Programming Languages and Compilers (CS 421)

William Mansky

http://courses.engr.illinois.edu/cs421/

Based in part on slides by Mattox Beckman, as updated by Vikram Adve, Gul Agha, Elsa Gunter, and Dennis Griffith

Floyd-Hoare Logic

- A kind of axiomatic semantics
- Based on predicate logic
- A logical system built from *axioms* and *inference rules*
- Mainly suited to simple imperative programming languages

Floyd-Hoare Logic

- Used to formally prove a property (*post-condition*) of the *state* after the execution of a program, assuming some 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 kind 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 kind

- Compose axioms and inference rules to build proofs for complex programs

Floyd-Hoare Logic

- An expression {P} C {Q} is a *partial correctness* statement
- For *total correctness* ([P] C [Q]) must also prove that C terminates (i.e. doesn't run forever)
- We'll only consider partial correctness

Language

- We will give rules for our simple imperative language SIMPL

C ::= *I* := *E* | *C*; *C* | if *B* then *C* else *C*
| while *B* do *C*

- Could add more features, like for-loops

The Assignment Rule

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

Example:

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

The Assignment Rule

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

Example:

$$\overline{\{\boxed{\_}= 2\}\ x := y\ \{\boxed{x} = 2\}}$$

The Assignment Rule

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

Example:

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

The Assignment 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 precondition should we use for

$$x := x + y \ \{x + y = w - x\}?$$

$$\{\qquad\qquad ? \qquad\qquad\}$$

$$x := x + y$$

$$\{x + y = w - x\}$$

The Assignment Rule – Your Turn

- What precondition should we use for

$$x := x + y \; \{x + y = w - x\}?$$

$$\{(x + y) + y = w - (x + y)\}$$

$$x := x + y$$

$$\{x + y = w - x\}$$

Precondition Strengthening

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

- If we can show that P implies P' and {P'} C {Q}, then we know that {P} C {Q}
- P is *stronger* than P' means P → P'

Example

How do we prove

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

given

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

Example

How do we prove

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

We have

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

but (x = n + 1) [x + 1 / x] isn't x = n

Example

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

Precondition Strengthening

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

$$\frac{\text{True} \rightarrow 2 = 2 \quad \{2 = 2\}\ x := 2\ \{x = 2\}}{\{\text{True}\}\ x := 2\ \{x = 2\}}$$

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

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?

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

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

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

Sequencing

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

Sequencing

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

- Example:

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

- Note: $C_1; C_2; C_3$ can be $(C_1; C_2); C_3$ or $C_1; (C_2; C_3)$

Postcondition Weakening

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

Example:

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

- Lets us summarize the goal of a program

Rule of Consequence

$$\frac{P \rightarrow P' \quad \{P'\}\ C\ \{Q'\} \quad Q' \rightarrow Q}{\{P\}\ C\ \{Q\}}$$

- Combination of Precondition Strengthening and Postcondition Weakening
- Note the direction of the implications!

If Then Else

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

- Example: Want

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

If Then Else Example

- Example: Want

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

Suffices to show:

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

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

If Then Else Example

$$\text{(3)} \qquad y = a \wedge x < 0 \Rightarrow ?$$

$$\text{(2)} \qquad \{?\}\ y := y - x\ \{y = a + |x|\}$$

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

(1) Reduces to (2) and (3) by Precondition Strengthening

If Then Else Example

(3) $y = a \wedge x \geq 0 \rightarrow y - x = a + |x|$

$$\frac{(2)\ \{y - x = a + |x|\}\ y := y - x\ \{y = a + |x|\}}{(1)\ \{y = a \wedge x < 0\}\ y := y - x\ \{y = a + |x|\}}$$

(1) Reduces to (2) and (3) by Precondition Strengthening

(2) Follows from assignment axiom

(3) Because $x < 0 \rightarrow |x| = -x$

If Then Else Example

(6) $y = a \wedge x \geq 0 \rightarrow y + x = a + |x|$

$$\frac{(5)\ \{y + x = a + |x|\}\ y := y + x\ \{y = a + |x|\}}{(4)\ \{y = a \wedge x \geq 0\}\ y := y + x\ \{y = a + |x|\}}$$

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

(5) Follows from assignment axiom

(6) Because $x \geq 0 \rightarrow |x| = x$

Example

$$\frac{(1)\ \{y = a \wedge x < 0\}\ y := y - x\ \{y = a + |x|\} \qquad (4)\ \{y = a \wedge x \geq 0\}\ y := y + x\ \{y = a + |x|\}}{\{y = a\}\ \text{if } x < 0 \text{ then } y := y - x \text{ else } y := y + x\ \{y = a + |x|\}}$$

By the if_then_else rule

While

- We need a rule to be able to make assertions about **while** loops
  - Premise must involve the body
  - Let's start with:

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

While

- The loop may never be executed, so let's try:

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

While

- If all we know is P when we enter the **while** loop, then we all we know when we enter the body is (P and B)
- P must hold when we finish the loop body:

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

While

- When the loop is finished, **not B** also holds
- Final **while** rule:

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

While

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

- P satisfying this rule is called a *loop invariant* because it must hold before and after the each iteration of the loop

While

- **While** rule generally needs to be used together with precondition strengthening and postcondition weakening
- There is NO algorithm for computing the correct P; it requires intuition and an understanding of why the program works

Example

- Let us prove

  $\{x \geq 0 \wedge x = a\}$

  fact := 1;

  while x > 0 do (fact := fact * x; x := x – 1)

  $\{fact = a!\}$

Example

- We need to find a condition P that is true both before and after the loop is executed, and such that

$$P \land \neg(x > 0) \Rightarrow fact = a!$$

Example

- First attempt:

