Small-step Operational Semantics

CS 152 (Spring 2020)

Harvard University

Tuesday, February 4, 2020

Abstract Syntax

Abstract Syntax

$$\begin{aligned} x, y, z &\in \textbf{Var} \\ n, m &\in \textbf{Int} \\ e &\in \textbf{Exp} \end{aligned}$$

Abstract Syntax

$$x, y, z \in \textbf{Var}$$

Var is the set of program variables (e.g., foo, bar, baz, i, etc.).

Abstract Syntax

$$n, m \in \textbf{Int}$$

Int is the set of constant integers (e.g., 42, -40, 7).

Abstract Syntax

$$e \in \mathbf{Exp}$$

Exp is the domain of expressions, which we specify using a BNF (Backus-Naur Form) grammar.

Expressions

$$
\begin{aligned}
e ::= \; & x \\
\mid \; & n \\
\mid \; & e_1 + e_2 \\
\mid \; & e_1 \times e_2 \\
\mid \; & x := e_1; e_2
\end{aligned}
$$

Assignment

$$x := e_1; e_2$$

Informally, the expression $x := e_1; e_2$ means that x is assigned the value of e_1 before evaluating e_2. The result of the entire expression is that of e_2.

Abstract Syntax Tree

$1 + 2 \times 3$

Abstract Syntax Tree

$1 + 2 \times 3$

$1 + (2 \times 3)$

$(1 + 2) \times 3$

Abstract Syntax Tree

$1 + 2 \times 3$

```
    +
   / \
  1   *
     / \
    2   3
```

$1 + (2 \times 3)$ $\qquad$ $(1 + 2) \times 3$

Abstract Syntax Tree

$$1 + 2 \times 3$$

```
     +
    / \
   1   *
      / \
     2   3
```

$$1 + (2 \times 3)$$

```
     *
    / \
   +   3
  / \
 1   2
```

$$(1 + 2) \times 3$$

Semantics

Styles of Semantics

Operational Semantics
Denotational Semantics
Axiomatic Semantics
Algebraic Semantics

Operational Semantics

Small-Step

Large-Step

Small-Step Operational Semantics

Configuration (State of Machine)

$$\begin{aligned} \textbf{Config} &= \textbf{Exp} \times \textbf{Store} \\ \textbf{Store} &= \textbf{Var} \to \textbf{Int} \end{aligned}$$

Step Relation

$$\longrightarrow \subseteq \mathbf{Config} \times \mathbf{Config}$$

Notation

$$(< e_1, \sigma_1 >, < e_2, \sigma_2 >) \in \longrightarrow$$

$$< e_1, \sigma_1 > \longrightarrow < e_2, \sigma_2 >$$

Rules Var

$$\textsc{Var}\ \frac{\text{where } n = \sigma(x)}{< x, \sigma > \longrightarrow < n, \sigma >}$$

Rules Add

$$\textsc{LAdd}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 + e_2, \sigma > \longrightarrow < e_1' + e_2, \sigma' >}$$

$$\textsc{RAdd}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n + e_2, \sigma > \longrightarrow < n + e_2', \sigma' >}$$

$$\textsc{Add}\ \frac{\text{where } p \text{ is the sum of } n \text{ and } m}{< n + m, \sigma > \longrightarrow < p, \sigma >}$$

Rules Mul

$$\textsc{LMul}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 \times e_2, \sigma > \longrightarrow < e_1' \times e_2, \sigma' >}$$

$$\textsc{RMul}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n \times e_2, \sigma > \longrightarrow < n \times e_2', \sigma' >}$$

$$\textsc{Mul}\ \frac{\text{where } p \text{ is the product of } n \text{ and } m}{< n \times m, \sigma > \longrightarrow < p, \sigma >}$$

Rules Asg

$$\textsc{Asg1}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< x := e_1; e_2, \sigma > \longrightarrow < x := e_1'; e_2, \sigma' >}$$

$$\textsc{Asg}\ \frac{}{< x := n; e_2, \sigma > \longrightarrow < e_2, \sigma[x \mapsto n] >}$$

Store Update $\sigma[x \mapsto n]$

If f is the function $\sigma[x \mapsto n]$, then

$$f(y) = \begin{cases} n & \text{if } y = x \\ \sigma(y) & \text{otherwise} \end{cases}$$

