Ludwig-Maximilians-Universität München
Institut für Informatik
Lehrstuhl für Mobile und Verteilte Systeme
Prof. Dr. Claudia Linnhoff-Popien

mobile and distributed systems group

Betriebssysteme im Wintersemester 2017/2018
Übungsblatt 9

Abgabetermin: 08.01.2018, 18:00 Uhr

Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 18. – 22. Dezember 2017
Besprechung der H-Aufgaben in den Tutorien vom 08. – 12. Januar 2018

Aufgabe 39: (T) Petri-Netze: Multiprocessoring

(– Pkt.)

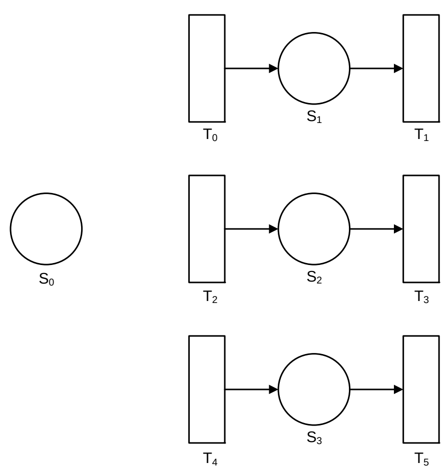
In dieser Aufgabe betrachten wir ein Zwei-Prozessor-System mit drei rechenbereiten Prozessen P_1, P_2 und P_3. Gehen Sie zunächst von folgenden Bedingungen aus:

– Auf einer CPU kann immer nur genau ein Prozess gleichzeitig rechnen.

– Ein Prozess kann auf einer oder aber auch gleichzeitig auf beiden CPUs abgearbeitet werden

a. Welches Thread-Konzept (KLT oder ULT) muss ein Betriebssystem umsetzen, damit die zweite der oben genannten Bedingungen erfüllt ist? Begründen Sie Ihre Antwort!

b. Im folgenden sollen das Zwei-Prozessor-System und die drei Prozesse mit einem Petrinetz modelliert werden. Die Stellen dieses Petrinetzes können als Zustände interpretiert werden. Dabei entspricht S_0 dem Zustand *"Ein bzw. zwei Prozessoren stehen zum Rechnen zur Verfügung"* und die Stelle S_1 (S_2, S_3) entspricht dem Zustand "P_1 *(*P_2, P_3*) rechnet auf einem bzw. zwei der Prozessoren"*. Ergänzen Sie nun das nachfolgend angedeutete Petrinetz um eine minimale Anzahl an weiteren Marken, Stellen, Transitionen und Kanten (insofern jeweils erforderlich), so dass zu jedem Zeitpunkt entweder genau ein Prozess beide Prozessoren nutzen kann, oder zwei Prozesse genau je einen Prozessor nutzen können.



c. Geben Sie zu dem erarbeiteten Petrinetz den Erreichbarkeitsgraphen an.

d. Ist das System Deadlock-frei, teilweise verklemmt oder verklemmt? Begründen Sie Ihre Antwort mit Hilfe des eben skizzierten Erreichbarkeitsgraphen!

e. Um die Architektur des Systems besser debuggen zu können, sollen Prozesse so beschränkt werden, dass sie immer nur auf genau einer CPU zur selben Zeit ausgeführt werden können. Dazu muss also sichergestellt werden, dass auf den beiden Prozessoren zu jedem Zeitpunkt nur unterschiedliche Prozesse rechnen können.

   Man kann diese Einschränkung sehr einfach modellieren, indem man die Kapazität der Stellen S_1, S_2 und S_3 auf 1 limitiert. Erweitern Sie das Petrinetz aus Aufgabe b) so, dass Ihre Modellierung den selben Effekt erzielt, dabei aber keine Einschränkung der Kapazität einer Stelle voraussetzt. Lösen Sie das Problem durch Verwendung einer minimalen Anzahl an Stellen, Marken, Kanten und Transitionen.

f. Geben Sie zu dem erweiterten Petrinetz aus Aufgabe e) den Erreichbarkeitsgraphen an.

Aufgabe 40: (T) Apfelplantage

(– Pkt.)

Im folgenden betrachten wir den Arbeitsablauf und die dafür notwendigen Schritte auf einer Apfelplantage.
Auf der Apfelplantage arbeiten 2 Feldarbeiter und ein Koch. Feldarbeiter müssen essen, um Äpfel pflücken zu können. Eine Portion Apfelmus reicht aus, damit sich ein Feldarbeiter genug gestärkt fühlt, um 2 Äpfel pflücken zu können (außer Apfelmus gibt es nichts zu essen). Vor dem Pflücken muss ein Arbeiter satt sein. Nach dem Pflücken ist der Feldarbeiter wieder hungrig. Der Koch kann aus 12 Äpfeln 8 Portionen Apfelmus kochen. Er beginnt immer erst dann mit dem Kochen, wenn er mindestens 12 Äpfel hat. Um die schwere Kocharbeit verrichten zu können, muss sich der Koch selbst zuvor auch mit einer Portion Apfelmus stärken. Vor dem Kochen hat der Koch satt zu sein, danach ist er wieder hungrig.

a. Auf welches in der einschlägigen Literatur häufig zitiertes Problem lässt sich die Situation der Apfelplantage abbilden?

b. Gehen Sie nun von der folgenden Konstellation aus:
   Zu Beginn seien 3 Portionen Apfelmus und 6 Äpfel in der Vorratskammer der Apfelplantage vorhanden. Außerdem seien der Koch und beide Feldarbeiter zu Beginn hungrig.

