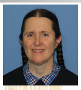
Programming Languages and Compilers (CS 421)

Elsa L Gunter
2112 SC, UIUC



https://courses.engr.illinois.edu/cs421/sp2023

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

Programming Languages & Compilers

III : Language Semantics

- Operational Semantics
- Lambda Calculus
- Axiomatic Semantics

Axiomatic Semantics

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

Axiomatic Semantics

- 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

Axiomatic Semantics

- 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}

Axiomatic Semantics

- *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

Axiomatic Semantics

- 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

Language

- We will give rules for simple imperative language

<command>
 ::= <variable> := <term>
 | <command>; … ;<command>
 | if <statement> then <command> else <command> fi
 | while <statement> do <command> od

- Could add more features, like for-loops

Substitution

- Notation: P[e/v] (sometimes P[v <- 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:

{ _ = 2 } x := y { x = 2}

The Assignment Rule

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

Example:

{ y = 2 } x := y { 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 is the weakest precondition of
  $x := x + y\ \{x + y = w - x\}$?

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

The Assignment Rule – Your Turn

- What is the weakest precondition of
  $x := x + y\ \{x + y = w - x\}$?

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

1725 minutes

Precondition Strengthening

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

- Meaning: If we can show that P implies P' (P ➔ P') and we can show that {P'} C {Q}, then we know that {P} C {Q}
- P is *stronger* than P' means P ➔ P'

Precondition Strengthening

- Examples:

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

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

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

Which Inferences Are Correct?

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

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

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

Which Inferences Are Correct?

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

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

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

Sequencing

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

- Example:

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

Sequencing

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

- Example:

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

Postcondition Weakening

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

Example:

$$\frac{\begin{array}{c}\{z = z \;\&\; z = z\}\ x := z;\ y := z\ \{x = z \;\&\; y = z\} \\ (x = z \;\&\; y = z) \Rightarrow (x = y)\end{array}}{\{z = z \;\&\; z = z\}\ x := z;\ y := z\ \{x = y\}}$$

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 $Q' \Rightarrow Q$

1750 minutes

If Then Else

$$\frac{\{P \text{ and } B\}\ C_1\ \{Q\} \quad \{P \text{ and } (\text{not } B)\}\ C_2\ \{Q\}}{\{P\}\ \textbf{if } B \textbf{ then } C_1 \textbf{ else } C_2 \textbf{ fi}\ \{Q\}}$$

- Example: Want

{y=a}

if x < 0 then y:= y-x else y:= y+x fi

{y=a+|x|}

Suffices to show:

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

(4) {y=a&not(x<0)} y:=y+x {y=a+|x|}

{y=a&x<0} y:=y-x {y=a+|x|}

(3) (y=a&x<0)➔ y-x=a+|x|

(2) {y-x=a+|x|} y:=y-x {y=a+|x|}

(1) {y=a&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 ➔ |x| = -x

{y=a&not(x<0)} y:=y+x {y=a+|x|}

(6) (y=a&not(x<0))➔(y+x=a+|x|)

(5) {y+x=a+|x|} y:=y+x {y=a+|x}}

(4) {y=a&not(x<0)} y:=y+x {y=a+|x|}

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

(5) Follows from assignment axiom

(6) Because not(x<0) ➔ |x| = x

If then else

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

(4) {y=a&not(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

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
  - Let's start with:

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

While

- The loop may never be executed, so if we want P to hold after, it had better hold before, so let's try:

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

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)
- If we need to know P when we finish the **while** loop, we had better know it when we finish the loop body:

$$\frac{\{\ P\ \text{and}\ B\}\ \ C\ \ \{\ P\ \}}{\{\ P\ \}\ \ \textbf{while}\ \ B\ \ \textbf{do}\ \ C\ \textbf{od}\ \ \{\ P\ \}}$$

While

- We can strengthen the previous rule because we also know that when the loop is finished, **not B** also holds
- Final **while** rule:

$$\frac{\{\ P\ \text{and}\ B\ \}\ \ C\ \ \{\ P\ \}}{\{\ P\ \}\ \textbf{while}\ \ B\ \ \textbf{do}\ \ C\ \textbf{od}\ \ \{\ P\ \text{and not}\ B\ \}}$$