Scheme1 Core Small-step Operational Semantics

November 7, 2021

The Scheme1 core small-step operational semantics is given as a two-place relation between expressions e and e', written $e \longrightarrow e'$, pronounced "e steps to e'". Formally, the small-step semantics is taken to be the smallest relation closed under the following rules:

Values

The Scheme1 Core values are the two Boolean values true and false, numbers, and functions of the form (fun $x\ e1$). The notation $[x \mapsto v]e$ denotes capture-avoiding substitution of value v for variable x in expression e. We let metavariables b and n (variously n_1, n_2, etc.) range over Boolean values and numbers, respectively.

Unary operators

$$\textsc{E-not1}\quad \frac{e \longrightarrow e'}{(\text{not } e) \longrightarrow (\text{not } e')} \qquad\qquad \textsc{E-not}\quad \frac{}{(\text{not } b) \longrightarrow \neg b}$$

Binary operators

$$\textsc{E-binop1}\quad \frac{e1 \longrightarrow e1'}{(op\, e_1\ e_2) \longrightarrow (op\, e_1'\ e_2)} \qquad\qquad \textsc{E-binop2}\quad \frac{e2 \longrightarrow e2'}{(op\, n_1\ e_2) \longrightarrow (op\, n_1\ e_2')}$$

$$\textsc{E-binop}\quad \frac{n_1\ op\ n_2 = v \qquad op \in \{+, *, -, /, =, <\}}{(op\, n_1\ n_2) \longrightarrow v}$$

Conditionals

E-IF1
$$\frac{e_1 \longrightarrow e_1'}{(\text{if } e_1\ e_2\ e_3) \longrightarrow (\text{if } e_1'\ e_2\ e_3)}$$

E-IF2
$$\frac{e_2 \longrightarrow e_2'}{(\text{if true } e_2\ e_3) \longrightarrow (\text{if true } e_2'\ e_3)}$$

E-IF3
$$\frac{e_3 \longrightarrow e_3'}{(\text{if false } e_2\ e_3) \longrightarrow (\text{if false } e_2\ e_3')}$$

E-IF-TRUE
$$\frac{}{(\text{if true } v_2\ e_3) \longrightarrow v_2}$$

E-IF-FALSE
$$\frac{}{(\text{if false } e_2\ v_3) \longrightarrow v_3}$$

Functions

E-APP1
$$\frac{e1 \longrightarrow e1'}{(e_1\ e_2) \longrightarrow (e_1'\ e_2)}$$

E-APP2
$$\frac{e2 \longrightarrow e2'}{((\text{fun } x\ e_1)\ e_2) \longrightarrow ((\text{fun } x\ e_1)\ e_2')}$$

E-APP
$$\frac{}{((\text{fun } x\ e_1)\ v_2) \longrightarrow [x \mapsto v_2]e_1}$$