Prof. Dr. Andreas Podelski
Christian Schilling

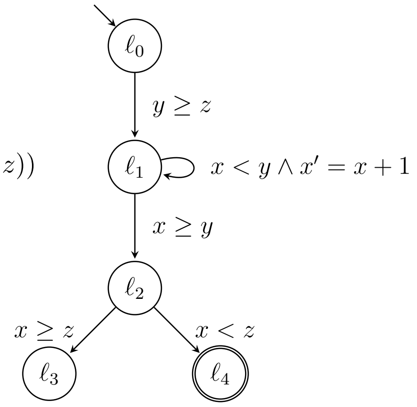
Last update: July 13, 2016
Delivery: July 19, 2016
Discussion: July 22, 2016

Tutorial for Cyber-Physical Systems - Hybrid Models
Exercise Sheet 11

Exercise 1: Counterexample-guided discovery of predicates

Consider the following program.

$$
\begin{aligned}
\varphi_{\text{init}} &\equiv pc = \ell_0 \\
\rho_1 &\equiv (move(\ell_0, \ell_1) \wedge y \geq z \wedge skip(x, y, z)) \\
\rho_2 &\equiv (move(\ell_1, \ell_1) \wedge x + 1 \leq y \wedge x' = x + 1 \wedge skip(y, z)) \\
\rho_3 &\equiv (move(\ell_1, \ell_2) \wedge x \geq y \wedge skip(x, y, z)) \\
\rho_4 &\equiv (move(\ell_2, \ell_3) \wedge x \geq z \wedge skip(x, y, z)) \\
\rho_5 &\equiv (move(\ell_2, \ell_4) \wedge x + 1 \leq z \wedge skip(x, y, z))
\end{aligned}
$$



Let Preds_{pc} be the set of all predicates on the program counter.

$$\mathsf{Preds}_{pc} = \{pc = \ell_1, pc = \ell_2, pc = \ell_3, pc = \ell_4, pc = \ell_5\}$$

(a) Given the path $\rho_1\rho_2\rho_3\rho_5$, provide a set of predicates Preds such that $\mathsf{Preds} \cup \mathsf{Preds}_{pc}$ is suffient to prove safety of the program, i.e., every abstract state returned by AbstReach($\mathsf{Preds} \cup \mathsf{Preds}_{pc}$) is disjoint from φ_{err} (the set of error states φ_{err} is $pc = \ell_4$).

Show that the predicates returned by your algorithm are sufficient to prove safety of the program by providing the abstract reachability graph.

(b) On page 31 of the handbook article you can find the (general) function RefinePath which is used in the function AbstRefineLoop and returns a set of predicates Preds given a path $\rho_1, \ldots, \rho_n$.

Let us implement RefinePath using the following idea:

> Let $\varphi_0 := \varphi_{\text{init}}$ and for the other predicates use the result from the application of *post* (e.g., $\varphi_1 := post(\varphi_0, \rho_1)$).

Observe that the predicates satisfy the required constraints of RefinePath.

Compute Preds for the above program and path $\rho_1\rho_2\rho_3\rho_5$ using this algorithm.

Are the predicates sufficient to prove safety?

(c) Imagine we had a smart implementation for RefinePath to find predicates sufficient to prove safety of any safe program. What are the implications? What can you conclude about the existence of such an algorithm?