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 27, 2020

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:

$$\{x = 1\}\ x := x + 1\ \{x = 2\}$$

Floyd-Hoare Logic

- **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

Partial vs Total Correctness

- An expression $\{P\}\ C\ \{Q\}$ is a partial correctness statement
- For total correctness must also prove that C terminates (i.e. doesnt run forever)
  - Written: $[P]\ C\ [Q]$
- Will only consider partial correctness here

- We will give rules for simple imperative language

$$
\begin{aligned}
&\langle \mathit{command} \rangle \ ::= \ \langle \mathit{variable} \rangle := \langle \mathit{term} \rangle \\
&\ \mid \langle \mathit{command} \rangle; \ \ldots; \ \langle \mathit{command} \rangle \\
&\ \mid \mathit{if} \ \langle \mathit{statement} \rangle \ \mathit{then} \ \langle \mathit{command} \rangle \ \mathit{else} \ \langle \mathit{command} \rangle \\
&\ \mid \mathit{while} \ \langle \mathit{statement} \rangle \ \mathit{do} \ \langle \mathit{command} \rangle
\end{aligned}
$$

- Could add more features, like for-loops

Substitution

- Notation: $P[e/v]$ (sometimes $P[v \to e]$)
- Meaning: Replace every v in P by e
- Example:

$$(x + 2)[y - 1/x] = ((y - 1) + 2)$$

The Assingment Rule

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

Example:

$$\frac{}{\left\{\quad ?\quad\right\}\ x\ :=\ y\ \{\,x\ = 2\,\}}$$

The Assingment Rule

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

Example:

$$\frac{}{\left\{\ \boxed{} = 2\right\}\ x\ :=\ y\ \{\boxed{x} = 2\}}$$

The Assingment Rule

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

Example:

$$\overline{\left\{\ \boxed{y} = 2\right\}\ x\ :=\ y\ \{\boxed{x} = 2\ \}}$$

The Assingment Rule

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

Examples:

$$\frac{}{\{y = 2\}\ x\ :=\ y\ \{x = 2\}}$$

$$\frac{}{\{y = 2\}\ x\ :=\ 2\ \{y = x\}}$$

$$\frac{}{\{x + 1 = n + 1\}\ x\ :=\ x + 1\ \{x = n + 1\}}$$

$$\frac{}{\{2 = 2\}\ x\ :=\ 2\ \{x = 2\}}$$

The Assignment Rule – Your Turn

- What is the weakest precondition of

$$x \ := \ x + y \ \{ x + y = wx \}?$$

$$\begin{array}{c} \{ \qquad\qquad\qquad ? \qquad\qquad\qquad \} \\ x \ := \ x + y \\ \{ x + y = wx \} \end{array}$$

The Assignment Rule – Your Turn

- What is the weakest precondition of

$$x := x + y \;\{x + y = wx\}?$$

$$\{\,(x + y) + y = w(x + y)\,\}$$
$$x := x + y$$
$$\{\,x + y = wx\,\}$$

Precondition Strengthening

$$\frac{(P \Rightarrow P') \qquad \{P'\}\ C\ \{Q\}}{\{P\}\ C\ \{Q\}}$$

- 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 $P \Rightarrow P'$

Precondition Strengthening

- Examples:

$$\frac{x = 3 \Rightarrow x < 7 \quad \{x < 7\}\ x\ :=\ x + 3\ \{x < 10\}}{\{x = 3\}\ x\ :=\ x + 3\ \{x < 10\}}$$

$$\frac{\mathit{True} \Rightarrow (2 = 2) \quad \{2 = 2\}\ x\ :=\ 2\ \{x = 2\}}{\{\mathit{True}\}\ x\ :=\ 2\ \{x = 2\}}$$

$$\frac{x = n \Rightarrow x + 1 = n + 1 \qquad \{x + 1 = n + 1\}\ x\ :=\ x + 1\ \{x = n + 1\}}{\{x = n\}\ x\ :=\ x + 1\ \{x = n + 1\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}$$

$$\frac{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}$$

$$\frac{\{x * x < 25\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}\ \textit{YES}$$

$$\frac{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}$$

$$\frac{\{x * x < 25\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}\ \mathit{YES}$$

$$\frac{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}\ \mathit{NO}$$

$$\frac{\{x * x < 25\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \wedge x < 5\}\ x\ :=\ x * x\ \{x < 25\}}$$

