Program Verification
While Loops

Alice Gao
Lecture 20
Outline

Program Verification: While Loops
  Learning Goals
  Proving Partial Correctness - Example 1
  Proving Partial Correctness - Example 2
  Proving Termination
  Revisiting the Learning Goals
Learning Goals

By the end of this lecture, you should be able to:
Partial correctness for while loops
  ▶ Determine whether a given formula is an invariant for a while loop.
  ▶ Find an invariant for a given while loop.
  ▶ Prove that a Hoare triple is satisfied under partial correctness for a program containing while loops.
Total correctness for while loops
  ▶ Determine whether a given formula is a variant for a while loop.
  ▶ Find a variant for a given while loop.
  ▶ Prove that a Hoare triple is satisfied under total correctness for a program containing while loops.
Proving Total Correctness of While Loops

- Partial correctness
- Termination
Proving Partial Correctness of While Loops

\[
P \\
i \\
\text{while } (B) \{ \\
\quad (I \land B) \\
\quad C \\
\quad I <\text{justify based on C — a subproof}> \\
\} \\
\quad (I \land (\neg B)) \\
\quad Q \\
\text{implied (B)}
\]

Proof of implied (A): \((P \rightarrow I)\)
Proof of implied (B): \(((I \land (\neg B)) \rightarrow Q)\)

\(I\) is called a loop invariant. We need to determine \(I\)!
What is a loop invariant?

A loop invariant is:

- A relationship among the variables. (A predicate formula involving the variables.)
- The word “invariant” means something that does not change.
- It is true before the loop begins.
- It is true at the start of every iteration of the loop and at the end of every iteration of the loop.
- It is true after the loop ends.
Proving partial correctness of while loops

Indicate the places in the program where the loop invariant is true.

\[(x \geq 0)\]

\[y = 1;\]

\[z = 0;\]

\textbf{while} \ (z \neq x) \ \{ \]

\[\quad z = z + 1;\]

\[\quad y = y \ast z;\]

\}\n
\[(y = x!)\]
Proving partial correctness of a while loop

Steps to follow:

▶ Find a loop invariant.
▶ Complete the annotations.
▶ Prove any implied’s.

How do we find a loop invariant???
How do we find a loop invariant?

First, we need to understand the purpose of an invariant.

- The postcondition is the ultimate goal of our while loop.
- At every iteration, we are making progress towards the postcondition.
- The invariant is describing the progress we are making at every iteration.
Example 1:
Prove that the following triple is satisfied under partial correctness.

\[(x \geq 0)\]
y = 1;
z = 0;
while \((z \neq x)\) {
z = z + 1;
y = y \times z;
}
\((y = x!))
Finding a loop invariant

Step 1: Write down the values of all the variables every time the while test is reached.

\( (x \geq 0) \)
\begin{align*}
y &= 1; \\
z &= 0; \\
\text{while } (z \neq x) \{ \\
&\quad z = z + 1; \\
&\quad y = y \times z; \\
\} \\
(y = x!)
\end{align*}
Finding a loop invariant

Step 2: Find relationships among the variables that are true for every while test. These are our candidate invariants. 
Come up with some invariants in the next 2 minutes.

\[(x \geq 0)\]
y = 1;
z = 0;
while (z != x) {
z = z + 1;
y = y * z;
}
(y = x!)

<table>
<thead>
<tr>
<th>x</th>
<th>z</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>1 = 0!</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1 = 1!</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2 = 2!</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6 = 3!</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>24 = 4!</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>120 = 5!</td>
</tr>
</tbody>
</table>
CQ 1: Is \((\neg (z = x))\) a loop invariant?

(A) Yes (B) No (C) I don’t know...

\[(x \geq 0)\]
\[
y = 1; \\
z = 0; \\
\textbf{while} \ (z \neq x) \ {\{ \\
\quad z = z + 1; \\
\quad y = y \ast z; \\
\}} \\
(y = x!)
\]

<table>
<thead>
<tr>
<th>x</th>
<th>z</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>1 = 0!</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1 = 1!</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2 = 2!</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6 = 3!</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>24 = 4!</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>120 = 5!</td>
</tr>
</tbody>
</table>
CQ 2: Is \((z \leq x)\) a loop invariant?

