Petrinetze

- Petrinetze (PN) eignen sich besonders für die Untersuchung und Darstellung asynchroner Systeme und simultaner Abläufe (parallele Prozesse).
- Insbesondere im Zuge des Versuchs, Rechenanlagen durch Parallelisierung der Ausführung zu beschleunigen, wurden Petrinetze zu einem sehr wichtigen und häufig benutzten Beschreibungsmittel.

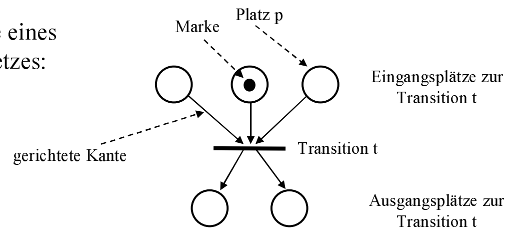
- Ein Standard-Petrinetz ist ein gerichteter, bipartiter Graph

PN = (P,T,A,M')

mit einer endlichen Menge von Plätzen $P = \{p_1,p_2,...,p_n\}$,
einer endlichen Menge von Transitionen $T = \{t_1,t_2,...,t_m\}$,
einer endlichen Menge gerichteter Kanten $A = \{a_1,a_2,...,a_k\}$,
und einer Anfangsmarkierung $M' = \{m'_1,m'_2,...,m'_n\}$.

- Ein bipartiter Graph ist ein Graph, dessen Knotenmenge in zwei Klassen (bei Petrinetzen Kreise- und Rechteckknoten) partitioniert ist und dessen Kanten jeweils nur zwischen unterschiedlich klassifizierten Knoten liegen.
- Die Kreisknoten werden als Plätze, die Rechteckknoten als Transitionen bezeichnet. Gerichtete Kanten verbinden Plätze mit Transitionen und Transitionen mit Plätzen.

Graphiksymbole eines Standard-Petrinetzes:



- Plätze können Marken beinhalten, die als schwarze Punkte gekennzeichnet sind.
- Der Zustand eines Petrinetzes ist definiert durch die Anzahl der Marken in jedem Platz und wird durch einen Vektor $M = \{m_1, m_2, ..., m_n\}$ repräsentiert, dessen i-te Komponente die Anzahl der Marken in Platz p_i angibt.
- Der Zustand eines Petrinetzes wird deshalb auch Markierung genannt.

- Der Anfangszustand eines Petrinetzes wird durch die Anfangsmarkierung $M' = \{m'_1, m'_2, ..., m'_n\}$ vorgegeben.
- Markierungswechsel ergeben sich aus der sogenannten Schaltregel, worin die Schaltbereitschaft und das Schalten einer Transition definiert werden.

- Eine Transition ist schaltbereit (aktiviert), wenn alle ihre Eingangsplätze markiert sind (mindestens eine Marke enthalten).
- Eingangsplätze einer Transition sind diejenigen Plätze, von denen ein Pfeil zur Transition hinführt.
- Schaltbereite Transitionen können schalten (feuern), wobei eine Marke aus jedem Eingangsplatz weggenommen und in jeden Ausgangsplatz eine Marke hinzugefügt wird.
- Ausgangsplätze einer Transition sind diejenigen Plätze, von denen ein Pfeil von der Transition ausgehend hinführt.

- Zur weiteren Einarbeitung in die theoretischen Grundlagen von Petrinetzen und deren Anwendungsmöglichkeiten eignen sich:
  - Baumgarten, Bernd: Petri-Netze: Grundlagen und Anwendungen; BI-Wiss.-Verl., 1990.
  - Reisig, Wolfgang: Petrinetze - Eine Einführung; Springer-Verlag Berlin Heidelberg, 1982.
  - Reisig, Wolfgang: Systementwurf mit Netzen; Springer-Verlag Berlin Heidelberg, 1985.

- Mit den Standard-Petrinetzen lassen sich zwar Abläufe, nicht aber deren zeitliches Verhalten quantitativ darstellen.
- Zur Behandlung von Zeitaspekten muß das Petrinetz-Konzept erweitert werden.
- Dies erfolgt i.a. durch Erweiterung der Schaltregel so, daß Verzögerungszeitdauern berücksichtigt werden.
- Diese können entweder fest vorgegeben (deterministisch) oder zufallsverteilt (stochastisch) sein.
- Zur Modellierung eignen sich beide Ansätze, wobei jedoch eine Tendenz zu den stochastischen Zeiten besteht.