Which Inferences Are Correct?

$$\frac{\{x > 0 \land x < 5\}\ x\ :=\ x * x\ \{x < 25\}}{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}\ \mathit{YES}$$

$$\frac{\{x = 3\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \land x < 5\}\ x\ :=\ x * x\ \{x < 25\}}\ \mathit{NO}$$

$$\frac{\{x * x < 25\}\ x\ :=\ x * x\ \{x < 25\}}{\{x > 0 \land x < 5\}\ x\ :=\ x * x\ \{x < 25\}}\ \mathit{YES}$$

Post Condition Weakening

$$\frac{\{P\}\ C\ \{Q'\} \qquad Q' \Rightarrow Q}{\{P\}\ C\ \{Q\}}$$

- Example:

$$\frac{\{x+y=5\}\ x:=x+y\ \{x=5\} \qquad (x=5) \Rightarrow (x<10)}{\{x+y=5\}\ x:=x+y\ \{x<10\}}$$

Rule of Consequence

$$\frac{P \Rightarrow P' \qquad \{P'\}\ C\ \{Q'\} \qquad Q' \Rightarrow Q}{\{P\}\ C\ \{Q\}}$$

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

Sequencing

$$\frac{\{P\}\ C_1\ \{Q\}\quad \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1;\ C_2\ \{R\}}$$

- Example:

$$\frac{\begin{array}{c}\{z = z \wedge z = z\}\ x := z\ \{x = z \wedge z = z\} \\ \{x = z \wedge z = z\}\ y := z\ \{x = z \wedge y = z\}\end{array}}{\{z = z \wedge z = z\}\ x := z;\ y := z\ \{x = z \wedge y = z\}}$$

If Then Else

$$\frac{\{P \wedge B\}\ C_1\ \{Q\} \quad \{P \wedge \neg B\}\ C_2\ \{Q\}}{\{P\}\ \mathit{if}\ B\ \mathit{then}\ C_1\ \mathit{else}\ C_2\ \{Q\}}$$

- Example:

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

  By If_Then_Else Rule suffices to show:
  - $(1)\ \{y = a \wedge x < 0\}\ y := y - x\ \{y = a + |x|\}$ and
  - $(4)\ \{y = a \wedge \neg(x < 0)\}\ y := y + x\ \{y = a + |x|\}$

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

$$\frac{\begin{array}{c}(3)\quad (y = a \wedge x < 0) \Rightarrow (y - x = a + |x|) \\ (2)\quad \{y - x = a + |x|\}\ y := y - x\ \{y = a + |x|\}\end{array}}{(1)\quad \{y = a \wedge x < 0\}\ y := y - x\ \{y = a + |x|\}}$$

- (1) reduces to (2) and (3) by Precondition Strengthening
- (2) instance of Assignment Axiom
- (3) holds since $x < 0 \Rightarrow |x| = -x$

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

$$\frac{(6)\quad (y = a \wedge \neg(x < 0)) \Rightarrow (y + x = a + |x|) \qquad (5)\quad \{y + x = a + |x|\}\ y := y + x\ \{y = a + |x\}}{(4)\quad \{y = a \wedge \neg(x < 0)\}\ y := y + x\ \{y = a + |x|\}}$$

- (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

$$\frac{\begin{array}{c}(1)\quad \{y = a \land x < 0\}\ y := y - x\ \{y = a + |x|\} \\ (4)\quad \{y = a \land \neg(x < 0)\}\ y := y + x\ \{y = a + |x|\}\end{array}}{\{y = a\}\ \mathit{if}\ x < 0\ \mathit{then}\ y := y - x\ \mathit{else}\ y := y + x\ \{y = a + |x|\}}$$

by the If_Then_Else Rule

While

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
- Lets start with:

$$\frac{\{\ ?\ \}\ C\ \{\ ?\ \}}{\{\ ?\ \}\ \mathit{while}\ B\ \mathit{do}\ C\ \{P\}}$$

