Prof. Dr. Ir. Dr. h. c. Joost-Pieter Katoen
Christian Hensel, Matthias Volk

Introduction to Model Checking (Summer Term 2018)

— Exercise Sheet 1 (due 30th April) —

General Remarks

- The exercises are to be solved in groups of *three* students. For sheet one, it is acceptable to form groups of two, but for the remaining sheets, we require you to form groups of three. You may use the L2P forum to search for group members.
- You may hand in your solutions for the exercises just before the exercise class starts at 12:15 or by dropping them into the "Introduction to Model Checking" box at our chair. Do *not* hand in your solutions via L2P or via e-mail.
- The solution for the first exercise sheet will be presented in the first exercise class on April 30th.
- Every sheet is worth 20 points. You need at least 40% of the exercise points to be admitted to the exam. If you gain at least 70% of the points *of the* ***marked**** *exercises* you get a 0.3 bonus on your grade for the exam. Note that this bonus cannot improve your grade if you failed the exam.

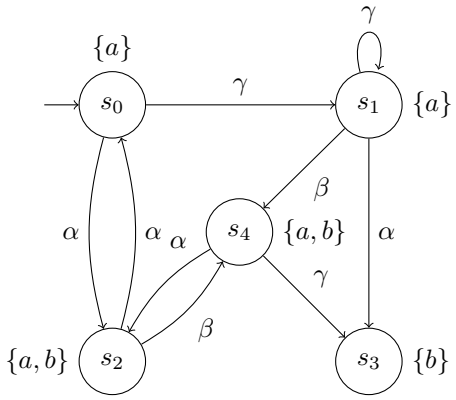
Exercise 1 (2 + 1 + 2 Points)

We call a transition system $\mathrm{TS} = (S, \mathrm{Act}, \to, S_0, \mathrm{AP}, L)$

- *action-deterministic* if $|\,S_0\,| \leq 1$ and $|\,\mathrm{Post}(s,\alpha)\,| \leq 1$ for all $s \in S$ and $\alpha \in \mathrm{Act}$, and
- *AP-deterministic* if $|\,S_0\,| \leq 1$ and $|\mathrm{Post}(s) \cap \{s' \in S \mid L(s') = A\}\,| \leq 1$ for all $s \in S$ and $A \in 2^{\mathrm{AP}}$,

where $\mathrm{Post}(s,\alpha) = \{s' \in S \mid \exists (s,\alpha,s') \in \to\}$ and $\mathrm{Post}(s) = \bigcup_{\alpha \in \mathrm{Act}} \mathrm{Post}(s,\alpha)$.

Let the transition system TS_1 be as follows.



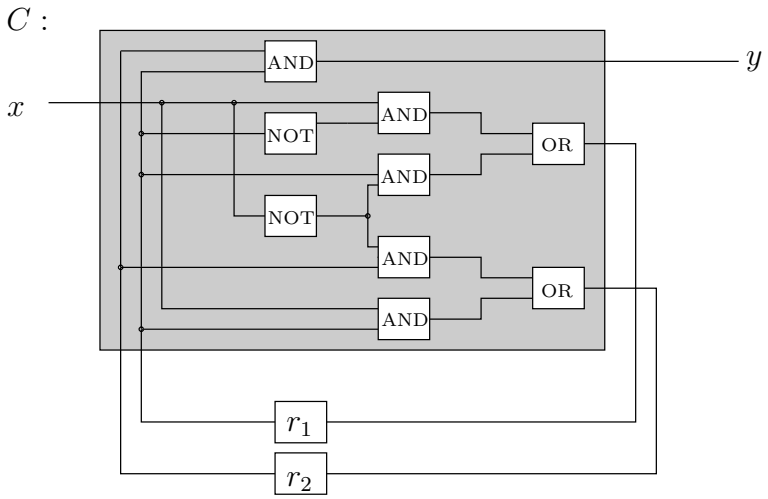
(a) Give the formal definition of TS_1.

(b) Specify a finite and an infinite execution of TS_1.

(c) Decide whether TS_1 is (i) *AP*-deterministic, and/or (ii) action-deterministic. Justify your answer.

Exercise 2 (1 + 1 + 3 + 1 Points)

Consider the following sequential hardware circuit C.



(a) Give the functions λ_y, δ_{r_1} and δ_{r_2} that govern the changes to the output and register bits depending on the input bits.

(b) Formally specify the state space of the transition system TS for the circuit C. Justify your answer.

(c) Draw the transition system TS for the circuit C using the set of labels $\text{AP} = \{x, r_1, r_2, y\}$ assuming that initially the values of the registers are 0. Make sure that ***all* formal components** of the transition system can be uniquely identified from your picture.

(d) Determine the set Reach(TS).

Exercise 3★ (2 + 5 + 1 + 1 Points)

Consider the following mutual exclusion algorithm for two processes P_0 and P_1.

The pseudo code of the algorithm for process P_i is given as

```
    int k := 0;
    b := [true, true];
ℓ0  while (true) do
      b[i] := false;
ℓ1    while (k != i) do
ℓ2      while (not b[1-i]) do
          k := i;
        end
      end
ℓ3    critical_section;
ℓ4    b[i] := true;
    end
```

where b is an array of two Boolean values which are initially true. k is a variable which is either 0 or 1, and initially 0.

(a) Give the program graph representation for a single process. Use the locations $\ell_0, \dots \ell_4$ corresponding to the indicated program fragments.

(b) Give the reachable part of the transition system of $P_0 ||| P_1$, i.e., $\mathrm{TS}(P_0 ||| P_1)$, over the state space $\langle \ell_i, \ell_j, k, b[0], b[1] \rangle$.
Hint: If you wish, you may treat the states $\langle \ell_i, \ell_j, 0, b[0], b[1] \rangle$ and $\langle \ell_j, \ell_i, 1, b[1], b[0] \rangle$ to be equivalent. In other words, whenever you encounter a transition to a state with $k = 1$, you may swap the locations of the processes P_0 and P_1, swap the values of $b[0]$ and $b[1]$, set k to 0, and make the transition target this "new" state instead. This operation considerably reduces the size of the resulting state space.

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

(d) Does the algorithm guarantee that every process eventually enters its critical section? Justify your answer.