universität innsbruck

Program Verification	SS 2023	LVA 703083+703084
Sheet 11		Deadline: June 8, 2022, 10am

- 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 *Equational Reasoning with Conditions* **12 p.**

In the lecture we proved soundness of insertion sort using some auxiliary properties which haven't been proven. The task of this exercise is to conduct one of these missing proofs.

1. As first step we want to be able to easily switch between the programmed conjunction and and the Boolean connective $\wedge$. Therefore, prove the equivalence $\forall x, y.\ \mathsf{and}(x, y) =_{\mathsf{Bool}} \mathsf{True} \longleftrightarrow x =_{\mathsf{Bool}} \mathsf{True} \wedge y =_{\mathsf{Bool}} \mathsf{True}$. Note that afterwards this property can be used as axiom in the same way as we used the equivalences for (dis)-equalities of constructors. (3 points)
2. As already mentioned in the lecture, transitivity of le is essential. Prove it.

   Hint: you need to perform case-analysis at one point, similar to slides 5/42–43. Here, the cases will be of the form Zero and $\mathsf{Succ}(x)$ for some fresh variable x. (5 points)
3. Prove $\vec{\forall}\,\mathsf{le}(x, y) \longrightarrow \mathsf{sorted}(\mathsf{Cons}(y, zs)) \longrightarrow \mathsf{all\_le}(x, \mathsf{Cons}(y, zs))$.

   You should use the equivalence of the previous part and also the transitivity lemma. (4 points)

Exercise 2 *Semantics of Imperative Programs* **8 p.**

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

$$\frac{}{(x := e, \alpha) \to \alpha[x := [\![e]\!]_\alpha]} \qquad \frac{(C_1, \alpha) \to \beta \quad (C_2, \beta) \to \gamma}{(C_1; C_2, \alpha) \to \gamma}$$

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

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

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

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

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

Here you may use an auxiliary lemma: (4 points)

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