   Beginnt man nun den Ablauf der Arbeitsschritte auf der Plantage auszuführen, landet man sehr schnell in einer Deadlock Situation.

   Im folgenden sollen Sie sich diesen Sachverhalt klar machen.

   Zeichnen Sie zunächst das Petrinetz, das die Situation auf der Apfelplantage modelliert. Überlegen Sie sich dazu erst, welche Zustände Sie für den Koch und die Feldarbeiter bzw. den genannten Ressourcen als Stellen modellieren müssen. Vergeben Sie an alle Stellen und Transitionen Bezeichner, so dass deren Semantik klar wird.
   Hinweis: Sie benötigen für die Modellierung gewichtete Transitionen.

c. Geben Sie nun einen repräsentativen Auschnitt des Erreichbarkeitsgraphen des Petri-Netzes aus Aufgabe b) an, der verdeutlicht, dass hier ein Deadlock vorliegt. Der Ausschnitt des Graphen soll mindestens 2 Pfade enthalten, die zu einem Deadlock führen. Geben Sie außerdem eine Abschätzung über die Anzahl der möglichen Pfade des Erreichbarkeitsgraphen an. Versuchen Sie im Erreichbarkeitsgraphen (wenn möglich) Markierungsmengen wieder zu verwerten, d.h. duplizieren Sie diese Menge nicht, wenn dieselbe Menge auf zwei oder mehr verschieden Pfaden erreichbar ist.

d. Warum bleibt das Apfelplantagen-Beispiel sicher in einem Deadlock stecken?

e. Geben Sie 2 mögliche Änderungen des Apfelplantagen-Beispiels an, so dass der Ablauf *möglicherweise* deadlockfrei läuft.

Aufgabe 41: (H) Petri-Netze: Druckerwarteschlange und Erreichbarkeitsgraph

(18 Pkt.)

a. *Modellierung einer Druckerwarteschlange*

Gehen Sie für die folgenden beiden Teilaufgaben von folgender Situation aus:

Ein Computer ist mit einem Drucker verbunden. Um zu vermeiden, dass der Computer einen Druckauftrag an den Drucker sendet, während er gerade noch mit einem anderen Druckauftrag beschäftigt ist, schickt der Computer seine Druckaufträge an eine Warteschlange. Wenn die Warteschlange nicht leer ist, entfernt der Drucker einen Druckauftrag aus der Warteschlange und druckt diesen.

(i) Modellieren Sie den oben beschriebenen Sachverhalt mit einem Petrinetz. Gehen Sie zudem von folgenden Bedingungen aus:

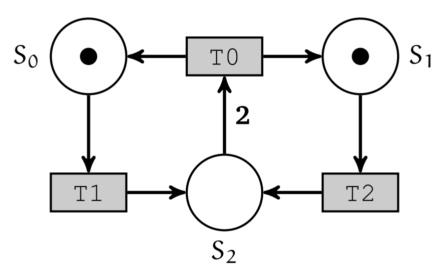
- Der Computer kann immer nur einen Druckauftrag in die Warteschlange stellen. Er kann diesen Schritt aber beliebig oft wiederholen.
- Der Drucker hat zwei Zustände: Entweder druckt er oder er wartet auf den nächsten Druckauftrag.
- Die Warteschlange besitzt eine unendliche Kapazität, d.h. es können beliebig viele Druckaufträge in die Warteschlange gestellt werden.
- Es spielt keine Rolle in welcher Reihenfolge der Drucker die Druckaufträge aus der Warteschlange entfernt.

Verwenden Sie zur Modellierung des Petrinetzes eine minimale Anzahl an Stellen, Marken und Transitionen. Beschriften Sie verwendeten Stellen sinnvoll.

(ii) Damit die Warteschlange nicht überläuft, soll deren Kapazität nun begrenzt werden. Erweitern sie Ihr Petrinetz nun so, dass die Warteschlange nur noch maximal drei Aufträge akzeptiert. Verwenden Sie zur Modellierung wieder eine minimale Anzahl an Stellen, Marken und Transitionen. Führen Sie dabei **keine** Begrenzung der Kapazität der Stellen ein.

b. *Erstellung eines Erreichbarkeitsgraphen:*

Bearbeiten Sie die folgenden beiden Teilaufgaben in Bezug auf das unten abgebildete Petrinetz.



