Name: ____ Vorname: ____ Matr.–Nr.: ____

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

WS 2002/2003
4. April 2003

Klausur zu Einführung in die Informatik III

(Gruppe A)

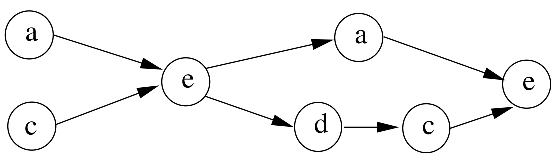
Hinweis: Das Aufgabenblatt muss abgegeben werden !

Aufgabe 1 Modellieren mit Hilfe von Petrinetzen (5 Punkte)

Modellieren Sie folgenden Teilprozess an einem (vereinfachten) Bankschalter mit Hilfe eines Petrinetzes: An einem Bankschalter warten in gebührendem Abstand N Kunden. Ist der Schalter frei, so tritt ein Kunde an den Schalter und der Bankangestellte bearbeitet das Anliegen des Kunden. Ist das Anliegen erledigt, so verläßt der Kunde die Bank und der nächste Kunde kann an den Schalter treten. Die Stellen und Transitionen sind hierbei aussagekräftig zu benennen!

Aufgabe 2 (2 + 2 + 3 = 7 Punkte)

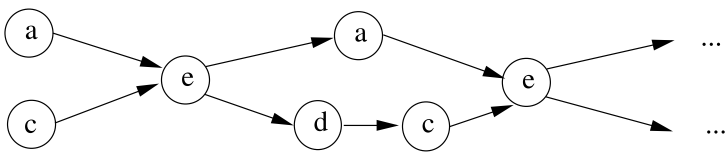
(a) Gegeben sei das folgende Aktionsdiagramm.



Geben Sie einen Agenten an, der genau dieses Aktionsdiagramm beschreibt.

(b) Geben Sie alle vollständigen Abläufe dieses Agenten an.

(c) Gegeben sei das folgende Aktionsdiagramm.



Geben Sie einen Agenten an, der genau dieses Aktionsdiagramm beschreibt.

Aufgabe 3 MI (10 Punkte)

Im Rahmen des Leser-Schreiber-Problems für mehrere lesende und höchstens einen schreibenden Prozess sei folgende Teilimplementierung gegeben:

```
var bool ws : = false;
var nat rs : = 0;

proc Writer =
 ⌈ var m x;
  while true do
    produce(x);
    await ¬ ws ∧ rs = 0 then ws := true endwait ;
    write(x);
    await true then ws := false endwait ;
  od ⌋
```

Geben Sie für die Prozedur `Writer` eine MI-Implentierung an; die Prozeduren `produce` und `write` können Sie dabei als gegeben annehmen.

Aufgabe 4 Bankier-Algorithmus (3 + 5 = 8 Punkte)

In einem Betriebsystem, das den *Bankier-Algorithmus* zur Vermeidung von Verklemmungen bei der Betriebsmittelvergabe einsetzt, existieren 3 Betriebsmittelklassen. Es seien 4 Prozesse im System, deren maximale Anforderungen durch die Matrix

$$Max = \begin{pmatrix} 3 & 2 & 5 \\ 2 & 3 & 1 \\ 5 & 9 & 2 \\ 1 & 8 & 9 \end{pmatrix}$$

beschrieben sind. Es seien jeweils 6, 10, 12 Instanzen der einzelnen Betriebsmittelklassen vorhanden.

(a) Kann das System in den Zustand gelangen, der durch die Matrix

$$Allocation = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 5 & 0 & 0 \\ 0 & 4 & 2 \end{pmatrix}$$

beschrieben ist?

(b) Das System befinde sich in dem Zustand, der durch die Matrix

$$Allocation = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \end{pmatrix}$$

beschrieben ist. Wie muss sich das Betriebssystem verhalten, wenn nun Prozess 3 eine Einheit der 3. Betriebsmittelklasse anfordert?

Aufgabe 5 **Parallele Suche durch Bisektion in Java** **(2+3+5 = 10 Punkte)**

In der Vorlesung wurde folgender Algorithmus diskutiert, der in dem Feld f parallel nach der Zahl i sucht. Nach der Terminierung enthält z das Resultat **true** oder **false**. z ist bei der Suche gemeinsame Variable aller parallelen Abläufe. Sobald ein Prozess die Zahl i findet und z mit **true** besetzt, beenden alle Prozesse die Suche.

```
proc suche = ([1:n] array nat f, nat lowbound, nat highbound, nat i, var bool z):
⌈   var nat middle := (lowbound+highbound)/2;
    if lowbound ≤ highbound
    then if f[middle] = i
         then await true then z := true endwait
         else var bool b;
              await true then b := z endwait;
              if ¬b then ⌈⌈ suche(f,lowbound,middle-1,i,z) ||
                            suche(f,middle+1,highbound,i,z) ⌋⌋ fi
         fi
    fi   ⌋
```

Aufruf: **var bool** z := **false**; suche(f,1,n,i,z)

Überführen Sie dieses Programm in ein äquivalentes, paralleles Java-Programm (ohne `main`) in folgenden Schritten:

(a) Implementieren Sie eine Klasse `GemBool` mit den Methoden `get` und `set` für die gemeinsamen Ergebnisvariable z.

Realisieren Sie die parallele Suche in der Klasse `PSuche` wie folgt:

(b) Geben Sie die Java Klassendefinition für `PSuche` inklusive Attribute und Konstruktor an.

(c) Implementieren Sie die Methode `run` von `PSuche`.