Program Verification:
Assignments and Conditionals

HE Mingxin, Max

CS104: program07 @ yeah.net

CS108: mxhe1 @ yeah.net

I2ML(H) Spring 2023 (CS104|CS108)

Outline

Key Points

Key points to learn:

- Prove that a Hoare triple is satisfied under partial/total correctness for a program containing assignment and conditional statements.

Proving partial and total correctness

- Both problems are undecidable. No algorithm can solve them in all situations.
- Different techniques for proving partial and total correctness.
- For proving partial correctness, we will construct formal proofs using inference rules.
- For proving total correctness, we will prove partial correctness and termination separately.

Proving partial correctness

```
(| precondition |)
(| ... |)                   <justification>
y = 1;
(| ... |)                   <justification>
z = 0;
(| ... |)                   <justification>
while (z != x) {
  (| ... |)                 <justification>
  (| ... |)                 <justification>
  z = z + 1;
  (| ... |)                 <justification>
  y = y * z;
  (| ... |)                 <justification>
}
(| postcondition |)         <justification>
```

The assignment inference rule

$$\begin{array}{ll} (\!|\, Q[E/x] \,|\!) & \\ \mathtt{x\ =\ E;} & \\ (\!|\, Q \,|\!) & \quad\quad\quad \mathtt{assignment} \end{array}$$

- Q is a predicate formula.
- x is a variable in Q.
- E is a term.

An example of using the assignment inference rule

Example:

$(\!|\ ???\ |\!)$
```
x = 2;
```
$(\!|\ x = 2\ |\!)$ assignment

CQ 1 The assignment inference rule

$(\!|P|\!)$
```
x = 2;
```
$(\!|x = y|\!)$ assignment

Which of the following is the precondition P derived using the assignment inference rule?

(A) $x = 2$

(B) $y = 2$

(C) None of the above

CQ 2 The assignment inference rule

$(\!|P|\!)$

```
x = x + 1;
```

$(\!| x+1 = n+1 |\!)$ assignment

Which of the following is the precondition P derived using the assignment inference rule?

(A) $x+1=n$

(B) $x=n+1$

(C) None of the above

CQ 3 The assignment inference rule

$(\!| P |\!)$
`x = y;`
$(\!| (\exists k \; (x = y * k)) |\!)$ assignment

Which of the following is the precondition P derived using the assignment inference rule?

(A) $(\exists k \; (x = y * k))$

(B) $(\exists k \; (y = y * k))$

(C) $(\exists k \; (x = x * k))$

(D) None of the above

Notes on the assignment inference rule

- For assignments, we work bottom-up from the postcondition. Sometimes, we call this pushing up the assignments.
- Treat E as one expression and do not worry about what's inside.
- If there is an equality in $Q[E/x]$, do not switch the two sides of the inequality.
- Do not simplify $Q[E/x]$ in any way.

Exercise on the assignment inference rule

Show that the following triple is satisfied under partial correctness.

$(\!| \, y = 6 \, |\!)$
`x = y + 1;`
$(\!| \, x = 7 \, |\!)$

The if-then-else inference rule

```
(| P |)
if ( B ) {
    (| (P ∧ B) |)              if-then-else
    C1
    (| Q |)   <justify based on C1 - a subproof>
} else {
    (| (P ∧ (¬B)) |)           if-then-else
    C2
    (| Q |)   <justify based on C2 - a subproof>
}
(| Q |)                        if-then-else
```

Example of if-then-else

Show that the following triple is satisfied under partial correctness.

$\llparenthesis\, true \,\rrparenthesis$

```
if ( x > y ) {

    max = x;

} else {

    max = y;

}
```

$\llparenthesis\, (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y))) \,\rrparenthesis$

Notes on the if-then-else inference rule

1. Move the precondition into the if and else blocks, adding on the corresponding if/else condition.
2. Copy the post condition into the last lines of the if and else blocks.
3. Complete the annotations for the two subproofs, one for the if block, and one for the else block.
4. Prove any implied's.

CQ 4 The if-then-else inference rule

$(\!|(x = 3)|\!)$

```
if (x > 0) {
  x = 0;
} else {
  x = −1;
  (|φ|)
}
```

$(\!|(x \leq 0)|\!)$

Which of the following is the correct formula for φ based on the if-then-else inference rule?

(A) $((x = 3) \land (x > 0))$

(B) $(x = 0)$

(C) $((x = 3) \land (\neg(x > 0)))$

(D) $(x = -1)$

(E) $(x \leq 0)$

CQ 5 The if-then-else inference rule

$(\!|\, (x = 3) \,|\!)$
```
if (x > 0) {
   x = 0;
   (| (x ≤ 0) |)
} else {
   (| φ |)
   x = −1;
   (| (x ≤ 0) |)
}
(| (x ≤ 0) |)
```

Let $(\!|\, \varphi \,|\!)$ be the annotation immediately below "else". Which of the following is the correct formula for φ based on the if-then-else inference rule?

(A) $((x = 3) \wedge (x > 0))$

(B) $(x = 0)$

(C) $((x = 3) \wedge (\neg(x > 0)))$

(D) $(x = -1)$

(E) $(x \leq 0)$

CQ 6 The if-then-else inference rule

$$
\begin{array}{ll}
(\!|\,(x = 3)\,|\!) & \\
\texttt{if (x > 0) \{} & \\
\quad (\!|\,((x = 3) \wedge (x > 0))\,|\!) & \textbf{if}\texttt{-then-}\textbf{else} \\
\quad \texttt{x = 1;} & \\
\quad (\!|\,(x \geq 0)\,|\!) & \texttt{assignment} \\
\texttt{\} }\textbf{else}\texttt{ \{} & \\
\quad (\!|\,((x = 3) \wedge (\neg(x > 0)))\,|\!) & \textbf{if}\texttt{-then-}\textbf{else} \\
\quad \texttt{x = 0;} & \\
\quad (\!|\,(x \geq 0)\,|\!) & \texttt{assignment} \\
\texttt{\}} & \\
(\!|\,(x \geq 0)\,|\!) & \textbf{if}\texttt{-then-}\textbf{else}
\end{array}
$$

Are we done with annotating this program?

(A) Yes

(B) No

(C) I'm not sure.

The if-then inference rule

$(\!| P |\!)$

if (B) {

  $(\!| (P \land B) |\!)$ **if**-then

  C1

  $(\!| Q |\!)$ <justify based on C1 - a subproof>

}

$(\!| Q |\!)$ **if**-then

  implied $((P \land (\neg B)) \rightarrow Q)$

Example of if-then

Show that the following triple is satisfied under partial correctness.

$(\!|\, true \,|\!)$

```
if ( max < x ) {

    max = x;

}
```

$(\!|\, max \geq x \,|\!)$

Notes on the if-then inference rule

1. Move the precondition into the if block, adding on the if condition.
2. Copy the post condition into the last line of the if block.
3. Complete the annotations for the subproof in the if block.
4. Write down the implied condition for the implicit "else" block.
5. Prove any implied's.

Summary

Key points learnt:

- Prove that a Hoare triple is satisfied under partial/total correctness for a program containing assignment and conditional statements.