Michał Majewski, 347108

Wywołanie funkcji dostaje w argumentach store oraz trójkę (lokacja argumentu, lokacja kopii argumentu, flaga w jakim jesteśmy trybie). Instrukcje też otrzymują między innymi taką trójkę. Domyślna flaga to *VAL* (przekazywanie przez wartość), domyślne lokacje to **??**, a używanie takich lokacji powoduje błąd. Zakładam, że błąd propaguje się przez wszystkie instrukcje.

$\mathbf{Env} = \mathbf{VAL} \to \mathbf{Loc}$
$\mathbf{PEnv} = \mathbf{PVAL} \to \mathbf{Proc}$
$\mathbf{Proc} = \mathbf{Store} \to (\mathbf{Loc} \times \mathbf{Loc} \times \{IO, VAL\}) \to \mathbf{Store}$
$\mathbf{Store} = \mathbf{Loc} \to \mathbf{Int}$
$\mathbf{Loc} = \{1, 2, 3, \ldots\} \cup \{\mathbf{??}\}$
$\mathbf{alloc} = \mathbf{Store} \to \mathbf{Loc}$
$\mathbf{save} = \mathbf{Loc} \to \mathbf{Loc} \to \mathbf{Store} \to \mathbf{Store}$

$\mathcal{N}[\![\_]\!].\texttt{Num} \to \mathbf{Int}$
$\mathcal{E}[\![\_]\!].\texttt{Expr} \to \mathbf{Env} \to \mathbf{Store} \to \mathbf{Int}$
$\mathcal{D}[\![\_]\!].\texttt{Dec} \to \mathbf{PEnv} \to \mathbf{Env} \to \mathbf{Store} \to (\mathbf{PEnv} \times \mathbf{Env} \times \mathbf{Store})$
$\mathcal{S}[\![\_]\!].\texttt{Instr} \to \mathbf{PEnv} \to \mathbf{Env} \to \mathbf{Store} \to (\mathbf{Loc} \times \mathbf{Loc} \times \{IO, VAL\}) \to (\mathbf{Store} \times \{IO, VAL\})$

$\rho, \rho' \in \mathbf{Env}$
$\pi, \pi' \in \mathbf{PEnv}$
$s, s', s'' \in \mathbf{Store}$
$l, l', l_{tmp}, l'_{tmp} \in \mathbf{Loc}$
$m, m' \in \{IO, VAL\}$
$\rho_p \in (\mathbf{Loc} \times \mathbf{Loc} \times \{IO, VAL\})$

$\mathbf{save} = \lambda l.\ \lambda l_{tmp}.\ \lambda s.\ s[l \to v]$
  where $v = s\ l_{tmp}$

$\mathcal{E}[\![\mathrm{n}]\!] = \lambda\rho.\ \lambda s.\ \mathcal{N}[\![\mathrm{n}]\!]$
$\mathcal{E}[\![x]\!] = \lambda\rho.\ \lambda s.\ s\ l$
  where $l = \rho\ x$
$\mathcal{E}[\![E_1 + E_2]\!] = \lambda\rho.\ \lambda s.\ e_1 + e_2$
  where
    $e_1 = \mathcal{E}[\![E_1]\!]\ \rho\ s$
    $e_2 = \mathcal{E}[\![E_2]\!]\ \rho\ s$

$\mathcal{D}[\![\texttt{int}\ x := E]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ (\pi, \rho[x \to l], s[l \to \mathcal{E}[\![E]\!]\ \rho\ s])$
  where $l = \ \mathbf{alloc}\ s$
$\mathcal{D}[\![D_1; D_2]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \mathcal{D}[\![D_2]\!]\ \pi'\ \rho'\ s'$
  where $(\pi', \rho', s') = \mathcal{D}[\![D_1]\!]\ \pi\ \rho\ s$
$\mathcal{D}[\![\texttt{proc}\ p(x)\ I]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ (\pi[p \to \text{fix}\ P], \rho, s)$
  where $P = \lambda P'.\ \lambda s'.\ \lambda(l, l_{tmp}, m).$
  let $(s'', m') = \mathcal{S}[\![I]\!]\ \pi[p \to P']\ \rho[x \to l_{tmp}]\ s'\ (l, l_{tmp}, m)$ in
    if $m' = VAL$ then $s''$
    else $\mathbf{save}\ l\ l_{tmp}\ s''$

$\mathcal{S}[\![\texttt{skip}]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).\ (s, m)$
$\mathcal{S}[\![\texttt{cio}]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).\ (s, IO)$
$\mathcal{S}[\![\texttt{cbv}]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).$
  if $m = VAL$ then $(s, VAL)$
  else $(\mathbf{save}\ l\ l_{tmp}\ s, VAL)$

$\mathcal{S}[\![x := E]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).\ (s[l' \to \mathcal{E}[\![E]\!]\ \rho\ s], m)$
  where $l' = \rho\ x$
$\mathcal{S}[\![I_1; I_2]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).\ \mathcal{S}[\![I_2]\!]\ \pi\ \rho\ s'\ (l, l_{tmp}, m')$
  where $(s', m') = \mathcal{S}[\![I_1]\!]\ \pi\ \rho\ s\ (l, l_{tmp}, m)$
$\mathcal{S}[\![\texttt{if}\ E = 0\ \texttt{then}\ I_1\ \texttt{else}\ I_2\ \texttt{fi}]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda\rho_p.$
  let $v = \mathcal{E}[\![E]\!]\ \rho\ s$ in
    if $v = 0$ then $\mathcal{S}[\![I_1]\!]\ \pi\ \rho\ s\ \rho_p$
    else $\mathcal{S}[\![I_2]\!]\ \pi\ \rho\ s\ \rho_p$
$\mathcal{S}[\![\texttt{begin}\ D; I\ \texttt{end}]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda\rho_p.\ \mathcal{S}[\![I]\!]\ \pi'\ \rho'\ s'\ \rho_p$
  where $(\pi', \rho', s') = \mathcal{D}[\![D]\!]\ \pi\ \rho\ s$

$$\mathcal{S}[\![\texttt{call } p(x)]\!] = \lambda\pi.\ \lambda\rho.\ \lambda s.\ \lambda(l, l_{tmp}, m).\ (P\ s[l'_{tmp} \to v]\ (l', l'_{tmp}, VAL), m)$$

where

$v = \mathcal{E}[\![x]\!]\ \rho\ s$

$l' = \rho\ x$

$l'_{tmp} = \mathbf{alloc}\ s$

$P = \pi\ p$