Prof. Dr. Andreas Podelski
Dominik Klumpp
Elisabeth Henkel

Hand in until November 16th, 2022
23:59 via ILIAS
Discussion: November 21st/22nd, 2022

Tutorial for Cyber-Physical Systems - Discrete Models

Exercise Sheet 4

Exercise 1: Railroad Crossing 6 Points

The goal of this exercise is to demonstrate how synchronous communication (handshaking) can be used to control the possible behaviours of a transition system.

In the lecture on 2$^{\text{nd}}$ November we discussed a model of a railroad crossing involving a train, a controller and a gate. We saw that the transition system *Train* $\|$ *Controller* $\|$ *Gate* for this model can reach a configuration where the train is on the crossing (state *in*) but the gate has not been lowered (state *up*). This is of course undesirable, the train should only be on the crossing when the gate is lowered.

(a) Identify the design flaw that causes this problem.

Hint: Consider for instance the case where the gate is stuck and cannot be lowered.

(b) How can this design flaw be fixed? Give the transition systems $Train'$, $Controller'$ and $Gate'$ for the repaired components.

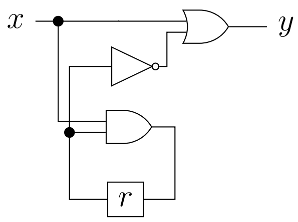
Hint: You can introduce additional states as well as new actions, which can be used to synchronize components and reduce nondeterminism. It is not necessary to change the gate.

(c) Draw the transition system for the new model, i.e., the parallel composition of the three components $Train' \parallel Controller' \parallel Gate'$.

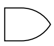
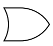
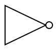
Exercise 2: Hardware Circuit and Transition System 4 Points

The goal of this exercise is to go from a pictorial representation of a hardware system to a formal model.

Consider the following sequential hardware circuit.



Draw the transition system of the hardware circuit. That is, the states are the valuations of the input x and the register r. The transitions represent the stepwise behavior where the value of the input bit x may or may not change in each step.
You may assume that initially the register r has the value `false`.

For your reference:  = AND gate,  = OR gate,  = NOT gate

Exercise 3: Parallelism - Interleaving 6 Points

The goal of this exercise is to construct the interleaving of two program graphs and the corresponding transition system.

Consider the following parallel system consisting of two processes. The programs for each process are given in a low-level programming language.

Algorithm 1:	**Algorithm 2:**
$r_1 := x + 1;$	$r_2 := 3 * x;$
$x := r_1;$	$x := r_2;$

(a) Draw the program graphs P_1 and P_2 for each process.

(b) Draw the interleaving of the program graphs $P_1 \parallel\!\mid P_2$.

(c) Draw the reachable part of the transition system $\mathcal{T}_{P_1 \parallel\!\mid P_2}$. We assume that the initial value of x is 1 and the initial values of r_1 and r_2 are both 0. Use it to determine which values can finally be stored in x.

Note: This program is an example, why one cannot assume that an update statement in a high-level language is an atomic action (with two processes consisting of just the statements $x := x + 1$ and $x := 3 * x$).