COMSM0067: Advanced Topics in Programming Languages
Problem Sheet for Hoare Logic II — Solutions

Invariants, weakest preconditions

What we need

Problem 1: rules of Hoare logic.

$$\frac{}{\{Q[a/x]\}\ x := a\ \{Q\}}\ \textsc{Assign} \qquad \frac{\{P\}\ c_1\ \{R\} \qquad \{R\}\ c_2\ \{Q\}}{\{P\}\ c_1;\ c_2\ \{Q\}}\ \textsc{Seq}$$

$$\frac{\{P \wedge b\}\ c_1\ \{Q\} \qquad \{P \wedge \neg b\}\ c_2\ \{Q\}}{\{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}\ \textsc{If} \qquad \frac{P \Rightarrow P' \qquad \{P'\}\ c\ \{Q'\} \qquad Q' \Rightarrow Q}{\{P\}\ c\ \{Q\}}\ \textsc{Conseq}$$

Problems 2 and 3: weakest preconditions.

$$\mathsf{wp}(x := a,\ Q) = Q[a/x] \qquad \mathsf{wp}(c_1;\ c_2,\ Q) = \mathsf{wp}(c_1,\ \mathsf{wp}(c_2,\ Q))$$

Problems 2 and 3: verification conditions for a loop. To prove

$$\{P\}\ c_0;\ \textbf{while}\ g\ \textbf{do}\ \mathit{body}\ \{Q\}$$

with a loop invariant I, we show the three verification conditions from the back of the sheet (we write g for the guard, because Problem 2 has a program variable called b):

VC1	(init)	$P \Rightarrow \mathsf{wp}(c_0, I)$
VC2	(preserve)	$I \wedge g \Rightarrow \mathsf{wp}(\mathit{body}, I)$
VC3	(exit)	$I \wedge \neg g \Rightarrow Q$

1. Derivations

1(i)

We derive $\{x \geq 0 \wedge y \geq 0\}\ x := x + 1;\ y := y + x\ \{y > 0\}$.

Work backwards from the postcondition with Assign:

$$\{y + x > 0\}\ y := y + x\ \{y > 0\} \qquad \{y + (x + 1) > 0\}\ x := x + 1\ \{y + x > 0\}$$

Join them with Seq; call this derivation D:

$$\frac{\dfrac{}{\{x + y + 1 > 0\}\ x := x + 1\ \{y + x > 0\}}\ \textsc{Assign} \qquad \dfrac{}{\{y + x > 0\}\ y := y + x\ \{y > 0\}}\ \textsc{Assign}}{\{x + y + 1 > 0\}\ x := x + 1;\ y := y + x\ \{y > 0\}}\ \textsc{Seq}$$

Then use Conseq to strengthen the precondition, since $x \geq 0 \wedge y \geq 0 \Rightarrow x + y + 1 > 0$:

$$\frac{x \geq 0 \wedge y \geq 0 \Rightarrow x + y + 1 > 0 \qquad D}{\{x \geq 0 \wedge y \geq 0\}\ x := x + 1;\ y := y + x\ \{y > 0\}}\ \textsc{Conseq}$$

1(ii)

We derive $\{\mathsf{True}\}$ **if** $x < y$ **then** $m := x$ **else** $m := y$ $\{m = \min(x, y)\}$.

The If rule needs one derivation per branch, each assuming the guard or its negation.

