Selam specification

nymphium

August 17, 2019

1 syntax

$$
\begin{array}{rcl}
x & \in & \mathit{Variables} \\
a & \in & \mathit{Atomics} \\
v & ::= & a \mid \text{¡}\lambda x.e \mid \text{¡}x \mid \lambda x.e \mid x \\
e & ::= & v \mid e@e \mid \texttt{¡let}\ x = e\ \texttt{in}\ e \\
 & & \mid e\ e \mid \texttt{let}\ x = e\ \texttt{in}\ e \\
 & & \\
w & ::= & a \mid \texttt{¡clos}_x\left(e, E, \text{¡}E\right) \mid \texttt{clos}_x\left(e, E\right) \\
F & ::= & w\ \square \mid w@\square \mid \left(\square@e, E, \text{¡}E\right) \mid \left(\square\ e, E\right) \\
 & & \mid \left(\texttt{let}\ x = \square\ \texttt{in}\ e, E\right) \mid \left(\texttt{¡let}\ x = \square\ \texttt{in}\ e, E\right) \\
C & ::= & e \mid w \\
E & ::= & [] \mid \left(x = w\right), E \\
\text{¡}E & ::= & [] \mid \left(x = w\right), \text{¡}E \\
K & ::= & [] \mid F, K \\
 & & \\
\tau & ::= & A \mid \tau \rightarrow \tau \mid \tau \multimap \tau \\
\Gamma & ::= & [] \mid \left(x = \tau\right), \Gamma \\
\Delta & ::= & [] \mid \left(x = \tau\right), \Delta
\end{array}
$$

2 semtntics

$$\boxed{\langle C;\ E;\ \text{¡}E;\ K\rangle \mapsto \langle C';\ E';\ \text{¡}E';\ K'\rangle}$$

$$\langle x;\ E;\ \text{¡}E;\ K\rangle \mapsto \langle \mathit{lookup}\left(x, E\right);\ E;\ \text{¡}E;\ K\rangle \qquad (\textsc{Lookup})$$

$$\langle \text{¡}x;\ E;\ \text{¡}E;\ K\rangle \mapsto \langle \mathit{lookup}\left(\text{¡}x, \text{¡}E\right);\ E;\ \mathit{drop}\left(\text{¡}x, \text{¡}E\right);\ K\rangle \qquad (\text{¡}\textsc{Lookup})$$

$$\langle \texttt{let}\ x = e\ \texttt{in}\ e';\ E;\ \text{¡}E;\ K\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ \left(\left(\texttt{let}\ x = \square\ \texttt{in}\ e', E\right), K\right)\rangle \qquad (\textsc{PushLet})$$

$$
\begin{aligned}
\mathit{lookup}\,(x, []) &= \text{undefined} \\
\mathit{lookup}\,(x, ((x = v)\,, \_)) &= v \\
\mathit{lookup}\,(x, ((y = \_)\,, \mathit{ls})) &= \mathit{lookup}\,(x, \mathit{ls})
\end{aligned}
$$

$$
\begin{aligned}
\mathit{drop}\,(x, []) &= [] \\
\mathit{drop}\,(x, ((x = \_)\,, \mathit{ls})) &= \mathit{ls} \\
\mathit{drop}\,(x, ((y = v)\,, \mathit{ls})) &= (y = v)\,, \mathit{drop}\,(x, \mathit{ls})
\end{aligned}
$$

Figure 1: utility functions

$$\langle \texttt{¡let}\ x = e\ \texttt{in}\ e';\ E;\ \text{¡}E;\ K\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ ((\texttt{¡let}\ x = \Box\ \texttt{in}\ e', E)\,, K)\rangle \quad (\textsc{Push¡Let})$$

$$\langle w;\ E;\ \text{¡}E;\ (\texttt{let}\ x = \Box\ \texttt{in}\ e', E')\,, K\rangle \mapsto \langle e';\ ((x = w)\,, E')\,;\ \text{¡}E;\ K\rangle \quad (\textsc{PopLet})$$

$$\langle w;\ E;\ \text{¡}E;\ ((\texttt{¡let}\ x = \Box\ \texttt{in}\ e, E')\,, K)\rangle \mapsto \langle e;\ E';\ ((x = w)\,, \text{¡}E')\,;\ K\rangle \quad (\textsc{Pop¡Let})$$

$$\langle \lambda x.e;\ E;\ \text{¡}E;\ K\rangle \mapsto \langle \texttt{clos}_x\,(e, E)\,;\ E;\ \text{¡}E;\ K\rangle \qquad (\textsc{Close})$$

