1 Concrete Semantics of Scheme

Syntax Domains:

$$
\begin{aligned}
e \in \mathsf{Exp} ::=\ & æ \\
& \mid (\texttt{if}\ e\ e\ e) \\
& \mid (\texttt{let}\ ([x\ e]\ ...)\ e) \\
& \mid (\texttt{call/cc}\ e) \\
& \mid (\texttt{set!}\ x\ e) \\
& \mid (\texttt{prim}\ op\ e\ ...) \\
& \mid (\texttt{apply-prim}\ op\ e) \\
& \mid (\texttt{apply}\ e\ e) \\
& \mid (e\ e\ ...) \\
æ \in \mathsf{AExp} ::=\ & x \mid lam \mid \mathbb{Z} \mid \texttt{\#t} \mid \texttt{\#f} \\
& \mid (\texttt{quote}\ e) \\
lam \in \mathsf{Lam} ::=\ & (\lambda\ (x...)\ e) \mid (\lambda\ x\ e) \\
x \in \mathsf{Var} \quad & \text{A set of identifiers} \\
op \in \mathsf{Prim} \quad & \text{A set of primitives}
\end{aligned}
$$

Atomic Evaluation:

$$\mathcal{A} : \Sigma_E \rightharpoonup \mathit{Val}$$

$$
\begin{aligned}
\mathcal{A}(\langle lam, \rho, ...\rangle) &\triangleq (lam, \rho) \\
\mathcal{A}(\langle (\texttt{quote}\ e), ...\rangle) &\triangleq \mathbf{quote}(e) \\
\mathcal{A}(\langle x, \rho, \sigma, ...\rangle) &\triangleq \sigma(\rho(x)) \\
\mathcal{A}(\langle æ, ...\rangle) &\triangleq æ
\end{aligned}
$$

Injection:

$$\mathcal{I} : \mathsf{Exp} \to \Sigma_E$$

$$\mathcal{I}(e) \triangleq E\langle e, \varnothing, \varnothing, \{(0,0) : \mathbf{mt}\}, (0,0)\rangle$$

Store Joining:

$$\sigma \sqcup [a \mapsto v] \triangleq \sigma[a \mapsto v]$$

Collecting Semantics:

$$eval(e) = \{\varsigma \mid \mathcal{I}(e) \mapsto^* \varsigma\}$$

Semantic Domains:

$$
\begin{aligned}
\varsigma \in \Sigma &\triangleq E\langle \mathit{Eval}\rangle + A\langle \mathit{Apply}\rangle \\
\mathit{Eval} &\triangleq \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{Addr} \\
\mathit{Apply} &\triangleq \mathit{Val} \times \mathit{Store} \times \mathit{Addr} \\
\rho \in \mathit{Env} &\triangleq \mathsf{Var} \rightharpoonup \mathit{Addr} \\
\sigma \in \mathit{Store} &\triangleq \mathit{BAddr} \rightharpoonup \mathit{Val} \\
& \quad \times \mathit{KAddr} \rightharpoonup \mathit{Kont} \\
v \in \mathit{Val} &\triangleq \mathit{Clo} + \mathit{Kont} + \mathbb{Z} \\
& \quad + \{\texttt{\#t}, \texttt{\#f}, \mathit{Null}, \mathit{Void}\} \\
& \quad + \{\mathbf{quote}(e), \mathbf{cons}(v, v)\} \\
clo \in \mathit{Clo} &\triangleq \mathsf{Lam} \times \mathit{Env} \\
a \in \mathit{BAddr} &\triangleq \mathbb{N} \times \mathbb{N} \\
a_\kappa \in \mathit{KAddr} &\triangleq \mathbb{N} \\
\kappa \in \mathit{Kont} &::= \mathbf{mt} \\
& \mid \mathbf{ifk}(e, e, \rho, a_\kappa) \\
& \mid \mathbf{letk}(vars, done, todo, e, \rho, a_\kappa) \\
& \mid \mathbf{callcck}(a_\kappa) \\
& \mid \mathbf{setk}(x, \rho, a_\kappa) \\
& \mid \mathbf{primk}(op, done, todo, \rho, a_\kappa) \\
& \mid \mathbf{appprimk}(op, a_\kappa) \\
& \mid \mathbf{appk}(done, todo, \rho, a_\kappa) \\
& \mid \mathbf{appappk}(val?, e, \rho, a_\kappa) \\
done &\triangleq \mathit{Val}^* \\
todo &\triangleq \mathsf{Exp}^* \\
vars &\triangleq \mathsf{Var}^*
\end{aligned}
$$

