Warm-Up Problem

1. What is the definition of a Hoare triple satisfying partial correctness?
2. Recall the rule for assignment:

$$\frac{}{(\!|\, Q[E/x] \,|\!)\ \ (\mathtt{x} = E)\ \ (\!|\, Q \,|\!)}\ \text{(assignment)}$$

   Why is this the correct rule and not the following rule?

$$\frac{}{(\!|\, Q \,|\!)\ \ (\mathtt{x} = E)\ \ (\!|\, Q[E/x] \,|\!)}\ \text{(assignment)}$$

1. What is the definition of a Hoare triple satisfying partial correctness?
2. Recall the rule for assignment:

$$\frac{}{(\!|\, Q[E/x] \,|\!)\ \ (\mathtt{x} = E)\ \ (\!|\, Q \,|\!)}\ \text{(assignment)}$$

   Why is this the correct rule and not the following rule?

$$\frac{}{(\!|\, Q \,|\!)\ \ (\mathtt{x} = E)\ \ (\!|\, Q[E/x] \,|\!)}\ \text{(assignment)}$$

   Solution: With the second rule, triples like
   $(\!|\, x = 3 \,|\!)$ `x=2` $(\!|\, 2 = 3 \,|\!)$ are provable which would allow us to prove false statements.

Program Verification
Conditionals

Carmen Bruni

Lecture 19

Based on slides by Jonathan Buss, Lila Kari, Anna Lubiw and Steve Wolfman with thanks to B. Bonakdarpour, A. Gao, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek

Last Time

- Give reasons for performing formal verification vs testing.
- Define a Hoare Triple.
- Define Partial Correctness.
- Prove that a Hoare triple is satisfied under partial correctness for a program containing assignment statements.

Learning Goals

- Understand and use implied statements as needed.
- Prove that a Hoare triple is satisfied under partial correctness for a program containing assignment and conditional statements.

How do we show that the following Hoare triple is satisfied under partial correctness?

$\llparenthesis\, (y = 6) \,\rrparenthesis$

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

$\llparenthesis\, (x = 7) \,\rrparenthesis$

How do we show that the following Hoare triple is satisfied under partial correctness?

$\llparenthesis\, (y = 6) \,\rrparenthesis$

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

$\llparenthesis\, (x = 7) \,\rrparenthesis$

Assignment here only gives us that

$\llparenthesis\, ((y + 1) = 7) \,\rrparenthesis$

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

$\llparenthesis\, (x = 7) \,\rrparenthesis$

However, we know that $((y + 1) = 7)$ implies that $(y = 6)$. How do we write this?

Inference Rules with Implications

Rule of "Precondition strengthening":

$$\frac{(P \to P') \qquad (\!|\, P' \,|\!)\ C\ (\!|\, Q \,|\!)}{(\!|\, P \,|\!)\ C\ (\!|\, Q \,|\!)} \ \textit{(implied)}$$

Rule of "Postcondition weakening":

$$\frac{(\!|\, P \,|\!)\ C\ (\!|\, Q' \,|\!) \qquad (Q' \to Q)}{(\!|\, P \,|\!)\ C\ (\!|\, Q \,|\!)} \ \textit{(implied)}$$

Example of use:

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

More Questions

What if my code has multiple lines? Why are we allowed to correctly compose annotations?

Inference Rule for Sequences of Instructions

$$\frac{(\!|\, P \,|\!)\ C_1\ (\!|\, Q \,|\!), \quad (\!|\, Q \,|\!)\ C_2\ (\!|\, R \,|\!)}{(\!|\, P \,|\!)\ C_1; C_2\ (\!|\, R \,|\!)}\ \textit{(composition)}$$

In order to prove $(\!|\, P \,|\!)\ C_1;\ C_2\ (\!|\, R \,|\!)$, we need to find a midcondition Q for which we can prove $(\!|\, P \,|\!)\ C_1\ (\!|\, Q \,|\!)$ and $(\!|\, Q \,|\!)\ C_2\ (\!|\, R \,|\!)$.

