

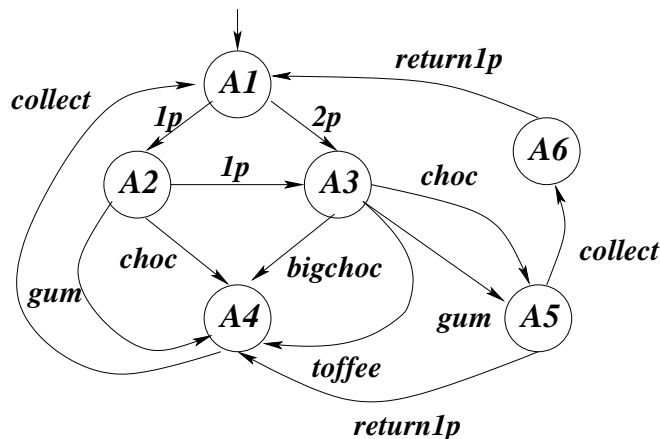
# An Introduction to Specification and Verification

## Exercise 5, Feb. 15th 2008

1. Draw the labelled transition systems that correspond following Lotos expressions. Use Lotos expressions as state names.

- (a) `c;d;stop`
- (b) `a;b;stop [] i;stop [] d;i;stop`
- (c) `hide b, c in a;b;stop [] i;c;stop [] d;i;stop`
- (d) `i;c; exit [] d;i;exit >> a;exit`
- (e) `hide a in (c;a;b;stop |[a,b]| b;c;stop [] a;stop)`
- (f) `(b;c;stop [> a;stop) ||| i;stop`

2. Design in Basic LOTOS a vending machine from the exercise two. The labelled transition system is below.



3. Design in Basic Lotos a variable having two different values: 1, and 2. Design in Basic Lotos a boolean variable having two different values: *false*, and *true*.
4. Design in Basic Lotos Hyman's mutual exclusion algorithm. Pseudocode is following:

```
boolean in1,in2;  
int k;
```

At the beginning  $in1 = in2 = false$ , and  $k = 1$ . The variables can be modelled as Lotos processes. Boolean variables can have two different values: *false*, and *true*. A variable  $k$  can have two different values: 1, and 2.

The code of the process P1:

```
while (true)
{
    /* the non-critical part of the program */
    in1 = true;
    while (k != 1)
    {
        while (in2) {skip};
        k = 1
    }
    /* the critical part of the program */
    in1 = false;
```

The command *skip* is not doing anything, it can be modelled with Lotos *i*-action.

The code of the process P2:

```
while (true)
{
    /* the non-critical part of the program */
    in2 = true;
    while (k != 2)
    {
        while (in1) {skip};
        k = 2
    }
    /* the critical part of the program */
    in2 = false;
```

Model the critical section of the process *P1* using two actions *in<sub>cs1</sub>* and *out<sub>cs1</sub>*. The first one (*in<sub>cs1</sub>*) is at the beginning of the critical section and the second (*out<sub>cs1</sub>*) before going out from the critical section. Use actions *in<sub>cs2</sub>* and *out<sub>cs2</sub>* for process *P2*.

Give a service description (using a labelled transition system) that describes a situation where *processes are never at the same time in the critical section*.

5. Make the global state graph of the assignment 4 using CAPD.

6. Design using Basic Lotos the adjacent faulty version of the sender and receiver of the FE-protocol. The missing of the messages are modelled using  $\tau$ -transitions. Write a LOTOS expression which describes the FE-system. Make a gobal state graph using CADP.

