RM 31/1/84

Operational Semantics of an ML subset

This subset restrict declarations to simple value and exception bindings, and allows only matches of the form $x.e$. Types are not considered. A simple "abstract" syntax is adopted — e.g. expressions are not subdivided into atomic and non-atomic expressions. The syntax classes of Identifiers and Constructors are assumed to be disjoint (but Identifiers may be bound both to values and to exceptions).

Identifiers $x \in Id$ Constructors $c \in Con$ Addresses $\alpha \in Addr$

Expressions $e \in Exp$, with the following forms:

$$e ::= x \mid c \mid (e_1, e_2) \mid (e_1, \cdots, e_n) \mid \underline{fun}\, x.e \mid$$
$$\underline{let}\ \underline{val}\ x = e\ \underline{in}\ e'\ \underline{end} \mid \underline{let}\ \underline{exception}\ x\ \underline{in}\ e\ \underline{end} \mid$$
$$\underline{raise}\ x\ e \mid e\ \underline{handle}\ x\ x'.e' \mid e\ ?\ e'$$

Values $v \in Val = Con + Con \times Val + \sum_{n \geq 2} Val^n + Addr + Clos$

Closures $Clos = Id \times Exp \times Valenv \times Excenv$ $\{$notation : $\langle x, e, \rho, \eta \rangle \}$

Value environments $\rho \in Valenv = Id \xrightarrow[fin]{} Val$

Exception environments $\eta \in Excenv = Id \xrightarrow[fin]{} Exc$ $\{$fin is finite functions$\}$

Exceptions $exc \in Exc$

Stores $\sigma \in Store = Mem \times FIN(Exc)$ $\{$FIN is finite subsets$\}$

Memories $\mu \in Mem = Addr \xrightarrow[fin]{} Val$

Packets $p \in Pack = Exc \times Val$ $\{$notation : $\langle exc, v \rangle \}$

Results $r \in Result = Val + Pack$

Rules

Evaluations have the form $\rho, \eta \vdash e, \sigma \longrightarrow r, \sigma'$. A result r may be a value v or a packet p. In all expression forms except the "handle" and "?" forms, a packet result for any subform aborts the evaluation. Thus, in the following rules (except those for "handle" and "?"), any rule of the shape

$$\frac{\begin{array}{l}\rho_1, \eta_1 \vdash e_1, \sigma_0 \longrightarrow v_1, \sigma_1 \\ \quad \rho_2, \eta_2 \vdash e_2, \sigma_1 \longrightarrow v_2, \sigma_2 \\ \qquad \cdots \\ \qquad\quad \rho_n, \eta_n \vdash e_n, \sigma_{n-1} \longrightarrow v_n, \sigma_n\end{array}}{\rho, \eta \vdash \mathrm{FORM}[e_1, \cdots, e_n], \sigma \longrightarrow r, \sigma'}$$

is understood to be supplemented by n rules, representing abortion by e_j for $1 \leq j \leq n$:

$$\frac{\begin{array}{l}\rho_1, \eta_1 \vdash e_1, \sigma_0 \longrightarrow v_1, \sigma_1 \\ \quad \cdots \\ \qquad \rho_{j-1}, \eta_{j-1} \vdash e_{j-1}, \sigma_{j-2} \longrightarrow v_{j-1}, \sigma_{j-1} \\ \qquad\quad \rho_j, \eta_j \vdash e_j, \sigma_{j-1} \longrightarrow p, \sigma_j\end{array}}{\rho, \eta \vdash \mathrm{FORM}[e_1, \cdots, e_n], \sigma \longrightarrow p, \sigma_j} \qquad [1 \leq j \leq n]$$

This device avoids cluttering the rules for each form with particular treatment of exceptions, and underlines the uniformity of their treatment.

The appearance of $v // p$ in a rule (in one or more places) indicates two rules, one with v in each of these places and one with p in each of these places.

