

Small-step Operational Semantics

CS 152 (Spring 2025)

Harvard University

Thursday, January 30, 2025

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

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

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

$e \in \mathbf{Exp}$

Abstract Syntax

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

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

Abstract Syntax

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

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

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

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 `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`?

Semantics

Styles of Semantics

Operational Semantics

Denotational Semantics

Axiomatic Semantics

Algebraic Semantics

Styles of Semantics

Operational Semantics : execution

Denotational Semantics : mathematical structures

Axiomatic Semantics : predicate transformers

Algebraic Semantics : laws

Operational Semantics

Small-Step

Large-Step

Small-Step Operational Semantics

Configuration (State of Machine)

Config = Exp \times Store

Store = Var \rightarrow Int

Step Relation

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

Notation

$$(\langle e_1, \sigma_1 \rangle, \langle e_2, \sigma_2 \rangle) \in \longrightarrow$$

$$\langle e_1, \sigma_1 \rangle \longrightarrow \langle e_2, \sigma_2 \rangle$$

Rules Var

$$\text{VAR} \frac{}{\langle x, \sigma \rangle \longrightarrow \langle n, \sigma \rangle} \text{ where } n = \sigma(x)$$

Rules Add

$$\text{LADD} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 + e_2, \sigma \rangle \longrightarrow \langle e'_1 + e_2, \sigma' \rangle}$$

$$\text{RADD} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n + e_2, \sigma \rangle \longrightarrow \langle n + e'_2, \sigma' \rangle}$$

$$\text{ADD} \frac{\text{where } p \text{ is the sum of } n \text{ and } m}{\langle n + m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle}$$

Interlude: Shape of a Rule

NAME $\frac{\textit{premise}_1 \dots \textit{premise}_k}{\textit{conclusion}}$ where *side conditions*

Rules Mul

$$\text{LMUL} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 \times e_2, \sigma \rangle \longrightarrow \langle e'_1 \times e_2, \sigma' \rangle}$$

$$\text{RMUL} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n \times e_2, \sigma \rangle \longrightarrow \langle n \times e'_2, \sigma' \rangle}$$

$$\text{MUL} \frac{\text{where } p \text{ is the product of } n \text{ and } m}{\langle n \times m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle}$$

Rules Asg

$$\text{ASG1} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle x := e_1; e_2, \sigma \rangle \longrightarrow \langle x := e'_1; e_2, \sigma' \rangle}$$

$$\text{ASG} \frac{}{\langle x := n; e_2, \sigma \rangle \longrightarrow \langle e_2, \sigma[x \mapsto n] \rangle}$$

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)

$$\text{LADD} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 + e_2, \sigma \rangle \longrightarrow \langle e'_1 + e_2, \sigma' \rangle}$$

$$\text{RADD} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n + e_2, \sigma \rangle \longrightarrow \langle n + e'_2, \sigma' \rangle}$$

$$\text{ADD} \frac{}{\langle n + m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle} \text{ where } p \text{ is the sum of } n \text{ and } m$$

$$\text{LMUL} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 \times e_2, \sigma \rangle \longrightarrow \langle e'_1 \times e_2, \sigma' \rangle}$$

$$\text{RMUL} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n \times e_2, \sigma \rangle \longrightarrow \langle n \times e'_2, \sigma' \rangle}$$

$$\text{MUL} \frac{}{\langle n \times m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle} \text{ where } p \text{ is the product of } n \text{ and } m$$

Semantic Rules (2/2)

$$\text{VAR} \frac{}{\langle x, \sigma \rangle \longrightarrow \langle n, \sigma \rangle} \text{ where } n = \sigma(x)$$

$$\text{ASG1} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle x := e_1; e_2, \sigma \rangle \longrightarrow \langle x := e'_1; e_2, \sigma' \rangle}$$

$$\text{ASG} \frac{}{\langle x := n; e_2, \sigma \rangle \longrightarrow \langle e_2, \sigma[x \mapsto n] \rangle}$$

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)

$$\text{VAR} \frac{}{\langle x, \sigma \rangle \longrightarrow \langle n, \sigma \rangle} \text{ where } n = \sigma(x)$$

$$\text{ADD} \frac{}{\langle n + m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle} \text{ where } p \text{ is the sum of } n \text{ and } m$$

$$\text{MUL} \frac{}{\langle n \times m, \sigma \rangle \longrightarrow \langle p, \sigma \rangle} \text{ where } p \text{ is the product of } n \text{ and } m$$

$$\text{ASG} \frac{}{\langle x := n; e_2, \sigma \rangle \longrightarrow \langle e_2, \sigma[x \mapsto n] \rangle}$$

Semantic Rules (Congruence)

$$\text{LADD} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 + e_2, \sigma \rangle \longrightarrow \langle e'_1 + e_2, \sigma' \rangle}$$

$$\text{RADD} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n + e_2, \sigma \rangle \longrightarrow \langle n + e'_2, \sigma' \rangle}$$

$$\text{LMUL} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle e_1 \times e_2, \sigma \rangle \longrightarrow \langle e'_1 \times e_2, \sigma' \rangle}$$

$$\text{RMUL} \frac{\langle e_2, \sigma \rangle \longrightarrow \langle e'_2, \sigma' \rangle}{\langle n \times e_2, \sigma \rangle \longrightarrow \langle n \times e'_2, \sigma' \rangle}$$

$$\text{ASG1} \frac{\langle e_1, \sigma \rangle \longrightarrow \langle e'_1, \sigma' \rangle}{\langle x := e_1; e_2, \sigma \rangle \longrightarrow \langle x := e'_1; e_2, \sigma' \rangle}$$

