

Elements of Programming Languages

Tutorial 7: Small-step semantics

Solution notes

1. Comparing large-step and small-step derivations

(a) For large-step:

$$\frac{\lambda x.x + 1 \Downarrow \lambda x.x + 1 \quad 42 \Downarrow 42 \quad \frac{42 \Downarrow 42 \quad 1 \Downarrow 1}{42 + 1 \Downarrow 43}}{(\lambda x.x + 1) 42 \Downarrow 43}$$

For small-step, the step derivations are:

$$\frac{}{(\lambda x.x + 1) 42 \mapsto 42 + 1}$$

$$\frac{}{42 + 1 \mapsto 43}$$

(b) For large-step:

$$\frac{f \Downarrow f \quad 42 \Downarrow 42 \quad \frac{42 \Downarrow 42 \quad 1 \Downarrow 1}{42 == 1 \Downarrow \text{false}} \quad \frac{42 \Downarrow 42 \quad 1 \Downarrow 1}{42 + 1 \Downarrow 43}}{\frac{\text{if } 42 == 1 \text{ then } 2 \text{ else } 42 + 1 \Downarrow 43}}{(\lambda x.\text{if } x == 1 \text{ then } 2 \text{ else } x + 1) 42 \Downarrow 43}}$$

For small-step, the step derivations are:

$$\frac{}{(\lambda x.\text{if } x == 1 \text{ then } 2 \text{ else } x + 1) 42 \mapsto \text{if } 42 == 1 \text{ then } 2 \text{ else } 42 + 1}$$

$$\frac{\frac{}{42 == 1 \mapsto \text{false}}}{\text{if } 42 == 1 \text{ then } 2 \text{ else } 42 + 1 \mapsto \text{if } \text{false} \text{ then } 2 \text{ else } 42 + 1}}$$

$$\frac{}{\text{if } \text{false} \text{ then } 2 \text{ else } 42 + 1 \mapsto 42 + 1}$$

$$\frac{}{42 + 1 \mapsto 43}$$

2. Small-step derivations that go wrong

(a) The small-step derivation is:

$$\begin{aligned} ((\lambda x.\lambda y.\text{let } z = x + y \text{ in } z + 1) 42) \text{ true} &\mapsto (\lambda y.\text{let } z = 42 + y \text{ in } z + 1) \text{ true} \\ &\mapsto \text{let } z = 42 + \text{true} \text{ in } z + 1 \end{aligned}$$

which is stuck because the next subexpression to evaluate, $42 + \text{true}$, cannot take a step.

(b) The small-step derivation is:

$$\begin{aligned} (\lambda x.\text{if } x \text{ then } x + 1 \text{ else } x + 2) \text{ true} &\mapsto \text{if } \text{true} \text{ then } \text{true} + 1 \text{ else } \text{true} + 2 \\ &\mapsto \text{true} + 1 \end{aligned}$$

3. Small-step rules for L_{Data}

(a) Here are some possible rules:

$$\boxed{e \mapsto e'}$$

$$\frac{\frac{\frac{e_1 \mapsto e'_1}{(e_1, e_2) \mapsto (e'_1, e_2)}}{\mathbf{fst}(v_1, v_2) \mapsto v_1}}{\mathbf{fst} e \mapsto \mathbf{fst} e'} \quad \frac{\frac{\frac{e_2 \mapsto e'_2}{(v_1, e_2) \mapsto (v_1, e'_2)}}{\mathbf{snd}(v_1, v_2) \mapsto v_2}}{\mathbf{snd} e \mapsto \mathbf{snd} e'} \quad \frac{e \mapsto e'}{\mathbf{left}(e) \mapsto \mathbf{left}(e')} \quad \frac{e \mapsto e'}{\mathbf{right}(e) \mapsto \mathbf{right}(e')}}}{\frac{e \mapsto e'}{\mathbf{case} e \text{ of } \{\mathbf{left}(x) \Rightarrow e_1; \mathbf{right}(y) \Rightarrow e_2\} \mapsto \mathbf{case} e' \text{ of } \{\mathbf{left}(x) \Rightarrow e_1; \mathbf{right}(y) \Rightarrow e_2\}}}$$

$$\frac{\mathbf{case} \mathbf{left}(v) \text{ of } \{\mathbf{left}(x) \Rightarrow e_1; \mathbf{right}(y) \Rightarrow e_2\} \mapsto e_1[v/x]}{\mathbf{case} \mathbf{right}(v) \text{ of } \{\mathbf{left}(x) \Rightarrow e_1; \mathbf{right}(y) \Rightarrow e_2\} \mapsto e_2[v/y]}$$

One example of a design choice is whether to evaluate the subexpressions of a pair left-to-right, or right-to-left (or not specify the order, or not evaluate them eagerly at all...)

- (b) i. The step derivations are:

$$\frac{}{(\lambda p.(\mathbf{snd} p, \mathbf{fst} p + 2)) (17, 42) \mapsto (\mathbf{snd} (17, 42), \mathbf{fst} (17, 42) + 2)}$$

$$\frac{\mathbf{snd} (17, 42) \mapsto 42}{(\mathbf{snd} (17, 42), \mathbf{fst} (17, 42) + 2) \mapsto (42, \mathbf{fst} (17, 42) + 2)}$$

$$\frac{\mathbf{fst} (17, 42) \mapsto 17}{\mathbf{fst} (17, 42) + 2 \mapsto 17 + 2}$$

$$\frac{}{(42, \mathbf{fst} (17, 42) + 2) \mapsto (42, 17 + 2)}$$

$$\frac{}{(42, 17 + 2) \mapsto (42, 19)}$$

- ii. The step derivations are:

$$\frac{}{(\lambda x. \mathbf{case} x \text{ of } \{\mathbf{left}(y).y + 1; \mathbf{right}(z).z\}) (\mathbf{left}(42)) \mapsto \mathbf{case} \mathbf{left}(42) \text{ of } \{\mathbf{left}(y).y + 1; \mathbf{right}(z).z\}}$$

$$\frac{}{\mathbf{case} \mathbf{left}(42) \text{ of } \{\mathbf{left}(y).y + 1; \mathbf{right}(z).z\} \mapsto 42 + 1}$$

$$\frac{}{42 + 1 \mapsto 43}$$

4. (★) Type soundness for nondeterminism

- (a) Preservation means that if $\vdash e : \tau$ holds and $e \mapsto e'$ then $\vdash e' : \tau$ holds.

To prove preservation, suppose we have a well-formed nondeterministic choice expression:

$$\frac{\Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 \square e_2 : \tau}$$

such that $e_1 \square e_2 \mapsto e'$.

There are two cases. If $e_1 \square e_2 \mapsto e_1$ then we know $\vdash e_1 : \tau$ by assumption. Similarly, if $e_1 \square e_2 \mapsto e_2$ then we know $\vdash e_2 : \tau$.

- (b) Progress means that if $\vdash e : \tau$ holds then either e is a value, or e can take a step ($e \mapsto e'$).

To prove progress for a nondeterministic expression $\vdash e_1 \square e_2 : \tau$, this is immediate since such an expression can definitely step to e_1 (or e_2). Note that progress does not require that the step taken is unique; indeed, type soundness can hold for a nondeterministic language as long as both alternatives in a choice expression have the same type.