Variables $\quad \rho, \eta \vdash x, \sigma \longrightarrow v, \sigma \qquad (\rho(x) = v)$

Constructors $\quad \rho, \eta \vdash c, \sigma \longrightarrow c, \sigma$

Applications

$$\frac{\rho, \eta \vdash e_1, \sigma \to v_1, \sigma' \qquad \rho, \eta \vdash e_2, \sigma' \to v_2, \sigma''}{\rho, \eta \vdash (e_1 e_2), \sigma \longrightarrow (v_1 v_2), \sigma''} \qquad (v_1 \notin Clos)$$

$$\frac{\rho, \eta \vdash e_1, \sigma \to \langle x, e', \rho', \eta' \rangle, \sigma' \qquad \rho, \eta \vdash e_2, \sigma' \to v_2, \sigma'' \qquad \rho'[x \mapsto v_2], \eta' \vdash e', \sigma'' \to v, \sigma'''}{\rho, \eta \vdash (e_1 e_2), \sigma \longrightarrow v, \sigma'''}$$

Tuples

$$\frac{\rho, \eta \vdash e_1, \sigma_0 \to v_1, \sigma_1 \qquad \rho, \eta \vdash e_2, \sigma_1 \to v_2, \sigma_2 \qquad \cdots \qquad \rho, \eta \vdash e_n, \sigma_{n-1} \to v_n, \sigma_n}{\rho, \eta \vdash (e_1, \cdots, e_n), \sigma_0 \longrightarrow (v_1, \cdots, v_n), \sigma_n}$$

Functions $\quad \rho, \eta \vdash (\underline{\text{fun}}\, x . e), \sigma \longrightarrow \langle x, e, \rho, \eta \rangle, \sigma$

Value bindings

$$\frac{\rho, \eta \vdash e, \sigma \to v, \sigma' \qquad \rho[x \mapsto v], \eta \vdash e', \sigma' \to v', \sigma''}{\rho, \eta \vdash (\underline{\text{let}}\ \underline{\text{val}}\ x = e\ \underline{\text{in}}\ e'\ \underline{\text{end}}), \sigma \longrightarrow v', \sigma''}$$

Exception bindings

$$\frac{\rho, \eta[x \to exc] \vdash e, (\mu, excs \cup \{exc\}) \to v, \sigma}{\rho, \eta \vdash (\underline{\text{let}}\ \underline{\text{exception}}\ x\ \underline{\text{in}}\ e\ \underline{\text{end}}), (\mu, excs) \longrightarrow v, \sigma} \qquad (exc \notin excs)$$

Exception raising

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow v, \sigma'}{\rho, \eta \vdash (\underline{\text{raise}}\ x\ e), \sigma \longrightarrow \langle exc, v\rangle, \sigma'} \quad (exc = \eta(x))$$

Exception handling

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow v, \sigma'}{\rho, \eta \vdash (e\ \underline{\text{handle}}\ x\ x'.e'), \sigma \longrightarrow v, \sigma'}$$

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow \langle exc, v\rangle, \sigma'}{\rho, \eta \vdash (e\ \underline{\text{handle}}\ x\ x'.e'), \sigma \longrightarrow \langle exc, v\rangle, \sigma'} \quad (exc \neq \eta(x))$$

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow \langle \eta(x), v'\rangle, \sigma' \qquad \rho[x' \mapsto v'], \eta \vdash e', \sigma' \longrightarrow v /\!/ p, \sigma''}{\rho, \eta \vdash (e\ \underline{\text{handle}}\ x\ x'.e'), \sigma \longrightarrow v /\!/ p, \sigma''}$$

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow v, \sigma'}{\rho, \eta \vdash (e\,?\,e'), \sigma \longrightarrow v, \sigma'}$$

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow p, \sigma' \qquad \rho, \eta \vdash e', \sigma' \longrightarrow v /\!/ p, \sigma''}{\rho, \eta \vdash (e\,?\,e'), \sigma \longrightarrow v /\!/ p, \sigma''}$$

Standard functions concerning references (addresses)

ref :

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow v, (\mu, excs)}{\rho, \eta \vdash (\text{ref}\ e), \sigma \longrightarrow \alpha, (\mu[\alpha \mapsto v], excs)} \quad (\alpha \notin \text{dom}\ \mu)$$

! :

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow \alpha, (\mu, excs)}{\rho, \eta \vdash (!\, e), \sigma \longrightarrow v, (\mu, excs)} \quad (v = \mu(\alpha))$$

:= :

$$\frac{\rho, \eta \vdash e, \sigma \longrightarrow \alpha, \sigma' \qquad \rho, \eta \vdash e', \sigma' \longrightarrow v, (\mu, excs)}{\rho, \eta \vdash (e := e'), \sigma \longrightarrow (), (\mu[\alpha \mapsto v], excs)}$$