universität
innsbruck

Program Verification — SS 2023 — LVA 703083+703084

Sheet 13 — Deadline: June 21, 2023, 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 *Semantics of Imperative Programs* **6 p.**

Prove the other direction of the equivalence of big-step semantics (see exercise sheet 11) and small-step semantics:

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

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 *Soundness of Hoare-Calculus* **8 p.**

In the lecture we only considered partial correctness of the Hoare-calculus, i.e., we proved:

$$\vdash \llparenthesis \varphi \rrparenthesis \, P \, \llparenthesis \psi \rrparenthesis \longrightarrow \models \llparenthesis \varphi \rrparenthesis \, P \, \llparenthesis \psi \rrparenthesis$$

In this exercise we consider total correctness.

1. We say that a relation $\to$ is deterministic, if for all a there is at most one b such that $a \to b$. Prove that for deterministic $\to$, termination is equivalent to normalization, i.e., there is no infinite $\to$-sequence starting from a is equivalent to $\exists b.\ a \to^! b$. (3 points)
2. Provide a definition of $\models_{total} \llparenthesis \varphi \rrparenthesis \, P \, \llparenthesis \psi \rrparenthesis$, i.e., a semantic notion of total correctness. You can exploit that $\hookrightarrow$ is deterministic. (2 points)
3. How would you try to prove $\vdash \llparenthesis \varphi \rrparenthesis \, P \, \llparenthesis \psi \rrparenthesis \longrightarrow \models_{total} \llparenthesis \varphi \rrparenthesis \, P \, \llparenthesis \psi \rrparenthesis$ for the Hoare-calculus with while-total rule? Just state the main property you would try to prove, and state which proof principle (induction, proof by contradiction, etc.) you would apply, with a brief justification why this looks like a promising attempt. (3 points)

Exercise 3 *Programming by Contract* **6 p.**

1. A contract contains an entry "modifies only," containing a list of variables $v_1, \ldots, v_n$. Now consider a generic method of the following form:

```
int method_name(int x_1, .., int x_m) {
  int y_1, .., int y_k; // local variables
  P; // imperative program as defined in lecture
  return e;
}
```

   Define Hoare-triples whose verification ensures that the "modifies only"-criterion is satisfied, i.e., that the value for all other global variables is unchanged when invoking the program. You can assume that method-parameters and local variables hide visibility of global variables with the same name. (2 points)

2. Apply your approach on the following code to verify that the procedure only modifies the global variable z, where the precondition is $y \leq 501$. (4 points)

```
int foo(int y) {
   z := 0;
   x := x + 501 - y;
   while (y <= 500) {
     z := z + 1;
     y := y + 1
   }
   while (z != 0) {
     z := z - 1;
     y := x + y * 3;
     x := x - 1
   }
   return x + y + z;
}
```