Technische Universität München (I7)
J. Křetínský / M. Azeem / K. Grover

Summer Semester 2022
18.07.2022

Model Checking – Exercise sheet 12

Exercise 12.1

Consider the following program with a Boolean variable `x`. Initially, the value of `x` is `false`. The question mark stands for a nondeterministic value.

```
1 x = ?;
2 while (x)
3         x = ?;
4 while (true) {}
```

Let $AP = \{x\}$, where x is true only in states where the variable `x` is `true`.

(a) Construct a Kripke structure $\mathcal{K} = (S, \to, r, AP, \nu)$ for the above program.

(b) Let $\approx$ be an equivalence relation on S such that for all $s \approx t$ we have $\nu(s) = \nu(t)$. Construct from $\mathcal{K}$ the abstracted Kripke structure $\mathcal{K}'$ w.r.t. $\approx$.

(c) Model check the following formulas with $\mathcal{K}'$. Refine the abstraction if necessary.

(i) $\neg x \, \mathbf{W} \, x$

(ii) $\mathbf{G}(\neg x \to \mathbf{X} \neg x)$

(iii) $\mathbf{X}(\neg x \to \mathbf{G} \neg x)$

Exercise 12.2

We consider the following program, over the integer variables x and y:

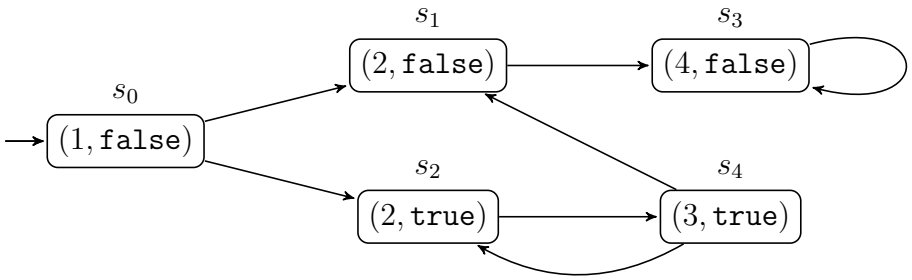
```
1 if (x >= 0) x = -x;
2 if (y >= 0) y = -y;
3 if (x + y > 0) error;
4 end
```

1. Give the set of configurations of the program (some may not be reachable).
2. Draw the abstract transition system with the predicates l_1, l_2, l_3, l_4 and "error".
3. Give a path ρ in the abstract transition system reaching a state where "error" holds.
4. What is the longest prefix (denoted ρ') of ρ that can be concretized?
5. Denote q the state in the abstract transition system reached by ρ'. Give a predicate that separates configurations reachable by ρ' from configurations that admit a successor.
6. Draw the abstract transition system with that additional predicate.

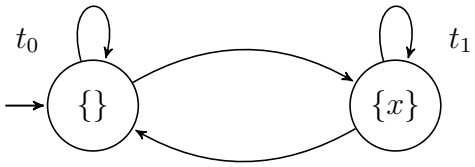
7. How many times do we have to repeat the abstraction refinement technique to exhibit an abstract transition system that does not reach the error state? Draw that transition system, how many predicates have we introduced?

Solution 12.1

(a) Each state of the following Kripke structure $\mathcal{K}$ is a pair of a program location and a valuation of `x`.



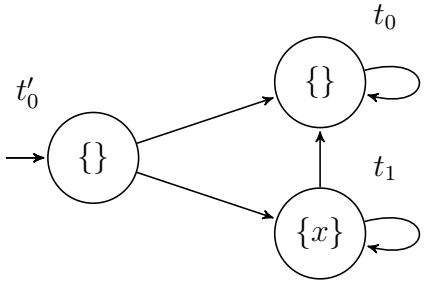
(b) Let $t_0 = [s0] = \{s_0, s_1, s_3\}$ and $t_1 = [s1] = \{s_2, s_4\}$. The abstraction $\mathcal{K}'$ is as follows:



(c) (i) $\mathcal{K}' \models \neg x \, \mathbf{W} \, x$

(ii) $\mathcal{K}' \not\models \mathbf{G}(\neg x \to \mathbf{X}\neg x)$. A counterexample in $\mathcal{K}'$ is $t_0 t_1 t_1^\omega$, which corresponds to the run $s_0 s_2 (s_4 s_2)^\omega$ in $\mathcal{K}$. So, $\mathcal{K} \not\models \mathbf{G}(\neg x \to \mathbf{X}\neg x)$.

