

# Home Assignments 3

## for Programming Theory (TDDA 43)

Deadline: Friday week 21 (22 May 2009) at noon.  
Homework reporting session: Monday week 22.

General requirements and hints concerning correctness proofs:

All steps in your proofs should be explained. For instance, if rule [cons] uses  $P \Rightarrow Q[x \mapsto a]$  as a premise then you should write what  $Q[x \mapsto a]$  is and show that the implication holds. An example of insufficient explanation is the sentence “It is easy to verify that  $(\neg(x = 1) \wedge INV) \Rightarrow (INV[x \mapsto x-1])[y \mapsto y * x]$ ” from p. 218 of our textbook.

Writing proofs as inference trees may be cumbersome; it is better to present them as proof outlines (see the slides) and separately justify the employed implications etc.

We suggest using the intensional way of expressing assertions (like at the lectures, see the slides and notes), not the extensional one (as in the book). The former is easier. Also it is usually (always?) simpler to use the total correctness rule for **while** given at the lectures, not that in the book.

For each assertion (pre- or postcondition in the terminology of the book) that you use in your proofs, convince yourself that the assertion indeed holds whenever the computation reaches the corresponding program point. This is a test able to disclose some errors in the proof. You are not required to write this down.

A. Consider the program

$$S_A = x := x + y; (y := x - y; x := x - y)$$

Find out what the program does. Express this by a pre- and postcondition and prove its partial correctness.

B. Assume that a new loop statement

**repeat**  $S$  **until**  $b$

is added to the language **While**. ( $S$  is executed repetitively at least once. The loop is terminated when the value of  $b$  is true.)

Augment the partial correctness axiomatic semantics of **While** by a proof rule for this statement. Explain (informally) that your rule is correct.

Your rule should not refer to the **while** statement.

Is your rule sufficient to prove the following partial correctness formulae?

$$\begin{aligned} \{x = 1\} \text{ repeat } x := x + 10 \text{ until } x > 50 \{x = 51\} \\ \{x = 7\} \text{ repeat } x := x + 10 \text{ until } x > 0 \{x = 17\} \end{aligned}$$

For each of them construct a proof or explain the reason of failure.

C. Find a precondition  $P_t$  under which program

$$S_C = \text{while } \neg(x = 100) \text{ do } x := x + y$$

terminates, and a precondition  $P_l$  under which the program loops. Prove the non-termination by proving partial correctness  $\{P_l\} S_C \{\mathbf{false}\}$ . Prove the termination by proving total correctness  $\{P_t\} S_C \{\Downarrow \mathbf{true}\}$ .

Each precondition should imply that  $y > 1$ . Each precondition should allow infinitely many distinct values of  $x$ .

If you find this problem too easy, here is a more ambitious version. Replace the program by

$$y := 0; \text{ while } x \neq y \text{ do } (y := y + 1; x := x + y).$$

You may use  $S(n)$  to denote the sum  $\sum_{i=1}^n i = (n+1)n/2$ .

D.1. Let  $F(i)$  denote the  $i$ -th Fibonacci number. (The first two Fibonacci numbers are  $F(1) = F(2) = 1$ , the remaining ones are defined by  $F(i) = F(i-2) + F(i-1)$ , for  $i = 3, 4, \dots$ ) Construct, together with a total correctness proof, a program establishing the postcondition

$$Q = \exists i: F(i) < n \leq F(i+1) = y$$

In other words, the program produces  $y$  which is the first Fibonacci number not smaller than the given number  $n$ . Use  $P = (n > 1)$  as the precondition. (The program should not change the value of  $n$ .)

Hints: Add  $x = F(i)$  to  $Q$  in order to obtain the postcondition of the loop in the program. (This is a design decision – to keep the previous Fibonacci number in  $x$ .) Obtain the loop invariant by weakening the postcondition. Find a reasonable bound function  $z$ ; construct the loop body so that it decreases  $z$  (in other words, comes closer to the solution) and preserves the invariant.

2. Introduce an error to your program and explain why it now cannot be proven correct. Try to use a kind of error that often happens in programming.

Put your solutions, addressed to Jonas Wallgren, in the “Post till IDA” slot at Café Java. Keep a copy of your solutions for the homework reporting session. Your answers may be in English or Swedish.

It is allowed to discuss the exercises with others, but you are supposed to solve each exercise individually. It is absolutely not allowed to copy solutions from others.

The maximal marks for the problems A, B, C and D are, respectively, 2, 3, 3 and 7 (with 5 for the more difficult version of C). To pass you need at least 8 points.