Axiomatic Semantics

CS 4301/6371: Advanced Programming Languages

Kevin W. Hamlen

April 18, 2024

Roadmap

- Operational Semantics
  - large-step and small-step varieties
  - formally defines the *operation* of a machine that executes the program
- Denotational Semantics
  - defines the mathematical object (i.e., function) that a program *denotes*
- Static Semantics (Type Theory)
  - a *static analysis* that prevents certain runtime errors ("stuck states")
- Today: Axiomatic Semantics

Axiomatic Semantics

- Goal: We wish to prove complete correctness of mission-critical code.
  - Type-theory too weak* (just proves soundness)
  - Operational semantics requires us to step outside the derivation system to prove things about derivations. Non-derivation parts cannot be machine-checked.
  - Denotational semantics creates a massive mathematical object that encodes all memory states (too hard to reason about).
- Solution: Axiomatic Semantics
  - inference rules that encapsulate the entire correctness proof into a derivation
  - Derivation is fully machine-checkable, so no reliance on (error-prone) humans writing perfect proofs or perfectly checking proofs.

*Actually, advanced type systems like λ_C encode an entire axiomatic semantics into the type system, but let's classify that as type theory + axiomatic semantics.

Two Kinds of Correctness

- Partial Correctness
  - Notation: $\{A\}c\{B\}$ (called a *Hoare triple*)
  - If A is true before executing c, and if c terminates, then B is true after executing c.
  - A is *precondition*, and B is *postcondition*
- Total Correctness
  - Notation: $[A]c[B]$
  - If A is true before executing c, then c eventually terminates and B is true once it does.

Examples

1. $\{x \leq 10\}$**while** `x <= 10` **do** `x := x + 1`$\{?\}$

Examples

1. $\{x \leq 10\}$**while** x <= 10 **do** x := x + $1\{x = 11\}$

Examples

1. $\{x \leq 10\}\mathbf{while}\ \mathtt{x} \mathrel{\mathtt{<=}} 10\ \mathbf{do}\ \mathtt{x} \mathrel{\mathtt{:=}} \mathtt{x} + 1\{x = 11\}$
2. $[x \leq 10]\mathbf{while}\ \mathtt{x} \mathrel{\mathtt{<=}} 10\ \mathbf{do}\ \mathtt{x} \mathrel{\mathtt{:=}} \mathtt{x} + 1[?]$

Examples

1. $\{x \leq 10\}$**while** `x <= 10` **do** `x := x + 1`$\{x = 11\}$
2. $[x \leq 10]$**while** `x <= 10` **do** `x := x + 1`$[x = 11]$

Examples

1. $\{x \leq 10\}$**while** x <= 10 **do** x := x + 1$\{x = 11\}$
2. $[x \leq 10]$**while** x <= 10 **do** x := x + 1$[x = 11]$
3. $[T]$**while** x <= 10 **do** x := x + 1$[?]$

Examples

1. $\{x \leq 10\}$**while** $\mathtt{x} <= 10$ **do** $\mathtt{x} := \mathtt{x} + 1 \{x = 11\}$
2. $[x \leq 10]$**while** $\mathtt{x} <= 10$ **do** $\mathtt{x} := \mathtt{x} + 1 [x = 11]$
3. $[T]$**while** $\mathtt{x} <= 10$ **do** $\mathtt{x} := \mathtt{x} + 1 [x \geq 11]$

Examples

1. $\{x \leq 10\}$**while** `x <=` 10 **do** `x := x +` $1\{x = 11\}$
2. $[x \leq 10]$**while** `x <=` 10 **do** `x := x +` $1[x = 11]$
3. $[T]$**while** `x <=` 10 **do** `x := x +` $1[x \geq 11]$
4. $[x = \bar{i}]$**while** `x <=` 10 **do** `x := x +` $1[?]$

Examples

1. $\{x \leq 10\}$**while** x <= 10 **do** x := x + 1$\{x = 11\}$
2. $[x \leq 10]$**while** x <= 10 **do** x := x + 1$[x = 11]$
3. $[T]$**while** x <= 10 **do** x := x + 1$[x \geq 11]$
4. $[x = \bar{i}]$**while** x <= 10 **do** x := x + 1$[x = \max(11, \bar{i})]$

Examples

1. $\{x \leq 10\}$**while** x <= 10 **do** x := x + $1\{x = 11\}$
2. $[x \leq 10]$**while** x <= 10 **do** x := x + $1[x = 11]$
3. $[T]$**while** x <= 10 **do** x := x + $1[x \geq 11]$
4. $[x = \bar{i}]$**while** x <= 10 **do** x := x + $1[x = \max(11, \bar{i})]$
5. $\{T\}$**while true do skip**$\{F\}$