While

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

$$\frac{\{\ ?\ \}\ C\ \{\ ?\ \}}{\{P\}\ \mathit{while}\ B\ \mathit{do}\ C\ \{P\}}$$

While

- Loop may execute C; enf 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 $P \wedge B$ holds at start of C
- Third approximation:

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ while\ B\ do\ C\ \{P\}}$$

While

- Always know $\neg B$ when *while* loop finishes
- Final While rule:

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \mathit{while}\ B\ \mathit{do}\ C\ \{P \wedge \neg B\}}$$

While

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \mathit{while}\ B\ \mathit{do}\ C\ \{P \wedge \neg B\}}$$

- P satisfying this rule is called a loop invariant
- Must hold before and after the each iteration of the loop

While

- 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

Example

Prove:

$$\{n \geq 0\}$$
$$x := 0;\ \ y := 0;$$
$$\mathit{while}\ x < n\ \mathit{do}$$
$$(y := y + ((2 * x) + 1);$$
$$x := x + 1)$$
$$\{y = n * n\}$$

Example

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

$$(P \wedge \neg(x < n)) \Rightarrow y = n * n$$

Example

- First attempt:

$$y = x * x$$

- Motivation:
- Want $y = n * n$
- x counts up to n
- **Guess:** Each pass of loop calcuates next square

Example

By Post-condition Weakening, suffices to show:

(1) $\{n \geq 0\}$
$x := 0;\ y := 0;$
$while\ x < n\ do$
$(y := y + ((2 * x) + 1);\ x := x + 1)$
$\{y = x * x \wedge \neg(x < n)\}$

and

(2) $(y = x * x \wedge \neg(x < n)) \Rightarrow (y = n * n)$

Problem with (2)

- Want (2) $(y = x * x \wedge \neg(x < n)) \Rightarrow (y = n * n)$
- 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 ading $x \leq n$
- Then have $((x \leq n) \wedge \neg(x < n)) \Rightarrow (x = n)$
- Then have $x = n$ when loop done

Example

Second attempt:

$$P = ((y = x * x) \wedge (x \leq n))$$

Again by Post-condition Weakening, sufices to show:

$$
\begin{array}{ll}
(1) & \{n \geq 0\} \\
 & x := 0;\ \ y := 0; \\
 & \mathit{while}\ x < n\ \mathit{do} \\
 & (y := y + ((2 * x) + 1);\ \ x := x + 1) \\
 & \{(y = x * x) \wedge (x \leq n) \wedge \neg(x < n)\}
\end{array}
$$

and

$$(2)\ \ ((y = x * x) \wedge (x \leq n) \wedge \neg(x < n)) \Rightarrow (y = n * n)$$

Proof of (2)

- $(\neg(x < n)) \Rightarrow (x \geq n)$
- $((x \geq n) \wedge (x \leq n)) \Rightarrow (x = n)$
- $((x = n) \wedge (y = x * x)) \Rightarrow (y = n * n)$

Example

- For (1), set up While Rule using Sequencing Rule
- By Sequencing Rule, suffices to show

(3) $\{n \geq 0\}\ x := 0;\ y := 0\ \{(y = x * x) \wedge (x \leq n)\}$

and

(4) $\{(y = x * x) \wedge (x \leq n)\}$
$\textit{while } x < n \textit{ do}$
$(y := y + ((2 * x) + 1);\ x := x + 1)$
$\{(y = x * x) \wedge (x \leq n) \wedge \neg(x < n)\}$

Proof of (4)