Then-branch D_1:

$$\dfrac{\mathsf{True} \wedge x < y \Rightarrow x = \min(x,y) \qquad \dfrac{}{\{x = \min(x,y)\}\ m := x\ \{m = \min(x,y)\}}\ \textsc{Assign}}{\{\mathsf{True} \wedge x < y\}\ m := x\ \{m = \min(x,y)\}}\ \textsc{Conseq}$$

Else-branch D_2:

$$\dfrac{\mathsf{True} \wedge \neg(x < y) \Rightarrow y = \min(x,y) \qquad \dfrac{}{\{y = \min(x,y)\}\ m := y\ \{m = \min(x,y)\}}\ \textsc{Assign}}{\{\mathsf{True} \wedge \neg(x < y)\}\ m := y\ \{m = \min(x,y)\}}\ \textsc{Conseq}$$

Both side conditions hold: if $x < y$ the minimum is x; if $x \geq y$ it is y. Finally,

$$\dfrac{D_1 \qquad D_2}{\{\mathsf{True}\}\ \textbf{if}\ x < y\ \textbf{then}\ m := x\ \textbf{else}\ m := y\ \{m = \min(x,y)\}}\ \textsc{If}$$

2. Repeated addition

(i) A suitable invariant is

$$\boxed{I \equiv r = a * i \wedge 0 \leq i \leq b.}$$

The annotated program, showing where each verification condition comes from:

$\{P \equiv a \geq 0 \wedge b \geq 0\}$	
$\quad r := 0;\ i := 0;$	c_0
$\{I \equiv r = a * i \wedge 0 \leq i \leq b\}$	VC1
$\quad$ **while** $i < b$ **do**	$g \equiv i < b$
$\qquad \{I \wedge g \equiv r = a * i \wedge 0 \leq i \leq b \wedge i < b\}$	
$\qquad\quad r := r + a;\ i := i + 1$	*body*
$\qquad \{I \equiv r = a * i \wedge 0 \leq i \leq b\}$	VC2
$\{I \wedge \neg g \equiv r = a * i \wedge 0 \leq i \leq b \wedge \neg(i < b)\}$	
$\{Q \equiv r = a * b\}$	VC3

(ii) We prove the three verification conditions.

VC1 (init): I **is established by** $r := 0;\ i := 0$**.** We must show

$$a \geq 0 \wedge b \geq 0 \Rightarrow \mathsf{wp}(r := 0;\ i := 0,\ I).$$

Pushing I up through the two assignments:

$$\mathsf{wp}(i := 0,\ I) = r = a * 0 \wedge 0 \leq 0 \leq b,$$
$$\mathsf{wp}(r := 0,\ r = a * 0 \wedge 0 \leq 0 \leq b) = 0 = a * 0 \wedge 0 \leq 0 \leq b \ \equiv\ b \geq 0.$$

Since $a \geq 0 \wedge b \geq 0 \Rightarrow b \geq 0$, VC1 holds.

VC2 (preserve): I is preserved by the loop body. We must show

$$r = a * i \wedge 0 \le i \le b \wedge i < b \Rightarrow \mathsf{wp}(r := r + a;\ i := i + 1,\ I).$$

Pushing I up through the body:

$$\mathsf{wp}(i := i + 1,\ I) = r = a * (i + 1) \wedge 0 \le i + 1 \le b,$$
$$\mathsf{wp}(r := r + a,\ r = a * (i + 1) \wedge 0 \le i + 1 \le b) = r + a = a * (i + 1) \wedge 0 \le i + 1 \le b.$$

Now assume the left-hand side:

- $r = a * i$ gives $r + a = a * i + a = a * (i + 1)$;
- $0 \le i$ gives $0 \le i + 1$;
- $i < b$ gives $i + 1 \le b$, since i and b are integers.

So VC2 holds.

VC3 (exit): on exit, I implies $r = a * b$. We must show

$$r = a * i \wedge 0 \le i \le b \wedge \neg(i < b) \Rightarrow r = a * b.$$

From $i \le b$ and $\neg(i < b)$ we get $i = b$, so $r = a * i = a * b$. So VC3 holds.

3. Fibonacci

(i) A suitable invariant is

$$\boxed{I \equiv x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i + 1) \wedge 0 \le i \le n.}$$

It must mention y: the body sets x to the old y, so to know $x = \mathsf{fib}(i + 1)$ after an iteration we need $y = \mathsf{fib}(i + 1)$ before it.

