- Prepare your solutions on paper.
- Mark the exercises in OLAT before the deadline.
- Marking an exercise means that a significant part of that exercise has been treated.

Exercise 1 *Semantics of Imperative Programs* **14 p.**

An alternative semantics of imperative programs is the big-step semantics. It is defined as follows:

$$\frac{}{(x := e, \alpha) \to \alpha[x := [\![e]\!]_\alpha]}$$

$$\frac{(C_1, \alpha) \to \beta \quad (C_2, \beta) \to \gamma}{(C_1; C_2, \alpha) \to \gamma}$$

$$\frac{[\![b]\!]_\alpha = \mathsf{true} \quad (C_1, \alpha) \to \beta}{(\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2, \alpha) \to \beta} \qquad \frac{[\![b]\!]_\alpha = \mathsf{false} \quad (C_2, \alpha) \to \beta}{(\texttt{if } b \texttt{ then } C_1 \texttt{ else } C_2, \alpha) \to \beta}$$

$$\frac{[\![b]\!]_\alpha = \mathsf{false}}{(\texttt{while } b\ \{C\}, \alpha) \to \alpha} \qquad \frac{[\![b]\!]_\alpha = \mathsf{true} \quad (C, \alpha) \to \beta \quad (\texttt{while } b\ \{C\}, \beta) \to \gamma}{(\texttt{while } b\ \{C\}, \alpha) \to \gamma}$$

$$\frac{}{(\texttt{skip}, \alpha) \to \alpha}$$

1. Consider the program *Fact* on slide 6/9.
   (a) Calculate 1! by evaluating $(\mathit{Fact}, [x := 1])$ using the small-step semantics. (2 points)
   (b) Calculate 1! by evaluating $(\mathit{Fact}, [x := 1])$ using the big-step semantics. (2 points)
2. Both the big- and the small-step semantics have their respective advantages when conducting proofs. However, to be able to switch between both semantics one needs to show that they are equivalent.

   Prove one part of this equivalence, namely (4 points)

   $$(C, \alpha) \to \beta \longrightarrow (C, \alpha) \hookrightarrow^* (\texttt{skip}, \beta)$$

   Here you may use an auxiliary lemma:

   $$(C_1, \alpha) \hookrightarrow^* (C_1', \beta) \longrightarrow (C_1; C_2, \alpha) \hookrightarrow^* (C_1'; C_2, \beta)$$

3. Prove the other direction of the equivalence: (6 points)

   $$(C, \alpha) \hookrightarrow^* (\texttt{skip}, \beta) \longrightarrow (C, \alpha) \to \beta$$

   Clearly state which kind of induction you are using.

   Hint: In the proof you will most likely figure out one required auxiliary property of $\hookrightarrow$ that you should clearly state as lemma, but don't need to prove.

Exercise 2 *Proof Tableaux* **6 p.**

Consider the following program

```
q := 0;
while (x >= y) {
  q := q + 1;
  x := x - y
}
```

1. What does the program compute? Formulate a sensible specification in the form of a Hoare triple. (2 points)
2. Prove partial correctness using a proof tableaux, and adjust the specification on demand. (4 points)