Address Allocation:

$$
\begin{aligned}
balloc &: \Sigma \times \mathbb{N} \to \mathit{BAddr} \\
kalloc &: \Sigma \to \mathit{KAddr} \\
balloc(\langle \sigma, ...\rangle, n) &\triangleq (|\sigma|, n) \\
kalloc(\langle \sigma, ...\rangle) &\triangleq (|\sigma|)
\end{aligned}
$$

Eval Rules

Rules for when the control is an expression

$\varsigma = E\langle e_\varsigma, \rho, \sigma, a_\kappa\rangle$ Proceed by matching on e_ς

$$\text{æ} \rightsquigarrow A\langle v, \sigma, a_\kappa, \rangle$$

$$(\texttt{if}\ e_c\ e_t\ e_f) \rightsquigarrow E\langle e_c, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{ifk}(e_t, e_f, \rho, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(\texttt{let}\ ()\ e_b) \rightsquigarrow E\langle e_b, \rho, \sigma, a_\kappa\rangle$$

$$(\texttt{let}\ \mathit{binds}\ e_b) \rightsquigarrow E\langle e_0, \rho, \sigma', a'_\kappa\rangle$$
where $\mathit{binds} = ([x_0\ e_0]\ [x_s\ e_s]\ ...)$
$$a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$$
$$\kappa \triangleq \mathbf{letk}(x_0 :: x_s, [\,], e_s, e_b, \rho, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(\texttt{call/cc}\ e) \rightsquigarrow E\langle e, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{callcck}(a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a_\kappa \mapsto \kappa]$$

$$(\texttt{set!}\ x\ e) \rightsquigarrow E\langle e, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$a \triangleq \rho(x)$$
$$\kappa \triangleq \mathbf{setk}(a, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(\texttt{prim}\ \mathit{op}) \rightsquigarrow A\langle v, \sigma, a_\kappa, \rangle$$
where $v = \mathit{op}$ applied to 0 arguments

$$(\texttt{prim}\ \mathit{op}\ e_0\ e_s\ ...) \rightsquigarrow E\langle e_0, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{primk}(\mathit{op}, [\,], e_s, \rho, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(\texttt{apply-prim}\ \mathit{op}\ e) \rightsquigarrow E\langle e, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{appprimk}(\mathit{op}, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(\texttt{apply}\ e_f\ e_x) \rightsquigarrow E\langle e_f, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{appappk}(\varnothing, e_x, \rho, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

$$(e_f\ e_s\ ...) \rightsquigarrow E\langle e_f, \rho, \sigma', a'_\kappa\rangle$$
where $a'_\kappa \triangleq \mathit{kalloc}(\varsigma)$
$$\kappa \triangleq \mathbf{appk}([\,], e_s, \rho, a_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]$$

Apply Rules

Rules for when the control is a value

$\varsigma = A\langle v, \sigma, a_\kappa, \rangle$

$\kappa \triangleq \sigma(a_\kappa)$

Proceed by matching on κ

$$\mathbf{mt} \rightsquigarrow \varsigma$$

$$\mathbf{ifk}(\_, e_f, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_f, \rho_\kappa, \sigma, a'_\kappa\rangle$$
where $v = \texttt{\#f}$

$$\mathbf{ifk}(e_t, \_, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_t, \rho_\kappa, \sigma, a'_\kappa\rangle$$
where $v \neq \texttt{\#f}$

$$\mathbf{letk}(vars, done, [\,], e_b, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_b, \rho'_\kappa, \sigma', a'_\kappa\rangle$$
where $a_i \triangleq balloc(\varsigma, i)$

$$done' \triangleq done \mathbin{+\!\!+} [v]$$

$$\rho'_\kappa \triangleq \rho_\kappa[vars_0 \mapsto a_0 \ ... \ vars_{n-1} \mapsto a_{n-1}]$$

$$\sigma' \triangleq \sigma \sqcup [a_0 \mapsto done'_0 \ ... \ a_{n-1} \mapsto done'_{n-1}]$$

$$\mathbf{letk}(vars, done, e_h :: e_t, e_b, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_h, \rho_\kappa, \sigma', a''_\kappa\rangle$$
where $a''_\kappa \triangleq kalloc(\varsigma)$

$$\kappa' \triangleq \mathbf{letk}(vars, done \mathbin{+\!\!+} [v], e_t, e_b, \rho_\kappa, a'_\kappa)$$