(A) Yes (B) No (C) I don’t know...

\[
\begin{align*}
(x \geq 0) \\
y &= 1; \\
z &= 0; \\
\textbf{while} \ (z \neq x) \ {\} \quad &\begin{array}{ccc}
\hline
x & z & y \\
5 & 0 & 1 = 0! \\
5 & 1 & 1 = 1! \\
5 & 2 & 2 = 2! \\
5 & 3 & 6 = 3! \\
5 & 4 & 24 = 4! \\
5 & 5 & 120 = 5! \\
\hline
\end{array}
\end{align*}
\]
CQ 3: Is \((y = z!)\) a loop invariant?

(A) Yes (B) No (C) I don’t know...

\[(x \geq 0)\]
\(y = 1;\)
\(z = 0;\)
\textbf{while} \ (z != x) \ \{\]
\hspace{1em} \(z = z + 1;\)
\hspace{1em} \(y = y \times z;\)
\}
\((y = x!);\)

\begin{tabular}{c|c|c}
\hline
\(x\) & \(z\) & \(y\) \\
\hline
5 & 0 & 1 = 0! \\
5 & 1 & 1 = 1! \\
5 & 2 & 2 = 2! \\
5 & 3 & 6 = 3! \\
5 & 4 & 24 = 4! \\
5 & 5 & 120 = 5! \\
\hline
\end{tabular}
CQ 4: Is \((y = x!)\) a loop invariant?

(A) Yes  (B) No  (C) I don’t know...

\[(x \geq 0)\]
\[y = 1;\]
\[z = 0;\]
\[\textbf{while} \ (z \neq x) \ {\{ \}
\[z = z + 1;\]
\[y = y \ast z;\]
\}\]
\[(y = x!)\]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>z</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0!</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1!</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2!</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6</td>
<td>3!</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>24</td>
<td>4!</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>120</td>
<td>5!</td>
</tr>
</tbody>
</table>
CQ 5: Is \(((z \leq x) \land (y = z!))\) a loop invariant?

(A) Yes (B) No (C) I don’t know...

\[(x \geq 0)\]
\[y = 1;\]
\[z = 0;\]
\[\textbf{while} \; (z \neq x) \{\]
\[z = z + 1;\]
\[y = y \times z;\]
\}\n
\[(y = x!)\]
Finding a loop invariant

Step 3: Try each candidate invariant until we find one that works for our proof.

\( (x \geq 0) \)

\[
\begin{align*}
y &= 1; \\
z &= 0; \\
\text{while } (z \neq x) \{ \\
&\quad z = z + 1; \\
&\quad y = y \times z; \\
\} \\
(y = x!)
\end{align*}
\]

<table>
<thead>
<tr>
<th>x</th>
<th>z</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>1 = 0!</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1 = 1!</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2 = 2!</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6 = 3!</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>24 = 4!</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>120 = 5!</td>
</tr>
</tbody>
</table>
How do we find an invariant?

A recap of the steps to find an invariant:

- Write down the values of all the variables every time the while test is reached.
- Find relationships among the variables that are true for every while test. These are our candidate invariants.
- Try each candidate invariant until we find one that works for our proof.
Partial While - Example 1 \(((z \leq x) \text{ as the invariant})\)

\[
x \geq 0
\]
\[
0 \leq x
\]
\[
y = 1;
\]
\[
0 \leq x
\]
\[
z = 0;
\]
\[
0 \leq x
\]
\[
\text{while } (z \neq x) \{
\]
\[
((z \leq x) \land (\neg(z = x)))
\]
\[
(z + 1 \leq x)
\]
\[
z = z + 1;
\]
\[
(z \leq x)
\]
\[
y = y \times z;
\]
\[
(z \leq x)
\]
\[
}\n\]
\[
((z \leq x) \land (\neg(\neg(z = x))))
\]
\[
y = x!
\]

implied (A)

assignment

assignment

partial –while

implied (B)

assignment

assignment

partial –while

implied (C)
We used \((z \leq x)\) as the invariant.