(In our examples, the midcondition will usually be determined by a rule, such as assignment. But in general, a midcondition might be very difficult to determine.)

Proof Format: Annotated Programs

Interleave program statements with assertions, each justified by an inference rule.

The composition rule is implicit.

Assertions should hold true whenever the program reaches that point in its execution.

Proof Format: Annotated Programs

If implied inference rule is used, we must supply a proof of the implication.

- We'll do these proofs after annotating the program.

Each assertion should be an instance of an inference rule.
Normally,

- Don't simplify the assertions in the annotated program.
- Do the simplification while proving the implied conditions.

Example: Composition of Assignments

To show: the following is satisfied under partial correctness.

We work bottom-up for assignments...

$(\!|\ ((x = x_0) \land (y = y_0))\ |\!)$

```
t = x ;

x = y ;

y = t ;
```

$(\!|\ ((x = y_0) \land (y = x_0))\ |\!)$

To show: the following is satisfied under partial correctness.

We work bottom-up for assignments...

$$(\!|\ ((x = x_0) \land (y = y_0))\ |\!)$$

```
t = x ;

x = y ;
```

$(\!|\ ((x = y_0) \land (t = x_0))\ |\!)$ P_2 is $(\!|\ P[t/y]\ |\!)$

```
y = t ;
```

$(\!|\ ((x = y_0) \land (y = x_0))\ |\!)$ assignment $(\!|\ P\ |\!)$

Example: Composition of Assignments

To show: the following is satisfied under partial correctness.

We work bottom-up for assignments...

$$(\!|\, ((x = x_0) \wedge (y = y_0)) \,|\!)$$

```
t = x ;
```

$(\!|\, ((y = y_0) \wedge (t = x_0)) \,|\!)$ $\quad$ P_3 is $(\!|\, P_2[y/x] \,|\!)$

```
x = y ;
```

$(\!|\, ((x = y_0) \wedge (t = x_0)) \,|\!)$ $\quad$ assignment

```
y = t ;
```

$(\!|\, ((x = y_0) \wedge (y = x_0)) \,|\!)$ $\quad$ assignment

Example: Composition of Assignments

To show: the following is satisfied under partial correctness.

We work bottom-up for assignments...

$(\!\|\ ((x = x_0) \wedge (y = y_0))\ \|\!)$	
$(\!\|\ ((y = y_0) \wedge (x = x_0))\ \|\!)$	$(\!\|\ P_3[x/t]\ \|\!)$
`t = x ;`	
$(\!\|\ ((y = y_0) \wedge (t = x_0))\ \|\!)$	assignment
`x = y ;`	
$(\!\|\ ((x = y_0) \wedge (t = x_0))\ \|\!)$	assignment
`y = t ;`	
$(\!\|\ ((x = y_0) \wedge (y = x_0))\ \|\!)$	assignment

To show: the following is satisfied under partial correctness.

We work bottom-up for assignments...

$(\!| ((x = x_0) \wedge (y = y_0)) |\!)$	
$(\!| ((y = y_0) \wedge (x = x_0)) |\!)$	implied *[proof required]*
`t = x ;`	
$(\!| ((y = y_0) \wedge (t = x_0)) |\!)$	assignment
`x = y ;`	
$(\!| ((x = y_0) \wedge (t = x_0)) |\!)$	assignment
`y = t ;`	
$(\!| ((x = y_0) \wedge (y = x_0)) |\!)$	assignment

Finally, show $(\!| ((x = x_0) \wedge (y = y_0)) |\!)$ implies $(\!| ((y = y_0) \wedge (x = x_0)) |\!)$.

Example

Show that the following Hoare triple is satisfied under partial correctness.

$\llparenthesis\, ((((2 \cdot x) + (3 \cdot y)) \geq 7) \land ((x + (2 \cdot y)) \geq 0))\, \rrparenthesis$

```
x = x + y;
y = x + y;
x = x + y;
```

$\llparenthesis\, ((x \geq 5) \land (y \geq 0))\, \rrparenthesis$

Show that the following Hoare triple is satisfied under partial correctness.

