CS 4110 – Programming Languages and Logics
Lecture #4: Large-step semantics

1 Large-step operational semantics

In the last lecture we defined a semantics for our language of arithmetic expressions using a small-step evaluation relation $\rightarrow \subseteq \mathbf{Config} \times \mathbf{Config}$ (and its reflexive and transitive closure $\rightarrow^*$). In this lecture we will explore an alternative approach—*large-step* operational semantics—which yields the final result of evaluating an expression directly.

Defining a large-step semantics boils down to specifying a relation $\Downarrow$ that captures the evaluation of an expression. The $\Downarrow$ relation has the following type:

$$\Downarrow \subseteq (\mathbf{Store} \times \mathbf{Exp}) \times (\mathbf{Store} \times \mathbf{Int}).$$

We write $\langle \sigma, e \rangle \Downarrow \langle \sigma', n \rangle$ to indicate that $((\sigma, e), (\sigma', n)) \in \Downarrow$. In other words, the expression e with store σ evaluates in one big step to the final store σ' and integer n.

We define the relation $\Downarrow$ inductively, using inference rules:

$$\frac{}{\langle \sigma, n \rangle \Downarrow \langle \sigma, n \rangle}\ \textsc{Int} \qquad\qquad \frac{n = \sigma(x)}{\langle \sigma, x \rangle \Downarrow \langle \sigma, n \rangle}\ \textsc{Var}$$

$$\frac{\langle \sigma, e_1 \rangle \Downarrow \langle \sigma', n_1 \rangle \qquad \langle \sigma', e_2 \rangle \Downarrow \langle \sigma'', n_2 \rangle \qquad n = n_1 + n_2}{\langle \sigma, e_1 + e_2 \rangle \Downarrow \langle \sigma'', n \rangle}\ \textsc{Add}$$

$$\frac{\langle \sigma, e_1 \rangle \Downarrow \langle \sigma', n_1 \rangle \qquad \langle \sigma', e_2 \rangle \Downarrow \langle \sigma'', n_2 \rangle \qquad n = n_1 \times n_2}{\langle \sigma, e_1 * e_2 \rangle \Downarrow \langle \sigma'', n \rangle}\ \textsc{Mul}$$

$$\frac{\langle \sigma, e_1 \rangle \Downarrow \langle \sigma', n_1 \rangle \qquad \langle \sigma'[x \mapsto n_1], e_2 \rangle \Downarrow \langle \sigma'', n_2 \rangle}{\langle \sigma, x := e_1 \,;\, e_2 \rangle \Downarrow \langle \sigma'', n_2 \rangle}\ \textsc{Assgn}$$

To illustrate the use of these rules, consider the following proof tree, which shows that evaluating $\langle \sigma, \mathit{foo} := 3\,;\ \mathit{foo} * \mathit{bar} \rangle$ using a store σ such that $\sigma(\mathit{bar}) = 7$ yields $\sigma' = \sigma[\mathit{foo} \mapsto 3]$ and 21 as a result:

$$\frac{\dfrac{}{\langle \sigma, 3 \rangle \Downarrow \langle \sigma, 3 \rangle}\ \textsc{Int} \qquad \dfrac{\dfrac{}{\langle \sigma', \mathit{foo} \rangle \Downarrow \langle \sigma', 3 \rangle}\ \textsc{Var} \qquad \dfrac{}{\langle \sigma', \mathit{bar} \rangle \Downarrow \langle \sigma', 7 \rangle}\ \textsc{Var}}{\langle \sigma', \mathit{foo} * \mathit{bar} \rangle \Downarrow \langle \sigma', 21 \rangle}\ \textsc{Mul}}{\langle \sigma, \mathit{foo} := 3\,;\ \mathit{foo} * \mathit{bar} \rangle \Downarrow \langle \sigma', 21 \rangle}\ \textsc{Assgn}$$

A closer look to this structure reveals the relation between small step and large-step evaluation: a depth-first traversal of the large-step proof tree yields the sequence of one-step transitions in small-step evaluation.

2 Equivalence of semantics

A natural question to ask is whether the small-step and large-step semantics are equivalent. The next theorem answers this question affirmatively.