By While Rule

$$\frac{(5)\ \ \begin{array}{l}\{(y = x * x) \wedge (x \leq n) \wedge (x < n)\} \\ y := y + ((2 * x) + 1);\ \ x := x + 1 \\ \{(y = x * x) \wedge (x \leq n)\}\end{array}}{\begin{array}{l}\{(y = x * x) \wedge (x \leq n)\} \\ \mathit{while}\ x < n\ \mathit{do} \\ (y := y + ((2 * x) + 1);\ \ x := x + 1) \\ \{(y = x * x) \wedge (x \leq n) \wedge \neg(x < n)\}\end{array}}$$

Proof of (5)

By Sequencing Rule

$$\frac{\begin{array}{l}(6)\ \{(y = x * x) \wedge (x \leq n) \\ \quad \wedge (x < n)\} \\ \quad y := y + ((2 * x) + 1) \\ \quad \{(y = (x+1) * (x+1)) \\ \quad \wedge ((x+1) \leq n)\}\end{array} \qquad \begin{array}{l}(7)\ \{(y = (x+1) * (x+1)) \\ \quad \wedge ((x+1) \leq n)\} \\ \quad x := x + 1 \\ \quad \{(y = x * x) \wedge (x \leq n)\}\end{array}}{\begin{array}{l}\{(y = x * x) \wedge (x \leq n) \wedge (x < n)\} \\ y := y + ((2 * x) + 1);\ x := x + 1 \\ \{(y = x * x) \wedge (x \leq n)\}\end{array}}$$

(7) holds by Assignment Axiom

Proof of (6)

By Precondition Strengthening

$$\frac{(8)\quad \begin{array}{l}((y = x * x) \\ \wedge (x \leq n) \wedge (x < n)) \Rightarrow \\ (((y + ((2 * x) + 1)) \\ \quad = (x + 1) * (x + 1)) \\ \wedge ((x + 1) \leq n))\end{array} \qquad (9)\quad \begin{array}{l}\{((y + ((2 * x) + 1)) \\ \quad = ((x + 1) * (x + 1))) \\ \wedge ((x + 1) \leq n)\} \\ y := y + ((2 * x) + 1) \\ \{(y = (x + 1) * (x + 1)) \\ \wedge ((x + 1) \leq n)\}\end{array}}{\begin{array}{l}\{(y = x * x) \wedge (x \leq n) \\ \wedge (x < n)\} \\ y := y + ((2 * x) + 1) \\ \{(y = (x + 1) * (x + 1)) \\ \wedge ((x + 1) \leq n)\}\end{array}}$$

Have (9) by Assignment Axiom

Proof of (8)

- (Assuming x integer) $(x < n) \Rightarrow ((x+1) \leq n)$
- $(y = x * x) \Rightarrow \begin{array}{l} ((y + ((2 * x) + 1)) \\ = ((x * x) + ((2 * x) + 1)) \\ = ((x + 1) * (x + 1))) \end{array}$

- That finishes (8), and thus (6) and thus (5) and thus (4) (*while*)
- Need (3) $\{n \geq 0\}\ x := 0;\ y := 0\ \{(y = x * x) \wedge (x \leq n)\}$

Proof of (3)

By Sequencing

$$\frac{(10)\quad \begin{array}{l}\{n \geq 0\}\\ x := 0\\ \{(0 = x * x) \wedge (x \leq n)\}\end{array} \qquad (11)\quad \begin{array}{l}\{(0 = x * x) \wedge (x \leq n)\}\\ y := 0\\ \{(y = x * x) \wedge (x \leq n)\}\end{array}}{\{n \geq 0\}\ x := 0;\ y := 0\ \{(y = x * x) \wedge (x \leq n)\}}$$

Have (11) by Assignment Axiom

Proof of (10)

By Precondition Strengthening

$$\frac{(12)\ (n \geq 0) \Rightarrow ((0 = 0 * 0) \wedge (0 \leq n)) \qquad (13)\ \begin{array}{l} \{(0 = 0 * 0) \wedge (0 \leq n)\} \\ x := 0 \\ \{(0 = x * x) \wedge (x \leq n)\} \end{array}}{\{n \geq 0\}\ x := 0;\ y := 0\ \{(0 = x * x) \wedge (x \leq n)\}}$$

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