Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann

Preference deadline: July 12, 2018
Discussion: July 16, 2018

Tutorial for Program Verification
Exercise Sheet 11

Exercise 1: Least fixed point of $post^{\#}$ 2 Points

Let S be a set of states. Let the concrete domain D be the powerset of S, i.e., $D := \mathcal{P}(S)$. Let $D^{\#} \subseteq D$ be the abstract domain. Let $\alpha : D \to D^{\#}$ be defined as follows.

$$\alpha(x) := \bigcap \{y \in D^{\#} \mid x \subseteq y\}$$

For transition relation ρ and $\phi_{\text{init}} \in D$ define $post^{\#}(s, \rho) := \alpha(\phi_{\text{init}}) \cup \alpha(post(s, \rho))$.

In the lecture (slides 23-26) you have seen a proof by induction that the least fixed point[1] of $post^{\#}$ is the smallest (i.e., most precise) element of the abstract domain that is inductive under $post$ w.r.t. ϕ_{init}.

(a) Give a more elegant proof that does not use induction.

(b) In the lecture we have seen several properties of α. Which ones did you need in the proof?

Hint: It suffices to show that $\text{lfp}(post^{\#}) \subseteq \phi$ holds for any ϕ with the following properties:

(1) ϕ is an element of the abstract domain, i.e., $\phi \in D^{\#}$.

(2) ϕ is inductive under $post$ w.r.t. ϕ_{init}, i.e., $\phi_{\text{init}} \subseteq \phi$ and $post(\phi, \rho) \subseteq \phi$.

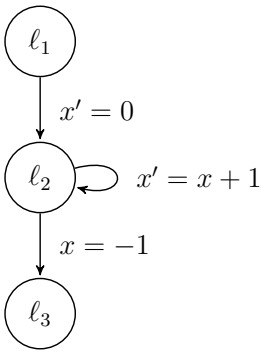
Exercise 2: Apply AbstRefineLoop 2 Points

Consider the following program

$$P = (V, pc, \varphi_{init}, \mathcal{R}, \varphi_{err})$$

where the tuple of program variables V is (pc, x), the inital condition φ_{init} is $pc = \ell_1$, the error condition φ_{err} is $pc = \ell_3$, and the set of transition relations $\mathcal{R}$ contains the following transitions.

$$\rho_1 = (move(\ell_1, \ell_2) \wedge x' = 0)$$
$$\rho_2 = (move(\ell_2, \ell_2) \wedge x' = x + 1)$$
$$\rho_3 = (move(\ell_2, \ell_3) \wedge x = -1)$$



[1] Let $f : L \to L$ be a function over some domain L. The least fixed point of f, written $\text{lfp}(f)$, is a smallest set X such that $f(X) = X$. In this exercise the least fixed point is unique.

As usually, we presume that the domain of the variable x is the set of all integers $\mathbb{Z}$.

(a) Is the program P correct?

(b) Apply the algorithm AbstRefineLoop for three iterations. Write down the set of predicates in each iteration. Write down the result of AbstReach for some iteration of your choice.

(c) Will the algorithm AbstRefineLoop terminate? Why?

(d) Propose an optimization for RefinePath such that AbstRefineLoop terminates.

(e) Consider the program P' that is very similar to P but where

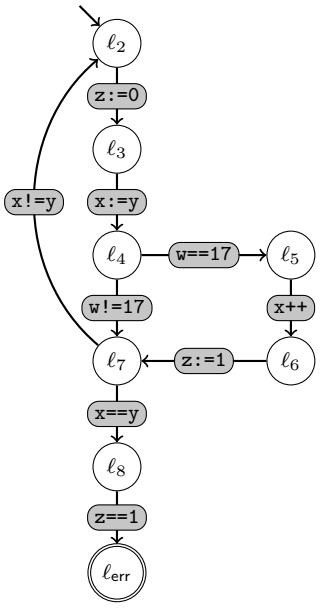
- the domain of the variable x is the set of all integers that are greater than or equal to -2^{31} and smaller than or equal to $2^{31} - 1$ and where
- the plus operator has the same semantics as in Java (e.g., if you add one to the largest number in the domain you get the smallest number of the domain).

Is the program P' correct? Will the algorithm AbstRefineLoop terminate on P'? Why? How many iteration are needed?

Exercise 3: Apply trace abstraction 2 Points

Consider the following program and the corresponding control automaton $\mathcal{A}_\mathcal{P}$.

```
   int x, y, z, w;
   void foo() {
1:     do {
2:         z := 0;
3:         x := y;
4:         if (w == 17){
5:             x++;
6:             z := 1;
           }
7:     } while(x != y)
8:     assert (z != 1);
   }
```



Give two error traces π_1, π_2 and construct corresponding interpolant automata $\mathcal{A}_1, \mathcal{A}_2$ such that the inclusion $\mathcal{L}(\mathcal{A}_\mathcal{P}) \subseteq \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$ holds.

Remark: We call a trace π infeasible if $post(\mathsf{true}, \pi) = \mathsf{false}$ holds.

Exercise 4: Regular traces 1 Point

Consider the program whose set of control flow traces is given by the following regular expression.

`assume(x is prime))` (`x--`)* `assume(x = 0)`

(a) Consider the pre-/postcondition pair $(true, true)$.

(i) Is the set of correct control flow traces a regular language?

(ii) Is the set of feasible correct control flow traces a regular language?

(iii) Is the set of infeasible correct control flow traces a regular language?

(b) Consider the pre-/postcondition pair $(true, false)$. Answer the same questions as above.