$$\sigma' \triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa']$$

$$\mathbf{setk}(a, a'_\kappa) \rightsquigarrow A\langle Void, \sigma', a'_\kappa, \rangle$$
where $\sigma' \triangleq \sigma \sqcup [a \mapsto v]$

$$\mathbf{callcck}(a'_\kappa) \rightsquigarrow E\langle e, \rho'_\lambda, \sigma', a'_\kappa\rangle$$
where $v = ((\lambda\ (x)\ e), \rho_\lambda)$

$$a \triangleq balloc(\varsigma, 0)$$

$$\kappa' \triangleq \sigma(a'_\kappa)$$

$$\rho'_\lambda \triangleq \rho_\lambda[x \mapsto a]$$

$$\sigma' \triangleq \sigma \sqcup [a \mapsto \kappa']$$

$$\mathbf{callcck}(\_) \rightsquigarrow A\langle \kappa, \sigma', a'_\kappa, \rangle$$
where $v = \kappa'$

$$a'_\kappa \triangleq kalloc(\varsigma)$$

$$\sigma' \triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa']$$

$$\mathbf{appprimk}(op, a'_\kappa) \rightsquigarrow A\langle v', \sigma, a'_\kappa, \rangle$$
where $v' \triangleq op$ applied to v

$$\mathbf{primk}(op, done, [\,], \_, a'_\kappa) \rightsquigarrow A\langle v', \sigma, a'_\kappa, \rangle$$
where $v' \triangleq op$ applied to $(done \mathbin{+\!\!+} [v])$

$$\mathbf{primk}(op, done, e_h :: e_t, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_h, \rho_\kappa, \sigma', a''_\kappa\rangle$$
where $a''_\kappa \triangleq kalloc(\varsigma)$

$$\kappa' \triangleq \mathbf{primk}(op, done \mathbin{+\!\!+} [v], e_t, \rho_\kappa, a'_\kappa)$$

$$\sigma' \triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa']$$

More Apply Rules

Rules for when the control is a value

$$\mathbf{appappk}(v_f, \_, \_, a'_\kappa) \rightsquigarrow E\langle e_b, \rho'_\lambda, \sigma', a'_\kappa\rangle$$
where $v_f = ((\lambda\ (x_s...)\ e_b), \rho_\lambda)$
$$\begin{aligned} a_i &\triangleq balloc(\varsigma, i) \\ \rho'_\lambda &\triangleq \rho_\lambda[x_0 \mapsto a_0\ ... \\ &\qquad x_{n-1} \mapsto a_{n-1}] \\ \sigma' &\triangleq \sigma \sqcup [a_0 \mapsto v_0\ ... \\ &\qquad a_{n-1} \mapsto v_{n-1}] \end{aligned}$$

$$\mathbf{appappk}(v_f, \_, \_, a'_\kappa) \rightsquigarrow E\langle e_b, \rho'_\lambda, \sigma', a'_\kappa\rangle$$
where $v_f = ((\lambda\ x\ e_b), \rho_\lambda)$
$$\begin{aligned} a &\triangleq balloc(\varsigma, 0) \\ \rho'_\lambda &\triangleq \rho_\lambda[x \mapsto a] \\ \sigma' &\triangleq \sigma \sqcup [a \mapsto v] \end{aligned}$$

$$\mathbf{appappk}(\varnothing, e, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e, \rho_\kappa, \sigma', a''_\kappa\rangle$$
where $a''_\kappa \triangleq kalloc(\varsigma)$
$$\begin{aligned} \kappa' &\triangleq \mathbf{appappk}(v, e, \rho_\kappa, a'_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa'] \end{aligned}$$

$$\mathbf{appk}(done, [\,], \_, a'_\kappa) \rightsquigarrow E\langle e_b, \rho'_\lambda, \sigma', a'_\kappa\rangle$$
where $done = ((\lambda\ (x_s...)\ e_b), \rho_\lambda) :: v_s$
$$\begin{aligned} a_i &\triangleq balloc(\varsigma, i) \\ v'_s &\triangleq v_s +\!\!+ [v] \\ \rho'_\lambda &\triangleq \rho_\lambda[x_0 \mapsto a_0\ ... \\ &\qquad x_{n-1} \mapsto a_{n-1}] \\ \sigma' &\triangleq \sigma \sqcup [a_0 \mapsto v'_0\ ... \\ &\qquad a_{n-1} \mapsto v'_{n-1}] \end{aligned}$$

