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

**Example:** \{x = 1\} x := x + 1 \{x = 2\}

- Goal: Derive statements of form \{P\} C \{Q\}
  - P, Q logical statements about state, P precondition, Q postcondition, C program

**Approach:** For each type of language statement, give an axiom or inference rule stating how to derive assertions of form \{P\} C \{Q\}

- 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

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

- We will give rules for simple imperative language

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

- 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\}

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

\[
\begin{align*}
\{ \quad ? \quad \} \\
\{ x := x + y \} \\
\{ x + y = w - x \}
\end{align*}
\]

Precondition Strengthening

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

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

Examples:

\[
\begin{align*}
x &= 3 \Rightarrow x < 7 \{x < 7\} \\
x := x + 3 \{x < 10\} \\
\{x = 3\} \\
x := x + 3 \{x < 10\} \\
\{x < 7\} \\
x := x + 3 \{x < 25\}
\end{align*}
\]

\[
\begin{align*}
\text{True} \Rightarrow 2 = 2 \{2 = 2\} \\
x := 2 \{x = 2\} \\
\{\text{True}\} \\
x := 2 \{x = 2\} \\
x = n \Rightarrow x + 1 = n + 1 \{x + 1 = n + 1\} \\
x := x + 1 \{x = n + 1\} \\
\{x = n\} \\
x := x + 1 \{x = n + 1\}
\end{align*}
\]

Which Inferences Are Correct?

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

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

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

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

Example:
\[ \{z = z \land z = z\} \ x := z \ \{x = z \land z = z\} \]
\[ \{x = z \land z = z\} \ y := z \ \{x = z \land y = z\} \]
\[ \{z = z \land z = z\} \ x := z \ ; y := z \ \{x = z \land y = z\} \]

Postcondition Weakening

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

Example:
\[ \{z = z \land z = z\} \ x := z \ \{x = z \land z = z\} \]
\[ \{x = z \land z = z\} \ y := z \ \{x = z \land y = z\} \]
\[ \{z = z \land z = z\} \ x := z \ ; y := z \ \{x = z \land y = z\} \]

Rule of Consequence

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

Logically equivalent to the combination of Precondition Strengthening and Postcondition Weakening.

Uses \[ P \Rightarrow P \] and \[ Q \Rightarrow Q \]

If Then Else

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

Example: Want
\[ \{y = a\} \]
if \( x < 0 \) then \( y := y - x \) else \( y := y + x \)
\[ \{y = a + |x|\} \]
Suffices to show:
1. \( \{y = a \land x < 0\} \ y := y - x \quad \{y = a + |x|\} \)
2. \( \{y - x = a + |x|\} \ y := y - x \quad \{y = a + |x|\} \)
3. \( \{y = a \land x < 0\} \ y := y - x \quad \{y = a + |x|\} \)

(1) Reduces to (2) and (3) by Precondition Strengthening.
(2) Follows from assignment axiom.
(3) Because \( x < 0 \Rightarrow |x| = -x \)
\{y=a&\not(x<0)\} \ y:=y+x \ \{y=a+|x|\}

(6) \ (y=a&\not(x<0)) \Rightarrow (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) \Rightarrow |x| = x

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

By the if_then_else rule

\{P\} \ while \ B \ do \ C \ \{P\}

\{P \ and \ B\} \ C \ \{P\}

\{P\} \ while \ B \ do \ C \ \{P \ and \ not \ B\}

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:

\{ ? \} \ C \ \{ ? \}
\{ ? \} \ while \ B \ do \ C \ \{ P \}

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

\{ ? \} \ C \ \{ ? \}
\{ P \} \ while \ B \ do \ C \ \{ P \}

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

\{ P \ and \ B \} \ C \ \{ P \}
\{ P \} \ while \ B \ do \ C \ \{ P \ and \ not \ B \}
\[ \text{While} \quad \{P \land B\} \quad C \quad \{P\} \quad \text{while} \quad B \quad \text{do} \quad C \quad \{P \land \neg B\} \]

- P satisfying this rule is called a \textit{loop invariant} because it must hold before and after the each iteration of the loop.

\textbf{Example}

- Let us prove \(\{x \geq 0 \land x = a\}\)
  \[
  \text{fact} := 1;
  \text{while}\ x > 0 \text{ do (fact := fact} \times x; \ x := x - 1)\]
  \(\{\text{fact} = a!\}\)

\textbf{Example}

- We need to find a condition P that is true both before and after the loop is executed, and such that
  \[
  (P \land \neg x > 0) \implies (\text{fact} = a!) \]

\textbf{Example}

- First attempt:
  \(\{a! = \text{fact} \times (x!)\}\)
  
  - Motivation:
  - What we want to compute: a!
  - What we have computed: \text{fact}
    which is the sequential product of a down through \((x + 1)\)
  - What we still need to compute: x!

