COMSM0067: Advanced Topics in Programming Languages
Problem Sheet 8 — Solutions

These solutions follow the rules and proof style introduced in Lecture 8: Assign works backwards by substitution, Seq threads midpoint assertions together, If requires both branches to establish a common postcondition, Conseq handles logical implication, and While checks a supplied invariant.

1. Derivations

(i)

$$\{x = 3\}\ y := x + 1\ \{y = 4\}.$$

The assignment axiom gives

$$\frac{}{\{x + 1 = 4\}\ y := x + 1\ \{y = 4\}}\ \textsc{Assign}.$$

Since

$$x = 3 \Rightarrow x + 1 = 4,$$

the rule of consequence gives

$$\frac{x = 3 \Rightarrow x + 1 = 4 \qquad \{x + 1 = 4\}\ y := x + 1\ \{y = 4\}}{\{x = 3\}\ y := x + 1\ \{y = 4\}}\ \textsc{Conseq}.$$

(ii)

$$\{\mathsf{True}\}\ x := 1;\ y := x + 1\ \{y = 2\}.$$

For the second assignment, working backwards from $y = 2$,

$$\frac{}{\{x + 1 = 2\}\ y := x + 1\ \{y = 2\}}\ \textsc{Assign}.$$

Since

$$x = 1 \Rightarrow x + 1 = 2,$$

we obtain

$$\{x = 1\}\ y := x + 1\ \{y = 2\}\quad \textsc{Conseq}.$$

For the first assignment,

$$\frac{}{\{1 = 1\}\ x := 1\ \{x = 1\}}\ \textsc{Assign}.$$

Since $\mathsf{True} \Rightarrow 1 = 1$,

$$\{\mathsf{True}\}\ x := 1\ \{x = 1\}.$$

Now apply sequencing:

$$\frac{\{\mathsf{True}\}\ x := 1\ \{x = 1\} \qquad \{x = 1\}\ y := x + 1\ \{y = 2\}}{\{\mathsf{True}\}\ x := 1;\ y := x + 1\ \{y = 2\}}\ \textsc{Seq}.$$

(iii)

$$\{x = A\}\ \textbf{if}\ x \leq 0\ \textbf{then}\ y := -x\ \textbf{else}\ y := x\ \{y = |A|\}.$$

For the then-branch we need

$$\{x = A \wedge x \leq 0\}\ y := -x\ \{y = |A|\}.$$

The assignment axiom gives

$$\{-x = |A|\}\ y := -x\ \{y = |A|\}.$$

From

$$x = A \wedge x \leq 0$$

we have $A \leq 0$, hence

$$-x = -A = |A|.$$

Therefore, by Conseq,

$$\{x = A \wedge x \leq 0\}\ y := -x\ \{y = |A|\}.$$

For the else-branch we need

$$\{x = A \wedge x > 0\}\ y := x\ \{y = |A|\}.$$

The assignment axiom gives

$$\{x = |A|\}\ y := x\ \{y = |A|\}.$$

From $x = A \wedge x > 0$, we have $A > 0$, so

$$x = A = |A|.$$

Hence

$$\{x = A \wedge x > 0\}\ y := x\ \{y = |A|\}.$$

Applying If,

$$\frac{\{x = A \wedge x \leq 0\}\ y := -x\ \{y = |A|\} \qquad \{x = A \wedge x > 0\}\ y := x\ \{y = |A|\}}{\{x = A\}\ \textbf{if}\ x \leq 0\ \textbf{then}\ y := -x\ \textbf{else}\ y := x\ \{y = |A|\}}\ \textsc{If}.$$

(iv)

$$\{\mathsf{False}\}\ x := x + 1\ \{x = 100\}.$$

The assignment axiom gives

$$\frac{}{\{x + 1 = 100\}\ x := x + 1\ \{x = 100\}}\ \textsc{Assign}.$$

Since False implies every assertion, in particular

$$\mathsf{False} \Rightarrow x + 1 = 100,$$

the rule of consequence gives

$$\frac{\mathsf{False} \Rightarrow x + 1 = 100 \qquad \{x + 1 = 100\}\ x := x + 1\ \{x = 100\}}{\{\mathsf{False}\}\ x := x + 1\ \{x = 100\}}\ \textsc{Conseq}.$$

The same two steps work for *any* postcondition: with precondition False the triple is derivable whatever the program is meant to establish, because there is no starting state to check.

2. Conditional swap-and-order program

We prove

$$\{x = A \wedge y = B\}$$
$$\textbf{if } x > y \textbf{ then } (t := x;\ x := y;\ y := t) \textbf{ else skip}$$
$$\{x \le y \wedge x + y = A + B\}.$$

Let

$$Q \equiv x \le y \wedge x + y = A + B.$$

Then branch

Work backwards through

$$t := x;\ x := y;\ y := t.$$

Starting from

$$Q \equiv x \le y \wedge x + y = A + B,$$

before $y := t$ we need

$$\boxed{x \le t \wedge x + t = A + B}.$$

Before $x := y$ we therefore need

$$\boxed{y \le t \wedge y + t = A + B}.$$

Before $t := x$ we need

$$\boxed{y \le x \wedge y + x = A + B}.$$

Thus three applications of Assign and two of Seq give

$$\{y \le x \wedge y + x = A + B\}\ t := x;\ x := y;\ y := t\ \{Q\}.$$

The actual branch precondition is

$$x = A \wedge y = B \wedge x > y.$$

It implies the calculated precondition because

$$x > y \Rightarrow y \le x$$

and

$$x = A \wedge y = B \Rightarrow y + x = B + A = A + B.$$

Hence, by Conseq,

$$\boxed{\{x = A \wedge y = B \wedge x > y\}\ t := x;\ x := y;\ y := t\ \{Q\}.}$$

Else branch

The branch assumption is

$$x = A \wedge y = B \wedge \neg(x > y).$$

Since $\neg(x > y)$ means $x \leq y$, and $x = A, y = B$ imply

$$x + y = A + B,$$

we have

$$x = A \wedge y = B \wedge \neg(x > y) \Rightarrow Q.$$

The Skip axiom gives

$$\{Q\}\ \textbf{skip}\ \{Q\},$$

so Conseq yields

$$\boxed{\{x = A \wedge y = B \wedge \neg(x > y)\}\ \textbf{skip}\ \{Q\}.}$$

Now both branches establish the same postcondition Q, so If gives

$$\boxed{\{x = A \wedge y = B\}\ \textbf{if}\ x > y\ \textbf{then}\ (t := x;\ x := y;\ y := t)\ \textbf{else skip}\ \{x \leq y \wedge x + y = A + B\}.}$$

The rule used to bring the two branches to a common postcondition is Conseq; once they share that postcondition, If combines them.