Program Semantics

17-654/17-765
Analysis of Software Artifacts
Jonathan Aldrich

Why Semantics?

- Semantics describe formally what a program means
  - Typically, how the program executes
- Framework for analysis
  - Precise definitions
  - Proofs of correctness
- Semantics in practice
  - Difficult to define for full languages
    - But see Standard ML!
  - Very useful for thinking about how analysis applies to the "core" of a language
    - Extension to full language is assumed to be easy
      - **Sometimes true, sometimes not!**

Forms of Program Semantics

- Big-Step Reduction Semantics
  - Shows result of program
  - Depends on environment *E* : **Var → Value**
  - Forms: $E \vdash a \Downarrow n \qquad E \vdash S \Downarrow E'$
    - In environment E, expression a reduces to number n
    - In environment E, statement S executes to a new environment E'
  - The primary semantics used in this course
- Small-Step Reduction Semantics
  - Shows step-by-step execution of program
  - Form: $(E, e) \mapsto (E', e')$
    - In environment E, expression e steps to expression e' and produces a new environment E'
- Denotational Semantics
  - Form: $[\![P]\!] = O$
    - The meaning of program P is mathematical object O

The WHILE Language

- A simple procedural language with:
  - assignment
  - statement sequencing
  - conditionals
  - while loops
- Used in early papers (e.g. Hoare 69) as as a "sandbox" for thinking about program semantics
- We will use it to illustrate several different kinds of analysis

WHILE Syntax

- Categories of syntax
  - S ∈ **Stmt** statements
  - a ∈ **AExp** arithmetic expressions
  - x,y ∈ **Var** variables
  - n ∈ **Num** number literals
  - P ∈ **BExp** boolean expressions
- Syntax
  - $S ::= x := a \mid \text{skip} \mid S_1;\ S_2$
    $\mid\ \text{if } P \text{ then } S_1 \text{ else } S_2 \mid \text{while } P \text{ do } S$
  - $a ::= x \mid n \mid a_1\ op_a\ a_2$
  - $op_a ::= + \mid - \mid * \mid / \mid \ldots$
  - $P ::= \text{true} \mid \text{false} \mid \text{not } P \mid P_1\ op_b\ P_2 \mid a_1\ op_r\ a_2$
  - $op_b ::= \text{and} \mid \text{or} \mid \ldots$
  - $op_r ::= < \mid \leq \mid = \mid > \mid \geq \mid \ldots$

Example WHILE Program

```
y := x;
z := 1;
while y>1 do
   z := z * y;
   y := y – 1
```

Computes the factorial function, with the input in x and the output in z

Big-Step Semantics

- We use big-step to show expression evaluation
- Inference rule format
  - Premises above the line
  - Conclusion below the line
  - Read, "If premises, then conclusion"

$$\frac{premise1 \quad premise2}{conclusion}$$

- Example: operators
  - If expression a evaluates to a number n
  - And expression a' evaluates to a number n'
  - Then the expression a + a' evaluates to n **+** n'
    - In the rule, we distinguish the textual + operator from the mathematical **+** operator (for which we use boldface font)

$$\frac{E \vdash a \Downarrow n \qquad E \vdash a' \Downarrow n'}{E \vdash a + a' \Downarrow n \mathbf{+} n'}$$

WHILE Expression Big-Step Semantics

- Values evaluate to themselves
  - *n*, *true*, *false*
- Variables x evaluate to the value in the environment
  - $E\{x\}$
- Operators evaluate according to mathematical operators
  - +, -, *, /, *not*, *and*, *or*, <, ≤, =, …
  - **math::** indicates mathematical operators
  - **boldface** indicates mathematical values
  - *italics* indicates program text

$$E \vdash n \Downarrow n$$

$$\frac{E\{x\} = n}{E \vdash x \Downarrow n}$$

$$\frac{E \vdash a \Downarrow n \qquad E \vdash a' \Downarrow n'}{E \vdash a \ \mathit{op} \ a' \Downarrow n \ \mathbf{op} \ n'}$$

$$E \vdash \mathit{true} \Downarrow \mathbf{true}$$

$$E \vdash \mathit{false} \Downarrow \mathbf{false}$$

$$\frac{E \vdash P \Downarrow b}{E \vdash !P \Downarrow \mathbf{!}b}$$

$$\frac{E \vdash P \Downarrow b \qquad E \vdash P' \Downarrow b'}{E \vdash P \ \mathit{op} \ P' \Downarrow b \ \mathbf{op} \ b'}$$