\textbf{CQ 7:} Is there a proof for implied (A)?

\[ ((x \geq 0) \rightarrow (0 \leq x)) \]

\begin{itemize}
  \item [(A)] Yes
  \item [(B)] No
  \item [(C)] I don't know.
\end{itemize}
CQ 8: Is there a proof for implied (B)?

We used \((z \leq x)\) as the invariant.

\textbf{CQ 8:} Is there a proof for implied (B)?

\[
((z \leq x) \land \neg(z = x)) \rightarrow (z + 1 \leq x)
\]

\begin{itemize}
  \item[(A)] Yes
  \item[(B)] No
  \item[(C)] I don’t know.
\end{itemize}
CQ 9 Is there a proof for implied (C)?

We used \((z \leq x)\) as the invariant.

CQ 9: Is there a proof for implied (C)?

\[
((z \leq x) \land (\neg (\neg (z = x)))) \rightarrow (y = x!))
\]

(A) Yes
(B) No
(C) I don’t know.
Example 2:
Prove that the following triple is satisfied under partial correctness.

\[(x \geq 0)\]
\[y = 1;\]
\[z = 0;\]
\[\text{while } (z < x) \{\]
\[\quad z = z + 1;\]
\[\quad y = y \times z;\]
\[\}
\[(y = x!)\]

Let’s try using \((y = z!)) as the invariant in our proof.
Which invariant leads to a valid proof?

To check whether an invariant leads to a valid proof, we need to check whether all of the implied’s can be proved.
CQ 11 Is there a proof for implied (A)?

We used \((y = z!)\) as the invariant.

CQ 11: Is there a proof for implied (A)?

\[((x \geq 0) \rightarrow (1 = 0!))\]

(A) Yes
(B) No
(C) I don’t know.
CQ 12: Is there a proof for implied (B)?

We used \((y = z!)\) as the invariant.

CQ 12: Is there a proof for implied (B)?

\[ (((y = z!) \land (z < x)) \rightarrow (y \ast (z + 1) = (z + 1)!)) \]

(A) Yes
(B) No
(C) I don’t know.
CQ 13 Is there a proof for implied (C)?

We used \((y = z!\) as the invariant.

CQ 13: Is there a proof for implied (C)?

\[
(((y = z!) \land (\neg (z < x))) \rightarrow (y = x!))
\]

(A) Yes

(B) No

(C) I don't know.
We used \(((y = z!) \land (z \leq x))\) as the invariant.

**CQ 14: Is there a proof for implied (C)?**

\[((((y = z!) \land (z \leq x)) \land (\neg(z < x))) \rightarrow (y = x!))\]

(A) Yes  
(B) No  
(C) I don’t know.
Find an integer expression that

- is non-negative before the loop starts, at every iteration of the loop, and after the loop ends.
- decreases by at least 1 at every iteration of the loop.

This integer expression is called a **variant** (something that changes).

The loop must terminate because a non-negative integer can decrease by 1 a finite number of times.
Example 2: Finding a variant

Prove that the following program terminates.

\[
\begin{align*}
y &= 1; \\
z &= 0; \\
\textbf{while } (z < x) \{ \\
    &\quad z = z + 1; \\
    &\quad y = y \times z;
\}
\end{align*}
\]

How do we find a variant? The loop guard \((z < x)\) helps.
Example 2: Proof of Termination

Consider the variant \((x - z)\).

Before the loop starts, \((x - z) \geq 0\) because the precondition is \((x \geq 0)\) and the second assignment mutates \(z\) to be 0.

During every iteration of the loop, \((x - z)\) decreases by 1 because \(x\) does not change and \(z\) increases by 1.

Thus, \(x - z\) will eventually reach 0.

When \(x - z = 0\), the loop guard \(z < x\) will terminate the loop.
Revisiting the learning goals

By the end of this lecture, you should be able to:

Partial correctness for while loops

▶ Determine whether a given formula is an invariant for a while loop.
▶ Find an invariant for a given while loop.
▶ Prove that a Hoare triple is satisfied under partial correctness for a program containing while loops.

Total correctness for while loops

▶ Determine whether a given formula is a variant for a while loop.
▶ Find a variant for a given while loop.
▶ Prove that a Hoare triple is satisfied under total correctness for a program containing while loops.