Deductive Verification of Software
Exercise session

(6.0 VU Formal Methods in Computer Science)

Gernot Salzer

9 May 2018

Exercise 1

Prove that $\mathrm{sp}(F, \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}) = \mathrm{sp}(F \wedge e, p) \vee \mathrm{sp}(F \wedge \neg e, q)$.

Solution

Using the definition of sp:

$$\mathrm{sp}(\{\, F \,\}, p) = \{\, [p]\,\sigma \mid \sigma \in \{\, F \,\} \,\} = [p]\,(\{\, F \,\})$$

and the (natural) semantics of the if-statement:

$$[\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}]\,\sigma = \begin{cases} [p]\,\sigma & \text{if } [e]\,\sigma \neq 0 \\ [q]\,\sigma & \text{if } [e]\,\sigma = 0 \end{cases}$$

we obtain the strongest postcondition for the if-statement as follows:

$\mathrm{sp}(\{\, F \,\}, \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi})$	
$= [\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}]\,(\{\, F \,\})$	definition of sp
$= [\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}]\,(\{\, F \wedge e \,\} \cup \{\, F \wedge \neg e \,\})$	propositional logic and sets
$= [\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}]\,(\{\, F \wedge e \,\})$	property of functions and sets:
$\quad \cup\, [\mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}]\,(\{\, F \wedge \neg e \,\})$	$f(A \cup B) = f(A) \cup f(B)$
$= [p]\,(\{\, F \wedge e \,\}) \cup [q]\,(\{\, F \wedge \neg e \,\})$	semantics of the if-statement
$= \mathrm{sp}(\{\, F \wedge e \,\}, p) \cup \mathrm{sp}(\{\, F \wedge \neg e \,\}, q)$	definition of sp

If $\mathrm{sp}(\{\, F \wedge e \,\}, p)$ and $\mathrm{sp}(\{\, F \wedge \neg e \,\}, q)$ are given as formulas, we have

$$\mathrm{sp}(F, \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}) = \mathrm{sp}(F \wedge e, p) \vee \mathrm{sp}(F \wedge \neg e, q)\ .$$

Exercise 2

Determine the strongest postcondition of the weakest precondition of an assignment statement, i.e., compute $\mathrm{sp}(\mathrm{wlp}(v := e, G), v := e)$. Why is it different from G?

Solution

$$\begin{aligned}\mathrm{sp}(\mathrm{wlp}(v := e, G), v := e) &= \mathrm{sp}(G[v/e], v := e)\\ &= \exists v' \, (G[v/e][v/v'] \wedge v = e[v/v'])\end{aligned}$$

$G[v/e]$ contains only those occurrences of v that are introduced by e; all other occurrences have been replaced by e. Therefore $G[v/e][v/v']$ is the same as $G[v/e[v/v']]$.

$$= \exists v' \, (G[v/e[v/v']] \wedge v = e[v/v'])$$

By the second conjunct we know that $e[v/v']$ equals v.

$$\begin{aligned}&= \exists v' \, (G[v/v] \wedge v = e[v/v'])\\ &= \exists v' \, (G \wedge v = e[v/v'])\\ &= G \wedge \exists v' \, (v = e[v/v'])\end{aligned}$$

Why is the result different from G? G may admit states that cannot be obtained by executing the assignment and therefore do not appear in the strongest postcondition. The additional existential formula ensures that the value of v is the result of evaluating the expression e for some input state. For example, the postcondition $G = \text{true}$ after the assignment $x := 2x$ is also satisfied by states assigning an odd number to x; the additional formula $\exists x'(x = 2x')$ ('x is even') excludes such states.

Exercise 3

Show that the following assertion is totally correct. Describe the function computed by the program.

$\{\, \mathit{Pre}\colon x \geq 1 \wedge y \geq 2 \,\}$
$u := y;$
$z := 0;$
while $u \leq x$ **do**
  $u := u * y;$
  $z := z + 1$
od
$\{\, \mathit{Post}\colon y^z \leq x < y^{z+1} \,\}$

Hint: Use the invariant $\mathit{Inv}\colon u = y^{z+1} \wedge 2 \leq y \leq u \leq xy$.

Solution

We start by adding further assertions to the program, according to the rules of the annotation calculus. The formulas are numbered in the order in which they are added.