Examples

1. $\{x \leq 10\}$**while** `x <= 10` **do** `x := x + 1`$\{x = 11\}$
2. $[x \leq 10]$**while** `x <= 10` **do** `x := x + 1`$[x = 11]$
3. $[T]$**while** `x <= 10` **do** `x := x + 1`$[x \geq 11]$
4. $[x = \bar{i}]$**while** `x <= 10` **do** `x := x + 1`$[x = \max(11, \bar{i})]$
5. $\{T\}$**while true do skip**$\{F\}$
   - $\{$any $A\}$any non-terminating program$\{$any $B\}$

Examples

1. $\{x \leq 10\}$**while** x <= 10 **do** x := x + $1\{x = 11\}$
2. $[x \leq 10]$**while** x <= 10 **do** x := x + $1[x = 11]$
3. $[T]$**while** x <= 10 **do** x := x + $1[x \geq 11]$
4. $[x = \bar{i}]$**while** x <= 10 **do** x := x + $1[x = \max(11, \bar{i})]$
5. $\{T\}$**while true do skip**$\{F\}$
   - $\{$any $A\}$any non-terminating program$\{$any $B\}$
6. $\{F\}$any program$\{$any $B\}$

Language of Assertions

- First-order logic with arithmetic:

$$\begin{array}{ll} \text{arithmetic exps} & a ::= n \mid v \mid \bar{v} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \\ \text{assertions} & A ::= T \mid F \mid a_1 = a_2 \mid a_1 \leq a_2 \mid A_1 \wedge A_2 \\ & \quad \mid A_1 \vee A_2 \mid \neg A \mid A \Rightarrow A_2 \mid \forall \bar{v}.A \mid \exists \bar{v}.A \end{array}$$

- *Meta-variables* ($\bar{v}$) are **mathematical** variables (not program variables) that have fixed (arbitrary) integer values across all assertions.
- From these one can construct all functions and logical operators, so we will freely use extensions to the above.
    - But if you write something extremely exotic, I reserve the right to challenge you on whether it can actually be expressed using the above.

Hoare Logic

- First published by Tony Hoare [1969]
  - First and most famous axiomatic semantics
  - "An axiomatic basis for computer programming"
  - Often cited as one of the greatest CS papers of all time (only 6 pages long!)
  - Optional: read the original paper (linked from course web site)
- Adaption to SIMPL consists of...
  - six axioms (rules) describing SIMPL programs
  - inference rules of first-order logic
  - axioms of arithmetic (e.g., Peano arithmetic)

Skip Rule

$$\frac{}{\{A\}\mathbf{skip}\{\,?\,\}}(1)$$

Skip Rule

$$\frac{}{\{A\}\mathbf{skip}\{A\}}(1)$$

Sequence Rule

$$\frac{}{\{A\}c_1\,;c_2\{B\}}\ (2)$$

Sequence Rule

$$\frac{\{A\}c_1\{C\} \qquad \{C\}c_2\{B\}}{\{A\}c_1\,;c_2\{B\}}\text{(2)}$$

Conditional Rule

$$\frac{}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}}$$

Conditional Rule

$$\frac{\{A\}c_1\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}} \text{(3a)}$$

$$\frac{\{A\}c_2\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}} \text{(3b)}$$

Conditional Rule

$$\frac{\{A\}c_1\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}}\text{(3a)}$$

$$\frac{\{A\}c_2\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}}\text{(3b)}$$

Problem: These rules can derive false assertions (unsound)!

$$\frac{\{T\}\mathtt{x}:=0\{x=0\}}{\{T\}\textbf{if } \mathtt{x} <= 0 \textbf{ then } \mathtt{x}:=0 \textbf{ else } \textbf{skip}\{x=0\}}\text{(3a)}$$

Conditional Rule

$$\frac{\{A\}c_1\{B\} \qquad \{A\}c_2\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}} \text{(3)}$$

Conditional Rule

$$\frac{\{A\}c_1\{B\} \qquad \{A\}c_2\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}}(3)$$

Problem: This rule cannot derive some true assertions (incomplete)!

$$\frac{\dfrac{\vdots}{\{T\}\texttt{x := }0\{x \geq 0\}} \qquad \dfrac{?}{\{T\}\textbf{skip}\{x \geq 0\}}}{\{T\}\textbf{if } \texttt{x <= } 0 \textbf{ then } \texttt{x := } 0 \textbf{ else skip}\{x \geq 0\}}(3)$$

Conditional Rule

$$\frac{\{A \wedge b\}c_1\{B\} \qquad \{A \wedge \neg b\}c_2\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{B\}}(3)$$

Solves completeness problem without sacrificing soundness:

$$\frac{\{T \wedge x \leq 0\}\texttt{x:=}0\{x \geq 0\} \qquad \{T \wedge \neg(x \leq 0)\}\textbf{skip}\{x \geq 0\}}{\{T\}\textbf{if } \texttt{x <=} 0 \textbf{ then } \texttt{x:=}0 \textbf{ else } \textbf{skip}\{x \geq 0\}}(3)$$

Assignment Rule

$$\frac{}{\{A\} v := a \{\ ?\ \}} (4)$$

Assignment Rule

$$\frac{}{\{A\}v := a\{\ ?\ \}}\text{(4)}$$

Usage example:

$$\{x > 10\}\mathtt{x} := \mathtt{x} + 1\{x > 11\}$$

Assignment Rule

$$\frac{}{\{A\}v \,\texttt{:=}\, a\{B\}}(4)$$

where $B = A$ with all a's replaced with v?

