Prof. Dr. Andreas Podelski
Dominik Klumpp
Frank Schüssele

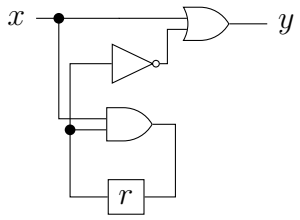
Hand in until November 24th, 2021
23:59 via ILIAS
Discussion: November 29th/30th, 2021

Tutorial for Cyber-Physical Systems - Discrete Models
Exercise Sheet 5

Exercise 1: 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.

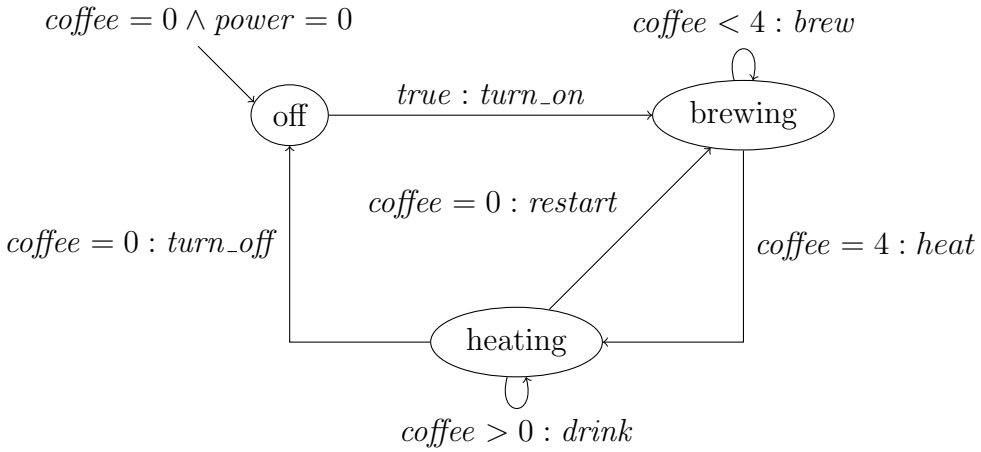


Draw the transition system of the hardware circuit. That is, the states are the valuations of the input x and the register r. The transitions represent the stepwise behavior where the value of the input bit x may or may not change in each step.
You may assume that initially the register r has the value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

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

Exercise 2: Coffee Machine and Transition System — 8 Points

The goal of this task is to provide some intuition on when the system described by a program graph satisfies given properties, by looking at the transition system.
The following program graph describes a simple coffee machine:



The effect of the operations is given by:

$$\begin{aligned}
\mathit{Effect}(\mathit{turn_on}, \eta) &= \eta[\mathit{power} := 1] \\
\mathit{Effect}(\mathit{turn_off}, \eta) &= \eta[\mathit{power} := 0] \\
\mathit{Effect}(\mathit{brew}, \eta) &= \eta[\mathit{coffee} := \mathit{coffee} + 1] \\
\mathit{Effect}(\mathit{drink}, \eta) &= \eta[\mathit{coffee} := \mathit{coffee} - 1] \\
\mathit{Effect}(\mathit{restart}, \eta) &= \eta \\
\mathit{Effect}(\mathit{heat}, \eta) &= \eta
\end{aligned}$$

(a) Draw the (reachable part of the) transition system corresponding to the program graph. Choose 3 transitions of the transition system, and justify their existence using the respective SOS-rule.

Use the SOS-rules to argue why the following transitions are *not* part of the transition system:

$$\langle \mathsf{off}, \{\mathit{coffee} \mapsto 0, \mathit{power} \mapsto 0\}\rangle \xrightarrow{\mathit{heat}} \langle \mathsf{heating}, \{\mathit{coffee} \mapsto 0, \mathit{power} \mapsto 0\}\rangle$$
$$\langle \mathsf{brewing}, \{\mathit{coffee} \mapsto 4, \mathit{power} \mapsto 1\}\rangle \xrightarrow{\mathit{brew}} \langle \mathsf{brewing}, \{\mathit{coffee} \mapsto 5, \mathit{power} \mapsto 1\}\rangle$$

(b) Use the transition system to explain which of the following properties are true for every execution of the coffee machine.

(i) If the machine is turned off ($\mathit{power} = 0$), it contains no coffee ($\mathit{coffee} = 0$).

(ii) If there are two cups of coffee ($\mathit{coffee} = 2$), there are either three or four cups of coffee in the next step ($\mathit{coffee} = 3$, $\mathit{coffee} = 4$).

(iii) There are always at most four cups of coffee ($\mathit{coffee} \leq 4$).

(iv) The coffee machine will be turned off (i.e., in location *off*) infinitely often.

(v) If there is no coffee ($\mathit{coffee} = 0$), there will be coffee after at most three steps.

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 $P_1 \mid\mid\mid P_2$.

(c) Draw the reachable part of the transition system $\mathcal{T}_{P_1 \| P_2}$. 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 $x := 3 * x$).