$\llparenthesis\ ((((2 \cdot x) + (3 \cdot y)) \geq 7) \wedge ((x + (2 \cdot y)) \geq 0))\ \rrparenthesis$

```
x = x + y;
y = x + y;
x = x + y;
```

$\llparenthesis\ ((x \geq 5) \wedge (y \geq 0))\ \rrparenthesis$

Why is it that $x = 1$ and $y = 2$ satisfies the pre-condition but not the post condition and yet the above Hoare triple is satisfied under partial correctness?

Example

Show that the following Hoare triple is satisfied under partial correctness.

$\llparenthesis\ ((((2 \cdot x) + (3 \cdot y)) \geq 7) \wedge ((x + (2 \cdot y)) \geq 0))\ \rrparenthesis$

```
x = x + y;
y = x + y;
x = x + y;
```

$\llparenthesis\ ((x \geq 5) \wedge (y \geq 0))\ \rrparenthesis$

Why is it that $x = 1$ and $y = 2$ satisfies the pre-condition but not the post condition and yet the above Hoare triple is satisfied under partial correctness?

Answer: The code will actually mutate x and y! The x and y at the end is not the same as the original.

Programs with Conditional Statements

Deduction Rules for Conditionals

`if-then-else`:

$$\frac{(\!|\, (P \wedge B) \,|\!)\ \ C_1\ \ (\!|\, Q \,|\!) \qquad (\!|\, (P \wedge (\neg B)) \,|\!)\ \ C_2\ \ (\!|\, Q \,|\!)}{(\!|\, P \,|\!)\ \ \texttt{if (B) } C_1 \texttt{ else } C_2\ \ (\!|\, Q \,|\!)}\ \text{(if-then-else)}$$

`if-then` (without `else`):

$$\frac{(\!|\, (P \wedge B) \,|\!)\ \ C\ \ (\!|\, Q \,|\!) \qquad ((P \wedge (\neg B)) \to Q)}{(\!|\, P \,|\!)\ \ \texttt{if (B) } C\ \ (\!|\, Q \,|\!)}\ \text{(if-then)}$$

Annotated program template for `if-then-else`:

$(\!|\ P\ |\!)$

`if (` B `) {`

    $(\!|\ (P \wedge B)\ |\!)$ if-then-else

    C_1

    $(\!|\ Q\ |\!)$ *(justify depending on* C_1*—a "subproof")*

`} else {`

    $(\!|\ (P \wedge (\neg B))\ |\!)$ if-then-else

    C_2

    $(\!|\ Q\ |\!)$ *(justify depending on* C_2*—a "subproof")*

`}`

$(\!|\ Q\ |\!)$ if-then-else [*justifies this* Q*, given previous two*]

Template for Conditionals Without `else`

Annotated program template for `if-then`:

$$
\begin{array}{ll}
(\!|\ P\ |\!) & \\
\texttt{if (} B \texttt{) \{} & \\
\quad (\!|\ (P \wedge B)\ |\!) & \text{if-then} \\
\quad C & \\
\quad (\!|\ Q\ |\!) & [\textit{add justification based on } C] \\
\texttt{\}} & \\
(\!|\ Q\ |\!) & \text{if-then} \\
 & \text{Implied: Proof of } ((P \wedge (\neg B)) \rightarrow Q)
\end{array}
$$

Example: Prove the following is satisfied under partial correctness.

$(\!\| \textit{true} \|\!)$	$(\!\| P \|\!)$
`if ( max < x ) {`	`if (` B `) {`
`    max = x ;`	C
`}`	`}`
$(\!\| (\textit{max} \geq x) \|\!)$	$(\!\| Q \|\!)$

First, let's recall our proof method....