$$\mathbf{appk}(done, [\,], \_, a'_\kappa) \rightsquigarrow E\langle e_b, \rho'_\lambda, \sigma', a'_\kappa\rangle$$
where $done = ((\lambda\ x\ e_b), \rho_\lambda) :: v_s$
$$\begin{aligned} a &\triangleq balloc(\varsigma, 0) \\ v'_s &\triangleq v_s +\!\!+ [v] \\ \rho'_\lambda &\triangleq \rho_\lambda[x \mapsto a] \\ \sigma' &\triangleq \sigma \sqcup [a \mapsto v'_s] \end{aligned}$$

$$\mathbf{appk}([\kappa_\lambda], [\,], \rho_\kappa, a'_\kappa) \rightsquigarrow A\langle v, \sigma', a''_\kappa, \rangle$$
where $a''_\kappa \triangleq kalloc(\varsigma)$
$$\sigma' \triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa_\lambda]$$

$$\mathbf{appk}(done, e_h :: e_t, \rho_\kappa, a'_\kappa) \rightsquigarrow E\langle e_h, \rho_\kappa, \sigma', a''_\kappa\rangle$$
where $a''_\kappa \triangleq kalloc(\varsigma)$
$$\begin{aligned} \kappa' &\triangleq \mathbf{appk}(done +\!\!+ [v], e_t, \rho_\kappa, a'_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa'] \end{aligned}$$

2 Abstract Semantics of Scheme

Abstract Semantic Domains:

$$\hat{\varsigma} \in \hat{\Sigma} \triangleq E\langle Eval\rangle + E\langle Apply\rangle$$
$$Eval \triangleq \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{KAddr}$$
$$Apply \triangleq \widehat{Val} \times \widehat{Store} \times \widehat{KAddr}$$
$$\hat{\rho} \in \widehat{Env} \triangleq \mathsf{Var} \rightharpoonup \widehat{Addr}$$
$$\hat{\sigma} \in \widehat{Store} \triangleq \widehat{BAddr} \rightharpoonup \widehat{Val} \times \widehat{KAddr} \rightharpoonup \mathcal{P}(\widehat{Kont})$$
$$\hat{v} \in \widehat{Val} \triangleq (\widehat{CVal} + \top + \bot) \times \mathcal{P}(\widehat{Clo})$$
$$\hat{cv} \in \widehat{CVal} \triangleq \widehat{Kont} + \mathbb{Z} + \{\texttt{\#t}, \texttt{\#f}, Null, Void\} + \{\mathbf{quote}(e), \mathbf{cons}(\hat{v}, \hat{v})\}$$
$$\widehat{clo} \in \widehat{Clo} \triangleq \mathsf{Lam} \times \widehat{Env}$$
$$\hat{a} \in \widehat{BAddr} \triangleq \mathsf{Var} \times \mathsf{Expr}$$
$$\hat{a}_{\hat{\kappa}} \in \widehat{KAddr} \triangleq \mathsf{Expr} \times \widehat{Env}$$
$$\widehat{done} \triangleq \widehat{Val}^*$$

$$\begin{aligned}
\hat{\kappa} \in \widehat{Kont} ::= {} & \mathbf{mt} \\
& \mid \mathbf{ifk}(e, e, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{letk}(e, vars, \widehat{done}, todo, e, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{callcck}(e, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{setk}(\hat{a}, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{primk}(op, \widehat{done}, todo, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{appprimk}(op, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{appk}(e, \widehat{done}, todo, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\
& \mid \mathbf{appappk}(e, \hat{v}?, e, \hat{\rho}, \hat{a}_{\hat{\kappa}})
\end{aligned}$$

Abstract Atomic Evaluation:

$$\mathcal{A} : Eval \to \widehat{Val}$$
$$\mathcal{A}(E\langle lam, \hat{\rho}, ...\rangle) \triangleq (\bot, \{(lam, \hat{\rho})\})$$
$$\mathcal{A}(æ) \triangleq (æ, \emptyset)$$

Allocation:

$$\widehat{balloc} : \mathsf{Var} \times \widehat{Env} \to \widehat{BAddr}$$
$$\widehat{balloc}(x, \hat{\rho}) \triangleq (x, \hat{\rho})$$
$$\widehat{kalloc} : \mathsf{Expr} \times \widehat{Env} \to \widehat{KAddr}$$
$$\widehat{kalloc}(e, \hat{\rho}) \triangleq (e, \hat{\rho})$$
$$\widehat{new\rho} : \mathsf{Lam} \times \widehat{Env} \times \mathsf{Lam} \times \widehat{Env} \to \widehat{Env}$$
$$\widehat{new\rho}(e_{call}, \hat{\rho}, e_\lambda, \hat{\rho}') \triangleq \begin{cases} first_m(call : \hat{\rho}) & e_\lambda \text{ is proc} \\ \hat{\rho}' & e_\lambda \text{ is kont} \end{cases}$$