$$\langle \text{¡}\lambda x.e;\ E;\ \text{¡}E;\ K\rangle \mapsto \langle \texttt{¡clos}_x\,(e, E, \text{¡}E)\,;\ E;\ \text{¡}E;\ K\rangle \qquad (\textsc{¡Close})$$

$$\langle e\ e';\ E;\ \text{¡}E;\ K\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ ((\Box\ e', E)\,, K)\rangle \qquad (\textsc{PushApp})$$

$$\langle e@e';\ E;\ \text{¡}E;\ K\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ ((\Box@e', E, \text{¡}E)\,, K)\rangle \qquad (\textsc{Push¡App})$$

$$\langle w;\ E;\ \text{¡}E;\ (\Box\ e', E')\,, K\rangle \mapsto \langle e';\ E';\ \text{¡}E;\ (w\ \Box, K)\rangle \qquad (\textsc{EvalArg})$$

$$\langle w;\ E;\ \text{¡}E;\ ((\Box@e', E', \text{¡}E')\,, K)\rangle \mapsto \langle e';\ E';\ \text{¡}E';\ (w@\Box, K)\rangle \quad (\textsc{Eval¡Arg})$$

$$\langle w;\ E;\ \text{¡}E;\ (\texttt{clos}_x\,(e, E')\,, K)\rangle \mapsto \langle e;\ ((x = w)\,, E')\,;\ \text{¡}E;\ K\rangle \qquad (\textsc{App})$$

$$\langle w;\ E;\ \text{¡}E;\ (\texttt{¡clos}_x\,(e, E', \text{¡}E')\,, K)\rangle \mapsto \langle e;\ E';\ ((x = w)\,, \text{¡}E')\,;\ K\rangle \quad (\textsc{¡App})$$

$$\langle a;\ \_;\ \_;\ \_\rangle \mapsto \text{Result: } a \qquad (\textsc{ResltAtom})$$

$$\langle \texttt{clos}_x\,(e, E)\,;\ \_;\ \_;\ \_\rangle \mapsto \text{Result: } \lambda x.e \qquad (\textsc{Result}\lambda)$$

$$\langle \texttt{¡clos}_x\,(e, E, ;)\ \_;\ \_;\ \_\rangle \mapsto \text{Result: } \text{¡}\lambda x.e \qquad (\textsc{Result¡}\lambda)$$

$$\Delta +\!\!+ \Delta' = \Delta''$$
$$\text{assumes } \forall x \in \Delta, \forall y \in \Delta'. \, (x \notin \Delta' \wedge (x \in \Delta'' \wedge y \in \Delta''))$$

3 Type System

$$\frac{}{\Gamma \mid \Delta \vdash a : A} \quad (\textsc{Atom})$$

$$\frac{(x = \tau)\,, \Gamma \mid [] \vdash e : \tau'}{\Gamma \mid \Delta \vdash \lambda x.e : \tau \to \tau'} \quad (\lambda)$$

$$\frac{\Gamma \mid (x = \tau)\,, \Delta \vdash e : \tau'}{\Gamma \mid \Delta \vdash \mathsf{i}\lambda x.e : \tau \multimap \tau'} \quad (\mathsf{i}\lambda)$$

$$\frac{\Gamma \mid [] \vdash e : \tau \to \tau' \qquad \Gamma \mid [] \vdash e' : \tau}{\Gamma \mid \Delta \vdash e\ e' : \tau'} \quad (\textsc{App})$$

$$\frac{\Gamma \mid \Delta \vdash e : \tau \multimap \tau' \qquad \Gamma \mid \Delta' \vdash e' : \tau}{\Gamma \mid \Delta +\!\!+ \Delta' \vdash e@e' : \tau'} \quad (\mathsf{i}\textsc{App})$$

$$\frac{(x = \tau) \in \Gamma}{\Gamma \mid \Delta \vdash x : \tau} \quad (\textsc{Var})$$

$$\frac{(x = \tau) \in \Delta}{\Gamma \mid \Delta \vdash x : \tau} \quad (\mathsf{i}\textsc{Var})$$

$$\frac{\Gamma \mid [] \vdash e : \tau \qquad (x = \tau)\,, \Gamma \mid \Delta \vdash e' : \tau'}{\Gamma \mid \Delta \vdash \texttt{let}\ x = e\ \texttt{in}\ e' : \tau'} \quad (\textsc{Let})$$

$$\frac{\Gamma \mid \Delta \vdash e : \tau \qquad (x = \tau)\,, \Gamma \mid \Delta' \vdash e' : \tau'}{\Gamma \mid \Delta +\!\!+ \Delta' \vdash \texttt{ilet}\ x = e\ \texttt{in}\ e' : \tau'} \quad (\mathsf{i}\textsc{Let})$$