Technische Universität München
Fakultät für Informatik
Prof. Dr. M. Broy

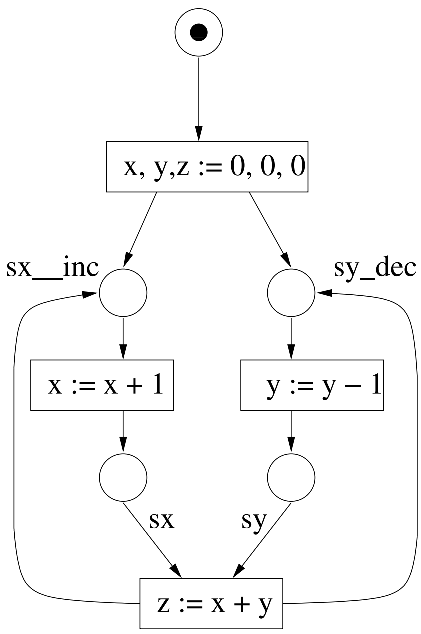
WS 2002/2003
Übungsblatt 6
29. November 2002

Übungen zur Vorlesung Einführung in die Informatik III

Aufgabe 20 Petri-Netz, Semaphore

Folgendes Programm und Boolesches Petri-Netz seien gegeben:

```
⌈var int x,y,z := 0,0,0 ;
 ⟦while true do x:=x+1 od
 ‖while true do y:=y-1 od
 ‖while true do z:=x+y od⟧⌋
```



Die einzelnen Anweisungen sind Transitionen des Petri-Netzes zugeordnet. Durch das Petri-Netz wird gewährleistet, dass

- z immer 0 bleibt,
- konfliktreiche Anweisungen sich gegenseitig ausschließen, und
- vor jeder Berechnung von z x inkrementiert und y dekrementiert wurden.

Ziel ist es, das Programm so zu ändern, dass es das gleiche Verhalten zeigt wie das Petri-Netz.

a) Betrachten Sie die Kanten bei den Stellen. In welcher Weise entsprechen sie P- und V- Operationen?

b) Fügen Sie Boolesche Semaphore im Programm ein, um das gleiche Verhalten wie im Petri-Netz zu erzielen.

Aufgabe 21 (H) Petri-Netz, Semaphore

Gegeben seien nachfolgend ein Programm und ein Boolesches Petri-Netz. Die Prozeduren u, uv, v und w verwenden keine gemeinsamen Variablen.

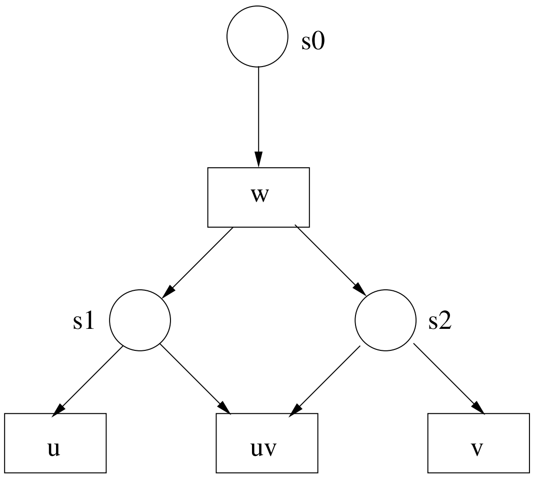
$$[\![\; w() \;||\; u() \;||\; uv() \;||\; v() \;]\!]$$

Ziel ist es, dieses Programm so zu ändern, dass es das gleiche Verhalten zeigt, wie das Petri-Netz.

a) Welche Kanten können nicht direkt in P- und V- Operationen umgesetzt werden? (Begründung)

b) Transformieren Sie das Petri-Netz so, dass die Kanten in P- und V-Operationen umgesetzt werden können. Dies kann durch Einführung einer zusätzlichen Stelle und einer Hilfstransition erreicht werden.

c) Fügen Sie dem transformierten Petri-Netz folgend im Programm Semaphore und Anweisungen ein, um das gleiche Verhalten wie in obigem Petri-Netz zu erreichen.



Aufgabe 22 Prozesssynchronisation

Gegeben sei der aus Aufgabe 13 bekannte Agent

$$t = (P||C)||_{\{w,r\}}B$$

mit

$$P = (\text{prod} :: p; w; \text{prod})$$
$$C = (\text{cons} :: r; c; \text{cons})$$
$$B = (\text{buff} :: w; r; \text{buff})$$

Die folgenden Programmstücke sind gleichwertige Programmstücke zu den Agenten P bzw. C:

```
var x; while true do produce(x); write(x) od
var y; while true do read(y); consume(y) od
```

Geben Sie ein zum Agenten t gleichwertiges Programm an, wobei Sie die Prozesse

(a) mit Hilfe des `await` Konstrukts

(b) mit Hilfe von Semaphoren

synchronisieren.

Aufgabe 23 (P) Parallele Matrixmultiplikation

Nutzen Sie das Java Thread-Konzept für die parallele Multiplikation von Matrizen.

(a) Auf der WWW-Seite der Übung zu „Einführung in die Informatik III“ finden Sie als Anhang zu Aufgabenblatt 6 eine Java-Implementierung zweidimensionaler Matrizen mit natürlich-zahligen Elementen. Sie können die Klasse `Matrix` zur Lösung dieser Aufgabe nutzen.

(b) *Wdh: Matrix-Multiplikation* $C_{m\times s} = A_{m\times n}, B_{n\times s}$; *Element* $c_{i,j} = \sum_{k=1}^{n} a_{i,k} * b_{k,j}$

Implementieren Sie eine Klasse `RowColumnMult` als Erweiterung der Klasse `Thread`, deren Instanzen genau ein Element der Ergebnismatrix berechnen.

Dem Konstruktor werden die Eingabematrizen A und B, die Ergebnismatrix C und die Zeile und Spalte des zu berechnenden Elements übergeben. In der `run` Methode wird das gewünschte Ergebniselement berechnet und in die Ergebnismatrix C geschrieben.

Fügen Sie der Klasse `RowColumnMult` die Methode `public void main(String args[])` hinzu, in dem Sie exemplarisch zwei Matrizen A und B erzeugen. Berechnen Sie alle Elemente der Ergebnismatrix C parallel, indem sie für jedes Ergebniselement einen Thread der Klasse `RowColumnMult` erzeugen und starten. Geben Sie das Multiplikations-Ergebnis auf dem Bildschirm aus.