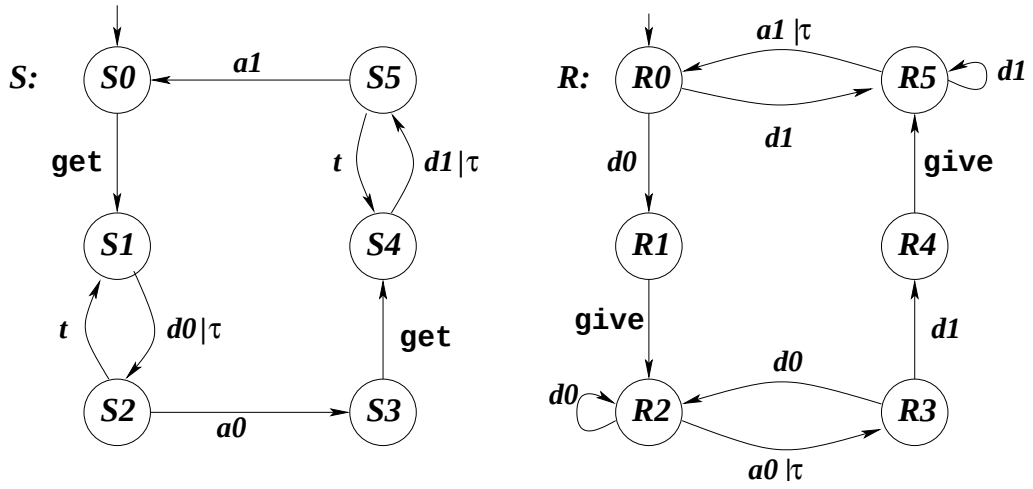
An Introduction to Specification and Verification

Exercise 5, Feb. 22th 2008

1. Design using Lotos AB-protocol $S|[d0, d1, a0, a1]|R$, when S and R are following labelled transition systems:



2. Make a global state graph of the AB-protocol using CADP. Does this protocol give the service we wanted?

3. In excercise 5.4 and 5.5 was ask to design in Lotos Hyman's mutual exclusion algorithm and to make the global state graph using CADP. Hide all the actions except the actions that model the critical section of the processes. Reduce the labelled transtion system using weak bisimulation (observational equivalence) and look at the result. Are the processes at the same time in the critical section?

4. Testing proberties can be done by specifying the proberty using Lotos-testprocess, which ends by action `testok`. This action tells that the test was successful. The test process is synchronized with the specification. In the synchronization list are all the actions of the test process except the action `testok`.

   If for example process S may execute the action sequence `ev1;ev2;ev3;exit` or the action sequence `ev1;ev2;ev4;exit` and we would like know if it possible that action `ev4` can follow action `ev2`, we can make following processes and expression for testing:

   process T[ev2, ev4,testok] : **exit** :=
   ev2;ev4;testok;**exit**
   endproc

process S[ev1,ev2,ev3,ev4] : **exit** :=
  ev1;ev2;ev3;**exit** [] ev1;ev2;ev4;**exit**
endproc

S[ev1,ev2,ev3,ev4] |[ev2, ev4]| T[ev2,ev4,testok]

Processes T and S are synchronized using actions `ev2` and `ev4`. After that we will examine all the traces of the system and if we find the action `testok` then the test is successfull.

How can you use the method to find out if processes are at the same time in the critical section? Try this method to Hyman's algorithm and to the Dekker's algorithm (pages 97-99, Lecture notes)

5. Let be $P \approx_{wbis} Q$. Is it true for all processes that R $R[> P \approx_{wbis} R[> Q$. (Because this is not true, it is enough to give an counterexample.)

6. (a) Process call is not exactly the same as the process body, where formal parameters are replaced by actual parameters used in call. Look at the following process and tell how you can demonstrate this.

```
process koe [a,b] :=
  apu[a,a]
 []
  apu[a,b]
where
  process apu[x,y] :=
   x; exit || y; exit
  endproc
endproc
```

   (b) Look at the following Full Lotos processes:

```
P[a,b,c] (n:Nat): noexit :=
 a!n; b?m:Nat !3; c?n:Nat; P[a,b,c](n)

Q[a,b,c]: noexit :=
 a?x:Nat; b!x !3; c!x+1; Q[a,b,c]
  []
 a!1; i; Q[a,b,c]
```

   Make a part of the global state graph of

```
P[a,b,c](0) || Q[a,b,c]
```

   so that you can see the overall structure.