

Musterklausur 1 für PMfE

(Musterklausur SS '94)

Aufgabe 1 (9 Punkte)

Fragenkatalog

- a) Erläutern Sie den Unterschied zwischen den Begriffen „Überlebenswahrscheinlichkeit“ und „Verfügbarkeit“.
- b) Erläutern Sie die Begriffe „heiße“, „kalte“ und „warme Redundanz“.
- c) Welche Bedeutung besitzt das Schlüsselwort „grain“ in dem Programmiersystem GranAda?
- d) Erläutern Sie den Unterschied zwischen den Auswertungsmethoden „Analyse“ und „Simulation“ und geben Sie jeweils einen Vorteil beider Methoden an.
- e) Wann ist eine Wahrscheinlichkeitsverteilung „gedächtnislos“? Geben Sie ein Beispiel für eine gedächtnislose Wahrscheinlichkeitsverteilung und zeichnen Sie ihren ungefähren Verlauf.
- f) Gegeben sei ein stochastischer Prozeß. Erläutern Sie den Unterschied zwischen den „transienten“ und „stationären Zustandswahrscheinlichkeiten“ des Prozesses.
- g) Nennen Sie drei verschiedene Arten von (nicht zeiterweiterten) „Petri-Netzen“ und erläutern Sie kurz die Unterschiede.
- h) Welche Arten von Transitionen gibt es in einem „generalized stochastic Petri net“ (GSPN) ? Welcher stochastische Prozeß liegt einem solchen Netz zugrunde?
- i) Wie wird die Glaubwürdigkeit von statistischen Schätzungen bestimmt? Nennen Sie zwei typische Probleme dieser Glaubwürdigkeitsbestimmung und übliche Lösungen.

Aufgabe 2 (15 Punkte)

Modellierung eines Ada-Programms mit zeiterweiterten Petri-Netzen

Im folgenden ist ein Ada-Programm eines Client-Server-Systems gegeben.

```
type DATA is private;

task SERVER is
  entry PROVIDE(ITEM : out DATA);
end SERVER;

task CLIENT;

task body SERVER is
begin
  loop
    select
      accept PROVIDE(ITEM : out DATA) do
        CREATE(ITEM);
      end SERVE;
    else
      SERVER_COMPUTE;
    end select
  end loop;
end SERVER;

task body CLIENT is
begin
  loop
    CLIENT_COMPUTE
    select
      SERVER.PROVIDE(ITEM);
    or
      delay 2.0;
      -- Timeout, Fehlerbehandlung:
      ERROR_HANDLING
    end select;
  end loop;
end CLIENT
```

Das Programm besteht aus zwei Prozessen, die den Client und den Server repräsentieren. Der Client-Prozess fordert zyklisch einen Dienst des Server-Prozesses an. Der Dienst ist durch ein

Rendezvous realisiert. Der Client-Prozeß führt eine Zeitüberwachung durch: falls der Dienst nicht innerhalb einer Zeitschranke angeboten wird, muß eine Fehlerbehandlung durchgeführt werden. Der Server-Prozeß führt interne Arbeiten durch, falls keine Dienstanforderung besteht.

a) Modellieren Sie das Ada-Programm als zeiterweitertes Petri-Netz. Dabei werde angenommen, daß folgende Anweisungen zeitbehaftet sind:

- SERVER_COMPUTE
- CLIENT_COMPUTE
- PROVIDE(ITEM)
- ERROR_HANDLING
- delay-Anweisung

Die Dauern aller anderen Anweisungen werden vernachlässigt. Vergeben Sie Namen für die Stellen und Transitionen, die ihre Bedeutungen erkennen lassen.

Hinweis: Modellieren Sie die Prozesse jeweils als eine Marke, die in einem Teilnetz zirkuliert. Das Rendezvous kann dann dargestellt werden durch Kanten, die beide Teilnetze verbinden.

b) Nennen Sie eine Stellen- und eine Transitionsinvariante Ihres Netzes, ohne die entsprechenden Gleichungssysteme zu lösen.

c) Belegen Sie nun die zeitbehafteten Transitionen mit Schaltzeiten. Dabei werden die folgenden Verteilungen angenommen:

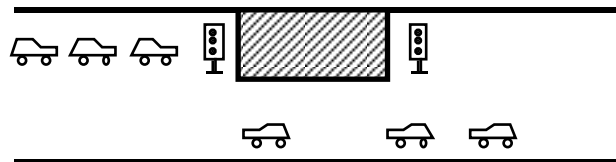
- SERVER_COMPUTE: exponentiell mit Mittelwert 0.1 Sekunden
- CLIENT_COMPUTE: exponentiell mit Mittelwert 1 Sekunde
- PROVIDE(ITEM): deterministisch 0.1 Sekunden
- ERROR_HANDLING: exponentiell mit Mittelwert 2 Sekunden
- delay-Anweisung: entsprechend der Angabe im Programm

d) Geben Sie einen Ausdruck in TimeNET-Notation an für die Wahrscheinlichkeit, daß der Client-Prozeß auf die Annahme eines Rendezvous wartet, während der Server-Prozeß mit internen Berechnungen beschäftigt ist.