{a! = fact * (x!)}

- Motivation:
- What we want to compute: **a!**
- What we have computed: **fact**

  which is the sequential product of **a** down through **(x + 1)**
- What we still need to compute: **x!**

Example

By post-condition strengthening suffices to show

1. $\{x \geq 0 \wedge x = a\}$

   fact := 1;

   while x > 0 do (fact := fact * x; x := x –1)

   $\{a! = fact * (x!) \wedge \neg(x > 0)\}$

and

2. $a! = fact * (x!) \wedge \neg(x > 0) \rightarrow fact = a!$

Problem

2. $a! = fact * (x!) \wedge \neg(x > 0) \rightarrow ?\ fact = a!$

- Not true if $x < 0$
- Need to know that $x = 0$ when loop terminates
- Loop invariant must include info about x
- If we add $x \geq 0$, then we'll have $x = 0$ when loop exits

Example

Second try, combine the two:

$$P = \{a! = fact * (x!) \wedge x \geq 0\}$$

Again, suffices to show

1. $\{x \geq 0 \wedge x = a\}$
   fact := 1;
   while x > 0 do (fact := fact * x; x := x –1)
   $\{P \wedge \neg(x > 0)\}$

and

2. $P \wedge \neg(x > 0) \rightarrow fact = a!$

Example

- For 2, we need

  $a! = fact * (x!) \land x \geq 0 \land \neg(x > 0) \Rightarrow fact = a!$

- But $x \geq 0 \land \neg(x > 0) \Rightarrow x = 0$ so

  $fact * (x!) = fact * (0!) = fact$

- Therefore

  $a! = fact * (x!) \land x \geq 0 \land \neg(x > 0) \Rightarrow fact = a!$

Example

- For 1, by the sequencing rule it suffices to show

3. $\{x \geq 0 \wedge x = a\}$
   fact := 1
   $\{a! = fact * (x!) \wedge x \geq 0 \}$

And

4. $\{a! = fact * (x!) \wedge x \geq 0\}$
   while x > 0 do
   (fact := fact * x; x := x –1)
   $\{a! = fact * (x!) \wedge x \geq 0 \wedge \neg(x > 0)\}$

Example

- Suffices to show that

  $$\{a! = fact * (x!) \wedge x \geq 0\}$$

  holds before the while loop is entered and that if

  $$\{(a! = fact * (x!)) \wedge x \geq 0 \wedge x > 0\}$$

  holds before we execute the body of the loop, then

  $$\{(a! = fact * (x!)) \wedge x \geq 0\}$$

  holds after we execute the body

Example

By the assignment rule, we have

$$\{a! = 1 * (x!) \wedge x \geq 0\}$$
$$\text{fact} := 1$$
$$\{a! = \text{fact} * (x!) \wedge x \geq 0\}$$

Therefore, to show (3), by precondition strengthening, it suffices to show

$$x \geq 0 \wedge x = a \rightarrow$$
$$a! = 1 * (x!) \wedge x \geq 0$$

Example

$$x \geq 0 \wedge x = a \rightarrow$$
$$a! = 1 * (x!) \wedge x \geq 0$$

holds because $x = a \rightarrow x! = a!$

Have that $\{a! = fact * (x!) \wedge x \geq 0\}$ holds at the start of the while loop

Example

To show (4):

$\{a! = fact * (x!) \wedge x \geq 0\}$
while $x > 0$ do
$(fact := fact * x; x := x - 1)$
$\{a! = fact * (x!) \wedge x \geq 0 \wedge \neg(x > 0)\}$

we need to show that

$$a! = fact * (x!) \wedge x \geq 0$$

is a loop invariant

Example

We need to show:

$$\{a! = fact * (x!) \wedge x \geq 0 \wedge x > 0\}$$
$$(\ fact = fact * x;\ x := x - 1\ )$$
$$\{a! = fact * (x!) \wedge x \geq 0\}$$

We will use assignment rule, sequencing rule and precondition strengthening

Example

By the assignment rule, we have

$$\{a! = fact * ((x-1)!) \wedge x - 1 \geq 0\}$$

$$x := x - 1$$

$$\{a! = fact * (x!) \wedge x \geq 0\}$$

By the sequencing rule, it suffices to show

$$\{a! = fact * (x!) \wedge x \geq 0 \wedge x > 0\}$$

$$fact = fact * x$$

$$\{a! = fact * ((x-1)!) \wedge x - 1 \geq 0\}$$

Example

By the assignment rule, we have that

$$\{a! = (fact * x) * ((x\text{-}1)!) \wedge x - 1 \geq 0\}$$

$$fact = fact * x$$

$$\{a! = fact * ((x\text{-}1)!) \wedge x - 1 \geq 0\}$$

By Precondition strengthening, it suffices to show that

$a! = fact * (x!) \wedge x \geq 0 \wedge x > 0 \rightarrow$

$a! = (fact * x) * ((x - 1)!)$ and $x - 1 \geq 0$

Example

However

$$fact * x * (x - 1)! = fact * x$$

and $(x > 0) \rightarrow x - 1 \geq 0$

since x is an integer, so

$$a! = fact * (x!) \wedge x \geq 0 \wedge x > 0 \rightarrow$$

$$a! = (fact * x) * ((x - 1)!)) \wedge x - 1 \geq 0$$

Example

Therefore, by precondition strengthening

$$\{a! = fact * (x!) \wedge x \geq 0 \wedge x > 0\}$$

$$fact = fact * x$$

$$\{a! = fact * ((x - 1)!) \wedge x - 1 \geq 0\}$$

QED!