(iii) $\mathcal{K}' \not\models \mathbf{X}(\neg x \to \mathbf{G}\neg x)$. A counterexample in $\mathcal{K}'$ is $t_0 t_0 t_1^\omega$. However, there are no corresponding runs in $\mathcal{K}$ because such paths must start with $s_0 s_1$, but no successors of s_1 are in t_1. Since $s_0 \in t_0$ and s_0 has a successor in t_1, we can refine the abstraction to distinguish s_0 from s_1. $t_0' = \{s_0\}$ and $t_0 = \{s_1, s_3\}$, and construct a new Kripke structure $\mathcal{K}''$ as follows.



We have $\mathcal{K}'' \models \mathbf{X}(\neg x \to \mathbf{G}\neg x)$.

Solution 12.2

1. $\{l_1, l_2, l_3, l_4, error\} \times \mathbb{Z} \times \mathbb{Z}$

2. The states of the abstract TS are $\{l_1, l_2, l_3, l_4, error\}$ and the transition relation is $\{(l_1, l_2), (l_2, l_3), (l_3, error), (l_3, l_4)\}$.

3. $\rho = l_1\ l_2\ l_3\ error$

4. $\rho' = l_1\ l_2\ l_3$

5. $q = l_3$. Configurations reachable by ρ' will satisfy the predicate $(x \leq 0) \wedge (y \leq 0)$ because if either of x or y are positive, l_1 and l_2 will make them non-positive. However, all configurations in l_3 admit a successor. (Computing S_0, S_1, S_2, and S_3 as per the lecture, we get $S_3 = \{\}$)

6. To avoid blow up in the number of states, we assume that if a state is not labeled by a propostion, we don't care (i.e. the proposition can either be **true** or **false**). If a proposition p is **false** in any of the state, the state has to be labeled by $\neg p$.

   New states will be $\{l_1, l_2, (l_3, p_1), (l_3, \neg p_1), l_4, error\}$ where p_1 is the predicate $(x \leq 0) \wedge (y \leq 0)$.

   To add an incoming transition, we look at feasibility of the concrete statement corresponding to the predecessor state and the predicates of the current state. Similarly, to add an outgoing transition, we look at if the concrete statement of the current state and the predicate of the successor are feasible. The transition relation would be $\{(l_1, l_2), (l_2, (l_3, p_1)), (l_2, (l_3, \neg p_1)), ((l_3, p_1), l_4), ((l_3, \neg p_1), error)\}$.

7. From the above refinement, we realize that there is a state $(l_3, \neg p_1)$, which admits a successor to an *error* state; however there is no concrete path which leads to $(l_3, \neg p_1)$. (Computing S_0, S_1 and S_2 as per lecture, we get $S_2 = \{\}$).

   We find that the predicate $(x \leq 0)$, say p_2, separates the configuration of l_2 such that $(l_2, \neg p_2)$ admits a successor to $(l_3, \neg p_1)$ but (l_2, p_2) does not admit a successor to $(l_3, \neg p_1)$.

   The updated set of states are $\{l_1, (l_2, p_2), (l_2, \neg p_2), (l_3, p_1), (l_3, \neg p_1), l_4, error\}$

   The new transition relation would be
   $\{(l_1, (l_2, p_2)), ((l_2, p_2), (l_3, p_1)), ((l_2, \neg p_2), (l_3, \neg p_1)), ((l_3, p_1), l_4), ((l_3, \neg p_1), error)\}$

   In this updated transition system, we do not find a path that leads to error.

Note: Abstraction refinement in such cases needs Hoare-like reasoning. For more details see: https://resources.mpi-inf.mpg.de/departments/rg1/conferences/vtsa10/slides/esparza-3.pdf