Programming Languages and Computation

Week 8: Loop Invariants

** 1. For each of the following statements, determine whether $q \leq x$ is a loop invariant. If so, justify why. Otherwise, provide a counter-example.

(a) while $y * q \leq x$ do $q \leftarrow q + 1$

(b) while $!(q \leq x)$ do $x \leftarrow x - 1$

(c) while $!(x \leq q * q)$ do $q \leftarrow q + 1$

Solution

(a) $q \leq x$ is not a loop invariant. Consider any state where [] where all variables are mapped to 0 and the branch condition $y * q \leq x$ is satisfied. We have that $\langle q \leftarrow q + 1, [] \rangle \rightarrow [q \mapsto 1]$ and thus the candidate loop invariant is not preserved.

(b) $q \leq x$ is a loop invariant. For it to be a loop invariant, it must be the case that $\{q \leq x \;\&\&\; !(q \leq x)\}\ x \leftarrow x - 1\ \{q \leq x\}$. This Hoare triple is vacuously valid — it has an unsatisfiable pre-condition, and thus trivially holds. In other words, when $q \leq x$, the loop is never entered and so the condition is trivially preserved.

(c) $q \leq x$ is a loop invariant. We have that $\{q \leq x \;\&\&\; !(x \leq q * q)\}\ q \leftarrow q + 1\ \{\exists q'. q' \leq x \;\&\&\; !(x \leq q' * q') \;\&\&\; q = q' + 1\}$ from the assignment rule. This post-condition is equivalent to $q - 1 \leq x \;\&\&\; (q - 1) * (q - 1) < x$. Consider two cases based on the first conjunct. If $q - 1 < x$, then we immediately have $q \leq x$. On the other hand, if $q - 1 = x$, then we have that $(q - 1) * (q - 1) = x^2$. However, $x^2 < x$ leads to a contradiction, as we are only dealing with the integers, and thus this case is absurd. Therefore, we have $q \leq x$ as required.

** 2. Find a loop invariant that can be used to demonstrate the following Hoare triple. You should justify your solution in relation to the rules of Hoare logic or the strongest post-condition.

$\{0 \leq x \;\&\&\; 0 < y\}$
$q \leftarrow 0;$
$r \leftarrow x;$
while $y \leq r$ do
  $r \leftarrow r - y;$
  $q \leftarrow q + 1;$
$\{x = q * y + r \;\&\&\; r < y\}$

Solution

After applying the assignment rule twice, we can reduce this problem to the following triple:

$$\begin{array}{l}\{r = x \;\&\&\; q = 0 \;\&\&\; 0 \leq x \;\&\&\; 0 < y\}\\ \text{while } y \leq r \text{ do}\\ \quad r \leftarrow r - y;\\ \quad q \leftarrow q + 1;\\ \{x = q * y + r \;\&\&\; r < y\}\end{array}$$

An invariant for this loop is $I(x, y, q, r) := x = q * y + r$. To verify that it is a loop invariant, we need to check that $\{I \;\&\&\; y \leq r\}\ r \leftarrow r - y;\ q \leftarrow q + 1\ \{I\}$. The strongest post-condition is $\exists q' r'.\, I(x, y, q', r') \;\&\&\; r = r' - y \;\&\&\; q = q' + 1$, which is equivalent to $x = (q-1) * y + r + y$ or just $I(x, y, q, r)$. So it is indeed an invariant.

Additionally, we have that $r = x \;\&\&\; q = 0 \;\&\&\; 0 \leq x \;\&\&\; 0 < y$ implies $I(x, y, q, r)$; and that $I(x, y, q, r) \;\&\&\; !(y \leq r)$ implies $x = q * y + r \;\&\&\; r < y$ as required.

