CS477 Formal Software Dev Methods

Elsa L Gunter
2112 SC, UIUC
egunter@illinois.edu
http://courses.engr.illinois.edu/cs477

Slides based in part on previous lectures
by Mahesh Vishwanathan, and by Gul Agha

February 24, 2018
Floyd-Hoare Logic

- Also called *Axiomatic Semantics*
- Based on formal logic (first order predicate calculus)
- Logical system built from *axioms* and *inference rules*
- Mainly suited to simple imperative programming languages
- Ideas applicable quite broadly
Floyd-Hoare Logic

- Used to formally prove a property (post-condition) of the state (the values of the program variables) after the execution of program, assuming another property (pre-condition) of the state holds before execution.
Floyd-Hoare Logic

- Goal: Derive statements of form

\[
\{P\} \ C \ \{Q\}
\]

- \(P, Q\) logical statements about state, \(P\) precondition, \(Q\) postcondition, \(C\) program

- Example:

\[
\{x = 1\} \ x := x + 1 \ \{x = 2\}
\]
**Approach:** For each type of language statement, give an axiom or inference rule stating how to derive assertions of form

\[
\{P\} \ C \ \{Q\}
\]

where \(C\) is a statement of that type

Compose axioms and inference rules to build proofs for complex programs
An expression \( \{ P \} \ C \ \{ Q \} \) is a partial correctness statement.

For total correctness must also prove that \( C \) terminates (i.e. doesn't run forever).

Written: \([P] \ C \ [Q]\)

Will only consider partial correctness here.
Simple Imperative Language

- We will give rules for simple imperative language

\[
\langle \text{command} \rangle ::= \langle \text{variable} \rangle ::= \langle \text{term} \rangle \\
| \langle \text{command} \rangle ; \ldots ; \langle \text{command} \rangle \\
| \text{if } \langle \text{statement} \rangle \text{ then } \langle \text{command} \rangle \text{ else } \langle \text{command} \rangle \\
| \text{while } \langle \text{statement} \rangle \text{ do } \langle \text{command} \rangle
\]

- Could add more features, like for-loops
Substitution

Notation: \( P[e/v] \) (sometimes \( P[v \rightarrow e] \))

Meaning: Replace every \( v \) in \( P \) by \( e \)

Example:

\[
(x + 2)[y - 1/x] = ((y - 1) + 2)
\]
The Assignment Rule

\{P[e/x]\} \ x \ := \ e \ \{P\}

Example:

\{\ ? \} \ x \ := \ y \ \{x = 2\}
The Assignment Rule

\[ \{ P[e/x] \} \ x := e \ \{ P \} \]

Example:

\[ \{ \quad = 2 \} \ x := \ y \ \{ \boxed{x} = 2 \} \]
The Assignment Rule

\[
\{P[e/x]\} \ x := e \ {P}
\]

Example:

\[
\{y = 2\} \ x := y \ \{x = 2\}
\]
The Assignment Rule

\[ \{P[e/x]\} \ x := e \ \{P\} \]

Examples:

\[ \{y = 2\} \ x := y \ \{x = 2\} \]

\[ \{y = 2\} \ x := 2 \ \{y = x\} \]

\[ \{x + 1 = n + 1\} \ x := x + 1 \ \{x = n + 1\} \]

\[ \{2 = 2\} \ x := 2 \ \{x = 2\} \]
What is the weakest precondition of

\[ x := x + y \{ x + y = wx \} \text{?} \]

\[
\begin{align*}
\{ & \quad ? \quad \} \\
\quad x := x + y \\
\{ & x + y = wx \}
\end{align*}
\]
What is the weakest precondition of

\[ x := x + y \{ x + y = wx \} \]?

\[ \{ (x + y) + y = w(x + y) \} \]
\[ x := x + y \]
\[ \{ x + y = wx \} \]
Meaning: If we can show that $P$ implies $P'$ (i.e. $(P \Rightarrow P')$ and we can show that $\{P\} \ C \ \{Q\}$, then we know that $\{P\} \ C \ \{Q\}$

$P$ is stronger than $P'$ means $P \Rightarrow P'$
Examples:

\[
x = 3 \Rightarrow x < 7 \quad \{ x < 7 \} \quad x := x + 3 \quad \{ x < 10 \}
\]

\[
\{ x = 3 \} \quad x := x + 3 \quad \{ x < 10 \}
\]

\[
\text{True} \Rightarrow (2 = 2) \quad \{ 2 = 2 \} \quad x := 2 \quad \{ x = 2 \}
\]

\[
\{ \text{True} \} \quad x := 2 \quad \{ x = 2 \}
\]

\[
x = n \Rightarrow x + 1 = n + 1 \quad \{ x + 1 = n + 1 \} \quad x := x + 1 \quad \{ x = n + 1 \}
\]

\[
\{ x = n \} \quad x := x + 1 \quad \{ x = n + 1 \}
\]
Which Inferences Are Correct?

