Deductive Verification of Software Exercises

(6.0 VU Formal Methods in Computer Science)

Gernot Salzer

WS 2013

If you want to have your solutions corrected until December 9 (discussion of solutions), please hand them in via Tuwel until Sunday, December 1.

Exercise 1 Let p be the following program:

```
x := x + y;
if x < 0 then
   abort
else
   while x ≠ y do
      x := x + 1;
      y := y + 2
   od
fi
```

(a) Show that p is syntactically correct with respect to the definition of TPL.

(b) Let σ be a state such that $\sigma(x) = 1$ and $\sigma(y) = 5$. Compute $[p]\,\sigma$, using

- the structural operational semantics
- the natural semantics

of TPL.

(c) Show that $\{\, x = 2y \wedge y > 2 \,\}\, p \,\{\, x = y \,\}$ is totally correct

- by computing the weakest precondition of the program;
- using the Hoare calculus;
- using annotation rules.

Exercise 2 Prove that $[p; q]\,\sigma = [q]\,[p]\,\sigma$ holds for all programs p and q and all states σ. Note that we have defined $[p; q]$, $[q]$ and $[p]$ via transition systems (structural operational semantics).

Exercise 3 Show that the two if-rules

$$\frac{\{\,F \wedge e\,\}\,p\,\{\,G\,\} \quad \{\,F \wedge \neg e\,\}\,q\,\{\,G\,\}}{\{\,F\,\}\ \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}\,\{\,G\,\}}\ (\mathrm{if})$$

$$\frac{\{\,F\,\}\,p\,\{\,H\,\} \quad \{\,G\,\}\,q\,\{\,H\,\}}{\{\,(e \Rightarrow F) \wedge (\neg e \Rightarrow G)\,\}\ \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}\,\{\,H\,\}}\ (\mathrm{if}')$$

are equivalent, i.e., that a complete calculus needs only one of the rules.

Hint: Show that each rule can be derived from the other one.

Exercise 4 Consider the following modified if-rule:

$$\frac{\{\,F\,\}\,p\,\{\,G\,\} \quad \{\,F\,\}\,q\,\{\,G\,\}}{\{\,F\,\}\ \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}\,\{\,G\,\}}\ (\mathrm{if}'')$$

(a) Show that the rule is admissible (for partial and total correctness).

Hint: Derive the new rule using rules that we alreay know to be admissible.

(b) Show that the Hoare calculus is no longer complete, if the regular if-rules (if) and (if′) are replaced by the rule (if″).

Hint: Find a correctness assertion that is correct (argue why it is!) but that cannot be derived in the modified calculus (explain why it can't!).

Exercise 5 Determine the strongest postcondition of while-loops, i.e., find a formula equivalent to $\mathrm{sp}(F, \mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od})$ similar to the weakest precondition in the course.

Exercise 6 Show that wp and wlp are dual to each other, i.e., show that $\mathrm{wlp}(p, G) = \neg\mathrm{wp}(p, \neg G)$ holds. Use this relationship to find a formula for $\mathrm{wlp}(\mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}, G)$ similar to the weakest precondition in the course.

Use your formula to compute the weakest liberal precondition of the program

$$z := 0;\ \mathsf{while}\ y \neq 0\ \mathsf{do}\ z := z + x;\ y := y - 1\ \mathsf{od}$$

with respect to the postcondition $z = x * y_0$. Compare the result to the weakest precondition computed in the course and explain the differences.

Exercise 7 Verify that the following program doubles the value of x. For which inputs does it terminate? Choose appropriate pre- and postconditions and show that the assertion is totally correct. Use $y = 2x_0 + x$ as a starting point for the invariant, where x_0 denotes the initial value of x.

$y := 3x;$
while $2x \neq y$ **do**
  $x := x + 1;$
  $y := y + 1;$
od

Exercise 8 Prove the total correctness of the following assertion. Describe the function computed by the program.

$\{ x \geq 0 \}$
$y := 0;$
$z := x + 1;$
while $y + 1 \neq z$ **do**
  $t := (y + z)/2;$
  if $t^2 \leq x$ **then**
    $y := t;$
  else
    $z := t$
  fi
od
$\{ y^2 \leq x < (y + 1)^2 \}$

Hint: Use the invariant $y < z \leq x + 1 \wedge y^2 \leq x < z^2$.

Exercise 9 Prove that the rule

$$\frac{\{ \mathit{Inv} \wedge e \}\, p \,\{ \mathit{Inv} \}}{\{ \mathit{Inv} \}\ \mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}\ \{ \mathit{Inv} \wedge \neg e \}}\ \text{(wh)}$$

is correct regarding partial correctness, i.e., show that $\{ \mathit{Inv} \}$ **while** e **do** p **od** $\{ \mathit{Inv} \wedge \neg e \}$ is partially correct whenever $\{ \mathit{Inv} \wedge e \}\, p \,\{ \mathit{Inv} \}$ is partially correct.

Exercise 10 Extend TPL by statements of the form "**assert** e". When the condition e evaluates to true, the program continues, otherwise the program aborts.

Specify the syntax and semantics of the extended language. Determine the weakest precondition, the weakest liberal precondition, the strongest postcondition, and Hoare rules (partial and total correctness) for **assert**-statements. Show that they are correct.

Treat the **assert**-statement as a first-class citizen, i.e., do not refer to other program statements in the final result. However, you may use other statements as intermediate steps when deriving the rules.