Tutorial 7 - Operational Semantics

November 4, 2023

1 Grammar

$\langle expr \rangle ::= \langle var \rangle$
| $\lambda\ \langle var \rangle\ .\ \langle expr \rangle$
| $\langle expr \rangle\ \langle expr \rangle$
| $(\langle expr \rangle, \langle expr \rangle)$
| fst $\langle expr \rangle$
| snd $\langle expr \rangle$
| true
| false
| and $\langle expr \rangle\ \langle expr \rangle$
| if $\langle expr \rangle$ then $\langle expr \rangle$ else $\langle expr \rangle$
| let $\langle var \rangle = \langle expr \rangle$ in $\langle expr \rangle$
| $(\langle expr \rangle)$

2 Call-By-Name

This is what was presented in tutorial; see `Lambda.hs` for a Haskell implementation.

$$\frac{e_1 \to e_1'}{e_1\ e_2 \to e_1'\ e_2} \qquad \frac{}{(\lambda x.e_1)\ e_2 \to e_1[e_2/x]}$$

$$\frac{}{\text{fst}\ (e_1, e_2) \to e_1} \qquad \frac{}{\text{snd}\ (e_1, e_2) \to e_2}$$

$$\frac{e_1 \to e_1'}{\text{and}\ e_1\ e_2 \to \text{and}\ e_1'\ e_2} \qquad \frac{}{\text{and true}\ e_2 \to e_2} \qquad \frac{}{\text{and false}\ e_2 \to \text{false}}$$

$$\frac{e_1 \to e_1'}{\text{if}\ e_1\ \text{then}\ e_2\ \text{else}\ e_3 \to \text{if}\ e_1\ \text{then}\ e_2\ \text{else}\ e_3} \qquad \frac{}{\text{if true then}\ e_2\ \text{else}\ e_3 \to e_2} \qquad \frac{}{\text{if false then}\ e_2\ \text{else}\ e_3 \to e_3}$$

$$\frac{}{\text{let}\ x = e_1\ \text{in}\ e_2 \to e_2[e_1/x]}$$

3 Call-By-Value

This was not presented in tutorial, and the Haskell implementation is left as an exercise. Recall that e denotes an arbitrary expression, and v denotes a normal form; in other words, an expression that cannot be reduced further.

$$\frac{e_1 \to e_1'}{e_1\ e_2 \to e_1'\ e_2} \qquad \frac{e_2 \to e_2'}{v_1\ e_2 \to v_1\ e_2'} \qquad \frac{}{(\lambda x.e_1)\ v_2 \to e_1[v_2/x]}$$

$$\frac{e_1 \to e_1'}{(e_1, e_2) \to (e_1', e_2)} \qquad \frac{e_2 \to e_2'}{(v_1, e_2) \to (v_1, e_2')} \qquad \frac{}{\text{fst } (v_1, v_2) \to v_1} \qquad \frac{}{\text{snd } (v_1, v_2) \to v_2}$$

$$\frac{e_1 \to e_1'}{\text{and } e_1\ e_2 \to \text{and } e_1'\ e_2} \qquad \frac{}{\text{and true } e_2 \to e_2} \qquad \frac{}{\text{and false } e_2 \to \text{false}}$$

$$\frac{e_1 \to e_1'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \to \text{if } e_1 \text{ then } e_2 \text{ else } e_3} \qquad \frac{}{\text{if true then } e_2 \text{ else } e_3 \to e_2} \qquad \frac{}{\text{if false then } e_2 \text{ else } e_3 \to e_3}$$

$$\frac{e_1 \to e_1'}{\text{let } x = e_1 \text{ in } e_2 \to \text{let } x = e_1' \text{ in } e_2} \qquad \frac{}{\text{let } x = v_1 \text{ in } e_2 \to e_2[v_1/x]}$$

4 Some Useful Exercises

If you are looking for further practice, look no further! Note that no marks will be given for completing these questions; they are just for practice.

★ Implement Call-By-Value in Haskell

★★ Add 'or' to the language by updating the BNF and operational semantics.

★ Implement 'or' in Haskell using your semantics.

★★★ Devise a type system for this language; write it out using typing judgements.

★★★★★ Prove progress and preservation for your type system.