Theorem (Equivalence of semantics)**.** *For all expressions e, stores σ and σ', and integers n we have:*

$$\langle \sigma, e\rangle \Downarrow \langle \sigma', n\rangle \textit{ if and only if } \langle \sigma, e\rangle \to^* \langle \sigma', n\rangle$$

To streamline the proof, we will work with the following definition of the multi-step relation:

$$\frac{}{\langle \sigma, e\rangle \to^* \langle \sigma, e\rangle}\ \textsc{Refl}$$

$$\frac{\langle \sigma, e\rangle \to \langle \sigma', e'\rangle \qquad \langle \sigma', e'\rangle \to^* \langle \sigma'', e''\rangle}{\langle \sigma, e\rangle \to^* \langle \sigma'', e''\rangle}\ \textsc{Trans}$$

Proof sketch. We show each direction separately.

$\Longrightarrow$**:** We want to prove that the following property P holds for all expressions $e \in \mathbf{Exp}$:

$$P(e) \triangleq \forall \sigma, \sigma' \in \mathbf{Store}.\ \forall n \in \mathbf{Int}.\ \langle \sigma, e\rangle \Downarrow \langle \sigma', n\rangle \Longrightarrow \langle \sigma, e\rangle \to^* \langle \sigma', n\rangle$$

We proceed by structural induction on e. We have to consider each of the possible axioms and inference rules for constructing an expression.

Case $e = x$**:** Assume that $\langle \sigma, x\rangle \Downarrow \langle \sigma', n\rangle$. That is, there is some derivation in the large-step operational semantics whose conclusion is $\langle \sigma, x\rangle \Downarrow \langle \sigma, n\rangle$. There is only one rule whose conclusion matches the configuration $\langle \sigma, x\rangle$: the large-step rule Var. Thus, we have $n = \sigma(x)$ and $\sigma' = \sigma$. By the small-step rule Var, we also have $\langle \sigma, x\rangle \to \langle \sigma, n\rangle$. By the Refl and Trans rules, we conclude that $\langle \sigma, x\rangle \to^* \langle \sigma, n\rangle$, which finishes the case.

Case $e = n$**:** Assume that $\langle \sigma, n\rangle \Downarrow \langle \sigma', n'\rangle$. There is only one rule whose conclusion matches $\langle \sigma, n\rangle$: the large-step rule Int. Thus, we have $n' = n$ and $\sigma' = \sigma$ and so $\langle \sigma, n\rangle \to^* \langle \sigma, n\rangle$ by the Refl rule.

Case $e = e_1 + e_2$**:** This is an inductive case. We want to prove that if $P(e_1)$ and $P(e_2)$ hold, then $P(e)$ also holds. Let's write out $P(e_1)$, $P(e_2)$, and $P(e)$ explicitly.

$$\begin{aligned} P(e_1) &= \forall n, \sigma, \sigma'.\ \langle \sigma, e_1\rangle \Downarrow \langle \sigma', n\rangle \Longrightarrow \langle \sigma, e_1\rangle \to^* \langle \sigma', n\rangle \\ P(e_2) &= \forall n, \sigma, \sigma'.\ \langle \sigma, e_2\rangle \Downarrow \langle \sigma', n\rangle \Longrightarrow \langle \sigma, e_2\rangle \to^* \langle \sigma', n\rangle \\ P(e) &= \forall n, \sigma, \sigma'.\ \langle \sigma, e_1 + e_2\rangle \Downarrow \langle \sigma', n\rangle \Longrightarrow \langle \sigma, e_1 + e_2\rangle \to^* \langle \sigma', n\rangle \end{aligned}$$

Assume that $P(e_1)$ and $P(e_2)$ hold. Also assume that there exist σ, σ' and n such that $\langle \sigma, e_1 + e_2\rangle \Downarrow \langle \sigma', n\rangle$. We need to show that $\langle \sigma, e_1 + e_2\rangle \to^* \langle \sigma', n\rangle$.

We assumed that $\langle \sigma, e_1 + e_2\rangle \Downarrow \langle \sigma', n\rangle$. This means that there is some derivation whose conclusion is $\langle \sigma, e_1 + e_2\rangle \Downarrow \langle \sigma', n\rangle$. By inspection, we see that only one rule has a conclusion of this form: the Add rule. Thus, the last rule used in the derivation was Add and it must be the case that $\langle \sigma, e_1\rangle \Downarrow \langle \sigma'', n_1\rangle$ and $\langle \sigma'', e_2\rangle \Downarrow \langle \sigma', n_2\rangle$ hold for some n_1 and n_2 with $n = n_1 + n_2$.

