Programming Languages
0. Imperative Programming and Hoare Logic
Exercises

Shin-Cheng Mu

March, 2021

Guarded Command Language Basics

1. Which of the following Hoare triples hold?
   (a) $\{x = 7\} skip \{odd\ x\}$;
   (b) $\{x > 60\} x := x \times 2 \{x > 100\}$;
   (c) $\{x > 40\} x := x \times 2 \{x > 100\}$;
   (d) $\{true\}$**if** $x \leqslant y \rightarrow y := y - x \mid x \geqslant y \rightarrow x := x - y$ **fi**$\{x \geqslant 0 \wedge y \geqslant 0\}$;
   (e) $\{even\ x \wedge even\ y\}$**if** $x \leqslant y \rightarrow y := y - x \mid x \geqslant y \rightarrow x := x - y$ **fi**$\{even\ x \wedge even\ y\}$.

2. Is it always true that $\{true\}\ x := E\ \{x = E\}$? If you think the answer is yes, explain why. If your answer is no, give a counter example.

3. Verify:

$$
\begin{array}{l}
\{x = X \wedge y = Y\} \\
x := x \not\equiv y \\
y := x \not\equiv y \\
x := x \not\equiv y \\
\{x = Y \wedge y = X\}
\end{array}
$$

   where x and y are boolean and $(\not\equiv)$ is the "not equal" or "exclusive or" operator. In fact, the code above works for any $(\otimes)$ that satisfies the properties that for all a, b, and c:

$$
\begin{array}{ll}
\text{associative}: & a \otimes (b \otimes c) = (a \otimes b) \otimes c\ , \\
\text{unipotent}: & a \otimes a = 1\ ,
\end{array}
$$

   where 1 is the unit of $(\otimes)$, that is, $1 \otimes b = b = b \otimes 1$.

4. Verify the following program:

$$
\begin{array}{l}
\textbf{var}\ r, b : Int \\
\{0 \leqslant r < 2 \times b\} \\
\textbf{if}\ b \leqslant r \rightarrow r := r - b \\
\mid\ r < b \rightarrow skip \\
\textbf{fi} \\
\{0 \leqslant r < b\}
\end{array}
$$

5. Verify:

$$\begin{array}{l}
\textbf{var } x, y : \mathit{Int} \\
\{\mathit{True}\} \\
x, y := x \times x, y \times y \\
\textbf{if } x \geqslant y \rightarrow x := x - y \\
\mid\ y \geqslant x \rightarrow y := y - x \\
\textbf{fi} \\
\{x \geqslant 0 \wedge y \geqslant 0\}\ \ .
\end{array}$$

6. Verify:

$$\begin{array}{l}
\textbf{var } a, b : \mathit{Bool} \\
\{\mathit{True}\} \\
\textbf{if } \neg\, a \vee b \rightarrow a := \neg\, a \\
\mid\ a \vee \neg\, b \rightarrow b := \neg\, b \\
\textbf{fi} \\
\{a \vee b\}\ \ .
\end{array}$$

Weakest Precondition

7. Given below is a list of statements and predicates. What are the weakest precondition for the predicates to be true after the statement?
   (a) $x := x \times 2$, $x > 100$;
   (b) $x := x \times 2$, $\mathit{even}\ x$;
   (c) $x := x \times 2$, $x > 100 \wedge \mathit{even}\ x$;
   (d) $x := x \times 2$, $\mathit{odd}\ x$.
   (e) skip, $\mathit{odd}\ x$.

8. Prove that $(\mathit{wp}\ S\ Q_0 \vee \mathit{wp}\ S\ Q_1) \Rightarrow \mathit{wp}\ S\ (Q_0 \vee Q_1)$.

9. Recall the definition of Hoare triple in terms of *wp*:

$$\{P\}\, S\, \{Q\}\ =\ P \Rightarrow \mathit{wp}\ S\ Q\ \ .$$

   Prove that

   1. $(\{P\}\, S\, \{Q\} \wedge (P_0 \Rightarrow P)) \Rightarrow \{P_0\}\, S\, \{Q\}$.
   2. $\{P\}\, S\, \{Q\} \wedge \{P\}\, S\, \{R\} \equiv \{P\}\, S\, \{Q \wedge R\}$.

10. Recall the weakest precondition of **if**:

$$\mathit{wp}\ (\textbf{if } B_0 \rightarrow S_0 \vee B_1 \rightarrow S_1\ \textbf{fi})\ Q\ =\ (B_0 \Rightarrow \mathit{wp}\ S_0\ Q) \wedge (B_1 \Rightarrow \mathit{wp}\ S_1\ Q) \wedge (B_0 \vee B_1)\ \ .$$

    Prove that

$$\begin{array}{l}
\{P\}\, \textbf{if } B_0 \rightarrow S_0 \vee B_1 \rightarrow S_1\ \textbf{fi}\, \{Q\}\ \equiv \\
\quad \{P \wedge B_0\}\, S\, \{Q\}\ \wedge\ \{P \wedge B_1\}\, S\, \{Q\}\ \wedge\ (P \Rightarrow (B_0 \vee B_1))\ \ .
\end{array}$$

    Note: having proved so shows that the way we annotate **if** is correct:

$$\begin{array}{l}
\{P\} \\
\textbf{if } B_0 \rightarrow \{P \wedge B_0\}\, S_0\, \{Q\} \\
\mid\ B_1 \rightarrow \{P \wedge B_1\}\, S_1\, \{Q\} \\
\textbf{fi} \\
\{Q\}\ \ .
\end{array}$$

11. Recall that $wp\ S\ Q$ stands for "the weakest precondition for program S to terminate in a state satisfying Q". What programs S, if any, satisfy each of the following conditions?

    1. $wp\ S\ True = True$.
    2. $wp\ S\ True = False$.
    3. $wp\ S\ False = True$.
    4. $wp\ S\ False = False$.