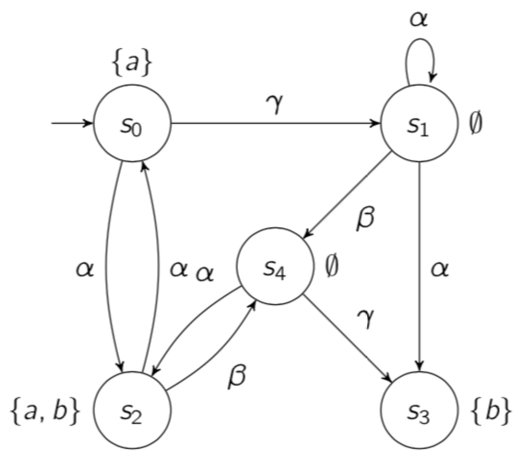
Verificação Formal de Software, 2019/20
FCUP

TP MC-1– Transition Systems and Communication between Processes

1. Consider the following transition system T_1:



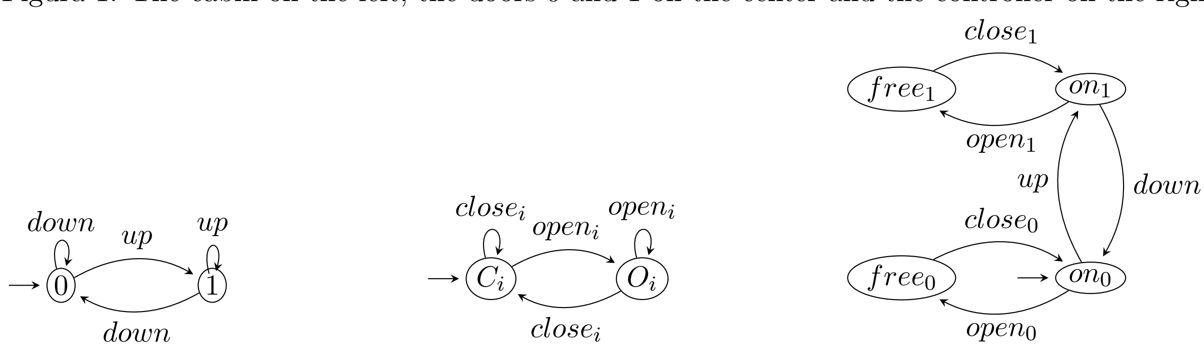
   (a) Formally define T_1.

   (b) Specify a finite and an infinite execution in T_1.

   (c) Compute $Pre(s_i)$ and $Post(s_i)$ for $i \in \{0, 1, 2, 3, 4\}$. Draw the first 4 levels of the computation tree starting at the initial state s_0 (ommitting the actions).

2. Consider an elevator in a two-story building and the following components: a cabin that goes up and down depending on the current floor; a door in each floor that can be open and closed and a controller that commands the doors and the cabin. The state $free_i$ of the controller corresponds to the situation that the cabin is in floor i with the door open; and the state on_i if the cabin is in floor i with the door closed. The transition systems of each components are described below.

Figura 1: The cabin on the left, the doors 0 and 1 on the center and the controller on the right.



   (a) Compute the synchronous message passing (handshaking)

$$cabin || door_0 || door_1 || controller$$

(b) Informally specify and verify the following properties

  a) The door on a given floor cannot open while the cabin is on a different floor

  b) The cabin cannot move while one of the doors is open

3. Determine the program graph and the transition system of each program fragment considering the locations and the conditions associated to each command.

   (a) Suppose that initially $x = 2$

   if $x > 2$ **then**
   $\quad x \leftarrow 0$
   $x \leftarrow x + 1$

   (b) Suppose that initially $x = 1$ and $y = 2$.

   while $x < 3$ **do**
   $\quad x \leftarrow x + 1$
   $\quad y \leftarrow y + x$

4. (Mutual Exclusion - Peterson Algorithm) Consider two processes with shared boolean variables b_1, b_2 and integer variable x which can take the values 1 or 2. Initially $b_1 = b_2 = False$. If both processes want to enter the critical section (waiting) the value of variable x decides which of the two processes may enter its critical section: if $x = i$ then P_i enters. The value of b_i is set if P_i wants to access the critical section.

   For process P_1 we have:

   while True **do**
   $\quad$ noncricital actions
   $\quad$ **atomic**($b_1 \leftarrow True; x \leftarrow 2$)
   $\quad$ **wait until** $x = 1 \vee b_2 = False$
   $\quad$ critical actions
   $\quad$ **atomic**($b_1 \leftarrow False$)

   and for P_2

   while True **do**
   $\quad$ noncricital actions
   $\quad$ **atomic**($b_2 \leftarrow True; x \leftarrow 1$)
   $\quad$ **wait until** $x = 2 \vee b_1 = False$
   $\quad$ critical actions
   $\quad$ **atomic**($b_2 \leftarrow False$)

   (a) Define the program graphs for two processes P_1 and P_2. Each graph PG_i has three locations n_i, w_i, c_i, corresponding to the colours green (noncritical) , blue (wait) and red (critical).

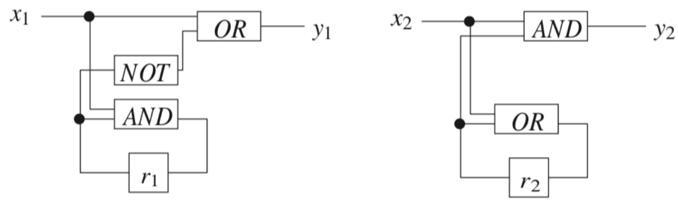
   (b) Compute $T(PG_1||PG_2)$. Note: only the reachable part and of b_i can be ommited. Initially $b_1 = b_2 = False$.

   (c) Check that mutual exclusion is ensured.

   (d) Check that every process that wants to access the critical section, enters it.

   (e) Implement in Promela and test with Spin.

5. (a) Let C_1 and C_2 be the following sequential circuits, each with one 1 bit register r_i, $i = 1, 2$.



(a) Suppose that initially that $r_1 = 0$ and $r_2 = 0$, determine the transition systems associated with C_1 e C_2. The values of the input bits are nondeterministically determined (i.e. each possible value must be considered).

(b) Determine the transition system of the synchronous product $C_1 \otimes C_2$. Recall that the synchronous product of two systems (omiting the actions) $T_i = (S_i, \longrightarrow_i, I_i, AP_i, L_i)$ for $i = 1, 2$ is $T_1 \otimes T_2 = (S_1 \times S_2, \longrightarrow, I_1 \times I_2, AP_1 \cup AP_2, L)$, with $L(\langle s_1, s_2 \rangle) = L(s_1) \cup L(s_2)$ and

$$\frac{s_1 \longrightarrow_1 s_1' \wedge s_2 \longrightarrow_2 s_2'}{\langle s_1, s_2 \rangle \longrightarrow \langle s_1', s_2' \rangle}$$