\[
\{x > 0 \land x < 5\} \quad x := x \ast x \quad \{x < 25\}
\]

\[
\{x = 3\} \quad x := x \ast x \quad \{x < 25\}
\]

\[
\{x = 3\} \quad x := x \ast x \quad \{x < 25\}
\]

\[
\{x > 0 \land x < 5\} \quad x := x \ast x \quad \{x < 25\}
\]

\[
\{x \ast x < 25\} \quad x := x \ast x \quad \{x < 25\}
\]

\[
\{x > 0 \land x < 5\} \quad x := x \ast x \quad \{x < 25\}
\]
Which Inferences Are Correct?

\[
\{ x > 0 \land x < 5 \} \quad x := x \ast x \quad \{ x < 25 \}
\]

\[
\{ x = 3 \} \quad x := x \ast x \quad \{ x < 25 \} \quad \text{YES}
\]

\[
\{ x = 3 \} \quad x := x \ast x \quad \{ x < 25 \}
\]

\[
\{ x > 0 \land x < 5 \} \quad x := x \ast x \quad \{ x < 25 \}
\]

\[
\{ x \ast x < 25 \} \quad x := x \ast x \quad \{ x < 25 \}
\]

\[
\{ x > 0 \land x < 5 \} \quad x := x \ast x \quad \{ x < 25 \}
\]
Which Inferences Are Correct?

\[
\begin{align*}
\{x > 0 \land x < 5\} & \quad x := x \times x \quad \{x < 25\} \\
\{x = 3\} & \quad x := x \times x \quad \{x < 25\} \\
\{x > 0 \land x < 5\} & \quad x := x \times x \quad \{x < 25\} \\
\{x \times x < 25\} & \quad x := x \times x \quad \{x < 25\} \\
\{x > 0 \land x < 5\} & \quad x := x \times x \quad \{x < 25\}
\end{align*}
\]

**YES**

**NO**
Which Inferences Are Correct?

\[
\begin{align*}
\{x > 0 \land x < 5\} & \quad x := x \times x \quad \{x < 25\} \quad \text{YES} \\
\{x = 3\} & \quad x := x \times x \quad \{x < 25\} \\
\{x > 0 \land x < 5\} & \quad x := x \times x \quad \{x < 25\} \\
\{x \times x < 25\} & \quad x := x \times x \quad \{x < 25\} \quad \text{YES}
\end{align*}
\]
Post Condition Weakening

\[
\begin{align*}
\{ P \} & \quad C & \quad \{ Q' \} & \quad Q' \Rightarrow Q \\
\{ P \} & \quad C & \quad \{ Q \}
\end{align*}
\]

**Example:**

\[
\begin{align*}
\{ x + y = 5 \} & \quad x := x + y & \quad \{ x = 5 \} & \quad (x = 5) \Rightarrow (x < 10) \\
\{ x + y = 5 \} & \quad x := x + y & \quad \{ x < 10 \}
\end{align*}
\]
Rule of Consequence

\[ P \Rightarrow P' \quad \{P'\} \quad C \quad \{Q'\} \quad Q' \Rightarrow Q \]

\[ \{P\} \quad C \quad \{Q\} \]

- Logically equivalent to the combination of Precondition Strengthening and Postcondition Weakening
- Uses \( P \Rightarrow P \) and \( Q \Rightarrow Q \)
Sequencing

\[
\{P\} \ C_1 \ \{Q\} \ \{Q\} \ C_2 \ \{R\}
\]

\[
\{P\} \ C_1; \ C_2 \ \{R\}
\]

Example:

\[
\begin{align*}
\{z = z \land z = z\} & \quad x := z & \quad \{x = z \land z = z\} \\
\{x = z \land z = z\} & \quad y := z & \quad \{x = z \land y = z\}
\end{align*}
\]

\[
\begin{align*}
\{z = z \land z = z\} & \quad x := z; \quad y := z & \quad \{x = z \land y = z\}
\end{align*}
\]
If Then Else

\[
\frac{\{ P \land B \} \quad C_1 \quad \{ Q \} \quad \{ P \land \neg B \} \quad C_2 \quad \{ Q \}}{\{ P \} \quad \text{if } B \text{ then } C_1 \text{ else } C_2 \quad \{ Q \}}
\]

Example:

\[
\{ y = a \} \quad \text{if } x < 0 \text{ then } y := y - x \quad \text{else } y := y + x \quad \{ y = a + |x| \}
\]

By If_Then_Else Rule suffices to show:

1. \( \{ y = a \land x < 0 \} \quad y := y - x \quad \{ y = a + |x| \} \) and
2. \( \{ y = a \land \neg(x < 0) \} \quad y := y + x \quad \{ y = a + |x| \} \)
\[(1) \quad \{ y = a \land x < 0 \} \quad y := y - x \quad \{ y = a + |x| \}\]

\[\begin{align*}
(3) & \quad (y = a \land x < 0) \Rightarrow (y - x = a + |x|) \\
(2) & \quad \{ y - x = a + |x| \} \quad y := y - x \quad \{ y = a + |x| \} \\
(1) & \quad \{ y = a \land x < 0 \} \quad y := y - x \quad \{ y = a + |x| \}
\end{align*}\]

- (1) reduces to (2) and (3) by Precondition Strengthening
- (2) instance of Assignment Axiom
- (3) holds since \( x < 0 \Rightarrow |x| = -x \)
(4) \{ y = a \land \neg(x < 0) \} \ y := y + x \ \{ y = a + |x| \}

(6) \ (y = a \land \neg(x < 0)) \Rightarrow (y + x = a + |x|) \\
(5) \ {y + x = a + |x|} \ y := y + x \ \{ y = a + |x| \}

(4) reduces to (5) and (6) by Precondition Strengthening

(5) Follows from Assignment Axiom

(6) since \neg(x < 0) \Rightarrow |x| = x
If Then Else

(1) \( \{ y = a \land x < 0 \} \ y := y - x \ \{ y = a + |x| \} \)

(4) \( \{ y = a \land \neg(x < 0) \} \ y := y + x \ \{ y = a + |x| \} \)

\( \{ y = a \} \ if \ x < 0 \ then \ y := y - x \ else \ y := y + x \ \{ y = a + |x| \} \)

by the If_Then_Else Rule
We need a rule to be able to make assertions about *while* loops.

- Inference rule because we can only draw conclusions if we know something about the body
- Let's start with:

\[
\begin{align*}
\{ \ ? \ \} & \quad C & \quad \{ \ ? \ \} \\
\{ \ ? \ \} & \quad \text{while} \ B \ \text{do} \ C & \quad \{ \ P \ \}
\end{align*}
\]
While

- Loop may never execute
- To know $P$ holds after, it had better hold before
- Second approximation:

$$\begin{align*}
\{ \ ? \ \} & \quad C & \quad \{ \ ? \ \} \\
\{ P \} & \quad \text{while } B \text{ do } C & \quad \{ P \}
\end{align*}$$
Loop may execute $C$; end of loop is of $C$

$P$ holds at end of while means $P$ holds at end of loop $C$

$P$ holds at start of while; loop taken means $P \land B$ holds at start of $C$

Third approximation:

$$\{ P \land B \} \ C \ { \{ P \} }$$

$$\{ P \} \ \text{while} \ B \ \text{do} \ C \ \{ P \}$$
Always know $\neg B$ when \textit{while} loop finishes

Final \textbf{While} rule:

\[
\begin{align*}
\{P \land B\} & \quad C \quad \{P\} \\
\{P\} & \quad \text{while } B \text{ do } C \quad \{P \land \neg B\}
\end{align*}
\]
While

\[ \{ P \land B \} \quad C \quad \{ P \} \]

\[ \{ P \} \quad \text{while } B \quad \text{do } C \quad \{ P \land \neg B \} \]

- \( P \) satisfying this rule is called a loop invariant
- Must hold before and after the each iteration of the loop
• **While** rule generally used with precondition strengthening and postcondition weakening

• **No** algorithm for computing $P$ in general

• Requires intuition and an understanding of why the program works
Prove:

\[
\{ n \geq 0 \} \\
x := 0; \ y := 0; \\
while x < n do \\
\quad (y := y + ((2 \times x) + 1); \\
\quad x := x + 1) \\
\{ y = n \times n \} 
\]
Example

Need to find $P$ that is true before and after loop is executed, such that

$$(P \land \lnot (x < n)) \Rightarrow y = n \times n$$
Example

First attempt:

\[ y = x \times x \]

Motivation:

Want \( y = n \times n \)

\( x \) counts up to \( n \)

**Guess:** Each pass of loop calculates next square
By Post-condition Weakening, suffices to show:

(1) \( \{ n \geq 0 \} \)
    \hspace{1em} x := 0; \ y := 0;
    \hspace{1em} while \ x < n \ do
    \hspace{1em} (y := y + ((2 \times x) + 1); \ x := x + 1)
    \hspace{1em} \{ y = x \times x \land \neg (x < n) \} 

and

(2) \( (y = x \times x \land \neg (x < n)) \Rightarrow (y = n \times n) \)
Problem with (2)

- Want (2) \( y = x \times x \land \neg(x < n) \Rightarrow (y = n \times n) \)
- From \( \neg(x < n) \) have \( x \geq n \)
- Need \( x = n \)
- Don’t know this; from this could have \( x > n \)
- Need stronger invariant
- Try adding \( x \leq n \)
- Then have \( ((x \leq n) \land \neg(x < n)) \Rightarrow (x = n) \)
- Then have \( x = n \) when loop done
Second attempt:

\[ P = ( (y = x \times x) \land (x \leq n)) \]