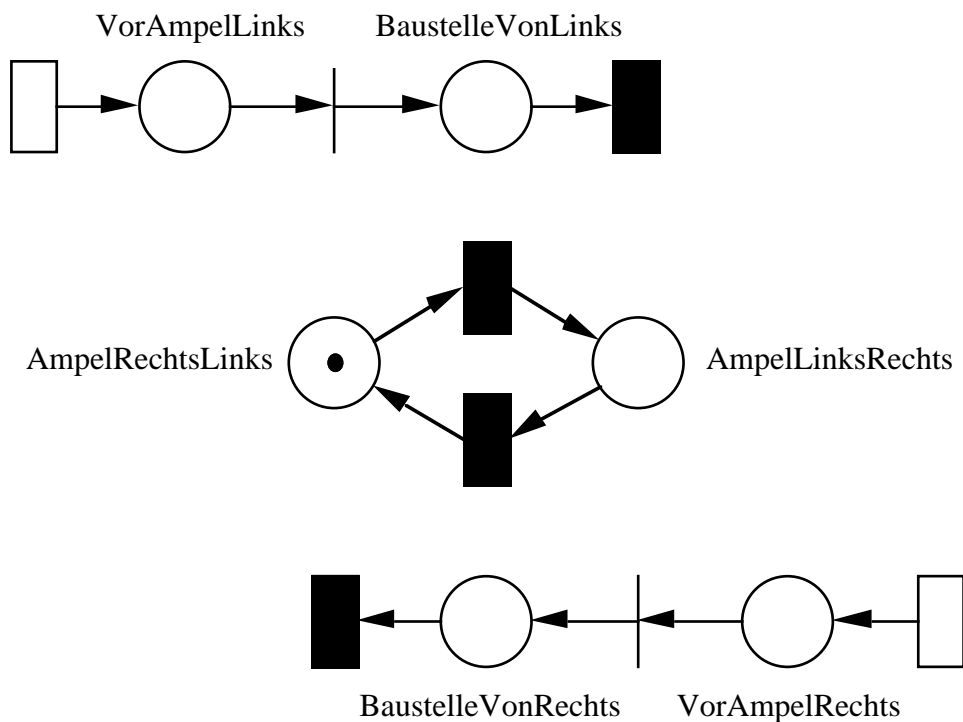
Aufgabe 3 (8 Punkte)

Synchronisation in Petri-Netzen

Das folgende Bild zeigt eine Situation im Straßenverkehr.



Abgebildet ist eine Straße mit Verkehr in beiden Richtungen. Eine Baustelle verengt die Fahrbahn, so daß Fahrzeuge nur in einer der beiden Richtungen fahren können. Eine Ampel regelt den Verkehr. Das folgende Bild zeigt ein DSPN-Modell der Situation. Dem DSPN fehlen noch Elemente, um die Synchronisation der Verkehrsströme korrekt nachzubilden.



Der Verkehr von links nach rechts wird durch das obere Teilnetz mit den Stellen *VorAmpelLinks* und *BaustelleVonLinks* modelliert. Der Verkehr von rechts nach links wird

durch das untere Teilnetz mit den Stellen *VorAmpelRechts* und *BaustelleVonRechts* modelliert. Das mittlere Teilnetz mit den Stellen *AmpelRechtsLinks* und *AmpelLinksRechts* repräsentiert die Ampelschaltung.

Marken in *VorAmpelLinks* repräsentieren Fahrzeuge, die vor der linken Ampel warten und Marken in *BaustelleVonLinks* Fahrzeuge, die den Streckenabschnitt der Baustelle in Richtung von links nach rechts befahren. Entsprechend repräsentieren Marken in *VorAmpelRechts* Fahrzeuge vor der rechten Ampel und Marken in *BaustelleVonRechts* Fahrzeuge, die den Streckenabschnitt der Baustelle in Richtung von rechts nach links befahren. Befindet sich eine Marke auf der Stelle *AmpelRechtsLinks*, so ist der Verkehr von rechts nach links freigegeben. Befindet sich die Marke auf der Stelle *AmpelLinksRechts*, so ist der Verkehr von links nach rechts freigegeben. Eine Gelbphase ist nicht modelliert.

a) Ergänzen Sie das DSPN um weitere Elemente, um die Synchronisation der Verkehrsströme zu modellieren. Folgende Eigenschaften sollen im Modell repräsentiert sein:

1. Fahrzeuge dürfen nur bei grüner Ampel den Streckenabschnitt der Baustelle befahren,
2. Fahrzeuge dürfen den Streckenabschnitt der Baustelle nicht befahren, falls sich noch Autos aus der Gegenrichtung auf ihm befinden,
3. maximal drei Fahrzeuge können den Streckenabschnitt der Baustelle befahren.