By the induction hypothesis $P(e_1)$, as $\langle \sigma, e_1 \rangle \Downarrow \langle \sigma'', n_1 \rangle$, we must have $\langle \sigma, e_1 \rangle \rightarrow^* \langle \sigma'', n_1 \rangle$. Likewise, by induction hypothesis $P(e_2)$, we have $\langle \sigma'', e_2 \rangle \rightarrow^* \langle \sigma', n_2 \rangle$. By Lemma 1 below, we have,

$$\langle \sigma, e_1 + e_2 \rangle \rightarrow^* \langle \sigma'', n_1 + e_2 \rangle,$$

and by another application of Lemma 1 we have:

$$\langle \sigma'', n_1 + e_2 \rangle \rightarrow^* \langle \sigma', n_1 + n_2 \rangle$$

Then, using the small-step ADD rule and the multi-step TRANS rule, we have:

$$\dfrac{\dfrac{n = n_1 + n_2}{\langle \sigma', n_1 + n_2 \rangle \rightarrow \langle \sigma', n \rangle}\ \textsc{Add} \qquad \dfrac{}{\langle \sigma', n \rangle \rightarrow^* \langle \sigma', n \rangle}\ \textsc{Refl}}{\langle \sigma', n_1 + n_2 \rangle \rightarrow^* \langle \sigma', n \rangle}\ \textsc{Trans}$$

Finally, by two applications of Lemma 2, we obtain $\langle \sigma, e_1 + e_2 \rangle \rightarrow^* \langle \sigma', n \rangle$, which finishes the case.

Case $e = e_1 * e_2$**.** Similar to case for $e_1 + e_2$ above.

Case $e = x := e_1; e_2$**.** Omitted. Try it as an exercise.

$\Longleftarrow$**:** We proceed by induction on the derivation of $\langle \sigma, e \rangle \rightarrow^* \langle \sigma', n \rangle$ with a case analysis on the last rule used.

Case REFL: Then $e = n$ and $\sigma' = \sigma$. We immediately have $\langle \sigma, n \rangle \Downarrow \langle \sigma, n \rangle$ by the large-step rule INT.

Case TRANS: Then $\langle \sigma, e \rangle \rightarrow \langle \sigma'', e'' \rangle$ and $\langle \sigma'', e'' \rangle \rightarrow^* \langle \sigma', n \rangle$. In this case, the induction hypothesis gives $\langle \sigma'', e'' \rangle \Downarrow \langle \sigma', n \rangle$. The result follows from Lemma 3 below.

□

Lemma 1. *If* $\langle \sigma, e \rangle \rightarrow^* \langle \sigma', n \rangle$*, then the following hold:*

- $\langle \sigma, e + e_2 \rangle \rightarrow^* \langle \sigma', n + e_2 \rangle$
- $\langle \sigma, e * e_2 \rangle \rightarrow^* \langle \sigma', n * e_2 \rangle$
- $\langle \sigma, n_1 + e \rangle \rightarrow^* \langle \sigma', n_1 + n \rangle$
- $\langle \sigma, n_1 * e \rangle \rightarrow^* \langle \sigma', n_1 * n \rangle$
- $\langle \sigma, x := e\,;\ e_2 \rangle \rightarrow^* \langle \sigma', x := n\,;\ e_2 \rangle$

Proof. Omitted; try it as an exercise. □

Lemma 2. *If* $\langle \sigma, e \rangle \rightarrow^* \langle \sigma', e' \rangle$ *and* $\langle \sigma', e' \rangle \rightarrow^* \langle \sigma'', e'' \rangle$*, then* $\langle \sigma, e \rangle \rightarrow^* \langle \sigma'', e'' \rangle$*.*

Proof. Omitted; try it as an exercise. □

Lemma 3. *If* $\langle \sigma, e \rangle \rightarrow \langle \sigma'', e'' \rangle$ *and* $\langle \sigma'', e'' \rangle \Downarrow \langle \sigma', n \rangle$*, then* $\langle \sigma, e \rangle \Downarrow \langle \sigma', n \rangle$*.*

Proof. Omitted; try it as an exercise. □