- Stochastische Petrinetze (SPN) gehen aus den Standard-Petrinetzen hervor, wobei jeder Transition eine exponentiell verteilte Schaltrate zugewiesen wird.
- Ein stochastisches Petrinetz ist ebenfalls ein gerichteter, bipartiter Graph

SPN = (P,T,A,M',R)

mit P, T, A und M' wie bereits eingeführt und $R = \{r_1,r_2,...,r_m\}$.

- R ist die Menge der Schaltraten, wobei r_i den Mittelwert der zur Transition t_i gehörenden, exponentiell verteilten Schaltrate darstellt.

- Für die analytische Modellierung ist die Exponentialverteilung die wichtigste und auch die am leichtesten handhabbare Verteilung, da sie als einzige kontinuierliche Verteilung die Markov-Eigenschaft der Gedächtnislosigkeit (memoryless property) besitzt.

- Schaltraten können von der jeweiligen Markierung des Petrinetzes abhängen. Man spricht dann von markierungsabhängigen Schaltraten.

- Oft ist es nicht erforderlich, jeder Transition eine exponentiell verteilte Schaltrate zuzuordnen.
- Eine dahingehende, auf den SPN´s aufbauende Erweiterung läßt sowohl zeitbehaftete als auch zeitlose Transitionen zu.
- Zeitlose Transitionen schalten ohne Verzögerung, wenn sie schaltbereit sind.
- Diese Erweiterung läuft unter dem Namen Generalized Stochastic Petri Nets und wird im folgenden mit GSPN abgekürzt.

- Bei einem GSPN handelt es sich ebenfalls um einen gerichteten, bipartiten Graph

  GSPN = (P,T,A,M',R')

  mit P, T, A und M' wie bereits eingeführt und
  $R' = \{r_1, r_2, \ldots, r_{m'}\}$.
- Dies entspricht im wesentlichen einem stochastischen Petrinetz bis auf die Menge R', die jetzt nur noch $m'<m$ Elemente beinhaltet, wobei mit m' die Zahl der zeitbehafteten Transitionen im Netz angegeben wird.

- Auflösung von Konfliktsituationen in GSPNs (mehrere Transitionen sind gleichzeitig aktiviert):
- Umfaßt die Menge der gleichzeitig aktivierten Transitionen H ausschließlich Zeittransitionen feuert Transition t_i ($i \in H$) mit der Wahrscheinlichkeit:

$$\frac{r_i}{\sum_{k \in H} r_k}$$

- Falls die Menge H der aktivierten Transitionen sowohl Zeittransitionen als auch zeitlose Transition umfaßt, feuert eine der zeitlosen Transitionen.
- Bei $k \geq 1$ gleichzeitig aktivierten zeitlosen Transitionen feuert, falls nicht vorher anders festgelegt, jede dieser Transitionen mit der Wahrscheinlichkeit:

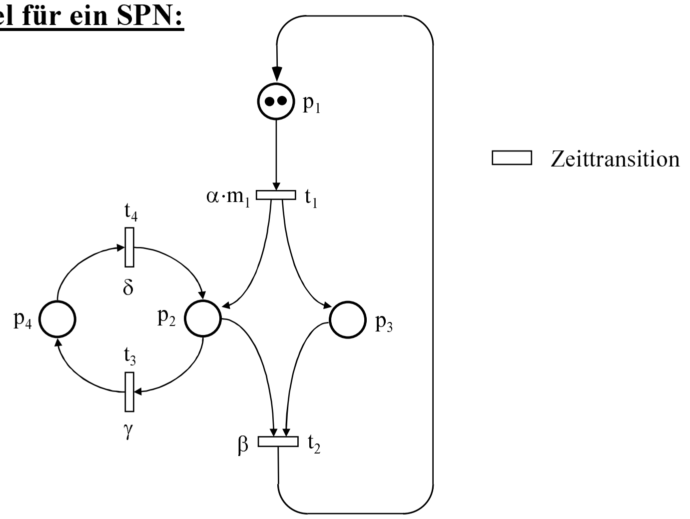
$$\frac{1}{k}$$

- Ziel der Auswertung von SPN- und GSPN-Modellen ist vor allem die Bestimmung der Zustandswahrscheinlichkeiten aller möglichen Systemzustände (Netzmarkierungen) im Gleichgewicht (sämtliche Einschwingvorgänge sind abgeklungen und die Leistungsgrößen sind zeitunabhängig).
- Aus diesen Zustandswahrscheinlichkeiten lassen sich dann, wie bei den Warteschlangennetzen die Mittelwerte aller anderen Leistungskenngrößen des Netzes ableiten. Beispiele dafür sind:
  - $p(m_i=k)$: Wahrscheinlichkeit, daß sich k Marken in Platz p_i befinden; mit m_i: Anzahl von Marken in Platz p_i
  - Mittlere Anzahl von Marken in jedem Platz p_i des Netzes:

$$\overline{p}_i = \sum_{k=1}^{\infty} k \cdot p(m_i = k)$$

Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow) 15

Beispiel für ein SPN:



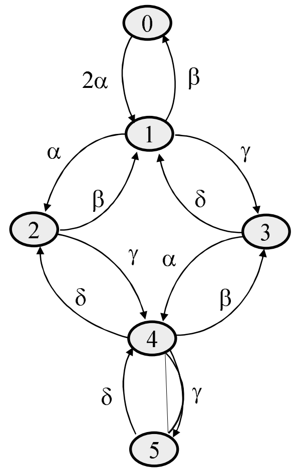
Vorlesung „Simulationstechnik"
(Dr. M. Syrjakow) 16

Mögliche Netzzustände (m_1,m_2,m_3,m_4)

- Zustandsraum

  R = {0 = (2,0,0,0),
  1 = (1,1,1,0),
  2 = (0,2,2,0),
  3 = (1,0,1,1),
  4 = (0,1,2,1),
  5 = (0,0,2,2)}

- Zustandsübergangsdiagramm



- Eine umfassende Einführung in die theoretischen Grundlagen, Analyseverfahren und Anwendungsmöglichkeiten stochastischer Petrinetze (SPN und GSPN) gibt:

  - Marsan, M.A.; Balbo, G.; Conte, G.:
    Performance Models of Multiprocessor Systems;
    The MIT Press, 1986.

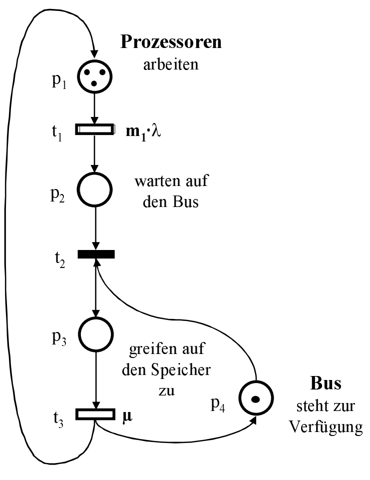
- Petrinetze können mit Rechnerunterstützung "ausgeführt" werden.
- Ein auf GSPN basierendes Modellierungswerkzeug ist das Programmpaket GreatSPN (http://www.di.unito.it/~greatspn/).
- Es besteht aus einem Graphikeditor zur Eingabe und Modifikation von GSPN-Modellen und aus Programmen zur Modellüberprüfung und Modellabwicklung.

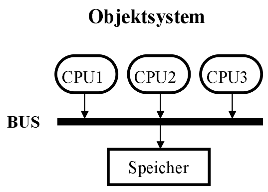
- Bei der Modellüberprüfung werden Struktureigenschaften eines Netzes (Lebendigkeit, Sicherheit und Invarianten) bestimmt.
- Die Modellabwicklung zur Ermittlung der gesuchten Leistungskenngrößen kann sowohl analytisch als auch simulativ durchgeführt werden.
- Der Benutzer hat zudem die Möglichkeit, jeden Zustandswechsel im Netz (Schalten einer Transition) von Hand auszuführen, wodurch ein schrittweises Verfolgen des Netzablaufs ermöglicht wird.

- Ergebnis der Modellabwicklung durch GreatSPN ist die Bestimmung der Zustandswahrscheinlichkeiten aller möglichen Systemzustände (Netzmarkierungen) im Gleichgewicht (sämtliche Einschwingvorgänge sind abgeklungen und die Leistungsgrößen sind zeitunabhängig).

- Aus diesen Zustandswahrscheinlichkeiten lassen sich analog zu den Warteschlangennetzen die Mittelwerte aller anderen Leistungsgrößen des Netzes (Durchsatz, Auslastung, etc.) ableiten.