Store Joining:

$$\sigma \sqcup [\hat{a}_{\hat{\kappa}} \mapsto \hat{\kappa}] \triangleq \sigma[\hat{a}_{\hat{\kappa}} \mapsto \sigma(\hat{a}_{\hat{\kappa}}) \cup \{\hat{\kappa}\}]$$
$$\sigma \sqcup [a \mapsto (\hat{cv}_i, \widehat{clo}_i)] \triangleq$$
match on $\sigma[a]$

$$\sigma[a \mapsto \begin{cases} (\hat{cv}_i, \widehat{clo}_i) & \text{if } \texttt{empty} \\ (\hat{cv}_i, \widehat{clo}_i \cup \widehat{clo}_e) & \text{if } (\bot, \widehat{clo}_e) \\ (\hat{cv}_e, \widehat{clo}_i \cup \widehat{clo}_e)) & \text{if } (\hat{cv}_e, \widehat{clo}_e) \\ & \quad \wedge\ \hat{cv}_i = \bot \\ (\hat{cv}_i, \widehat{clo}_i \cup \widehat{clo}_e) & \text{if } (\hat{cv}_e, \widehat{clo}_e) \\ & \quad \wedge\ \hat{cv}_i = \hat{cv}_e \\ (\top, \widehat{clo}_i \cup \widehat{clo}_e) & \text{if } (\_, \widehat{clo}_e) \end{cases}]$$

Abstract Eval Rules

Rules for when the control is an expression

$$\hat{\varsigma} = E\langle e_{\hat{\varsigma}}, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle$$

Proceed by matching on $e_{\hat{\varsigma}}$

$$\text{æ} \rightsquigarrow A\langle \hat{v}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle$$
$$\text{where } \hat{v} \triangleq \hat{\mathcal{A}}(\hat{\varsigma})$$

$$(\texttt{if}\ e_c\ e_t\ e_f) \rightsquigarrow E\langle e_c, \hat{\rho}, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{ifk}(e_t, e_f, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{let}\ ()\ e_b) \rightsquigarrow E\langle e_b, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle$$

$$(\texttt{let}\ bnds\ e_b) \rightsquigarrow E\langle e_0, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } bnds &= ([x_0\ e_0]\ [x_s\ e_s]\ ...) \\ \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{letk}(e_{\hat{\varsigma}}, x_0 :: x_s, [\,], \\ &\qquad e_s, e_b, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{call/cc}\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{callcck}(e_{\hat{\varsigma}}, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{set!}\ x\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{a} &\triangleq \hat{\rho}(x) \\ \hat{\kappa} &\triangleq \textbf{setk}(\hat{a}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{prim}\ op) \rightsquigarrow A\langle \hat{v}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle$$
$$\text{where } \hat{v} = op \text{ applied to 0 arguments}$$

$$(\texttt{prim}\ op\ e_0\ e_s\ ...) \rightsquigarrow E\langle e_0, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{primk}(op, [\,], e_s, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{apply-prim}\ op\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{appprimk}(op, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(\texttt{apply}\ e_f\ e_x) \rightsquigarrow E\langle e_f, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{appappk}(e_{\hat{\varsigma}}, \varnothing, e_x, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

$$(e_f\ e_s\ ...) \rightsquigarrow E\langle e_f, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$
$$\begin{aligned} \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}) \\ \hat{\kappa} &\triangleq \textbf{appk}(e_{\hat{\varsigma}}, [\,], e_s, \hat{\rho}, \hat{a}_{\hat{\kappa}}) \\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}] \end{aligned}$$

Apply Rules

Rules for when the control is a value

$\hat{\varsigma} = A\langle \hat{v}, \hat{\sigma}, \hat{a}_{\kappa\varsigma} \rangle$

$\kappa_\varsigma \in \hat{\sigma}(\hat{a}_{\kappa\varsigma})$

Proceed by matching on κ_ς

$$\mathbf{mt} \rightsquigarrow \emptyset$$

$$\mathbf{ifk}(\_, e_f, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_f, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle\}$$
where $\hat{v} = (\texttt{\#f}, \emptyset)$

$$\mathbf{ifk}(e_t, \_, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_t, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle\}$$
where $\hat{v} = (\mathit{not}\ \texttt{\#f}\ \mathit{nor}\ \top, \_)$

$$\mathbf{ifk}(e_t, e_f, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_t, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle, E\langle e_f, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle\}$$
where $\hat{v} \in \{(\texttt{\#f}, \mathit{not}\ \emptyset), (\top, \_)\}$

$$\mathbf{letk}(e_{\hat{\varsigma}}, \mathit{vars}, \widehat{\mathit{done}}, [\,], e_b, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_b, \hat{\rho}'_{\hat{\kappa}}, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}$$
where $\hat{a}_i \triangleq \widehat{\mathit{balloc}}(\mathit{vars}_i, e_{\hat{\varsigma}})$

