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.,
\[ \text{wp}((C_1 ; C_2) ; C_3, \phi) \equiv \text{wp}(C_1 ; (C_2 ; C_3), \phi) \]
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{align*}
C_1 : & \text{if } x > 0 \text{ then } x := 1 \text{ else } x := 2 \\
C_2 : & y := 1 \\
C_3 : & x := x + y
\end{align*}
\( \phi : x = 3 \)

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.
\[ \text{while } b \text{ do } C_0 \equiv \text{if } b \text{ then } C_0 ; \text{while } b \text{ do } C_0 \text{ else skip} \]
(a) Use the operational semantics of commands (\( \Rightarrow \)) to show that the preceding equivalence holds, i.e., show that the following equation is valid.
\[ [\text{while } b \text{ do } C_0] = [\text{if } b \text{ then } C_0 ; \text{while } b \text{ do } C_0 \text{ else skip}] \]
(b) Use the weakest precondition \( wp(\cdot, \cdot) \) to state a recursive equation for a loop invariant \( \theta \) of a while loop \( \text{while } b \text{ do } C_0 \).
Hint: Start computing \( wp \) for both sides. Finally, the right-hand side of the equation should be a first-order logic formula that contains \( b, \theta \), and \( \text{wp}(C_0, \phi) \) for some suitable first-order logic formula \( \phi \).
Exercise 3: Hoare logic derivation – Multiplication

(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

Consider the following program $P$.

\[
\begin{array}{l}
\{true\} \\
x := i; \\
y := j; \\
while x \neq 0 do \{ \theta \} \{ \\
x := x - 1 \\
y := y - 1 \\
\} \\
\{ i = j \rightarrow y = 0 \}
\end{array}
\]

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

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

2