Prof. Dr. Andreas Podelski
Matthias Heizmann

16.11.2011
Submission: 22.11.2011
at the beginning of the lecture

Tutorials for Program Verification
Exercise sheet 4

Exercise 1: Hoare logic derivation and weakest precondition 1+1 points

Consider the program Fact that was presented in the lecture.

$$
\begin{array}{l}
\{n \geq 0\} \\
f := 1; \\
i := 1; \\
\textbf{while } i \leq n \textbf{ do } \{f = \mathit{fact}(i-1) \land i \leq n+1\} \ \{ \\
\quad f := f \times i \\
\quad i := i + 1 \\
\} \\
\{f = \mathit{fact}(n)\}
\end{array}
$$

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

(b) Compute the weakest precondition $wp(\mathsf{Fact}, f = \mathit{fact}(n))$.

Exercise 2: Hoare logic derivation 1+1+3 points

(a) Write down a partial correctness specification (i.e. precondition and postcondition for a program C that computes the integer divison of integers x and y, and stores the quotient in z and the remainder in w.

(b) Write down the program C. Use the pseudo code language introduced in the lecture.

(c) Annotate the while loop of your program with a suitable loop invariant and construct a Hoare logic derivation that proves that your program C fulfills your correctness specification.

In the last exercise it was probably not easy to guess a suitable inductive invariant for the while loop. In the next exercise we derive a recursive equation for the loop invariant of a while loop. This equation might be useful to guess inductive loop invariants.

Exercise 3: Recursive equation for loop invariants 2 + 1 points

Consider the following equivalence of commands.

$$\textbf{while } b \textbf{ do } C \qquad \equiv \qquad \textbf{if } b \textbf{ then } C \,;\, \textbf{while } b \textbf{ do } C \textbf{ else skip}$$

(a) Use the operational semantics of commands to show that the preceeding equivalence holds. I.e., show that the following equation is valid.

$$[\![\textbf{while } b \textbf{ do } C]\!] \quad = \quad [\![\textbf{if } b \textbf{ then } C \,;\, \textbf{while } b \textbf{ do } C \textbf{ else skip}]\!]$$

(b) Use the weakest precondition $wp(\cdot,\cdot)$ to state a recursive equation for the loop invariant θ of a while loop **while** b **do** C. The right hand side of the equation should be a first order logic formula that contains b, θ, and $wp(C, \phi)$ for some first order logic formula ϕ.