The annotated program, showing where each verification condition comes from:

$\{P \equiv n \ge 0\}$	
$\quad x := 0;\ y := 1;\ i := 0;$	c_0
$\{I \equiv x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i + 1) \wedge 0 \le i \le n\}$	VC1
$\quad$ **while** $i < n$ **do**	$g \equiv i < n$
$\quad\quad \{I \wedge g \equiv x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i + 1) \wedge 0 \le i \le n \wedge i < n\}$	
$\quad\quad\quad t := x + y;\ x := y;\ y := t;\ i := i + 1$	*body*
$\quad\quad \{I \equiv x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i + 1) \wedge 0 \le i \le n\}$	VC2
$\{I \wedge \neg g \equiv x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i + 1) \wedge 0 \le i \le n \wedge \neg(i < n)\}$	
$\{Q \equiv x = \mathsf{fib}(n)\}$	VC3

(ii) We prove the three verification conditions.

VC1 (init): I is established by $x := 0;\ y := 1;\ i := 0$. We must show

$$n \ge 0 \Rightarrow \mathsf{wp}(x := 0;\ y := 1;\ i := 0,\ I).$$

Pushing I up through the three assignments:

$$\begin{aligned}
&\text{through } i := 0: && x = \mathsf{fib}(0) \wedge y = \mathsf{fib}(1) \wedge 0 \le 0 \le n \\
&\text{through } y := 1: && x = \mathsf{fib}(0) \wedge 1 = \mathsf{fib}(1) \wedge 0 \le 0 \le n \\
&\text{through } x := 0: && 0 = \mathsf{fib}(0) \wedge 1 = \mathsf{fib}(1) \wedge 0 \le 0 \le n
\end{aligned}$$

Since $\mathsf{fib}(0) = 0$ and $\mathsf{fib}(1) = 1$, the last line is equivalent to $n \ge 0$, which is exactly P. So VC1 holds.

VC2 (preserve): I is preserved by the loop body. We must show

$$x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i+1) \wedge 0 \le i \le n \wedge i < n \Rightarrow \mathsf{wp}(t := x + y;\ x := y;\ y := t;\ i := i + 1,\ I).$$

Pushing I up through the body, right to left:

$$\begin{array}{ll}
\text{through } i := i + 1: & x = \mathsf{fib}(i+1) \wedge y = \mathsf{fib}(i+2) \wedge 0 \le i + 1 \le n \\
\text{through } y := t: & x = \mathsf{fib}(i+1) \wedge t = \mathsf{fib}(i+2) \wedge 0 \le i + 1 \le n \\
\text{through } x := y: & y = \mathsf{fib}(i+1) \wedge t = \mathsf{fib}(i+2) \wedge 0 \le i + 1 \le n \\
\text{through } t := x + y: & y = \mathsf{fib}(i+1) \wedge x + y = \mathsf{fib}(i+2) \wedge 0 \le i + 1 \le n
\end{array}$$

Now assume the left-hand side:

- $y = \mathsf{fib}(i+1)$ is given directly;
- $x + y = \mathsf{fib}(i) + \mathsf{fib}(i+1) = \mathsf{fib}(i+2)$, by the definition of fib, since $i \ge 0$;
- $0 \le i$ gives $0 \le i + 1$, and $i < n$ gives $i + 1 \le n$, since i and n are integers.

So VC2 holds.

VC3 (exit): on exit, I implies $x = \mathsf{fib}(n)$. We must show

$$x = \mathsf{fib}(i) \wedge y = \mathsf{fib}(i+1) \wedge 0 \le i \le n \wedge \neg(i < n) \Rightarrow x = \mathsf{fib}(n).$$

From $i \le n$ and $\neg(i < n)$ we get $i = n$, so $x = \mathsf{fib}(i) = \mathsf{fib}(n)$. So VC3 holds.