Prof. Dr. Andreas Podelski
Dr. Matthias Heizmann

Preference deadline: May 17, 2018
Discussion: May 28, 2018

Tutorial for Program Verification
Exercise Sheet 5

Exercise 1: Weakest precondition for sequential composition 2 Points

The weakest precondition of the sequential composition is independent of the way we add parentheses, i.e.,

$$\mathbf{wp}((\mathtt{C}_1 \,;\, \mathtt{C}_2) \,;\, \mathtt{C}_3, \phi) \;\equiv\; \mathbf{wp}(\mathtt{C}_1 \,;\, (\mathtt{C}_2 \,;\, \mathtt{C}_3), \phi)$$

Use the following program and postcondition to exemplarily show this fact, i.e., compute **wp** for both interpretations step by step and compare the results.

$$\begin{aligned} &\mathtt{C}_1 : \mathbf{if}\ x > 0\ \mathbf{then}\ x := 1\ \mathbf{else}\ x := 2 \\ &\mathtt{C}_2 : y := 1 \qquad\qquad\qquad\qquad \phi : x = 3 \\ &\mathtt{C}_3 : x := x + y \end{aligned}$$

Exercise 2: Recursive equation for loop invariants 2 Points

In this exercise we derive a recursive equation for the loop invariant of a while loop. This equation might be useful to guess inductive loop invariants.

Consider the following equivalence of commands.

$$\mathbf{while}\ \mathtt{b}\ \mathbf{do}\ \mathtt{C}_0 \qquad \equiv \qquad \mathbf{if}\ \mathtt{b}\ \mathbf{then}\ \mathtt{C}_0 \,;\, \mathbf{while}\ \mathtt{b}\ \mathbf{do}\ \mathtt{C}_0\ \mathbf{else\ skip}$$

(a) Use the operational semantics of commands ("$\rightsquigarrow$") to show that the preceding equivalence holds, i.e., show that the following equation is valid.

$$[\![\mathbf{while}\ \mathtt{b}\ \mathbf{do}\ \mathtt{C}_0]\!] \quad = \quad [\![\mathbf{if}\ \mathtt{b}\ \mathbf{then}\ \mathtt{C}_0 \,;\, \mathbf{while}\ \mathtt{b}\ \mathbf{do}\ \mathtt{C}_0\ \mathbf{else\ skip}]\!]$$

(b) Use the weakest precondition $\mathbf{wp}(\cdot,\cdot)$ to state a recursive equation for a loop invariant θ of a while loop **while** $\mathtt{b}$ **do** $\mathtt{C}_0$.

Hint: Start computing **wp** for both sides. Finally, the right-hand side of the equation should be a first-order logic formula that contains $\mathtt{b}$, θ, and $\mathbf{wp}(\mathtt{C}_0, \phi)$ for some suitable first-order logic formula ϕ.

Exercise 3: Hoare logic derivation – Multiplication 2 Points

(a) Write down a partial correctness specification (i.e., precondition and postcondition) for a program `C` that multiplies two integers m and n, where m is nonnegative, and stores the result in r.

(b) Write down a program `C` as specified above that only uses addition (but not multiplication). Use the command language introduced in the lecture.

Hint: Using an auxiliary variable may be helpful for the next part of the exercise.

(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.

Exercise 4: Loop invariants 1 Point

Consider the following program P.

$$
\begin{array}{l}
\{\mathit{true}\} \\
x := i; \\
y := j; \\
\textbf{while } x \neq 0 \textbf{ do } \{\theta\} \ \{ \\
\quad x := x - 1 \\
\quad y := y - 1 \\
\} \\
\{i = j \rightarrow y = 0\}
\end{array}
$$

(a) Find a suitable loop invariant θ such that $\mathit{true} \models \mathbf{wp}(P, i = j \rightarrow y = 0)$ holds.

(b) Give two examples for a loop invariant θ such that $\mathit{true} \models \mathbf{wp}(P, i = j \rightarrow y = 0)$ does not hold.