Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann
Christian Schilling

Tutorial for Program Verification
Bonus Sheet 13 (not discussed in the tutorial)

Exercise 1: Transition predicate abstraction
Consider the following modified version of the TPA algorithm. (Modification underlined: 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**
> $P^\# := P^\# \cup \{\alpha(T \circ \underline{\alpha(\rho_\tau)}) \mid T \in P^\#,\ \tau \in \mathcal{T},\ \underline{\alpha(T \circ \alpha(\rho_\tau))} \neq \emptyset\}$
> **until** no change

(a) Prove or refute the following claim:

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

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

Exercise 2: TPA with initial states
So far 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. Give an informal explanation why for each set of transition predicates $\mathcal{P}$ the set of abstract transitions $P^\#$ is not disjunctively well-founded.

(b) Assume you have a tool that does a 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 part (a).