1 Semantics of Basic Scheme CESK*

Syntax:

$$
\begin{aligned}
e \in \mathsf{Exp} &::= \text{æ} \\
&\mid (\texttt{if}\ e\ e\ e) \\
&\mid (\texttt{let}\ (x\ e)\ e) \\
&\mid (\texttt{prim}\ op\ e\ e...) \\
&\mid (e\ e\ ...) \\
\text{æ} \in \mathsf{AExp} &::= lam \mid \mathbb{Z} \mid \texttt{\#t} \mid \texttt{\#f} \\
lam \in \mathsf{Lam} &::= (\lambda\ (x...)\ e) \\
x \in \mathsf{Var} &\quad \text{A set of identifiers}
\end{aligned}
$$

Semantics:

$$
\begin{aligned}
\varsigma \in \Sigma &\triangleq \mathsf{Exp} \times Env \times Kont \\
\rho \in Env &\triangleq \mathsf{Var} \rightharpoonup Addr \\
\sigma \in Store &\triangleq Addr \rightharpoonup Val \\
v \in Val &\triangleq Clo + \mathbb{Z} + \{\texttt{\#t}, \texttt{\#f}\} \\
clo \in Clo &\triangleq \mathsf{Lam} \times Env \\
\kappa \in Kont &\triangleq \mathbf{mt} \mid \mathbf{appk}(done, todo, \rho, a) \\
&\quad \mid \mathbf{ifk}(e, e, \rho, a) \\
&\quad \mid \mathbf{letk}(x, e, \rho, a) \\
a, b, c \in Addr &\quad \text{A set of addreses} \\
done \triangleq Val* &\quad todo \triangleq \mathsf{Exp}*
\end{aligned}
$$

Atomic Evaluation Function:

$$
\begin{aligned}
\mathcal{A}(x, \rho, \sigma) &\triangleq \sigma(\rho(x)) \\
\mathcal{A}(lam, \rho, \sigma) &\triangleq (lam, \rho) \\
\mathcal{A}(\text{æ}, \rho, \sigma) &\triangleq \text{æ}
\end{aligned}
$$

Transition Function:

$$(\Sigma \times Store) \rightsquigarrow (\Sigma \times Store)$$

$(\varsigma \times \sigma) \to (\varsigma' \times \sigma)$, where $\kappa = \sigma(a), b = alloc(\varsigma)$

proceed by matching on ς

$\langle(\texttt{if}\ e_c\ e_t\ e_f), \rho, a\rangle$	$\langle e_c, \rho, b\rangle$ $\sigma[b \mapsto \mathbf{ifk}(e_t, e_f, \rho, a)]$
$\langle(\texttt{let}\ (x\ e_x)\ e_b), \rho, a\rangle$	$\langle e_x, \rho, b\rangle$ $\sigma[b \mapsto \mathbf{letk}(x, e_b, \rho, a)]$
$\langle(\texttt{prim}\ op\ e_0\ es...), \rho, a\rangle$	$\langle e_0, \rho, b\rangle$ $\sigma[b \mapsto \mathbf{appk}([op], es, \rho, a)]$
$\langle(e_f\ es...), \rho, a\rangle$	$\langle e_f, \rho, b\rangle$ $\sigma[b \mapsto \mathbf{appk}([], e_s, \rho, a)]$
$\langle æ, \rho, a\rangle$ let $v = \mathcal{A}(æ, \rho, \sigma)$ match on κ below	
mt	ς
$\mathbf{ifk}(e_t, e_f, \rho', c)$ when $v = \texttt{\#f}$	$\langle e_f, \rho', c\rangle$
$\mathbf{ifk}(e_t, e_f, \rho', c)$ when $v \neq \texttt{\#f}$	$\langle e_t, \rho', c\rangle$
$\mathbf{letk}(x, e_b, \rho', c)$	$\langle e_b, \rho'[x \mapsto b], c\rangle$ $\sigma[b \mapsto v]$
$\mathbf{appk}(done, e_h :: e_t, \rho', c)$	$\langle e_h, \rho', b\rangle$ $\sigma[b \mapsto \mathbf{appk}(done \mathbin{+\!\!+} [v], e_t, \rho', c)]$
$\mathbf{appk}(op :: v_s, [], \rho', c)$	$\langle v', \rho', c\rangle$ $v' = op$ applied to $(v_s \mathbin{+\!\!+} [v])$
$\mathbf{appk}(clo :: v_s, [\,], \rho', c)$ where $clo = ((\lambda\ (xs...)\ e_b), \rho_\lambda)$	$\langle e_b, \rho_\lambda[xs_0 \mapsto b_0 ... xs_i \mapsto b_i], c\rangle$ $v'_s = v_s \mathbin{+\!\!+} [v]$ $\sigma[b_0 \mapsto v'_{s0} ... b_i \mapsto v'_{si}]$

2 Formalization of an Abstract CESK* machine with basic Scheme features.