Usage example:

$$\{x > 10\}\texttt{x := x + }1\{x > 11\}$$

Assignment Rule

$$\frac{}{\{A\}v \mathtt{:=} a\{B\}}(4)$$

where $B = A$ with all a's replaced with v?

Usage example:

$$\{x > 10\}\mathtt{x := x + 1}\{x > 11\}$$

$$\text{equivalent} \Updownarrow$$

$$\{x + 1 > 11\}\mathtt{x := x + 1}\{x > 11\}$$

Assignment Rule

$$\frac{}{\{\quad ? \quad\} v := a \{B\}} \text{(4)}$$

Usage example:

$$\{x > 10\}\texttt{x := x + 1}\{x > 11\}$$

$$\text{equivalent} \Updownarrow$$

$$\{x + 1 > 11\}\texttt{x := x + 1}\{x > 11\}$$

Assignment Rule

$$\frac{}{\{B[a/v]\}v\mathtt{:=}a\{B\}}(4)$$

Usage example:

$$\{x > 10\}\mathtt{x:=x+}1\{x > 11\}$$

$$\text{equivalent}\Big\Updownarrow$$

$$\{x + 1 > 11\}\mathtt{x:=x+}1\{x > 11\}$$

While Rule

$$\{A\}\textbf{while } b \textbf{ do } c\{B\}$$

While Rule

$$\frac{\{A\}\textbf{if}\ b\ \textbf{then}\ (c;\textbf{while}\ b\ \textbf{do}\ c)\ \textbf{else}\ \textbf{skip}\{B\}}{\{A\}\textbf{while}\ b\ \textbf{do}\ c\{B\}}\quad(5)$$

While Rule

$$\dfrac{\dfrac{\{A \wedge b\}c;\textbf{while}\ b\ \textbf{do}\ c\{B\} \qquad \{A \wedge \neg b\}\textbf{skip}\{B\}}{\{A\}\textbf{if}\ b\ \textbf{then}\ (c;\textbf{while}\ b\ \textbf{do}\ c)\ \textbf{else}\ \textbf{skip}\{B\}}(3)}{\{A\}\textbf{while}\ b\ \textbf{do}\ c\{B\}}(5)$$

While Rule

$$\dfrac{\dfrac{\dfrac{\{A \wedge b\}c\{C\} \quad \{C\}\textbf{while } b \textbf{ do } c\{B\}}{\{A \wedge b\}c;\textbf{while } b \textbf{ do } c\{B\}}(2) \qquad \{A \wedge \neg b\}\textbf{skip}\{B\}}{\{A\}\textbf{if } b \textbf{ then } (c;\textbf{while } b \textbf{ do } c) \textbf{ else skip}\{B\}}(3)}{\{A\}\textbf{while } b \textbf{ do } c\{B\}}(5)$$

While Rule

$$
\dfrac{\dfrac{\dfrac{\{A \wedge b\}c\{C\} \qquad \dfrac{\vdots}{\{C\}\textbf{while } b \textbf{ do } c\{B\}}\,\text{(5)}}{\{A \wedge b\}c;\textbf{while } b \textbf{ do } c\{B\}}\,\text{(2)} \qquad \{A \wedge \neg b\}\textbf{skip}\{B\}}{\{A\}\textbf{if } b \textbf{ then } (c;\textbf{while } b \textbf{ do } c) \textbf{ else skip}\{B\}}\,\text{(3)}}{\{A\}\textbf{while } b \textbf{ do } c\{B\}}\,\text{(5)}
$$

While Rule

$$\{A\}\textbf{while } b \textbf{ do } c\{\quad ? \quad\}$$

While Rule

$$\frac{\{A \land b\}c\{A\}}{\{A\}\textbf{while } b \textbf{ do } c\{\quad ? \quad\}} \text{(5)}$$

While Rule

$$\frac{\{A \land b\} c \{A\}}{\{A\} \textbf{while } b \textbf{ do } c \{\qquad A\}} \text{(5)}$$

While Rule

$$\frac{\{A \land b\}c\{A\}}{\{A\}\textbf{while } b \textbf{ do } c\{\neg b \land A\}} \quad (5)$$

