

Natural Semantics Specification Of Evaluation for Project 5

Throughout this document, ρ is an environment mapping variable names to values. The term $\rho(\mathbf{x})$ is the value assigned to the variable \mathbf{x} . It is assumed that the environment is represented as an ordered list of variable name/value pairs, and that it is searched from the front with the first match returned. Similarly, σ is an environment storing the definitions of functions. The term $\sigma(\mathbf{f})$ is the definition of the function \mathbf{f} , represented as a pair of the list of formal parameters and the expression forming the body of the function.

Evaluation of literals and variables:

In these three rules, n is an integer literal, r is a floating point literal, and \mathbf{x} is an identifier.

$$\frac{}{(\mathbf{n}, \rho, \sigma) \downarrow (\text{Int } n)} \quad \frac{}{(\mathbf{r}, \rho, \sigma) \downarrow (\text{Real } r)} \quad \frac{}{(\mathbf{x}, \rho, \sigma) \downarrow \rho(\mathbf{x})}$$

Evaluation of built-in operators:

In the following four rules, \bullet on the left of an arrow is any one of $+$, $-$, or $*$; on the right of an arrow it is the corresponding one of $+$, $-$, or $*$.

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Int } i_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Int } i_2)}{((\bullet e_1 e_2), \rho, \sigma) \downarrow (\text{Int } (i_1 \bullet i_2))} \quad \frac{(e_1, \rho, \sigma) \downarrow (\text{Real } r_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Real } r_2)}{((\bullet e_1 e_2), \rho, \sigma) \downarrow (\text{Real } (r_1 \bullet r_2))}$$

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Real } r_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Int } i_2)}{((\bullet e_1 e_2), \rho, \sigma) \downarrow (\text{Real } (r_1 \bullet (i_2 \text{ as real})))} \quad \frac{(e_1, \rho, \sigma) \downarrow (\text{Int } i_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Real } r_2)}{((\bullet e_1 e_2), \rho, \sigma) \downarrow (\text{Real } ((i_1 \text{ as real}) \bullet r_2))}$$

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Int } i_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Int } i_2) \quad i_1 \bmod i_2 = 0}{((/ e_1 e_2), \rho, \sigma) \downarrow (\text{Int } (i_1 \text{ div } i_2))}$$

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Int } i_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Int } i_2) \quad i_1 \bmod i_2 \neq 0}{((/ e_1 e_2), \rho, \sigma) \downarrow (\text{Real } ((i_1 \text{ as real})/(i_2 \text{ as real})))}$$

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Real } r_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Real } r_2)}{((/ e_1 e_2), \rho, \sigma) \downarrow (\text{Real } (r_1/r_2))}$$

$$\frac{(e_1, \rho, \sigma) \downarrow (\text{Real } r_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Int } i_2)}{((/ e_1 e_2), \rho, \sigma) \downarrow (\text{Real } (r_1/(i_2 \text{ as real})))} \quad \frac{(e_1, \rho, \sigma) \downarrow (\text{Int } i_1) \quad (e_2, \rho, \sigma) \downarrow (\text{Real } r_2)}{((/ e_1 e_2), \rho) \downarrow (\text{Real } ((i_1 \text{ as real})/r_2))}$$

Evaluation of user-defined operators:

In the following rule, $@$ is the list append function, and zip is the function which, given two lists of the same length, returns a list of pairs of corresponding elements.

$$\frac{(e_1, \rho, \sigma) \downarrow v_1 \quad \cdots \quad (e_n, \rho, \sigma) \downarrow v_n \quad (2^{nd}(\sigma(\mathbf{f})), zip(1^{st}(\sigma(\mathbf{f})), [v_1, \dots, v_n])@ \rho, \sigma) \downarrow v}{((\mathbf{f} e_1 \cdots e_n), \rho, \sigma) \downarrow v}$$

Evaluation of Commands:

While expressions (in the context of variable and function definitions) evaluate to values, commands evaluate to new contexts.

$$\overline{((\mathbf{exit}), \rho, \sigma) \downarrow \perp} \quad (\text{The system shuts down.})$$

$$\overline{((\mathbf{bindings}), \rho, \sigma) \downarrow (\rho, \sigma)} \quad (\text{The current bindings are displayed.})$$

$$\frac{(exp, \rho, \sigma) \downarrow v}{\overline{((\mathbf{define} \text{ var } exp), \rho, \sigma) \downarrow ((var, v) :: \rho), \sigma)}$$

$$\overline{((\mathbf{defun} ((opr :: vars)) exp), \rho, \sigma) \downarrow (\rho, (opr, vars, exp) :: \sigma)}$$

Composition of Commands:

$$\frac{(c_1, \rho, \sigma) \downarrow (\rho', \sigma') \quad (c_2, \rho', \sigma') \downarrow (\rho'', \sigma'')}{\overline{(c_1; c_2), \rho, \sigma) \downarrow (\rho'', \sigma'')}}}$$