\textbf{Example}

- By post-condition weakening suffices to show
  1. \(\{x \geq 0 \land x = a\}\)
     \[
     \text{fact} := 1;
     \text{while}\ x > 0 \text{ do (fact := fact} \times x; \ x := x - 1)\]
     \(\{a! = \text{fact} \times (x!) \land \neg (x > 0)\}\)
     and
  2. \(\{a! = \text{fact} \times (x!) \land \neg (x > 0)\} \implies \{\text{fact} = a!\}\)
Problem

2. \{a! = \text{fact} \ast (x!) \text{ and not (x > 0)} \} \Rightarrow \{\text{fact} = a!\}
- Don’t know this if x < 0
- Need to know that x = 0 when loop terminates
- Need a new loop invariant
- Try adding x >= 0
- Then will have x = 0 when loop is done

Example

Second try, combine the two:
\[ P = \{a! = \text{fact} \ast (x!) \text{ and x >=0}\} \]
Again, suffices to show
1. \{x>=0 and x = a\}
   \text{fact} := 1;
   \text{while x > 0 do (fact} := \text{fact} \ast x; x := x –1)\}
   \{P \text{ and not x > 0}\}
   \text{and}
2. \{P \text{ and not x > 0} \Rightarrow \{\text{fact} = a!\}\}

Example

For 1, by the sequencing rule it suffices to show
3. \{x>=0 and x = a\}
   \text{fact} := 1
   \{a! = \text{fact} \ast (x!) \text{ and x >=0}\}
And
4. \{a! = \text{fact} \ast (x!) \text{ and x >=0}\}
   \text{while x > 0 do (fact} := \text{fact} \ast x; x := x –1)\}
   \{a! = \text{fact} \ast (x!) \text{ and x >=0 and not (x > 0)}\}

Example

Suffices to show that
\{a! = \text{fact} \ast (x!) \text{ and x >=0}\}
holds before the while loop is entered and that if
\{(a! = \text{fact} \ast (x!)) \text{ and x >=0 and x > 0}\}
holds before we execute the body of the loop, then
\{(a! = \text{fact} \ast (x!)) \text{ and x >=0}\}
holds after we execute the body

Example

By the assignment rule, we have
\{a! = 1 \ast (x!) \text{ and x >=0}\}
\text{fact} := 1
\{a! = \text{fact} \ast (x!) \text{ and x >=0}\}
Therefore, to show (3), by precondition strengthening, it suffices to show
\{(x>=0 and x = a) \Rightarrow \}
\{(a! = 1 \ast (x!) \text{ and x >=0})\}
Example

\begin{align*}
(x & \geq 0 \text{ and } x = a) \Rightarrow \\
(a & = 1 \ast (x!) \text{ and } x \geq 0) \\
\text{holds because } x = a & \Rightarrow x! = a!
\end{align*}

Have that \{a! = fact \ast (x!) \text{ and } x \geq 0\} holds at the start of the while loop.

Example

To show (4):
\[\{a! = fact \ast (x!) \text{ and } x \geq 0\}\]
while \(x > 0\) do
\[\text{fact} := \text{fact} \ast x; x := x - 1\]
\[\{a! = fact \ast (x!) \text{ and } x \geq 0 \text{ and not } (x > 0)\}\]
we need to show that
\[\{a! = fact \ast (x!) \text{ and } x \geq 0\}\]
is a loop invariant.

Example

We need to show:
\[\{(a! = fact \ast (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}\]
\[\text{fact} = \text{fact} \ast x; x := x - 1\]
\[\{(a! = fact \ast (x!)) \text{ and } x \geq 0\}\]

We will use assignment rule, sequencing rule and precondition strengthening.

Example

By the assignment rule, we have
\[\{a! = (\text{fact} \ast x) \ast ((x-1)!) \text{ and } x - 1 \geq 0\}\]
\[\text{fact} = \text{fact} \ast x\]
\[\{(a! = \text{fact} \ast ((x-1)!) \text{ and } x - 1 \geq 0\}\]
By Precondition strengthening, it suffices to show that
\[\{(a! = \text{fact} \ast (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}\]
\[\{(a! = (\text{fact} \ast x) \ast ((x-1)!) \text{ and } x - 1 \geq 0\}\]

Example

However
\[\text{fact} \ast x \ast (x - 1)! = \text{fact} \ast x\]
and
\[x > 0 \Rightarrow x - 1 \geq 0\]
since \(x\) is an integer, so
\[\{(a! = \text{fact} \ast (x!)) \text{ and } x \geq 0 \text{ and } x > 0\} \Rightarrow \]
\[\{(a! = (\text{fact} \ast x) \ast ((x-1)!) \text{ and } x - 1 \geq 0\}\]
Example

Therefore, by precondition strengthening

\{(a! = \text{fact} \times (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}

\text{fact} = \text{fact} \times x

\{(a! = \text{fact} \times ((x-1)!)) \text{ and } x - 1 \geq 0\}

This finishes the proof