(i) Geben Sie den Erreichbarkeitsgraphen für das dargestellte Petrinetz an. Verwenden Sie für eine Markierung M_j folgende Anordnung der Stellen $S_i (i \in \{0, \dots, 2\})$:

$$M_j = (S_0, S_1, S_2)$$

Der oben abgebildete Zustand entspricht also z.B. folgender Markierung:

$$M_0 = (1, 1, 0)$$

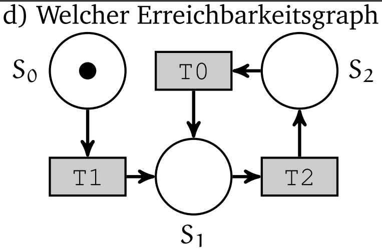
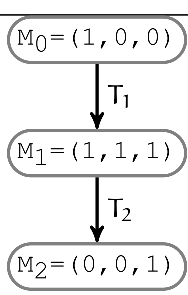
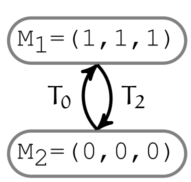
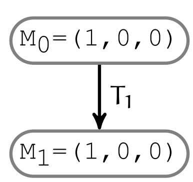
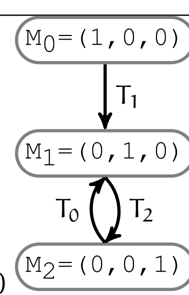
Kennzeichnen Sie jede Kante zwischen zwei Markierungen mit der entsprechenden Transition, die für den Übergang von einer Makierung zur anderen verantwortlich ist.

(ii) Kann im dargestellten Petrinetz ein Deadlock entstehen? Begründen Sie Ihre Antwort.

Aufgabe 42: (H) Einfachauswahlaufgabe: Multiprocessing

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

<table>
<tr><td colspan="4">a) Welche Aussage bezüglich neu hinzukommender Marken zu einer Stelle im Nachbereich einer schaltfähigen Transition ist korrekt?</td></tr>
<tr><td>(i) Die Anzahl der Stelle zugeordneten Marken nimmt ab.</td><td>(ii) Die Marken verändern die der Stelle zugeordnete Kapazität.</td><td>(iii) Die neu hinzukommenden Marken dürfen die Kapazität der Stelle nicht überschreiten.</td><td>(iv) Die neu hinzukommenden Marken dürfen die Kapazität der Stelle überschreiten.</td></tr>
<tr><td colspan="4">b) Angenommen eine Stelle s befindet sich ausschließlich im Vorbereich einer schaltenden Transition t ($s \in (\cdot t) \setminus (t \cdot)$). Wie berechnet sich allgemein die unmittelbare Folgemarkierung $M'(s)$ der Stelle s aus der vorhergehenden Markierung $M(s)$ und dem Kantengewicht $W(s,t)$ der zur Transition führenden Kante?</td></tr>
<tr><td>(i) $M(s) + W(s,t)$</td><td>(ii) $M(s) - W(s,t)$</td><td>(iii) $M(s) \cdot W(s,t)$</td><td>(iv) $M(s) \div W(s,t)$</td></tr>
<tr><td colspan="4">c) Was ist keines der drei Teilprobleme des klassischen Erzeuger/Verbraucher-Problems? Dabei handelt es sich zum Beispiel um zwei Prozesse, die über eine gemeinsam genutzte Datenstruktur (gemeinsam genutzten Speicherbereich) Informationen austauschen.</td></tr>
<tr><td colspan="4">(i) Der gemeinsam genutzt Speicherbereich muss immer genau zu 60% gefüllt sein.
(ii) Der Verbraucher kann nur so lange Objekte aus der Datenstruktur entnehmen, bis diese leer ist.
(iii) Erzeuger und Verbraucher dürfen nicht gleichzeitig auf die gemeinsam genutzt Datenstruktur (den gemeinsam genutzten Speicherbereich) zugreifen.
(iv) Der Erzeuger kann nur so viele Objekte in die Datenstruktur einfügen, wie der Speicherbereich Kapazität bietet.</td></tr>
<tr><td colspan="4">d) Welcher Erreichbarkeitsgraph gehört zu folgendem Petrinetz?

</td></tr>
<tr><td>(i) </td><td>(ii) </td><td>(iii) </td><td>(iv) </td></tr>
<tr><td colspan="4">e) Welche Aussage bezüglich des Petrinetzes aus Aufgabenteil d) ist korrekt?</td></tr>
<tr><td>(i) Es enthält eine teilweise Verklemmung.</td><td>(ii) Es enthält eine echte Verklemmung.</td><td>(iii) Es kann niemals eine Transition schalten.</td><td>(iv) Es existieren Markierungen, bei denen mehrere Transitionen schalten können.</td></tr>
</table>