The Steps of Creating a Proof

Three steps in doing a proof of partial correctness:

1. First annotate using the appropriate inference rules.
2. Then "back up" in the proof: add an assertion before each assignment statement, based on the assertion following the assignment.
3. Finally prove any "implieds":
   - Annotations from (1) above containing implications
   - Adjacent assertions created in step (2).

   Proofs here are written in Math 135 style. They can use Predicate Logic, basic arithmetic, or any other appropriate reasoning.

1. Add annotations for the `if-then` statement.

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

```
if ( max < x ) {
```

$(\!|\ (\mathit{true} \wedge (\mathit{max} < x))\ |\!)$ if-then

```
    max = x ;
```

$(\!|\ (\mathit{max} \geq x)\ |\!)$ $\longleftarrow$ *to be shown*

```
}
```

$(\!|\ (\mathit{max} \geq x)\ |\!)$ if-then

Implied: $\Big(\big(\mathrm{true} \wedge \neg((\mathit{max} < x))\big) \rightarrow (\mathit{max} \geq x)\Big)$

1. Add annotations for the `if-then` statement.
2. "Push up" for the assignments.

$(\!|\ \text{true}\ |\!)$

```
if ( max < x ) {
```

$(\!|\ (\textit{true} \wedge (\textit{max} < x))\ |\!)$ if-then

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

```
    max = x ;
```

$(\!|\ (\textit{max} \geq x)\ |\!)$ assignment

```
}
```

$(\!|\ (\textit{max} \geq x)\ |\!)$ if-then

Implied: $\Big(\big(\text{true} \wedge \neg((\textit{max} < x))\big) \to (\textit{max} \geq x)\Big)$

1. Add annotations for the `if-then` statement.
2. "Push up" for the assignments.
3. Identify "implieds" to be proven.

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

```
if ( max < x ) {
```

$(\!|\ (\mathit{true} \wedge (\mathit{max} < x))\ |\!)$ if-then

$(\!|\ (x \geq x)\ |\!)$ Implied (a)

```
    max = x ;
```

$(\!|\ (\mathit{max} \geq x)\ |\!)$ assignment

```
}
```

$(\!|\ (\mathit{max} \geq x)\ |\!)$ if-then

Implied (b): $\Big(\big(\text{true} \wedge \neg((\mathit{max} < x))\big) \rightarrow (\mathit{max} \geq x)\Big)$

Proving "Implied" Conditions

The auxiliary "implied" proofs can be done in Math 135 style (and assuming the necessary arithmetic properties). We will write them informally, but clearly.

Proof of Implied (a):

$$\emptyset \vdash \big((\textit{true} \wedge (\textit{max} < x)) \rightarrow (x \geq x)\big).$$

- The statement $(x \geq x)$ is a tautology since $\geq$ is reflexive and so the required implication holds.

Implied (b)

Proof of Implied (b): Show $\emptyset \vdash \big((P \wedge (\neg B)) \rightarrow Q\big)$, which in this case is

$$\emptyset \vdash \Big(\big(\textit{true} \wedge (\neg(\textit{max} < x))\big) \rightarrow (\textit{max} \geq x)\Big).$$

- The hypothesis, $\big(\textit{true} \wedge (\neg(\textit{max} < x))\big)$ can be simplified to $(\neg(\textit{max} < x))$.
- Then by properties of $\neg$, $<$ and $\geq$, the conclusion, $(\textit{max} \geq x)$, follows.

Example 2 for Conditionals

Prove the following is satisfied under partial correctness.