$\{\, \mathit{Pre}\colon x \geq 1 \wedge y \geq 2 \,\}$
$\{\, F_6\colon \mathit{Inv}[z/0][u/y] \,\}$
$u := y;$
$\{\, F_5\colon \mathit{Inv}[z/0] \,\}$
$z := 0;$
$\{\, F_1\colon \mathit{Inv} \,\}$
while $u \leq x$ **do**
  $\{\, F_2\colon \mathit{Inv} \wedge u \leq x \wedge t = t_0 \,\}$
  $\{\, F_8\colon (\mathit{Inv} \wedge (u \leq x \Rightarrow 0 \leq t < t_0))[z/z+1][u/uy] \,\}$
  $u := u * y;$
  $\{\, F_7\colon (\mathit{Inv} \wedge (u \leq x \Rightarrow 0 \leq t < t_0))[z/z+1] \,\}$
  $z := z + 1$
  $\{\, F_3\colon \mathit{Inv} \wedge (u \leq x \Rightarrow 0 \leq t < t_0) \,\}$
od
$\{\, F_4\colon \mathit{Inv} \wedge \neg(u \leq x) \,\}$
$\{\, \mathit{Post}\colon y^z \leq x < y^{z+1} \,\}$

It remains to show the validity of the three implications $\mathit{Pre} \Rightarrow F_6$, $F_4 \Rightarrow \mathit{Post}$, and $F_2 \Rightarrow F_8$.

***Pre* ⇒ F_6:**

$$(x \geq 1 \wedge y \geq 2) \Rightarrow \mathit{Inv}[z/0][u/y]$$
$$(x \geq 1 \wedge y \geq 2) \Rightarrow (y = y^{0+1} \wedge 2 \leq y \leq y \leq xy)$$

We have to show that each conjunct in the conclusion is true (under the assumption that the premises are true).

conclusion	why it holds
$y = y^{0+1}$	Equivalent to the valid formula $y = y$
$2 \leq y$	Is one of the premises.
$y \leq y$	Valid (reflexivity of $\leq$)
$y \leq xy$	Since y is positive (premise $y \geq 2$), multiplying the premise $x \geq 1$ by y yields the conclusion $y \leq xy$

***F*₄ ⇒ *Post*:**

$$\mathit{Inv} \wedge \neg(u \leq x) \Rightarrow \mathit{Post}$$
$$(u = y^{z+1} \wedge 2 \leq y \leq u \leq xy \wedge u > x) \Rightarrow y^z \leq x < y^{z+1}$$

conclusion	why it holds
$y^z \le x$	From the premises $u = y^{z+1}$ and $u \le xy$ we obtain $y^{z+1} \le xy$. Since y is positive (premise $y \ge 2$), dividing the inequality by y yields the conclusion $y^z \le x$.
$x < y^{z+1}$	Follows from the premises $u = y^{z+1}$ and $u > x$.

$F_2 \Rightarrow F_8$:

$$(Inv \wedge u \le x \wedge t = t_0) \Rightarrow (Inv \wedge (u \le x \Rightarrow 0 \le t < t_0))[z/z+1][u/uy]$$
$$(Inv \wedge u \le x \wedge t = t_0) \Rightarrow (Inv[z/z+1][u/uy] \wedge (uy \le x \Rightarrow 0 \le t[z/z+1][u/uy] < t_0))$$

Proving an implication $F \Rightarrow (G \wedge H)$ is equivalent to proving the implications $F \Rightarrow G$ and $F \Rightarrow H$ separately. Here the first implication corresponds to partial correctness and the second one to termination.

Partial correctness: We omit the premise $t = t_0$ as it cannot contribute anything to the proof; it just states that some term t equals a variable t_0 that does not occur anywhere else in the formula.

$$(Inv \wedge u \le x) \Rightarrow Inv[z/z+1][u/uy]$$
$$(u = y^{z+1} \wedge 2 \le y \le u \le xy \wedge u \le x) \Rightarrow (uy = y^{z+2} \wedge 2 \le y \le uy \le xy)$$

conclusion	why it holds
$uy = y^{z+2}$	Obtained by multiplying the premise $u = y^{z+1}$ by y.
$y \ge 2$	Is one of the premises.
$y \le uy$	Since y is positive (premise $y \ge 2$), its product with another positive number (the premises $y \le u$ and $y \ge 2$ imply $u \ge 2$) is greater than or equal to y.
$uy \le xy$	Since y is positive (premise $y \ge 2$), multiplying the premise $u \le x$ by y yields the conclusion $uy \le xy$.

