Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann
Christian Schilling

Delivery: November 14th, 2016
16:15 via the post boxes
Discussion: November 16th, 2016

Tutorial for Cyber-Physical Systems - Discrete Models

Exercise Sheet 4

Exercise 1: Peterson's mutual exclusion algorithm — 4 Points

In the lecture we analyzed a version of Peterson's mutual exclusion algorithm in which each of the two processes did the assignments to b_i and x in an atomic step. In this exercise we consider two variants of Peterson's algorithm. VARIANT 1 is depicted below. We note that there is a fourth location which allows us to do the above mentioned assignments non-atomically. VARIANT 2 is obtained by swapping for each process the statements at locations *nc* and *req*.

```
        while true {
              ......
$nc$ :        $b_1$ := true;
$req$ :       $x := 2$;
$wt$ :        wait until($x = 1 \lor \neg b_2$);
$cs$ :        ... critical section ...
              $b_1$ := false;
              ......
        }
```

```
        while true {
              ......
$nc$ :        $b_2$ := true;
$req$ :       $x := 1$;
$wt$ :        wait until($x = 2 \lor \neg b_1$);
$cs$ :        ... critical section ...
              $b_2$ := false;
              ......
        }
```

We say that a variant of Peterson's mutual exclusion algorithm *satisfies the mutual exclusion property* if there is no execution such that both processes are in the critical section at the same time (i.e. the location (cs_1, cs_2) in the interleaving of the program graphs is unreachable).

Analyze for each variant formally if the mutual exclusion property is satisfied.

We propose the following approach.

- Construct the interleaving of the program graphs for both processes.
- Translate the interleaving into a transition system (reachable states are sufficient).
- Check if there is some reachable state in which both processes are in the critical section.

In order to save you some work we introduce the following convention *for this exercise.* You only have to draw the interleaving and the transition system if the variant satisfies the property. In case the property is violated it is also sufficient if you explain an execution that shows the violation.

Exercise 2: Weakest precondition 1+3 Points

Let $\mathcal{T} = (S, Act, \to, S_0, AP, L)$ be a transition system and let $C \subseteq S$ be a set of states. We define the operators *Pre* and *Post* as follows.[1]

$$\begin{aligned} Pre(C) &= \{s \in S \mid \exists \alpha \in Act.\ s \xrightarrow{\alpha} s' \text{ and } s' \in C\} \\ Post(C) &= \{s' \in S \mid \exists \alpha \in Act.\ s \xrightarrow{\alpha} s' \text{ and } s \in C\} \end{aligned}$$

A similar operator, $wp(C)$ (weakest precondition), is defined as follows:

$$wp(C) = \{s \in S \mid \forall s' \in S, \alpha \in Act.\ s \xrightarrow{\alpha} s' \text{ implies } s' \in C\}$$

(a) Give an example for $wp(C) \neq Pre(C)$.

(b*) How can $wp(C)$ be defined in terms of $Pre(\cdot), Post(\cdot)$ and standard set operations (union, intersection, complement)?

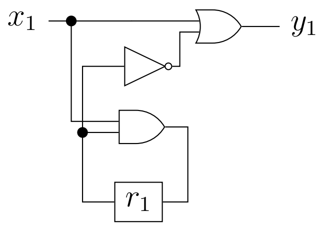
(c*) Prove or refute:

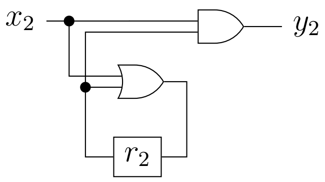
$$Post(C_1) \subseteq C_2 \iff C_1 \subseteq wp(C_2) \tag{1}$$

$$C_1 \subseteq Post(C_2) \iff wp(C_1) \subseteq C_2 \tag{2}$$

*Parts (b) and (c) are bonus exercises.

Exercise 3: Hardware circuits and synchronization operator 2 Points





Consider the sequentional hardware circuits above. The initial values of the registers are $r_1 = 0$ and $r_2 = 0$. In Exercise Sheet 1 you already constructed the transition system T_1 for the hardware circuit on the left.

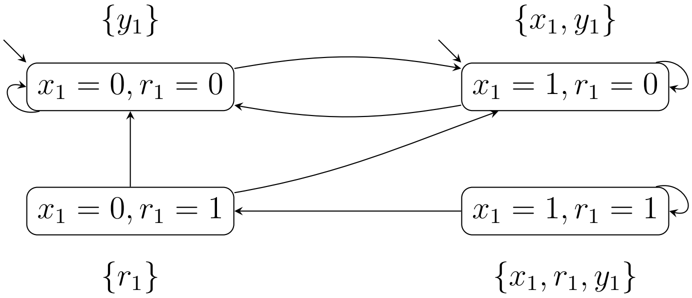
(a) Give the transition system T_2 for the hardware circuit on the right. As in Exercise Sheet 1 the states are the evaluations of the inputs and the registers. Furthermore, the atomic propositions are all input variables, registers and output variables whose value is one. In Exercise Sheet 1 you had the freedom to decide which values are changing in one step of the transition system. This time we fix the following convention.

In one step of the transition system, the values of the input variables may change nondeterministically and the new values of the output and all registers are computed based on the old values of the inputs and the old values of the registers.

Furthermore, all transitions are labelled with the action τ.

[1] We note that $s \xrightarrow{\alpha} s'$ is a shorthand denoting that the triple (s, α, s') is a transition of $\mathcal{T}$, i.e., $(s, \alpha, s') \in \to$.

(b) Apply the synchronization operator for transition systems "$\|$" and construct the reachable part of the transition system $T_1 \| T_2$. For transition system T_1 you may use our sample solution from Exercise Sheet 1 which is given below.



$\{y_1\}$
$\{x_1, y_1\}$
$x_1 = 0, r_1 = 0$
$x_1 = 1, r_1 = 0$
$x_1 = 0, r_1 = 1$
$x_1 = 1, r_1 = 1$
$\{r_1\}$
$\{x_1, r_1, y_1\}$