$\hat{\rho}'_{\hat{\kappa}} \triangleq \hat{\rho}_{\hat{\kappa}}[\mathit{vars}_0 \mapsto \hat{a}_0 \ldots \mathit{vars}_{n-1} \mapsto \hat{a}_{n-1}]$

$\widehat{\mathit{done}}' \triangleq \widehat{\mathit{done}} +\!\!+ [\hat{v}]$

$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a}_0 \mapsto \widehat{\mathit{done}}'_0 \ldots \hat{a}_{n-1} \mapsto \widehat{\mathit{done}}'_{n-1}]$

$$\mathbf{letk}(e_{\hat{\varsigma}}, \mathit{vars}, \widehat{\mathit{done}}, e_h :: e_t, e_b, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_h, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle\}$$
where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{\mathit{kalloc}}(e_{\hat{\varsigma}}, \hat{\rho}_{\hat{\kappa}})$

$\hat{\kappa} \triangleq \mathbf{letk}(e_{\hat{\varsigma}}, \mathit{vars}, \widehat{\mathit{done}} +\!\!+ [\hat{v}], e_t, e_b, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}})$

$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}]$

$$\mathbf{setk}(\hat{a}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{A\langle (\mathit{Void}, \emptyset), \hat{\sigma}', \hat{a}_{\hat{\kappa}}, \rangle\}$$
where $\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{v}]$

$$\mathbf{callcck}(e_{\hat{\varsigma}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_b, \hat{\rho}'_\lambda, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}$$
where $\hat{v} \ni (\_, ((\lambda\ (x)\ e_b), \hat{\rho}_\lambda))$

$\hat{a} \triangleq \widehat{\mathit{balloc}}(x, e_{\hat{\varsigma}})$

$\hat{\kappa} \in \hat{\sigma}(\hat{a}_{\hat{\kappa}})$

$\hat{\rho}'_\lambda \triangleq \hat{\rho}_\lambda[x \mapsto \hat{a}]$

$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{\kappa}]$

$$\mathbf{callcck}(\_, \_) \rightsquigarrow \{A\langle \kappa_\varsigma, \hat{\sigma}', \hat{a}_{\kappa\varsigma}, \rangle\}$$
where $\hat{v} = (\hat{\kappa}, \_)$

$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a}_{\kappa\varsigma} \mapsto \hat{\kappa}]$

$$\mathbf{appprimk}(\mathit{op}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{A\langle \hat{v}', \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle\}$$
where $\hat{v}' \triangleq \mathit{op}$ applied to $\hat{v}$

$$\mathbf{primk}(\mathit{op}, \widehat{\mathit{done}}, [\,], \_, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{A\langle \hat{v}', \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle\}$$
where $\hat{v}' \triangleq \mathit{op}$ applied to $(\widehat{\mathit{done}} +\!\!+ [\hat{v}])$

$$\mathbf{primk}(\mathit{op}, \widehat{\mathit{done}}, e_h :: e_t, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_h, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle\}$$
where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{\mathit{kalloc}}(e_h, \hat{\rho}_{\hat{\kappa}})$

$\hat{\kappa} \triangleq \mathbf{primk}(\mathit{op}, \widehat{\mathit{done}} +\!\!+ [\hat{v}], e_t, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}})$

$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}]$

More Apply Rules

Rules for when the control is a value

$$\begin{aligned}\textbf{appappk}(e_{\hat{\varsigma}}, \hat{v}_f, \_, \_, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_b, \hat{\rho}'_\lambda, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{v}_f &\ni (\_, ((\lambda\ (x_s...)\ e_b), \hat{\rho}_\lambda))\\ \hat{a}_i &\triangleq \widehat{balloc}(x_i, e_{\hat{\varsigma}})\\ \hat{\rho}'_\lambda &\triangleq \hat{\rho}_\lambda[x_0 \mapsto \hat{a}_0\ ...\\ &\qquad x_{n-1} \mapsto \hat{a}_{n-1}]\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}_0 \mapsto \hat{v}_0\ ...\\ &\qquad \hat{a}_{n-1} \mapsto \hat{v}_{n-1}]\end{aligned}$$

