1 Operational Semantics

1.1 Environments

E - a mapping from identifiers to values

F - a mapping from identifiers to function definitions.

A function definition has three fields:

prog: a statement of the form P (see context free grammar)

r: a statement of the form e that comes after the ret keyword.

p_list: a list of identifiers that make up the parameter list of the function definition.

1.2 The Fundamental crumbL Statement

$$\frac{E, F \vdash S_1 : E', F' \quad E', F' \vdash S_2 : E'', F''}{E, F \vdash S_1\ S_2 : E'', F''}$$

1.3 Constants

$$\frac{}{E, F \vdash \text{INT\_CONST} : \text{INT\_CONST}}$$

$$\frac{}{E, F \vdash \text{STRING\_CONST} : \text{STRING\_CONST}}$$

$$\frac{}{E, F \vdash \text{Nil} : \text{Nil}}$$

1.4 Arithmetic

$$\frac{E, F \vdash e_1 : i_1 \quad E, F \vdash e_2 : i_2}{E, F \vdash e_1 \oplus e_2 : i_1 \oplus i_2}$$

where $\oplus \in \{+, -, *, \%\}$

$$\frac{E, F \vdash e_1 : i_1 \quad E, F \vdash e_2 : i_2 \quad i_2 \neq 0}{E, F \vdash e_1 / e_2 : i_1 / i_2}$$

$$\frac{E, F \vdash e_1 : s_1 \quad E, F \vdash e_2 : s_2}{E, F \vdash e_1 :: e_2 : s_1 s_2}$$

1.5 Lists

$$\frac{E, F \vdash e_1 : v_1 (\text{not a list}) \quad E, F \vdash e_2 : v_2 \ (\text{not Nil})}{E, F \vdash e_1 @ e_2 : [v_1, v_2]}$$

$$\frac{E, F \vdash e_1 : v_1 (\text{not a list}) \quad E, F \vdash e_2 : \text{Nil}}{E, F \vdash e_1 @ e_2 : v_1}$$

$$\frac{E, F \vdash e : [v_1, v_2]}{E, F \vdash !e : v_1}$$

$$\frac{E, F \vdash e : [v_1, v_2]}{E, F \vdash \#e : v_2}$$

$$\frac{E, F \vdash e : v_1 (\text{not a list})}{E, F \vdash !e : v_1}$$

$$\frac{E, F \vdash e : v_1 (\text{not a list})}{E, F \vdash \#e : Nil}$$

1.6 Boolean Logic

$$\frac{E, F \vdash e_1 : i_1 \quad E, F \vdash e_2 : i_2}{E, F \vdash e_1 \odot e_2 : i_1 \odot i_2}$$

where $\odot \in \{<, >, <=, >=, ==, !=, and, or\}$

$$\frac{E, F \vdash e_1 : \text{nonzero int}}{E, F \vdash \text{not } e_1 : 0}$$

$$\frac{E, F \vdash e_1 : 0}{E, F \vdash \text{not } e_1 : 1}$$

$$\frac{E, F \vdash e_1 : Nil}{E, F \vdash \text{isNil } e_1 : 1}$$

$$\frac{E, F \vdash e_1 : \text{not Nil}}{E, F \vdash \text{isNil } e_1 : 0}$$

1.7 Conditional Statements

$$\frac{E, F \vdash C : \text{nonzero int} \quad E, F \vdash S_1 : E', F'}{E, F \vdash \text{if } (C) \text{ then } S_1 \text{ else } S_2 \text{ fi } : E', F'}$$

$$\frac{E, F \vdash C : 0 \quad E, F \vdash S_2 : E', F'}{E, F \vdash \text{if } (C) \text{ then } S_1 \text{ else } S_2 \text{ fi } : E', F'}$$

$$\frac{E, F \vdash C : 0}{E, F \vdash \text{while}(C) \text{ do } S \text{ ob } : E, F}$$

$$\frac{E, F \vdash C : \text{nonzero int} \quad E, F \vdash S : E', F \quad E', F \vdash \text{while}(C) \text{ do } S \text{ ob } : E'', F}{E, F \vdash \textit{while}(C) \text{ do } S \text{ od } : E'', F}$$

Note: Under this rule, the statement S must not change the function environment.

1.8 Identifiers and Functions

$$\frac{E, F \vdash e : v \quad E(id) \text{ does not exist}}{E, F \vdash id = e; : E[id \leftarrow v], F}$$

$$\frac{E, F \vdash v = E[id]}{E, F \vdash id : v}$$

$$\frac{fentry = \{\text{prog: P, r: e, p\_list: p\_list}\} \quad F' = F[\textbf{fname} \leftarrow fentry]}{E, F \vdash \text{func } \textbf{fname}(\text{p\_list}) \ P \text{ ret } e; \text{ cnuf } : E, F'}$$

$$\frac{fentry = F[\textbf{fname}] \quad E' = \text{apply}(fentry.\text{p\_list}, \text{ call\_list}) \quad p = fentry.\text{prog} \quad E', F \vdash p : E'' \quad E'', F \vdash fentry.\text{r} : v}{E, F \vdash \textbf{fname}(\text{call\_list}) : v}$$

Note: apply is just an operational semantics subroutine to construct a new environment for a called function, and is not useable from source code.

$$\frac{\begin{array}{c} E, F \vdash \text{p\_list} = [p_1, R_1] \\ E, F \vdash \text{call\_list} = [e_1, R_2] \\ E, F \vdash e_1 : v_1 \\ E, F \vdash \text{apply}(R_1, R_2) : E' \\ E'' = E'[p_1 \leftarrow v_1] \end{array}}{E, F \vdash \text{apply}(\text{p\_list}, \text{call\_list}) : E''}$$

$$\frac{}{\text{apply}(\epsilon, \epsilon) : \emptyset}$$

1.9 I/O

$$\frac{E, F \vdash e : v}{E, F \vdash \text{print}(e); : E, F, \text{ print out } v}$$

$$\frac{\begin{array}{c} T \vdash e_1 : \alpha_1 \\ T \vdash e_2 : \alpha_2 \\ \alpha_2 = \alpha_1 \text{ or } \alpha_2 = \alpha_1 List \end{array}}{T \vdash e_1 @ e_2 : \alpha_1 List}$$