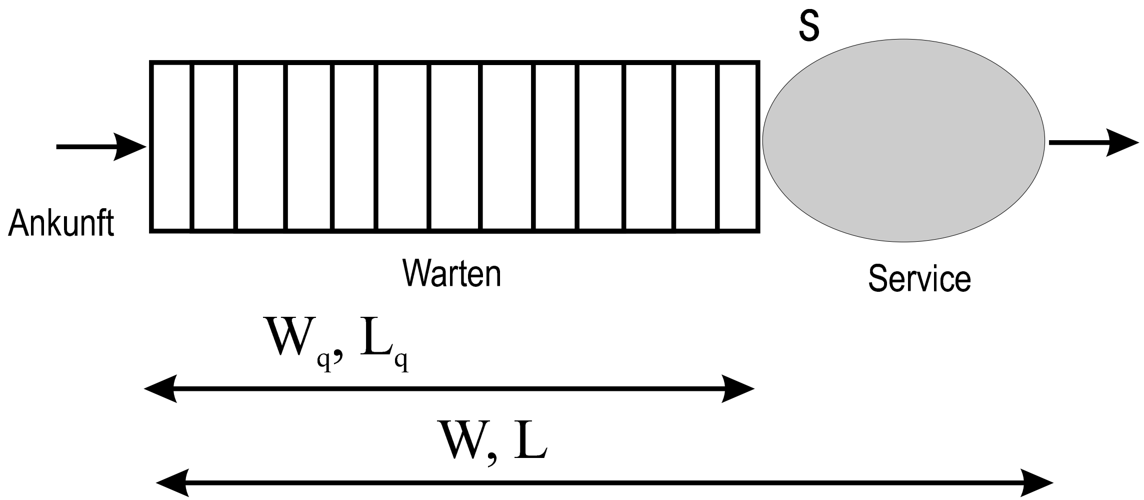
- ⇒ Annahme von Zufallsverteilungen (Modellannahme!)

Warteschlangenmodell



- λ: Durchschnittliche Ankunftsrate
  (durchschnittliche Anzahl der ankommenden Kunden pro Zeiteinheit)
- μ: Durchschnittliche Servicerate
  (durchschnittliche Anzahl von Kunden, die pro Zeiteinheit von einem Server bedient werden)
- s: Anzahl der Server

Metriken



W_q Durchschnittliche Zeit der Kunden in der Schlange

W Durchschnittliche Zeit der Kunden im System

L_q Durchschnittliche Anzahl der Kunden in der Warteschlange

L Durchschnittliche Anzahl von Kunden im System

KENDALL-Notation

A|B|C|K|m|Z

A Beschreibung des Ankunftsprozeßes: M - Exponentialverteilung, G - Normalverteilung, ...
B Beschreibung des Serviceprozeßes (wie A)
C Anzahl der Server
K maximale Kapazität der Server (normal: ∞)
m Nutzerpopulation (normal: ∞)
Z Serviceart (FIFO, LIFO, zufällig)

- Häufig werden die Parameter K, m und Z weggelassen, dann ist $\infty|\infty|$ FIFO gemeint