$$\begin{aligned}\textbf{appappk}(e_{\hat{\varsigma}}, \hat{v}_f, \_, \_, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_b, \hat{\rho}'_\lambda, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{v}_f &\ni (\_, ((\lambda\ x\ e_b), \hat{\rho}_\lambda))\\ \hat{a} &\triangleq \widehat{balloc}(x, e_{\hat{\varsigma}})\\ \hat{\rho}'_\lambda &\triangleq \hat{\rho}_\lambda[x \mapsto \hat{a}]\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{v}]\end{aligned}$$

$$\begin{aligned}\textbf{appappk}(e_{\hat{\varsigma}}, \varnothing, e_x, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_x, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_x, \hat{\rho}_{\hat{\kappa}})\\ \hat{\kappa} &\triangleq \textbf{appappk}(e_{\hat{\varsigma}}, \hat{v}, e_x, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}})\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}]\end{aligned}$$

$$\begin{aligned}\textbf{appk}(e_{\hat{\varsigma}}, \hat{v}_h :: \hat{v}_t, [\,], \_, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_b, \hat{\rho}'_\lambda, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{v}_h &\ni (\_, ((\lambda\ (x_s...)\ e_b), \hat{\rho}_\lambda))\\ \hat{a}_i &\triangleq \widehat{balloc}(x_i, e_{\hat{\varsigma}})\\ \hat{\rho}'_\lambda &\triangleq \hat{\rho}_\lambda[x_0 \mapsto \hat{a}_0...\\ &\qquad x_{n-1} \mapsto \hat{a}_{n-1}]\\ \hat{v}'_t &\triangleq \hat{v}_t +\!\!+ [\hat{v}]\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}_0 \mapsto \hat{v}'_0...\\ &\qquad \hat{a}_{n-1} \mapsto \hat{v}'_{n-1}]\end{aligned}$$

$$\begin{aligned}\textbf{appk}(e_{\hat{\varsigma}}, \hat{v}_h :: \hat{v}_t, [\,], \_, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_b, \hat{\rho}'_\lambda, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{v}_h &\ni (\_, ((\lambda\ x\ e_b), \hat{\rho}_\lambda))\\ \hat{a} &\triangleq \widehat{balloc}(x, e_{\hat{\varsigma}})\\ \hat{\rho}'_\lambda &\triangleq \hat{\rho}_\lambda[x \mapsto \hat{a}]\\ \hat{v}'_t &\triangleq \hat{v}_t +\!\!+ [\hat{v}]\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a} \mapsto \hat{v}'_t]\end{aligned}$$

$$\begin{aligned}\textbf{appk}(e_{\hat{\varsigma}}, [\hat{v}_f], [\,].\hat{\rho}_{\hat{\kappa}}, \_) &\rightsquigarrow \{A\langle \hat{v}, \hat{\sigma}', \hat{a}_{\hat{\kappa}}, \rangle\}\\ \text{where } \hat{v}_f &= (\hat{\kappa}_\lambda, \_)\\ \hat{a}_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho}_{\hat{\kappa}})\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}_{\hat{\kappa}} \mapsto \hat{\kappa}_\lambda]\end{aligned}$$

$$\begin{aligned}\textbf{appk}(e_{\hat{\varsigma}}, \widehat{done}, e_h :: e_t, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) &\rightsquigarrow \{E\langle e_h, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle\}\\ \text{where } \hat{a}'_{\hat{\kappa}} &\triangleq \widehat{kalloc}(e_h, \hat{\rho}_{\hat{\kappa}})\\ \hat{\kappa} &\triangleq \textbf{appk}(e_{\hat{\varsigma}}, \widehat{done} +\!\!+ [\hat{v}], e_t, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}})\\ \hat{\sigma}' &\triangleq \hat{\sigma} \sqcup [\hat{a}'_{\hat{\kappa}} \mapsto \hat{\kappa}]\end{aligned}$$

Concrete CESK* With Flat Closures

Syntactic Domains

$$
\begin{aligned}
e \in \mathsf{Exp} &::= \text{æ} \mid \mathit{call} \\
\text{æ} \in \mathsf{AExp} &::= x \mid \mathit{lam} \\
x \in \mathsf{Var} &\triangleq \text{The set of identifiers} \\
\mathit{call} \in \mathsf{Call} &::= (e\ e\ ...) \\
\mathit{lam} \in \mathsf{Lam} &::= (\lambda\ (x)\ e)
\end{aligned}
$$

