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

---

1. Give a derivation of each of the following triples, naming every rule you use.

   (i) $\{x \geq 0 \wedge y \geq 0\}\ x := x + 1;\ y := y + x\ \{y > 0\}$

   (ii) $\{\mathsf{true}\}\ \mathsf{if}\ x < y\ \mathsf{then}\ m := x\ \mathsf{else}\ m := y\ \{m = \min(x, y)\}$

2. Let c be the program

$$
\begin{array}{l}
r := 0; \\
i := 0; \\
\mathsf{while}\ i < b\ \mathsf{do}\ (\, r := r + a;\ \ i := i + 1\,)
\end{array}
$$

   i. State a loop invariant I sufficient to prove $\{a \geq 0 \wedge b \geq 0\}\ c\ \{r = a * b\}$.

   ii. Verify that your I is established by $r := 0;\ i := 0$, that it is preserved by the loop body, and that on exit it implies $r = a * b$.

3. (*) Let c be the program

$$
\begin{array}{l}
x := 0; \\
y := 1; \\
i := 0; \\
\mathsf{while}\ i < n\ \mathsf{do}\ (\, t := x + y;\ \ x := y;\ \ y := t;\ \ i := i + 1\,)
\end{array}
$$

   where fib is the usual Fibonacci function, $\mathrm{fib}(0) = 0$, $\mathrm{fib}(1) = 1$, and $\mathrm{fib}(k{+}2) = \mathrm{fib}(k{+}1) + \mathrm{fib}(k)$.

   i. State a loop invariant I sufficient to prove $\{n \geq 0\}\ c\ \{x = \mathrm{fib}(n)\}$. *Hint:* an invariant mentioning only x and i will not be preserved. Ask yourself what else the loop is maintaining.

   ii. Verify that your I is established by $x := 0;\ y := 1;\ i := 0$, that it is preserved by the loop body, and that on exit it implies $x = \mathrm{fib}(n)$.

Reference: weakest preconditions

Loop-free commands. $\mathsf{wp}(c, Q)$ is the weakest P such that $\{P\}\ c\ \{Q\}$; and $\{P\}\ c\ \{Q\}$ holds iff $P \Rightarrow \mathsf{wp}(c, Q)$.

$$
\begin{array}{lcl}
\mathsf{wp}(\mathsf{skip},\ Q) & = & Q \\
\mathsf{wp}(x := a,\ Q) & = & Q[a/x] \\
\mathsf{wp}(c_1;\ c_2,\ Q) & = & \mathsf{wp}(c_1,\ \mathsf{wp}(c_2,\ Q)) \\
\mathsf{wp}(\mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2,\ Q) & = & (b \Rightarrow \mathsf{wp}(c_1, Q))\ \wedge\ (\neg b \Rightarrow \mathsf{wp}(c_2, Q))
\end{array}
$$

Loops. There is no wp rule for while: we supply a loop invariant I instead. To prove

$$\{P\}\ \ c_0;\ \mathsf{while}\ b\ \mathsf{do}\ c\ \ \{Q\}$$

it suffices to show the three verification conditions

$$
\begin{array}{lll}
\text{VC1} & \text{(init)} & P \Rightarrow \mathsf{wp}(c_0,\ I) \\
\text{VC2} & \text{(preserve)} & I \wedge b \Rightarrow \mathsf{wp}(c,\ I) \\
\text{VC3} & \text{(exit)} & I \wedge \neg b \Rightarrow Q
\end{array}
$$