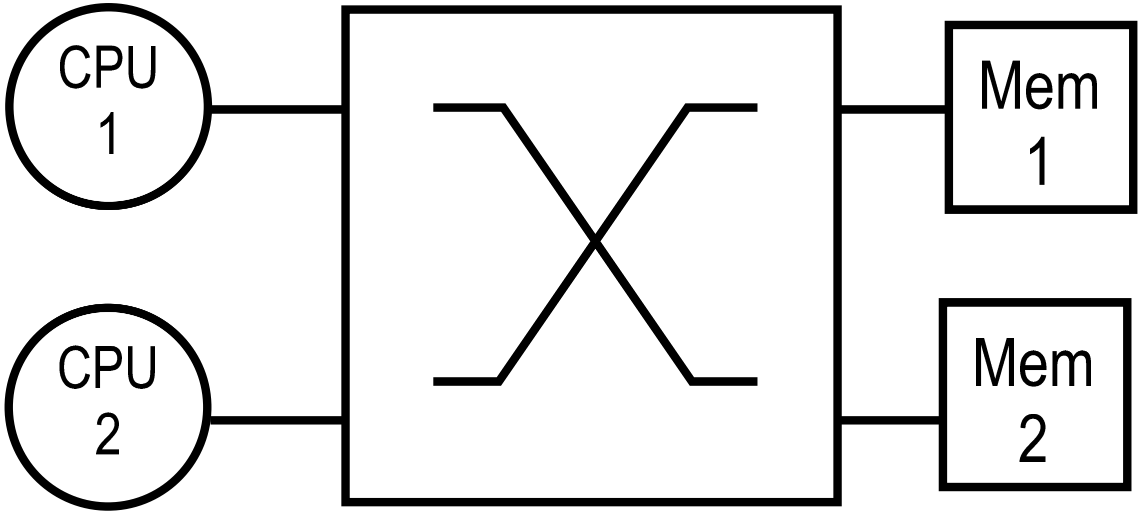
M | M | 1 - Warteschlangen

- Annahmen:
  - Zwischenankunftszeit ist exponentialverteilt
  - Servicezeit ist exponentialverteilt
  - Nur ein FIFO-Server
- Vorteile:
  - Viele Vorgänge in der Natur verhalten sich nach Exponentialverteilungen
  - Leichte Berechenbarkeit
- Wichtige Formeln:
  - $L = \max\left(\frac{\lambda}{\lambda-\mu}\right)$
  - $W = \frac{1}{\mu-\lambda}$

MARKOV-**Ketten**

- nach A.A. MARKOV (häufig auch MARKOW), russischer Mathematiker, 1856 – 1922
- Verallgemeinerung von Warteschlangen auf *Zustandsfolgen*
- System besitzt eine Menge von diskreten Zuständen
- Übergänge zwischen Zuständen besitzen Wahrscheinlichkeiten
- Die Wahrscheinlichkeit des Übergangs in einen Folgezustand hängt *ausschließlich* vom gegenwärtigen Zustand ab, also insbesondere *nicht* von schon durchlaufenen Zuständen (Markov-Eigenschaft)
- Unterscheidung von *zeitdiskreten* und *zeitkontinuierlichen* Markov-Ketten
- Wahrscheinlichkeiten werden nach BAYES verknüpft (bedingte Wahrscheinlichkeit)

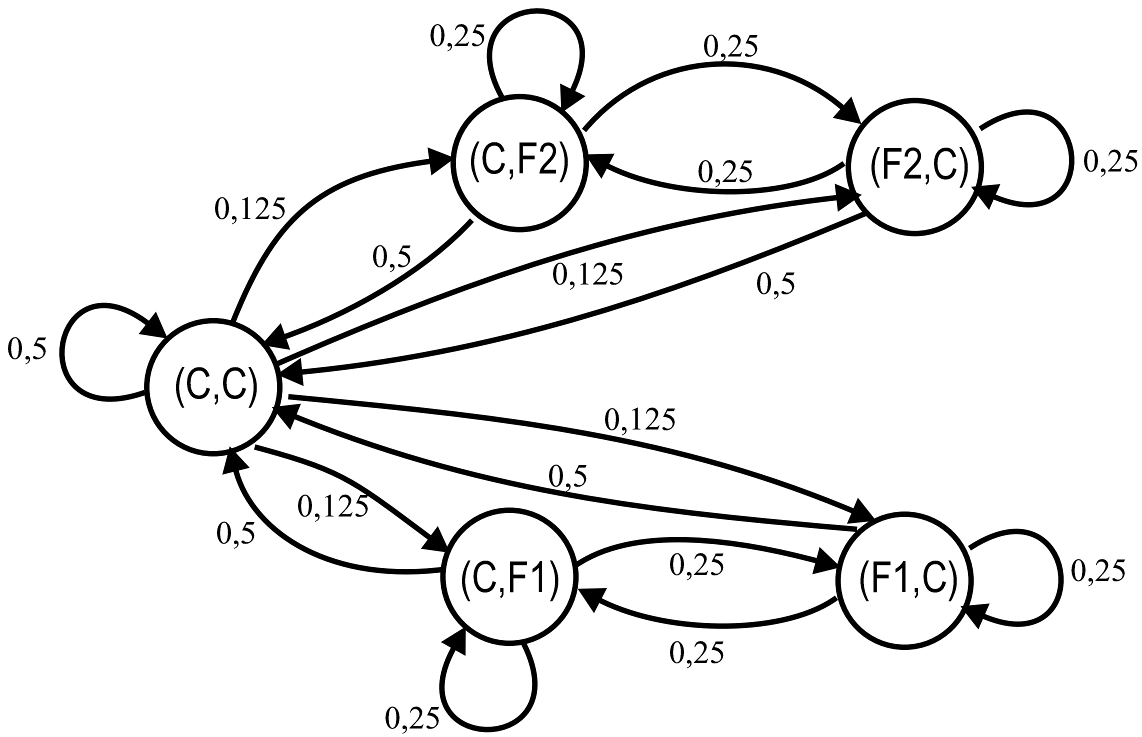
Beispiel für Markov-Ketten (I)