Injection

$$
\begin{gathered}
\mathit{inj} : \mathsf{Exp} \to \Sigma \\
\mathit{inj}(e) \triangleq (e, (0, \epsilon), \{\kappa_0 : \mathbf{mt}\}, \kappa_0)
\end{gathered}
$$

Allocation

$$
\begin{gathered}
\mathit{new}\rho : \mathsf{Call} \times \mathit{Env} \to \mathit{Env} \\
\mathit{new}\rho(\mathit{call}, (n, \overrightarrow{\mathit{call}})) \triangleq (n+1, \mathit{call} : \overrightarrow{\mathit{call}})
\end{gathered}
$$

Semantic Domains

$$
\begin{aligned}
\varsigma \in \Sigma &\triangleq E\langle \mathit{Eval} \rangle + A\langle \mathit{Apply} \rangle \\
\mathit{Eval} &\triangleq \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{KAddr} \\
\mathit{Apply} &\triangleq \mathit{Val} \times \mathit{Env} \times \mathit{Store} \times \mathit{KAddr} \\
\rho \in \mathit{Env} &\triangleq \mathbb{N} \times \mathsf{Call}^* \\
\sigma \in \mathit{Store} &\triangleq \mathit{BAddr} \rightharpoonup \mathit{Val} \\
&\quad \times \mathit{KAddr} \rightharpoonup \mathit{Kont} \\
a \in \mathit{BAddr} &\triangleq \mathsf{Var} \times \mathit{Env} \\
a_\kappa \in \mathit{KAddr} &\triangleq \mathsf{Exp} \times \mathit{Env} \\
v \in \mathit{Val} &\triangleq \mathit{Clo} \\
\mathit{clo} \in \mathit{Clo} &\triangleq \mathsf{Lam} \times \mathit{Env} \\
\kappa \in \mathit{Kont} &::= \mathbf{mt} \\
&\mid \mathbf{callk}(\mathit{call}, \mathit{done}, \mathit{todo}, \rho, a_\kappa) \\
\mathit{done} &\in \mathit{Val}^* \\
\mathit{todo} &\in \mathsf{Exp}*
\end{aligned}
$$

Semantics

$$E\langle æ, \rho, \sigma, a_\kappa\rangle \rightsquigarrow A\langle v, \rho, \sigma, a_\kappa\rangle$$
$$\text{where } v \triangleq \mathcal{A}(\varsigma)$$

$$E\langle call, \rho, \sigma, a_\kappa\rangle \rightsquigarrow E\langle e_f, \rho, \sigma', a'_\kappa\rangle$$
$$\begin{aligned} \text{where } call &= (e_f\ e_s\ ...) \\ a'_\kappa &\triangleq (e_f, \rho) \\ \kappa &\triangleq \mathbf{callk}(call, \epsilon, e_s, \rho, a_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa] \end{aligned}$$

$$A\langle\varsigma\rangle \rightsquigarrow \varsigma$$
$$\text{where } \sigma(a_\kappa) = \mathbf{mt}$$

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow E\langle e_b, \rho', \sigma', a'_\kappa\rangle$$
$$\begin{aligned} \text{where } \sigma(a_\kappa) &= \mathbf{callk}(call, done, \epsilon, \rho_\kappa, a'_\kappa) \\ done &= ((\lambda\ (x\ ...)\ e_b), \rho_\lambda) :: v' \\ v'' &\triangleq v' \mathbin{+\!\!+} v \\ \rho' &\triangleq new\rho(call, \rho) \\ a_{x_i} &\triangleq (x_i, \rho') \\ x'_j &\triangleq free((\lambda\ (x\ ...)\ e_b)) \\ a_{x'_j} &\triangleq (x'_j, \rho') \\ \sigma' &\triangleq \sigma \sqcup [a_{x_i} \mapsto v''_i] \sqcup [a_{x'_j} \mapsto \sigma(x'_j, \rho_\lambda)] \end{aligned}$$

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow E\langle e_h, \rho_\kappa, \sigma', a''_\kappa\rangle$$
$$\begin{aligned} \text{where } \sigma(a_\kappa) &= \mathbf{callk}(call, done, e_h :: e_t, \rho_\kappa, a'_\kappa) \\ a''_\kappa &\triangleq (e_h, \rho_\kappa) \\ \kappa &\triangleq \mathbf{callk}(call, done \mathbin{+\!\!+} [v], e_t, \rho_\kappa, a'_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa] \end{aligned}$$