Termination:

$$(Inv \wedge u \le x \wedge t = t_0) \Rightarrow (uy \le x \Rightarrow 0 \le t[z/z+1][u/uy] < t_0)$$
$$(Inv \wedge u \le x) \Rightarrow (uy \le x \Rightarrow 0 \le t[z/z+1][u/uy] < t)$$
$$(Inv \wedge u \le x \wedge uy \le x) \Rightarrow (0 \le t[z/z+1][u/uy] < t)$$

In the first step we use the equation $t = t_0$ to eliminate t_0 on the right-hand side; after that we do not need the equation anymore and omit it. In the second step we use the fact that $F \Rightarrow (G \Rightarrow H)$ is equivalent to $(F \wedge G) \Rightarrow H$.

To guess a suitable variant t we rewrite the loop condition $u \le x$ to $x - u \ge 0$ and choose $t \stackrel{\text{def}}{=} x - u$. The loop condition ensures that $t \ge 0$ holds within the loop; moreover, since the loop increases u we may expect that t decreases. It remains to prove these two properties formally by showing the validity of the implication.

$$(Inv \wedge u \le x \wedge uy \le x) \Rightarrow (0 \le t[z/z+1][u/uy] < t)$$
$$(u = y^{z+1} \wedge 2 \le y \le u \le xy \wedge u \le x \wedge uy \le x) \Rightarrow (0 \le x - uy < x - u)$$

conclusion	why it holds
$0 \leq x - uy$	Equivalent to the premise $uy \leq x$.
$x - uy < x - u$	This inequality is equivalent to $u < uy$. Since u is positive (the premises $y \leq u$ and $y \geq 2$ imply $u \geq 2$) and y is greater than one (premise $y \geq 2$), the product uy is strictly greater than u.

Therefore the given assertion is totally correct.

Function computed by the program: To see what the program does it suffices to analyze the postcondition and to express z as a function of x and y.

$$
\begin{aligned}
& y^z \leq x < y^{z+1} \\
& z \leq \log_y(x) < z + 1 && \text{(Take the logarithm to base } y.\text{)} \\
& z = \lfloor \log_y(x) \rfloor && (a = \lfloor b \rfloor \text{ is equivalent to } a \leq b < a + 1.)
\end{aligned}
$$

Thus the program computes the integer logarithm to base y of x.

Exercise 4

Show that the following rule is admissible regarding partial correctness.

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

Solution

We show that the conclusion of the rule (if″) can be derived from its premises using the rules (if) and (lc). Since the latter rules are correct for proving (partial and total) correctness, the former is also correct.

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

The premises $F \wedge e \Rightarrow F$ and $F \wedge \neg e \Rightarrow F$ are tautologies, hence the partial (or total) correctness of $\{\,F\,\}\,p\,\{\,G\,\}$ and $\{\,F\,\}\,q\,\{\,G\,\}$ implies the partial (or total) correctness of $\{\,F\,\}\ \mathsf{if}\ e\ \mathsf{then}\ p\ \mathsf{else}\ q\ \mathsf{fi}\ \{\,G\,\}$.

Exercise 5

Show that the Hoare calculus is no longer complete if we replace the regular if-rule by the following one.

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

Solution

We give a counter-example, i.e., we present a correctness assertion that is true but not derivable in the modified calculus. Consider the correctness assertion

$$\{\,\mathrm{true}\,\}\,\mathsf{if}\ x = y\ \mathsf{then}\ x := x+1\ \mathsf{else\ skip\ fi}\,\{\,x \neq y\,\}\ .$$

(This assertion is different from the one used in class and in the hand-written notes, since the following presentation was already prepared in advance. It is straight-forward to adapt the arguments below to the counter-example used in class.)

The above assertion is partially and totally correct, as we show using the annotation calculus.

$$\begin{array}{l}
\{\,F_1 \colon \mathrm{true}\,\}\\
\mathsf{if}\ x = y\ \mathsf{then}\\
\quad \{\,F_7 \colon \mathrm{true} \wedge x = y\,\} \quad \mathrm{if}{\downarrow}\\
\quad \{\,F_6 \colon x+1 \neq y\,\} \quad \mathrm{as}{\uparrow}\\
\quad x := x+1\\
\quad \{\,F_4 \colon x \neq y\,\} \quad \mathrm{fi}{\uparrow}\\
\mathsf{else}\\
\quad \{\,F_8 \colon \mathrm{true} \wedge x \neq y\,\} \quad \mathrm{if}{\downarrow}\\
\quad \{\,F_5 \colon x \neq y\,\} \quad \mathrm{sk}{\uparrow}\\
\quad \mathsf{skip}\\
\quad \{\,F_3 \colon x \neq y\,\} \quad \mathrm{fi}{\uparrow}\\
\mathsf{fi}\\
\{\,F_2 \colon x \neq y\,\}
\end{array}$$