Again by Post-condition Weakening, suffices to show:

(1) \( \{ n \geq 0 \} \)

\[
\begin{align*}
x & := 0; \quad y := 0; \\
while \ x < n \ do \\
& (y := y + ((2 \times x) + 1); \quad x := x + 1) \\
& \{ (y = x \times x) \land (x \leq n) \land \neg (x < n) \}
\end{align*}
\]

and

(2) \(((y = x \times x) \land (x \leq n) \land \neg (x < n)) \Rightarrow (y = n \times n))\]
Proof of (2)

\[ \neg (x < n) \Rightarrow (x \geq n) \]

\[ ((x \geq n) \land (x \leq n)) \Rightarrow (x = n) \]

\[ ((x = n) \land (y = x \times x)) \Rightarrow (y = n \times n) \]
For (1), set up While Rule using Sequencing Rule

By Sequencing Rule, suffices to show

\[(3) \{ n \geq 0 \} \ x := 0; \ y := 0 \ \{ (y = x \times x) \land (x \leq n) \}\]

and

\[(4) \ \{ (y = x \times x) \land (x \leq n) \}\]

while \( x < n \) do

\( y := y + ((2 \times x) + 1); \ x := x + 1 \)

\( \{ (y = x \times x) \land (x \leq n) \land \neg (x < n) \}\)
Proof of (4)

By While Rule

(5) \[(y = x \times x) \land (x \leq n) \land (x < n)\]

\[
y := y + ((2 \times x) + 1); \ x := x + 1
\]

\[
\{(y = x \times x) \land (x \leq n)\}
\]

while \(x < n\) do

\[
(y := y + ((2 \times x) + 1); \ x := x + 1)
\]

\[
\{(y = x \times x) \land (x \leq n) \land \neg(x < n)\}
\]
Proof of (5)

By Sequencing Rule

$$(6) \{ (y = x \times x) \land (x \leq n) \land (x < n) \}$$

$$y := y + ((2 \times x) + 1)$$

$$(7) \{ (y = (x + 1) \times (x + 1)) \land ((x + 1) \leq n) \}$$

$$x := x + 1$$

$$(y = (x + 1) \times (x + 1)) \land ((x + 1) \leq n)$$

$$(7) \text{ holds by Assignment Axiom}$$
Proof of (6)

By Precondition Strengthening

(8) \((y = x \times x) \land (x \leq n) \land (x < n) \Rightarrow (((y + ((2 \times x) + 1))) = (x + 1) \times (x + 1)) \land ((x + 1) \leq n))\)

\((9)\)
\[
\begin{align*}
&\{(y = x \times x) \land (x \leq n) \\
&\land (x < n)\} \\
&y := y + ((2 \times x) + 1) \\
&\{(y = (x + 1) \times (x + 1)) \\
&\land ((x + 1) \leq n)\}
\end{align*}
\]

Have (9) by Assignment Axiom
Proof of (8)

- (Assuming $x$ integer) $(x < n) \Rightarrow ((x + 1) \leq n)$
- $(y = x \times x) \Rightarrow ((y + ((2 \times x) + 1))$
  $= ((x \times x) + ((2 \times x) + 1))$
  $= ((x + 1) \times (x + 1)))$

That finishes (8), and thus (6) and thus (5) and thus (4) (while)

Need (3) \{n \geq 0\} x := 0; y := 0 \{(y = x \times x) \land (x \leq n)\}
Proof of (3)

By Sequencing

(10) \( \{ n \geq 0 \} \)

\( x := 0 \)

\( \{(0 = x \times x) \land (x \leq n)\} \)

(11) \( \{(0 = x \times x) \land (x \leq n)\} \)

\( y := 0 \)

\( \{(y = x \times x) \land (x \leq n)\} \)

\[ n \geq 0 \] \( x := 0; \ y := 0 \) \( \{(y = x \times x) \land (x \leq n)\} \)

Have (11) by Assignment Axiom
Proof of (10)

By Precondition Strengthening

(12) \( (n \geq 0) \Rightarrow ((0 = 0 \times 0) \land (0 \leq n)) \)

(13) \( \{ (0 = 0 \times 0) \land (0 \leq n) \} \)

\( x := 0 \)

\( \{ (0 = x \times x) \land (x \leq n) \} \)

\( \{ n \geq 0 \} \ x := 0 \); \ y := 0 \ \{ (0 = x \times x) \land (x \leq n) \} \)

- For (12), \( 0 = 0 \times 0 \) and \( (n \geq 0) \iff (0 \leq n) \)
- Have (13) by Assignment Axiom
- Finishes (10), thus (3), thus (1)