Using the Semantic Rules

Using the Semantic Rules

$\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle$

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

Using the Semantic Rules

$$\text{LM}_{\text{UL}} \frac{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle e'_1, \sigma \rangle}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow \langle e'_1 \times (\text{bar} + 1), \sigma \rangle}$$

Using the Semantic Rules

$$\text{LADD} \frac{\langle \text{foo}, \sigma \rangle \longrightarrow \langle e_1'', \sigma \rangle}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle e_1'' + 2, \sigma \rangle}$$

Using the Semantic Rules

$$\text{VAR} \frac{}{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle} \text{ where } \sigma(\text{foo}) = 4$$

Using the Semantic Rules

$$\begin{array}{c} \text{VAR} \frac{\text{where } \sigma(\text{foo}) = 4}{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle} \\ \text{LADD} \frac{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle 4 + 2, \sigma \rangle} \\ \text{LMUL} \frac{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle 4 + 2, \sigma \rangle}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow \langle (4 + 2) \times (\text{bar} + 1), \sigma \rangle} \end{array}$$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

LMUL $\frac{\quad}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \rightarrow}$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

$$\text{LMUL} \frac{\text{LADD} \frac{}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow}}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow}$$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

$$\text{LMUL} \frac{\text{LADD} \frac{\text{VAR} \frac{\text{where } \sigma(\text{foo}) = 4}{\langle \text{foo}, \sigma \rangle \longrightarrow}}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow}}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow}$$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

$$\begin{array}{l} \text{VAR} \frac{\text{where } \sigma(\text{foo}) = 4}{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle} \\ \text{LADD} \frac{\quad}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow} \\ \text{LMUL} \frac{\quad}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow} \end{array}$$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

$$\begin{array}{l} \text{VAR} \frac{\text{where } \sigma(\text{foo}) = 4}{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle} \\ \text{LADD} \frac{\quad}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle 4 + 2, \sigma \rangle} \\ \text{LMUL} \frac{\quad}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow} \end{array}$$

Writing Derivation:

Bottom Left, Go Up, Top Right, Go Down

$$\begin{array}{c} \text{VAR} \frac{\text{where } \sigma(\text{foo}) = 4}{\langle \text{foo}, \sigma \rangle \longrightarrow \langle 4, \sigma \rangle} \\ \text{LADD} \frac{\quad}{\langle \text{foo} + 2, \sigma \rangle \longrightarrow \langle 4 + 2, \sigma \rangle} \\ \text{LMUL} \frac{\quad}{\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow \langle (4 + 2) \times (\text{bar} + 1), \sigma \rangle} \end{array}$$

Using the Semantic Rules (Next Step)

$$\text{LMUL} \frac{\text{ADD} \frac{}{\langle 4 + 2, \sigma \rangle \rightarrow \langle 6, \sigma \rangle}}{\langle (4 + 2) \times (\text{bar} + 1), \sigma \rangle \rightarrow \langle 6 \times (\text{bar} + 1), \sigma \rangle}$$

Using the Semantic Rules (All Steps)

$$\begin{aligned} & \langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \\ \longrightarrow & \langle (4 + 2) \times (\text{bar} + 1), \sigma \rangle \\ \longrightarrow & \langle 6 \times (\text{bar} + 1), \sigma \rangle \\ \longrightarrow & \langle 6 \times (3 + 1), \sigma \rangle \\ \longrightarrow & \langle 6 \times 4, \sigma \rangle \\ \longrightarrow & \langle 24, \sigma \rangle \end{aligned}$$

Using the Semantic Rules (Multi-Steps)

$$\langle (\text{foo} + 2) \times (\text{bar} + 1), \sigma \rangle \longrightarrow^* \langle 24, \sigma \rangle.$$

Multi-Step: Reflexive Transitive Closure of Step

$$\text{REFL} \frac{}{c \longrightarrow^* c}$$

$$\text{TRANS} \frac{c \longrightarrow c' \quad c' \longrightarrow^* c''}{c \longrightarrow^* c''}$$

Expressing Program Properties

Progress

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

either $e \in \mathbf{Int}$ or $\exists e', \sigma'. \langle e, \sigma \rangle \longrightarrow \langle e', \sigma' \rangle$

Termination

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

Deterministic Result

$\forall e \in \mathbf{Exp}. \forall \sigma_0, \sigma, \sigma' \in \mathbf{Store}. \forall n, n' \in \mathbf{Int}.$
if $\langle e, \sigma_0 \rangle \longrightarrow^* \langle n, \sigma \rangle$ and
 $\langle e, \sigma_0 \rangle \longrightarrow^* \langle n', \sigma' \rangle$ then
 $n = n'$ 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?

What if

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