Selam specification

nymphium

August 17, 2019

1 syntax

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

2 semtntics

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

Figure 1: utility functions

$$\boxed{\langle C;\ E;\ \mathsf{i}E;\ K\rangle \mapsto \langle C';\ E';\ \mathsf{i}E';\ K'\rangle}$$

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

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

$$\langle \mathtt{let}\ x = e\ \mathtt{in}\ e';\ E;\ \mathsf{i}E;\ K\rangle \mapsto \langle e;\ E;\ [];\ \left(\left(\mathtt{let}\ x = \square\ \mathtt{in}\ e', E, \mathsf{i}E\right), K\right)\rangle \qquad (\textsc{PushLet})$$

$$
\begin{array}{c}
\langle \mathtt{ilet}\ x = e\ \mathtt{in}\ e';\ E;\ \mathsf{i}E;\ K\rangle \mapsto \langle e;\ E;\ \mathsf{i}E';\ \left(\left(\mathtt{ilet}\ x = \square\ \mathtt{in}\ e', E, \mathsf{i}E''\right), K\right)\rangle \\
\text{where} \quad \begin{aligned} \mathsf{i}E' &= \quad \mathsf{i}FV\left(e\right) \\ \mathsf{i}E'' &= \quad \mathsf{i}E \backslash \mathsf{i}E' \end{aligned}
\end{array} \qquad (\textsc{Push}\mathsf{i}\textsc{Let})
$$

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

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

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

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

$$\langle e\ e';\ E;\ \_;\ K\rangle \mapsto \langle e;\ E;\ [];\ ((\Box\ e', E, \text{¡}E), K)\rangle \tag{PushApp}$$

$$\begin{array}{c}\langle e@e';\ E;\ \text{¡}E;\ K\rangle \mapsto \langle e;\ E;\ \text{¡}E';\ ((\Box@e, E, \text{¡}E''), K)\rangle \\ \text{where}\quad \begin{array}{rl}\text{¡}E' = & \text{¡}FV(e) \\ \text{¡}E'' = & \text{¡}E\backslash\text{¡}E'\end{array}\end{array} \tag{Push¡App}$$

$$\langle w;\ \_;\ \_;\ (\Box\ e, E, \text{¡}E), K\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ ((w\ \Box), K)\rangle \tag{PushArg}$$

$$\langle w;\ \_;\ \_;\ ((\Box@e, E, \text{¡}E), K)\rangle \mapsto \langle e;\ E;\ \text{¡}E;\ ((w@\Box), K)\rangle \tag{Push¡Arg}$$

$$\langle w;\ \_;\ \_;\ (\texttt{clos}_x(e, E), K)\rangle \mapsto \langle e;\ ((x = w), E);\ [];\ K\rangle \tag{App}$$

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

$$\langle a;\ \_;\ \_;\ \_\rangle \mapsto \text{Result: } a \tag{ResltAtom}$$

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

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

3 Type System

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

$$\frac{}{\Gamma \mid \Delta \vdash a : A} \tag{Atom}$$

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

$$\frac{\Gamma \mid (x = \tau), \Delta \vdash e : \tau'}{\Gamma \mid \Delta \vdash \text{¡}\lambda x.e : \tau \multimap \tau'} \tag{¡λ}$$

$$\frac{\Gamma \mid [] \vdash e : \tau \to \tau' \qquad \Gamma \mid [] \vdash e' : \tau}{\Gamma \mid \Delta \vdash e\ e' : \tau'} \tag{App}$$

$$\frac{\Gamma \mid \Delta \vdash e : \tau \multimap \tau' \qquad \Gamma \mid \Delta' \vdash e' : \tau}{\Gamma \mid \Delta{+\!\!+}\Delta' \vdash e@e' : \tau'} \tag{¡App}$$

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

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

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

$$\frac{\Gamma \mid \Delta \vdash e : \tau \qquad (x = \tau), \Gamma \mid \Delta' \vdash e' : \tau'}{\Gamma \mid \Delta{+\!\!+}\Delta' \vdash \texttt{¡let}\ x = e\ \texttt{in}\ e' : \tau'} \tag{¡Let}$$