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
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 $\{P\} \rightarrow C \{Q\}$ is a partial correctness statement.

- For total correctness must also prove that $C$ terminates (i.e. doesn’t run forever).
  - Written: $[P] \rightarrow C [Q]$

- Will only consider partial correctness here.
Language

- We will give rules for simple imperative language

\[
\text{<command>} \\
\text{::=} \text{<variable>} := \text{<term>} \\
| \text{<command>;} \ldots ;\text{<command>} \\
| \text{if <statement> then <command> else <command> fi} \\
| \text{while <statement> do <command> od}
\]

- 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

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

Examples:

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

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

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

\[
\{2 = 2\} \ x := 2 \ {x = 2}
\]
The Assignment Rule – Your Turn

What is the weakest precondition of

\[ x := x + y \{x + y = w - x}\]?

\[
\begin{array}{c}
\{?\} \\
x := x + y \\
\{x + y = w - x\}
\end{array}
\]
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

\[ P \rightarrow P' \quad \{P'\} \subset \{Q\} \quad \{P\} \subset \{Q\} \]

- Meaning: If we can show that \( P \) implies \( P' \) \((P \rightarrow P')\) and we can show that \( \{P'\} \subset \{Q\} \), then we know that \( \{P\} \subset \{Q\} \).
- \( P \) is stronger than \( P' \) means \( P \rightarrow P' \).
Precondition Strengthening

Examples:

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

- 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?

\[
\begin{align*}
\{x > 0 \& x < 5\} \quad &x := x \times x \quad \{x < 25\} \\
\{x = 3\} \quad &x := x \times x \quad \{x < 25\} \\
\{x = 3\} \quad &x := x \times x \quad \{x < 25\} \\
\{x > 0 \& x < 5\} \quad &x := x \times x \quad \{x < 25\} \\
\{x \times x < 25\} \quad &x := x \times x \quad \{x < 25\} \\
\{x > 0 \& x < 5\} \quad &x := x \times x \quad \{x < 25\}
\end{align*}
\]
Which Inferences Are Correct?

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

\[
\begin{align*}
\{P\} & \ C_1 \ \{Q\} \quad \{Q\} \ C_2 \ \{R\} \\
\{P\} & \ C_1; \ C_2 \ \{R\}
\end{align*}
\]

Example:

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

\[
\{P\} \ C_1 \ {Q} \quad \{Q\} \ C_2 \ {R}\ \\
\{P\} \ C_1 ; \ C_2 \ {R}\n\]

Example:

\[
\{z = z & z = z\} \ x := z \quad \{x = z & z = z\}
\]

\[
\{x = z & z = z\} \ y := z \quad \{x = z & y = z\}
\]

\[
\{z = z & z = z\} \ x := z \ ; \ y := z \quad \{x = z & y = z\}
\]
Postcondition Weakening

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

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

Example:

\[
\{z = z & z = z\} \ x := z; \ y := z \ \{x = z & y = z\}
\]

\[
(x = z & y = z) \Rightarrow (x = y)
\]

\[
\{z = z & z = z\} \ x := z; \ y := z \ \{x = y\}
\]
Rule of Consequence

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

- Logically equivalent to the combination of Precondition Strengthening and Postcondition Weakening
- Uses \( P \rightarrow P' \) and \( Q' \rightarrow Q \)
1750 minutes
{P and B} C_1 {Q}   {P and (not B)} C_2 {Q} 

{P} if B then C_1 else C_2 fi {Q}

- Example: Want

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

\[
\{y=a\}
\]

Suffices to show:

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

(4) \{y=a\&\text{not}(x<0)\} y := y + x \{y=a+|x|\}
\{y = a & x < 0\} \ y := y - x \ \{y = a + |x|\} \\
(3) \quad (y = a & x < 0) \Rightarrow y - x = a + |x| \\
(2) \quad \{y - x = a + |x|\} \ y := y - x \ \{y = a + |x|\} \\
(1) \quad \{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 \Rightarrow |x| = -x
\{y=a \& \neg(x<0)\} \ y:=y+x \ \{y=a+|x|\}

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

(4) Reduces to (5) and (6) by Precondition Strengthening
(5) Follows from assignment axiom
(6) Because \(\neg(x<0) \Rightarrow |x| = x\)
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|\}

\begin{align*}
\{y=a\} \\
\text{if } x < 0 \text{ then } y := y - x \text{ else } y := y + x \\
\{y = a + |x|\}
\end{align*}

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:

\[
\{ ? \} \quad C \quad \{ ? \} \\
\{ ? \} \quad \textbf{while} \quad B \quad \textbf{do} \quad C \quad \textbf{od} \quad \{ P \} 
\]
The loop may never be executed, so if we want \( P \) to hold after, it had better hold before, so let’s try:

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

- If all we know is \( P \) when we enter the \textbf{while} loop, then we all we know when we enter the body is \((P \text{ and } B)\).

- If we need to know \( P \) when we finish the \textbf{while} loop, we had better know it when we finish the loop body:

\[
\{ P \text{ and } B \} \ C \ \{ P \} \\
\underline{\{ P \} \text{ while } B \text{ do } C \text{ od} \ \{ P \}}
\]
We can strengthen the previous rule because we also know that when the loop is finished, not B also holds.

Final **while** rule:

\[
\begin{align*}
\{ P \text{ and } B \} & \quad C & \{ P \} \\
\{ P \} & \quad \textbf{while} \; B \; \textbf{do} \; C \; \textbf{od} & \{ P \text{ and } \neg B \}
\end{align*}
\]