RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHRSTUHL FÜR INFORMATIK 2
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/

Prof. Dr. Ir. J.-P. Katoen
F. Gretz & F. Sher

Introduction to Model Checking
Winter term 2011/2012

– Series 1 –

Hand in on October 26th before the exercise class.

Exercise 1 **(3 points)**

For this exercise we give the following definition:

Definition 1. Deterministic Transition System

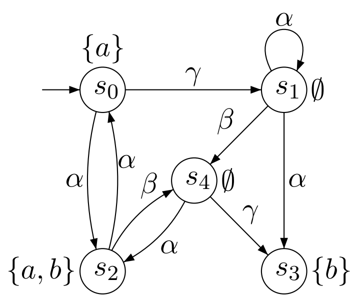
Let $\mathrm{T} = (S, Act, \rightarrow, I, AP, L)$ be a transition system.

a) T is called *action-deterministic* if $|\,I\,| \;\leq\; 1$ and $|\,Post(s, \alpha)\,| \;\leq\; 1$ for all states s and actions α.

b) T is called *AP-deterministic* if $|\,I\,| \;\leq\; 1$ and $|Post(s) \cap \{\, s' \in S \mid L(s') = A\,\}| \;\leq\; 1$ for all states s and $A \in 2^{AP}$.

■

Now let TS be the transition system depicted on the right.



a) Give the formal definition of TS.

b) Specify a finite and an infinite execution of TS.

c) Decide whether TS is an AP-deterministic or an action-deterministic transition system. Justify your answer!

Exercise 2 **(1 points)**

We are given three (primitive) processes P_1, P_2, and P_3 with shared integer variable x and local registers r_1, r_2 and r_3. The program of process P_i is as follows:

Algorithm 1 Process P_i

for $k_i = 1, \ldots, 10$ **do**
  LOAD($r_i \leftarrow x$);
  INC(r_i);
  STORE($r_i \rightarrow x$);
end for

That is, P_i executes ten times the assignment $x := x{+}1$. The assignment $x := x{+}1$ is realized using the three actions LOAD, INC and STORE. Consider now the parallel program:

Algorithm 2 Parallel program P

$x := 0;$
$P_1 \,||\, P_2 \,||\, P_3$

Question: Does P have an execution that halts with the terminal value $x = 2$?

Exercise 3 **(4 points)**

The following program is a mutual exclusion protocol for two processes due to Pnueli. There is a single shared variable s which is either 0 or 1, and initially 1. Besides, each process has a local Boolean variable y that initially equals 0. The program text for process P_i ($i = 0, 1$) is as follows:

l0: **loop forever do**
    begin
l1:   Noncritical section
l2: $(y_i, s) := (1, i)$;
l3:   **wait until** $((y_{1-i} = 0) \lor (s \neq i))$;
l4:   Critical section
l5: $y_i := 0$
    end.

Here, the statement $(y_i, s) := (1, i)$; is a *multiple assignment* in which variable $y_i := 1$ and $s := i$ is a single, atomic step.

Questions:

(a) Define the program graph of a process in Pnueli's algorithm.

(b) Determine the transition system for each process.

(c) Construct their parallel composition.

(d) Check whether the algorithm ensures mutual exclusion, i.e. both processes are never in their critical section at the same time.

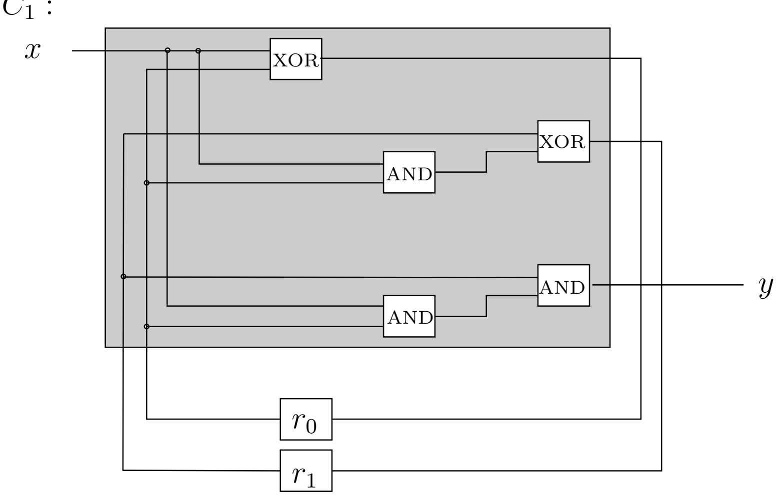
(e) Check whether the algorithm ensures starvation freedom, i.e. every time a process want to enter its critical section it can eventually do so.

The last two questions may be answered by inspecting the transition system.

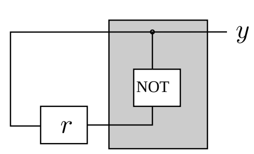
Exercise 4 **(2 points)**

The circuit C_1 describes the layout of a hardware adder that stores a 2-bit binary number represented by the registers r_0 and r_1. In each cycle, the value of x is added to the currently stored value; y is used as the carry bit:

C_1 :



C_2 :



a) Give the transition system representation TS_1 of the circuit C_1.

b) Let TS_2 be the transition system of the circuit C_2. Outline the transition system $TS_1 \otimes TS_2$.

Remark: The operator $\otimes$ denotes the synchronous product in which both systems always perform one step synchronously.