The two implications $\mathrm{true} \wedge x = y \Rightarrow x+1 \neq y$ ($F_7 \Rightarrow F_6$) and $\mathrm{true} \wedge x \neq y \Rightarrow x \neq y$ ($F_8 \Rightarrow F_5$) are obviously valid, which concludes the correctness proof.

To show that the assertion cannot be derived in the modified calculus we give a proof by contradiction: We assume that there is a derivation and argue that this assumption leads to a contradiction. Suppose the assertion can be derived. Then the derivation must have the form

$$\dfrac{\mathrm{true} \Rightarrow F \qquad \dfrac{\dfrac{\text{some derivation of } \{\,F\,\}\,x := x+1\,\{\,G\,\}}{\{\,F\,\}\,x := x+1\,\{\,G\,\}} \quad \dfrac{\text{some derivation of } \{\,F\,\}\,\mathsf{skip}\,\{\,G\,\}}{\{\,F\,\}\,\mathsf{skip}\,\{\,G\,\}}}{\{\,F\,\}\,\mathsf{if}\ x = y\ \mathsf{then}\ x := x+1\ \mathsf{else\ skip\ fi}\,\{\,G\,\}}\ {}^{(\mathrm{if}'')} \qquad G \Rightarrow x \neq y}{\{\,\mathrm{true}\,\}\,\mathsf{if}\ x = y\ \mathsf{then}\ x := x+1\ \mathsf{else\ skip\ fi}\,\{\,x \neq y\,\}}\ \mathrm{lc}$$

for some suitable formulas F and G. Note that lc and if$''$ are the only rules by which we may obtain the proposed counter-example. When constructing the derivation bottom-up, the modified if-rule has to be applied at some point since it is the only rule introducing if-statements. Before that, the lc-rule may be applied several times to strengthen the pre- and weaken the postcondition. Because of the transitivity of implication we may assume without loss of generality that there is exactly *one* application of the lc-rule; this subsumes the use of n successive applications of the rule, for all $n \geq 0$. (The case $n = 0$ corresponds to the situation where F and G are chosen identical to true and $x \neq y$, respectively.)

Now we observe that – no matter how we choose F and G – the implications true $\Rightarrow F$ and $G \Rightarrow x \neq y$ must be valid (otherwise it would not be justified to apply the rule lc) and the assertions $\{\, F \,\}\, x := x + 1 \,\{\, G \,\}$ and $\{\, F \,\}\, \mathsf{skip} \,\{\, G \,\}$ must be true (otherwise they cannot be derived in a correct calculus). We show that this leads to a contradiction. Consider a state σ with $\sigma(x) = 0$ and $\sigma(y) = 0$.

- Since true $\Rightarrow F$ is valid and σ satisfies true ($[\text{true}]\,\sigma = 1$), the state σ also satisfies F ($[F]\,\sigma = 1$).
- Since $\{\, F \,\}\, \mathsf{skip} \,\{\, G \,\}$ is true and σ satisfies F, the state after executing skip satisfies the postcondition, i.e., $[G]\,[\mathsf{skip}]\,\sigma = [G]\,\sigma = 1$.
- Since $G \Rightarrow x \neq y$ is valid and σ satisfies G, the state σ has also to satisfy $x \neq y$, which is obviously false: $[x \neq y]\,\sigma = 0$ since $\sigma(x) = \sigma(y)$.

Summarizing, the assertion above is correct, but cannot be derived in the modified Hoare calculus. Therefore the calculus is not complete.

Exercise 6

Characterize the programs p that satisfy the following conditions.

(a) $\text{wp}(p, \text{true}) = \text{true}$

(b) $\text{wp}(p, \text{true}) = \text{false}$

(c) $\text{wp}(p, \text{false}) = \text{true}$

(d) $\text{wp}(p, \text{false}) = \text{false}$

Solution

(a) A program p satisfies this property iff it terminates for all inputs.

(b) A program p satisfies this property iff it does not terminate for any input.

(c) No program p satisfies this property.

(d) This property holds for all programs p.