Prof. Dr. Andreas Podelski
Matthias Heizmann

8.2.2012
Submission: 14.2.2012
at the beginning of the lecture

Tutorials for Program Verification
Exercise sheet 14

Exercise 1: Hoare Logic 3 points

Consider the following program which computes the factorial of natural numbers.

```
procedure Fact(n) returns res
    {true}
    if(n ≤ 1){
        res := 1;
    } else{
        n_Fact := n − 1;
        res_Fact := call Fact(n_Fact);
        res := n · res_Fact;
    }
    {n ≥ 0 → res = n!}
    return res
```

Construct a Hoare logic derivation that proves that the program Fact fulfills the correctness specification.

Exercise 2: Trace Abstraction For Recursive Programs 3+3 points

Consider the following program which computes the factorial of natural numbers.

```
procedure Fact(n) returns res
    if(n ≤ 1){
        res := 1;
    } else{
        n_Fact := n − 1;
        res_Fact := call Fact(n_Fact);
        res := n · res_Fact;
    }
    assert (n ≥ 0 → res = n!);
    return res
```

(a) Construct the program automaton $\mathcal{A}_{\mathcal{P}}$ (i.e. a nested word automaton that accepts the error traces of the program.

(b) State a set of predicates $\mathsf{Pred} = \{p_1, \ldots, p_m\}$ and construct the predicate automaton $\mathcal{A}_{\mathsf{Pred}}$ such that the intersection $\mathcal{L}(\mathcal{A}_{\mathsf{Pred}}) \cap \mathcal{L}(\mathcal{A}_{\mathcal{P}})$ is empty.

Exercise 3: Pushdown systems 4 bonus points

We take a finite automaton defined as usual and give an equivalent characterization of acceptance, namely in terms of configurations (q, w) of the automaton.
Intuitively, (q, w) means that the automation is in state q and the word w is still to be read (from left to right). Thus, we have the transition $(q, w) \to (q', w')$ whenever $(q, a, q') \in \delta$ and $w = a.u$ and $w' = u$ for some word $u \in \Sigma^*$.
The word w is accepted if there exists a sequence of transitions that leads the configuration (q_0, w) to a configuration (q, v) where q is a final state and and v is empty.
We can view each transition as a pop transition if we view the word w as the contents of a stack.
It is obvious that the new notion of acceptance is equivalent to the old one; i.e., the set of accepted words is a regular language.
This still holds true if we add epsilon transitions. I.e., we add a relation $\delta' \subseteq Q \times Q$ and we have a transition $(q, w) \to (q', w)$ whenever $(q, q') \in \delta'$ and w in Σ^*.

Now, what if add push transitions? I.e., we add a relation $\delta'' \subseteq Q \times \Sigma \times Q$ and we have a transition $(q, w) \to (q', w')$ whenever $(q, a, q') \in \delta''$ (where $w' = a.w$ for any word w in Σ^*).
Show that we still obtain the same notion of acceptance; i.e., the set of accepted words is a regular language.