$(\!|\ \mathsf{true}\ |\!)$

```
if ( x > y ) {
    max = x;
} else {
    max = y;
}
```

$(\!|\ \left(((x > y) \wedge (\mathit{max} = x)) \vee ((x \leq y) \wedge (\mathit{max} = y))\right)\ |\!)$

$(\!|\, \textit{true} \,|\!)$

```
if ( x > y ) {
```

$(\!|\, (x > y) \,|\!)$ — if-then-else

```
    max = x ;
```

$(\!|\, (((x > y) \wedge (\textit{max} = x)) \vee ((x \leq y) \wedge (\textit{max} = y))) \,|\!)$

```
} else {
```

$(\!|\, (\neg(x > y)) \,|\!)$ — if-then-else

```
    max = y ;
```

$(\!|\, (((x > y) \wedge (\textit{max} = x)) \vee ((x \leq y) \wedge (\textit{max} = y))) \,|\!)$

```
}
```

$(\!|\, (((x > y) \wedge (\textit{max} = x)) \vee ((x \leq y) \wedge (\textit{max} = y))) \,|\!)$ — if-then-else

Example 2: Annotated Code

$(\!|\ \mathit{true}\ |\!)$

```
if ( x > y ) {
```

$(\!|\ (x > y)\ |\!)$ if-then-else

$(\!|\ (((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y)))\ |\!)$

```
    max = x ;
```

$(\!|\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \leq y) \wedge (\mathit{max} = y)))\ |\!)$ assignment

```
} else {
```

$(\!|\ (\neg(x > y))\ |\!)$ if-then-else

$(\!|\ (((x > y) \wedge (y = x)) \vee ((x \leq y) \wedge (y = y)))\ |\!)$

```
    max = y ;
```

$(\!|\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \leq y) \wedge (\mathit{max} = y)))\ |\!)$ assignment

```
}
```

$(\!|\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \leq y) \wedge (\mathit{max} = y)))\ |\!)$ if-then-else

Example 2: Annotated Code

$(\!\vert\ \mathit{true}\ \vert\!)$	
`if ( x > y ) {`	
$\quad(\!\vert\ (x > y)\ \vert\!)$	if-then-else
$\quad(\!\vert\ (((x > y) \wedge (x = x)) \vee ((x \le y) \wedge (x = y)))\ \vert\!)$	implied (a)
`    max = x ;`	
$\quad(\!\vert\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \le y) \wedge (\mathit{max} = y)))\ \vert\!)$	assignment
`} else {`	
$\quad(\!\vert\ (\neg(x > y))\ \vert\!)$	if-then-else
$\quad(\!\vert\ (((x > y) \wedge (y = x)) \vee ((x \le y) \wedge (y = y)))\ \vert\!)$	implied (b)
`    max = y ;`	
$\quad(\!\vert\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \le y) \wedge (\mathit{max} = y)))\ \vert\!)$	assignment
`}`	
$(\!\vert\ (((x > y) \wedge (\mathit{max} = x)) \vee ((x \le y) \wedge (\mathit{max} = y)))\ \vert\!)$	if-then-else

Example 2: Implied Conditions

(a) Prove $\Big((x > y) \rightarrow \Big(((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y))\Big)\Big)$.

- From $(x > y)$ and from the trivially true $(x = x)$, it follows that $((x > y) \wedge (x = x))$.
- From this it follows that $\Big(((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y))\Big)$.

(b) Prove $\Big((x \leq y) \rightarrow \Big(((x > y) \wedge (y = x)) \vee ((x \leq y) \wedge (y = y))\Big)\Big)$.

- From $(x \leq y)$ and from the trivially true $(y = y)$, it follows that $((x > y) \wedge (y = x))$.
- From this it follows that $\Big(((x > y) \wedge (y = x)) \vee ((x \leq y) \wedge (y = y))\Big)$.

Try Some on Your Own!

Show that the following two Hoare triples are satisfied under partial correctness:

$(\!|\, \mathrm{true} \,|\!)$

```
x = y ;
if (y > 1){
    x = x - 1;
} else {
    x = x * x + 1;
}
```

$(\!|\, (x > 0) \,|\!)$

$(\!|\, (x \geq -7) \,|\!)$

```
w = x + 1 ;
if (w >= 0){
    w = (w+1)*(w+1)+ 1;
}
```

$(\!|\, \left(\neg(\exists u\, ((u \cdot u) = w))\right) \,|\!)$

Is the left hand Hoare triple still satisfied under partial correctness if we replace $x = y$ with $y = x$?