Homework 1

To hand in on September 30th at the beginning of the exercise session, or by email to `leroux@lsv.fr`.

Exercise 1 (Mutual Exclusion)**.**

1. The following program is a mutual exclusion protocol for two processes. There is a shared boolean variable s, initialized to 1, and two shared boolean variables y_i, i in $\{0, 1\}$, initialized to 0. Each process P_i can read the values of s, y_0, and y_1, but only write a new value in s and y_i. Here is the code of process P_i in C-like syntax:

   ```
   while (true)
     {
       /* 1: Noncritical section. */
       atomic { $y_i = 1$; $s = i$; };
       /* 2: Wait for turn. */
       wait until (($y_{1-i}$ == 0) || ($s$ != $i$));
       /* 3: Critical section. */
       $y_i = 0$;
     }
   ```

   Draw the transition system of each process, and construct their parallel composition. Label the states appropriately using the atomic propositions w_i and c_i, holding when process P_i is waiting or in the critical section, respectively.

2. Does the algorithm ensure *mutual exclusion*, i.e. that the two processes can never be simultaneously inside the critical section?

3. Does the algorithm ensure *starvation freedom*, i.e. that every waiting process will eventually access the critical section, provided that the other process does not stay forever inside the critical section?