Semantic Rules (1/2)

$$\textsc{LAdd}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 + e_2, \sigma > \longrightarrow < e_1' + e_2, \sigma' >}$$

$$\textsc{RAdd}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n + e_2, \sigma > \longrightarrow < n + e_2', \sigma' >}$$

$$\textsc{Add}\ \frac{}{< n + m, \sigma > \longrightarrow < p, \sigma >}\ \text{where } p \text{ is the sum of } n \text{ and } m$$

$$\textsc{LMul}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 \times e_2, \sigma > \longrightarrow < e_1' \times e_2, \sigma' >}$$

$$\textsc{RMul}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n \times e_2, \sigma > \longrightarrow < n \times e_2', \sigma' >}$$

$$\textsc{Mul}\ \frac{}{< n \times m, \sigma > \longrightarrow < p, \sigma >}\ \text{where } p \text{ is the product of } n \text{ and } m$$

Semantic Rules (2/2)

$$\textsc{Var}\ \frac{}{< x, \sigma > \longrightarrow < n, \sigma >} \text{ where } n = \sigma(x)$$

$$\textsc{Asg1}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< x := e_1; e_2, \sigma > \longrightarrow < x := e_1'; e_2, \sigma' >}$$

$$\textsc{Asg}\ \frac{}{< x := n; e_2, \sigma > \longrightarrow < e_2, \sigma[x \mapsto n] >}$$

Semantic Rules Recap

- LAdd, RAdd, Add
- LMul, RMul, Mul
- Var
- Asg1, Asg

Semantic **Congruence** Rules Recap

- **LAdd**, **RAdd**, ADD
- **LMul**, **RMul**, MUL
- VAR
- **Asg1**, ASG

Semantic **Computation** Rules Recap

- LAdd, RAdd, **Add**
- LMul, RMul, **Mul**
- **Var**
- Asg1, **Asg**

Semantic Rules (Computation)

$$\textsc{Var}\ \frac{}{< x, \sigma > \longrightarrow < n, \sigma >} \text{ where } n = \sigma(x)$$

$$\textsc{Add}\ \frac{}{< n + m, \sigma > \longrightarrow < p, \sigma >} \text{ where } p \text{ is the sum of } n \text{ and } m$$

$$\textsc{Mul}\ \frac{}{< n \times m, \sigma > \longrightarrow < p, \sigma >} \text{ where } p \text{ is the product of } n \text{ and } m$$

$$\textsc{Asg}\ \frac{}{< x := n; e_2, \sigma > \longrightarrow < e_2, \sigma[x \mapsto n] >}$$

Semantic Rules (Congruence)

$$\textsc{LAdd}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 + e_2, \sigma > \longrightarrow < e_1' + e_2, \sigma' >}$$

$$\textsc{RAdd}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n + e_2, \sigma > \longrightarrow < n + e_2', \sigma' >}$$

$$\textsc{LMul}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< e_1 \times e_2, \sigma > \longrightarrow < e_1' \times e_2, \sigma' >}$$

$$\textsc{RMul}\ \frac{< e_2, \sigma > \longrightarrow < e_2', \sigma' >}{< n \times e_2, \sigma > \longrightarrow < n \times e_2', \sigma' >}$$

$$\textsc{Asg1}\ \frac{< e_1, \sigma > \longrightarrow < e_1', \sigma' >}{< x := e_1; e_2, \sigma > \longrightarrow < x := e_1'; e_2, \sigma' >}$$

Using the Semantic Rules

Using the Semantic Rules

$$< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma >$$

where $\sigma(\mathsf{foo}) = 4$ and $\sigma(\mathsf{bar}) = 3$

Using the Semantic Rules

$$\textsc{LMul}\ \frac{< \mathsf{foo} + 2, \sigma > \longrightarrow < e_1', \sigma >}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow < e_1' \times (\mathsf{bar} + 1), \sigma >}$$

Using the Semantic Rules

$$\textsc{LAdd}\ \frac{< \mathsf{foo}, \sigma > \longrightarrow < \mathsf{e}_1'', \sigma >}{< \mathsf{foo} + 2, \sigma > \longrightarrow < \mathsf{e}_1'' + 2, \sigma >}$$

Using the Semantic Rules

$$\textsc{Var}\ \frac{}{< \mathsf{foo}, \sigma > \longrightarrow < \mathsf{4}, \sigma >}$$

