Small-step Operational Semantics

CS 152 (Spring 2024)

Harvard University

Thursday, January 25, 2024

Today, we learn to

- define and use abstract syntax
- define and use small-step operational semantics
- express program properties
- using Math & using Coq

Abstract Syntax

Abstract Syntax

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

Abstract Syntax

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

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.

Syntax Break

- Is $\mathsf{foo} := 1$ a valid expression?
- What about $\mathsf{foo} := 1; \mathsf{bar} := \mathsf{foo}; \mathsf{bar}$?
  What is the unambiguous abstract syntax?
- What about $\mathsf{bar} := (\mathsf{foo} := 1; \mathsf{foo}); \mathsf{bar} + 1$?

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)

$$\mathbf{Config} = \mathbf{Exp} \times \mathbf{Store}$$

$$\mathbf{Store} = \mathbf{Var} \to \mathbf{Int}$$

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

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

Rules Add

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

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

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

Interlude: Shape of a Rule

$$\textsc{Name}\ \frac{\mathit{premise}_1 \quad \ldots \quad \mathit{premise}_k}{\mathit{conclusion}}\ \text{where } \mathit{side\ conditions}$$

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)

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

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

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

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

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

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

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

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

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

$$\textsc{LAdd}\ \frac{< \mathsf{foo}, \sigma > \longrightarrow < e_1'', \sigma >}{< \mathsf{foo} + 2, \sigma > \longrightarrow < e_1'' + 2, \sigma >}$$

Using the Semantic Rules

$$\textsc{Var}\ \frac{}{< \text{foo}, \sigma > \longrightarrow < 4, \sigma >}\ \text{where } \sigma(\text{foo}) = 4$$

Using the Semantic Rules

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

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

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

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

$$\textsc{LMul}\ \frac{\textsc{LAdd}\ \frac{\textsc{Var}\ \frac{\text{where } \sigma(\text{foo}) = 4}{< \text{foo}, \sigma > \longrightarrow}}{< \text{foo} + 2, \sigma > \longrightarrow}}{< (\text{foo} + 2) \times (\text{bar} + 1), \sigma > \longrightarrow}$$

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

$$\textsc{LMul}\ \frac{\textsc{LAdd}\ \frac{\textsc{Var}\ \frac{\text{where } \sigma(\text{foo}) = 4}{< \text{foo}, \sigma > \longrightarrow\ < 4, \sigma >}}{< \text{foo} + 2, \sigma > \longrightarrow}}{< (\text{foo} + 2) \times (\text{bar} + 1), \sigma > \longrightarrow}$$

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

$$\textsc{LMul}\ \frac{\textsc{LAdd}\ \frac{\textsc{Var}\ \frac{\text{where } \sigma(\text{foo}) = 4}{< \text{foo}, \sigma > \longrightarrow\ < 4, \sigma >}}{< \text{foo} + 2, \sigma > \longrightarrow\ < 4 + 2, \sigma >}}{< (\text{foo} + 2) \times (\text{bar} + 1), \sigma > \longrightarrow}$$

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

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

Using the Semantic Rules (Next Step)

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

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)

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

Multi-Step: Reflexive Transitive Closure of Step

$$\textsc{Refl}\ \frac{}{c \longrightarrow^* c} \qquad \textsc{Trans}\ \frac{c \longrightarrow c' \quad c' \longrightarrow^* c''}{c \longrightarrow^* c''}$$

Expressing Program Properties

Progress

$$\forall e \in \textbf{Exp}.\ \forall \sigma \in \textbf{Store}.$$

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

Termination

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

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

What if

- How would we add division to our language?
- How do we handle division by zero?
- Would the properties defined earlier still hold?