GSPN-Beispielmodell:





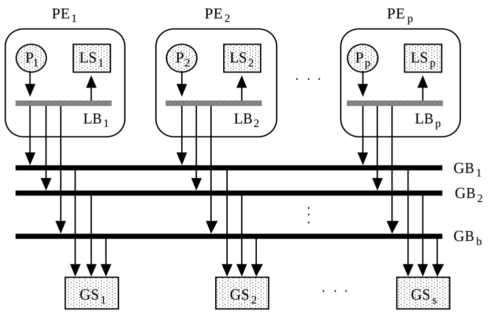
- Gegenstand der Modellierung in diesem Beispiel ist der Zugriff dreier Prozessoren auf einen gemeinsamen, globalen Speicherbereich über einen gemeinsamen Bus.
- Die drei Marken in p_1 repräsentieren die in ihrem lokalen Speicherbereich arbeitenden Prozessoren.
- Nach einer zufälligen, exponentiell verteilten Zeit mit Mittelwert $1/(m_1*\lambda)$ schaltet die Transition t_1 (markierungsabhängige Schaltrate) und eine Marke aus p_1 befindet sich nun in p_2 (einer der Prozessoren ist im Begriff auf den gemeinsamen Speicherbereich zuzugreifen).

- An Transition t_2 wird geprüft, ob der gemeinsame Bus (repräsentiert durch eine Marke in p_4) zur Verfügung steht.
- Ist das der Fall, schaltet t_2 und die Marken aus p_2 und p_4 verschwinden, während eine neue Marke in Platz p_3 hinzugefügt wird.
- Nach einer zufälligen, exponentiell verteilten Zeit mit Mittelwert $1/\mu$ (Modellierung der Speicherzugriffsdauer) schaltet die Transition t_3 und die Marke aus p_3 verschwindet, während in p_1 und p_4 jeweils eine neue Marke hinzugefügt wird (Ende des Speicherzugriffs und Busfreigabe).

- Ziel der Modellierung in diesem Beispiel könnte die Ermittlung der mittleren Anzahl der in ihrem lokalen Speicherbereich arbeitenden Prozessoren sein, welche hier der mittleren Anzahl von Marken in p_1 entspricht.

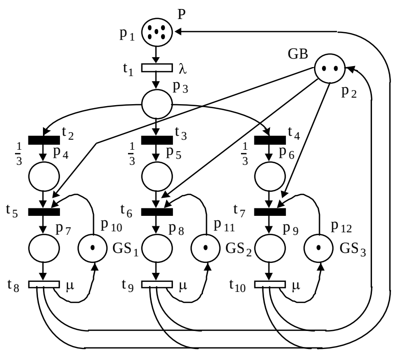
- Eine weitere interessante Leistungsgröße wäre die mittlere Anzahl der auf den Bus wartenden Prozessoren, die durch die mittlere Anzahl von Marken in p_2 gegeben ist.

Fallstudie: Speichergekoppeltes Multiprozessorsystem mit Mehrfachbus

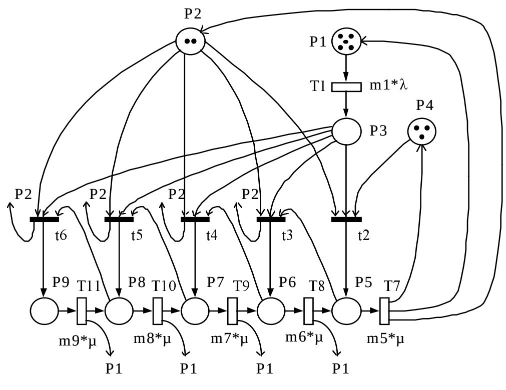


siehe auch http://goethe.ira.uka.de/people/syrjakow/mod_vorlesung/seiten/modvorl.html

GSPN-Modell des speichergekoppelten Multiprozessorsystems mit Mehrfachbus



Ein weiteres GSPN-Modell des speichergekoppelten Multiprozessorsystems mit Mehrfachbus



Gegenüberstellung GSPN - Erweiterte Warteschlangennetze

Vorteile GSPN:

+ sowohl analytisch als auch simulativ auswertbar
+ analytische Modellauswertung liefert exakte Ergebnisse

Nachteile GSPN:

– Marken repräsentieren sowohl Aufträge als auch Ressourcen, was zu schwer überschaubaren Modellen führt
– analytische Modellauswertung nur bei geringer Modellkomplexität praktisch durchführbar
– exponentiell verteilte Schaltraten sind nicht immer eine gute Approximation der tatsächlich vorliegenden Verteilungen

Vorteile erweiterter Warteschlangennetze:

+ übersichtliche Modelle mit leicht nachvollziehbaren Abläufen (Aufträge werden durch Jobs, Ressourcen durch passive Knoten modelliert)
+ die Jobvariable ermöglicht eine nähere Charakterisierung von Aufträgen

Nachteile erweiterter Warteschlangennetze:

– i.a. nur durch Simulation auswertbar (keine exakten Ergebnisse)
– die simulative Auswertung komplexer Modelle ist meist sehr rechen- und speicheraufwendig