Using the Semantic Rules

$$
\textsc{LMul}\ \dfrac{\textsc{LAdd}\ \dfrac{\textsc{Var}\ \dfrac{}{< \mathsf{foo}, \sigma > \longrightarrow < 4, \sigma >}}{< \mathsf{foo} + 2, \sigma > \longrightarrow < 4 + 2, \sigma >}}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow < (4 + 2) \times (\mathsf{bar} + 1), \sigma >}
$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$\textsc{LMul}\ \frac{}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow}$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$\cfrac{\textsc{LAdd}\ \cfrac{}{<\mathsf{foo} + 2, \sigma> \longrightarrow}}{<(\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma> \longrightarrow}\ \textsc{LMul}$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$
\textsc{LMul}\ \dfrac{\textsc{LAdd}\ \dfrac{\textsc{Var}\ \dfrac{}{< \mathsf{foo}, \sigma > \longrightarrow}}{< \mathsf{foo} + 2, \sigma > \longrightarrow}}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow}
$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$
\textsc{LMul}\ \dfrac{\textsc{LAdd}\ \dfrac{\textsc{Var}\ \dfrac{}{< \mathsf{foo}, \sigma > \longrightarrow\ < 4, \sigma >}}{< \mathsf{foo} + 2, \sigma > \longrightarrow}}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow}
$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$\textsc{LMul}\ \dfrac{\textsc{LAdd}\ \dfrac{\textsc{Var}\ \dfrac{}{<\mathsf{foo}, \sigma> \longrightarrow\ <4, \sigma>}}{<\mathsf{foo} + 2, \sigma> \longrightarrow\ <4 + 2, \sigma>}}{<(\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma> \longrightarrow}$$

Writing Derivation: Bottom Left, Go Up, Top Right, Go Down

$$\textsc{LMul}\ \dfrac{\textsc{LAdd}\ \dfrac{\textsc{Var}\ \dfrac{}{< \mathsf{foo}, \sigma > \longrightarrow\ < 4, \sigma >}}{< \mathsf{foo} + 2, \sigma > \longrightarrow\ < 4 + 2, \sigma >}}{< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow\ < (4 + 2) \times (\mathsf{bar} + 1), \sigma >}$$

Using the Semantic Rules (Next Step)

$$\textsc{LMul}\ \dfrac{\textsc{Add}\ \dfrac{}{< 4+2, \sigma > \longrightarrow < 6, \sigma >}}{< (4+2) \times (\mathsf{bar}+1), \sigma > \longrightarrow < 6 \times (\mathsf{bar}+1), \sigma >}$$

Using the Semantic Rules (All Steps)

$$
\begin{aligned}
&< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \\
\longrightarrow &< (4 + 2) \times (\mathsf{bar} + 1), \sigma > \\
\longrightarrow &< 6 \times (\mathsf{bar} + 1), \sigma > \\
\longrightarrow &< 6 \times (3 + 1), \sigma > \\
\longrightarrow &< 6 \times 4, \sigma > \\
\longrightarrow &< 24, \sigma >
\end{aligned}
$$

Using the Semantic Rules (Multi-Steps)

$$< (\mathsf{foo} + 2) \times (\mathsf{bar} + 1), \sigma > \longrightarrow^* < 24, \sigma >.$$

Expressing Program Properties

Progress

$$\forall e \in \textbf{Exp}.\ \forall \sigma \in \textbf{Store}.$$
$$\text{either } e \in \textbf{Int} \text{ or } \exists e', \sigma'.\ \ < e, \sigma > \longrightarrow < e', \sigma' >$$

Termination

$$\forall e \in \mathbf{Exp}.\ \forall \sigma_0 \in \mathbf{Store}.\ \exists \sigma \in \mathbf{Store}.\ \exists n \in \mathbf{Int}.$$
$$< e, \sigma_0 > \longrightarrow^* < n, \sigma >$$

Deterministic Result

$$\forall e \in \textbf{Exp}.\ \forall \sigma_0, \sigma, \sigma' \in \textbf{Store}.\ \forall n, n' \in \textbf{Int}.$$
$$\text{if } \ < e, \sigma_0 > \longrightarrow^* < n, \sigma > \ \text{ and}$$
$$< e, \sigma_0 > \longrightarrow^* < n', \sigma' > \ \text{ then}$$
$$n = n' \text{ and } \sigma = \sigma'.$$

Break

- Is foo $:=$ 1 a valid expression?
- What about foo $:=$ 1; bar $:=$ foo; bar?
  What is the unambiguous abstract syntax?
- What about bar $:=$ (foo $:=$ 1; foo); bar $+$ 1?
- What is the meaning of these expressions?