Prof. Dr. Andreas Podelski
Matthias Heizmann

18.1.2012
Submission: 24.1.2012
at the beginning of the lecture

Tutorials for Program Verification
Exercise sheet 11

Exercise 1: Transition Predicate Abstraction 3 point + 1 bonus points

Consider the following modified version of the TPA algorithm. (Modification: T composed with the *abstraction* of ρ_τ)

> ***Algorithm (TPA^{cl})***
>
> **Input:** program $P = (\Sigma, \mathcal{T}, \rho)$
> set of transition predicates $\mathcal{P}$
> abstraction α defined by $\mathcal{P}$
>
> **Output:** set of abstract transitions $P^\# = \{T_1, \ldots, T_n\}$
> such that $T_1 \cup \cdots \cup T_n$ is a transition invariant
>
> $P^\# := \{\alpha(\rho_\tau) \mid \tau \in \mathcal{T}\}$
> **repeat**
> $P^\# := P^\# \cup \{\alpha(T \circ \alpha(\rho_\tau)) \mid T \in P^\#,\ \tau \in \mathcal{T},\ \alpha(T \circ \alpha(\rho_\tau)) \neq \emptyset\}$
> **until** no change

(a) Prove or refute the following claim:

The set of abstract transitions computed by TPA^{cl} is a disjuctively well-founded transition invariant iff the set of abstract transitions computed by TPA is a disjuctively well-founded transition invariant.

(b) Think about a setting where we reapply the algorithm multiple times for the same set of transition predicates and the same set of transitions $\mathcal{T}$. What can be a possible advantage of TPA^{cl} over TPA?

Exercise 2: TPA with Initial States 2 point + 1 bonus points

In the lecuture we considered only programs $P = (\Sigma, \mathcal{T}, \rho)$ where every state is an initial state. Let us now consider programs $P = (\Sigma, \Sigma_{\mathsf{init}}, \mathcal{T}, \rho)$ where only the states in $\Sigma_{\mathsf{init}} \subseteq \Sigma$ are initial states.

(a) Give a program $P = (\Sigma, \Sigma_{\mathsf{init}}, \mathcal{T}, \rho)$ whose transition relation R_P is not well-founded, but R_P restricted to the reachable states of P is well-founded. Given an informal explanation why for each set of transition predicate $\mathcal{P}$ the set of abstract transitions $P^\#$ is not disjuctively well-founded.

(b) Assume you have a tool that does an reachability analysis and you have a tool that computes the TPA algorithm. Describe a termination analysis that uses both tools and can be used to show termination of your program stated in (a).

Exercise 3: Synthesis of Ranking Functions 4 points

Give a ranking function (i.e. a function whose value is decreased after each iteration of the loop) for the following while loop.

```
while (x>=0 && y>=0){
     tmp := x;
     x := y;
     y := tmp-1;
}
```

Apply the procedure given in the lecture to compute a ranking function that shows that the following transition[1] relation is well-founded.

$$x \geq 0 \wedge y \geq 0 \wedge x' \leq y \wedge y' \leq x - 1$$

Write down all your steps.

- Give a formula that is satisfiable iff the program has a linear ranking function.
- Apply Farkas' lemma to obtain an equivalent formula that has neither universal quantifiers nor non-liner terms.
- Give a satisfying assignment for the formula obtained after applying Farkas' lemma. State also satisfying assignments for the existentially quantified variables λ and μ.

[1]We obtained this relation from the programs transition relation, but we droped some conjucts to make your life easier.