*** 3. Find a loop invariant that can be used to demonstrate the following Hoare triple. You should justify your solution in relation to the rules of Hoare logic or the strongest post-condition.

$$\begin{array}{l}\{n \geq 0\}\\ i \leftarrow 0\\ s \leftarrow 0\\ \text{while } !(i = n) \text{ do}\\ \quad s \leftarrow s + 2 * i + 1\\ \quad i \leftarrow i + 1\\ \{s = n * n\}\end{array}$$

Solution

After applying the assignment rule twice, we can reduce this problem to the following triple:

$$\begin{array}{l}\{n \geq 0 \;\&\&\; i = 0 \;\&\&\; s = 0\}\\ \text{while } !(i = n) \text{ do}\\ \quad s \leftarrow s + 2 * i + 1;\\ \quad i \leftarrow i + 1\\ \{s = n * n\}\end{array}$$

An invariant is $s = i^2$ (this can be hypothesized by trialling a few iterations, or observing that $(x + 1)^2 = x^2 + 2x + 1$). To verify this invariant, we need to consider whether the triple $\{s = i^2\}\ s \leftarrow s + 2 * i + 1;\ i \leftarrow i + 1\ \{s = n^2\}$. The strongest post-condition is $\exists s'\ i'.\, s' = i'^2 \;\&\&\; s = s' + 2i' + 1 \;\&\&\; i = i' + 1$, which is equivalent to $s = (i-1)^2 + 2(i-1) + 1$, i.e. $s = i^2$. So the invariant is preserved.

Finally, we need to check that $n \geq 0 \;\&\&\; i = 0 \;\&\&\; s = 0$ implies $s = i^2$, which is of course true, and likewise that $s = i * i \;\&\&\; i = n$ implies $s = n * n$, which is again trivial.

*** 4. Find a pair of loop invariants (one for each loop) that together can be used to demonstrate the following Hoare triple. You should justify your solution in relation to the rules of Hoare logic or the strongest post-condition.

$$
\begin{array}{l}
\{0 \leq n\} \\
i \leftarrow 0; \\
j \leftarrow 0; \\
\textsf{while } i \leq n-1 \textsf{ do} \\
\quad j \leftarrow 0; \\
\quad \textsf{while } j \leq i \textsf{ do} \\
\quad\quad j \leftarrow j+1; \\
\quad i \leftarrow i+1 \\
\{j = n\}
\end{array}
$$

Solution

For the inner loop, the invariant is $0 \leq j \leq i+1 \leq n$. We can verify this invariant by considering the strongest post-condition of the loop's body $\exists j'.\ \leq j' \leq i+1 \leq n$ && $j' \leq i$ && $j = j'+1$, which implies $0 \leq j \leq i+1 \leq n$ as required. We therefore have:

$$
\begin{array}{l}
\{0 \leq j \leq i+1 \leq n\} \\
\textsf{while } j \leq i \textsf{ do} \\
\quad j \leftarrow j+1 \\
\{0 \leq j \leq i+1 \leq n \ \&\& \ j > i\} \\
\{0 \leq j = i+1 \leq n\} \\
i \leftarrow i+1; \\
\{0 \leq j = i \leq n\}
\end{array}
$$

For the outer loop, the invariant is $0 \leq i \leq n$ && $j = i$. After the assignment statement $j \leftarrow 0$, we have $0 \leq i \leq n$ && $j = 0$ && $i \leq n-1$ which can be weakened to $0 \leq j \leq i+1 \leq n$ (where $i \leq n-1$ is due to the outer loop's branch condition). And, as we have shown, this is preserved by the loop's body.

Therefore, we have that:

$$
\begin{array}{l}
\{0 \leq n\} \\
i \leftarrow 0; \\
j \leftarrow 0; \\
\textsf{while } i \leq n-1 \textsf{ do} \\
\quad j \leftarrow 0; \\
\quad \textsf{while } j \leq i-1 \textsf{ do} \\
\quad\quad j \leftarrow j+1; \\
\quad i \leftarrow i+1 \\
\{0 \leq i \leq n \ \&\& \ j = i \ \&\& \ i \geq n\}
\end{array}
$$

which can be weakened to get the post-condition $j = n$ as required.

** 5. Find a loop invariant for the following program that demonstrates that this program has non-terminating traces. You should justify your answer in relation to the definition of a Hoare triple.

$$\textsf{while } x > 0 \textsf{ do } x \leftarrow x + y$$

Solution

A possible invariant is simply $y \leq 0$ && $x < 0$. This allows us to conclude that $\{y \leq 0\}\ S\ \{\bot\}$. By the definition of a Hoare triple, we have that if σ satisfies $y \leq 0$ and $\langle S, \sigma\rangle \rightarrow^* \sigma'$ then σ' satisfies $\bot$. In other words, in any state σ such that $\sigma(y) \leq 0$, there is no terminal state.