Hinweis: Die Verwendung von sechs hemmenden Kanten ist ausreichend.

b) Ist der Erreichbarkeitsgraph des DSPNs endlich oder unendlich?

c) Ergänzen Sie das DSPN um weitere Elemente, um zu modellieren, daß maximal fünf Fahrzeuge vor jeder Ampel warten dürfen.

Hinweis: Die Verwendung von zwei hemmenden Kanten ist ausreichend.

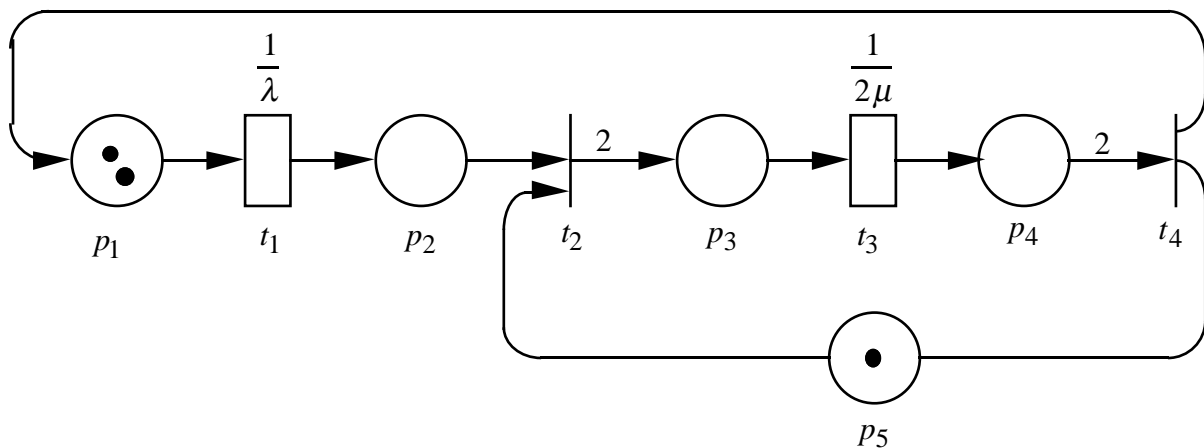
d) Geben Sie einen Ausdruck in TimeNET-Syntax an für die mittlere Anzahl der vor beiden Ampeln wartenden Fahrzeuge.

e) Kann das unter c) erhaltene DSPN numerisch analysiert werden?

Aufgabe 4 (18 Punkte)

Analyse von zeiterweiterten Petri-Netzen

Das folgende Bild zeigt ein zeiterweitertes Petri-Netz. Es stellt eine Warteschlange mit begrenztem Stauraum dar. Kundenankunftszeiten sind exponentiell verteilt mit der Rate λ . Kunden werden bedient, indem sie zwei exponentiell verteilte Zeitdauern mit der Rate 2μ durchlaufen.



- Um welche Klasse von zeiterweiterten Petri-Netzen handelt es sich?
- Erzeugen Sie den reduzierten Erreichbarkeitsgraphen und geben Sie eine Numerierung der Markierungen an.
- Ermitteln Sie die zugrundeliegende Markow-Kette. Geben Sie den Zustandsgraph und die Generatormatrix an.
- Geben Sie die Zustandsgleichungen an, die die transienten und stationären Zustandswahrscheinlichkeiten beschreiben.
- Berechnen Sie die stationären Zustandswahrscheinlichkeiten für $\lambda = 1$ und $\mu = \frac{1}{2}$.

- f) Ermitteln Sie die mittlere Anzahl von Marken auf p_1 und die mittlere Schaltfrequenz von t_1 .

Es werde nun angenommen, daß die Schaltzeit von t_1 deterministisch ist.

- g) In diesem Fall ist eine numerische analytische Lösung möglich. Warum?
- h) Geben Sie die zugeordnete Markow-Kette von t_1 an und erläutern Sie kurz Ihre Bedeutung.