- Zugriff auf jeden Speicher gleichwahrscheinlich
- Bei Konflikt wird zufällig entschieden
- Erfolgloser Zugriff wird wiederholt

- Wie groß ist der Durchsatz zu einer Zeit t?

Beispiel für Markov-Ketten (II)



(C, C) Erfolg bei beiden Prozessoren
(C, Fi) Mißerfolg bei Prozessor 2 beim Zugriff auf Speicher i
(Fi, C) Mißerfolg bei Prozessor 1 beim Zugriff auf Speicher i

Beispiel für Markov-Ketten (III)

Übergangsmatrix

$$P = \begin{bmatrix} 0.5 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.5 & 0.25 & 0 & 0.25 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0.25 \\ 0.5 & 0.25 & 0 & 0.25 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0.25 \end{bmatrix}$$

Zustandswahrscheinlichkeit nach n Schritten: $p_n = p_{n-1}P = p_0P^n$

- Im Beispielsfall ist die partielle Zustandswahrscheinlichkeit für (C, C) einfach: Egal, welcher Zustand zu $t = i$ war, mit der Wahrscheinlichkeit von 0.5 ist zu $t = i + 1$ der Zustand (C, C)
- Durchsatz: $D = 2 \cdot \Pr(C, C) + 1 \cdot (1 - \Pr(C, C))$
- $D = 0.75$

Andere Modellierungstechniken

Hier seien nur kurz noch ein paar wichtige Ansätze genannt:

Endliche Automaten Graphen, deren Knoten mit dem Vokabular der akzeptierten Sprache bezeichnet werden. Beispiele: Statecharts, Modecharts, Timed Automata

Temporallogiken erlauben logische Betrachtungen von Abläufen, haben ursprünglich aber keine Zeitbeschreibung. Erweiterungen u.a. zur Real Time Logic oder Real-Time Temporal Logic

Stochastische Netze „Zusammenschluß“ der Ideen von PN und Markov-Ketten ermöglichen genauere Beschreibung von Systemen mit stochastischen Vorgängen. Sie sind häufig schwer zu analysieren und dienen meist der Simulation. Beispiele: Stochastic PN, Stochastic Activity Nets (SAN), Stochastic Reward Nets

Tools

- Wesentlich bei der Bewertung einer Modellierungstechnik ist das Vorhandensein von Tools

- Frei verfügbar sind u.a.:

  Uppaal Verifikationstool für Timed Automata, erlaubt u.a. Zustandserreichbarkeit; graphische Nutzeroberfläche, keine vollständige Semantik

  tINA Analysator für zeitbehaftete PN

  UltraSAN Analysetool für SAN, graphische Nutzeroberfläche

- Für allgemeine Aussagen im Echtzeitbereich (Proofer) bisher wenig Tool-Unterstützung

Das Komplexitätsproblem