While Rule

$$\frac{\{I \wedge b\}c\{I\}}{\{I\}\textbf{while } b \textbf{ do } c\{\neg b \wedge I\}} \text{(5)}$$

I is called a **loop invariant**

Rule of Consequence

Recall that we earlier needed a way to prove (derive) equivalence of assertions:

$$\{x > 10\}\texttt{x := x + }1\{x > 11\}$$

$$\text{equivalent}\ \Big\Updownarrow$$

$$\{x + 1 > 11\}\texttt{x := x + }1\{x > 11\}$$

Rule of Consequence

Recall that we earlier needed a way to prove (derive) equivalence of assertions:

$$\{x > 10\}\mathtt{x:=x+}1\{x > 11\}$$

$$\text{equivalent}\Big\Updownarrow$$

$$\{x + 1 > 11\}\mathtt{x:=x+}1\{x > 11\}$$

Rule of Consequence:

$$\frac{\{A'\}c\{B'\}}{\{A\}c\{B\}} \quad (6)$$

Rule of Consequence

Recall that we earlier needed a way to prove (derive) equivalence of assertions:

$$\{x > 10\}\texttt{x:=x+1}\{x > 11\}$$

$$\text{equivalent}\Big\Updownarrow$$

$$\{x + 1 > 11\}\texttt{x:=x+1}\{x > 11\}$$

Rule of Consequence:

$$\frac{\models A \Rightarrow A' \qquad \{A'\}c\{B'\}}{\{A\}c\{B\}} \ (6)$$

$\models$ with nothing to the left means implication is **universally true** (i.e., not merely true in this program or loop)

- $\models A \Rightarrow A'$ $\longleftarrow$Assumptions may be safely **weakened**

Rule of Consequence

Recall that we earlier needed a way to prove (derive) equivalence of assertions:

$$\{x > 10\}\mathtt{x:=x+1}\{x > 11\}$$

$$\text{equivalent}\Big\Updownarrow$$

$$\{x + 1 > 11\}\mathtt{x:=x+1}\{x > 11\}$$

Rule of Consequence:

$$\frac{\models A \Rightarrow A' \qquad \{A'\}c\{B'\} \qquad \models B' \Rightarrow B}{\{A\}c\{B\}} (6)$$

$\models$ with nothing to the left means implication is **universally true** (i.e., not merely true in this program or loop)

- $\models A \Rightarrow A'$ $\longleftarrow$Assumptions may be safely **weakened**
- $\models B' \Rightarrow B$ $\longleftarrow$Conclusions (goals) may be safely **strengthened**

Rule of Consequence Example

$$\frac{}{\{x > 10\}\mathtt{x := x + 1}\{x > 11\}}$$

Rule of Consequence Example

$$\dfrac{\dfrac{\vdots}{\models x > 10 \Rightarrow x + 1 > 11} \quad \dfrac{}{\{x+1 > 11\}\texttt{x:=x+1}\{x > 11\}}(4) \quad \dfrac{\vdots}{\models x > 11 \Rightarrow x > 11}}{\{x > 10\}\texttt{x:=x+1}\{x > 11\}}(6)$$

Rule of Consequence Example

$$\dfrac{\dfrac{\vdots}{\models x > 10 \Rightarrow x + 1 > 11} \quad \dfrac{}{\{x + 1 > 11\}\mathtt{x:=x+1}\{x > 11\}}(4) \quad \dfrac{\vdots}{\models x > 11 \Rightarrow x > 11}}{\{x > 10\}\mathtt{x:=x+1}\{x > 11\}}(6)$$

When you write axiomatic derivations in this class:

- You are **not** required to write out the derivations of consequence premises ($\models A$).
- I assume those are derivable using the laws of propositional logic and integer arithmetic.
- But make sure your implications $X \Rightarrow Y$ are **universally true**!

Axiomatic Semantics of SIMPL

$$\frac{}{\{A\}\mathbf{skip}\{A\}}\text{(1)}$$

$$\frac{}{\{B[a/v]\}v := a\{B\}}\text{(4)}$$

$$\frac{\{A\}c_1\{C\} \quad \{C\}c_2\{B\}}{\{A\}c_1\,;c_2\{B\}}\text{(2)}$$

$$\frac{\{I \wedge b\}c\{I\}}{\{I\}\mathbf{while}\ b\ \mathbf{do}\ c\{\neg b \wedge I\}}\text{(5)}$$

$$\frac{\{A \wedge b\}c_1\{B\} \quad \{A \wedge \neg b\}c_2\{B\}}{\{A\}\mathbf{if}\ b\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2\{B\}}\text{(3)}$$

$$\frac{\models A \Rightarrow A' \quad \{A'\}c\{B'\} \quad \models B' \Rightarrow B}{\{A\}c\{B\}}\text{(6)}$$