Applying Semantic Rules

- A tree of inference rules forms a derivation
  - Rules at top are axioms; they have no premises
- Example:
  - $E = \{x \mapsto 3, y \mapsto 5\}$
  - $P = x + 3 > y$

$$\cfrac{\cfrac{\cfrac{E\{x\} = \mathbf{3}}{E \vdash x \Downarrow \mathbf{3}} \quad \cfrac{}{E \vdash 3 \Downarrow \mathbf{3}}}{E \vdash x + 3 \Downarrow \mathbf{6}} \quad \cfrac{E\{y\} = \mathbf{5}}{E \vdash y \Downarrow \mathbf{5}}}{E \vdash x + 3 > y \Downarrow \mathbf{true}}$$

Practice: Applying Semantic Rules

- Example:
  - $E = \{x \mapsto 3, y \mapsto 5\}$
  - $P = x > y$ *and true*

$$E \vdash n \Downarrow n$$

$$\frac{E\{x\} = n}{E \vdash x \Downarrow n}$$

$$\frac{E \vdash a \Downarrow n \quad E \vdash a' \Downarrow n'}{E \vdash a \ op \ a' \Downarrow n \ \mathbf{op} \ n'}$$

$$E \vdash \mathit{true} \Downarrow \mathbf{true}$$

$$E \vdash \mathit{false} \Downarrow \mathbf{false}$$

$$\frac{E \vdash P \Downarrow b}{E \vdash !P \Downarrow \mathbf{!}b}$$

$$\frac{E \vdash P \Downarrow b \quad E \vdash P' \Downarrow b'}{E \vdash P \ op \ P' \Downarrow b \ \mathbf{op} \ b'}$$

WHILE Statement Semantics

- In a statement semantics, we are not looking for a resulting value, but for updates to variables in the environment
- Example: assignment
  - If the right-hand side evaluates to a value n
  - Then the assignment generates a new environment where x maps to n

$$\frac{E \vdash a \Downarrow n}{E \vdash x{:=}a \Downarrow E\{x \mapsto n\}}$$

WHILE Statement Semantics

- sequences execute statements in order
- *skip* does not affect the environment
- *if* executes either first or second statement, depending on P
- *while* executes the body followed by the loop if P is **true**

$$\frac{E \vdash a \Downarrow n}{E \vdash x{:=}a \Downarrow E\{x \mapsto n\}}$$

$$\frac{E \vdash S_1 \Downarrow E' \quad E' \vdash S_2 \Downarrow E''}{E \vdash S_1; S_2 \Downarrow E''}$$

$$\frac{}{E \vdash \mathit{skip} \Downarrow E}$$

$$\frac{E \vdash P \Downarrow \mathbf{true} \quad E \vdash S_1 \Downarrow E'}{E \vdash \mathit{if}\ P\ \mathit{then}\ S_1\ \mathit{else}\ S_2 \Downarrow E'}$$

$$\frac{E \vdash P \Downarrow \mathbf{false} \quad E \vdash S_2 \Downarrow E'}{E \vdash \mathit{if}\ P\ \mathit{then}\ S_1\ \mathit{else}\ S_2 \Downarrow E'}$$

$$\frac{E \vdash P \Downarrow \mathbf{true} \quad E \vdash S;\ \mathit{while}\ P\ \mathit{do}\ S \Downarrow E'}{E \vdash \mathit{while}\ P\ \mathit{do}\ S \Downarrow E'}$$

$$\frac{E \vdash P \Downarrow \mathbf{false}}{E \vdash \mathit{while}\ P\ \mathit{do}\ S \Downarrow E}$$

WHILE Execution Example

(1)	$\{\} \vdash 5 \Downarrow 5$	by rule eval-num
(2)	$\{\} \vdash x := 5 \Downarrow \{x \mapsto 5\}$	by rule ex-assign on (1)
(3)	$\{x \mapsto 5\} \vdash x \Downarrow 5$	by rule eval-var
(4)	$\{x \mapsto 5\} \vdash 3 \Downarrow 3$	by rule eval-num
(5)	$\{x \mapsto 5\} \vdash x > 3 \Downarrow \mathbf{true}$	by rule eval-rop on (3),(4)
(6)	$\{x \mapsto 5\} \vdash 1 \Downarrow 1$	by rule eval-num
(7)	$\{x \mapsto 5\} \vdash y := 1 \Downarrow \{x \mapsto 5, x \mapsto 1\}$	by rule ex-assign on (6)
(8)	$\{x \mapsto 5\} \vdash \mathit{if}\ x > 3\ \mathit{then}\ y := 1\ \mathit{else}\ y := 5$ $\Downarrow \{x \mapsto 5, x \mapsto 1\}$	by rule ex-iftrue on (5),(7)
(9)	$\{\} \vdash x := 5;\ \mathit{if}\ x > 3\ \mathit{then}\ y := 1\ \mathit{else}\ y := 5$ $\Downarrow \{x \mapsto 5, x \mapsto 1\}$	by rule ex-seq on (2),(8)