Concrete Scheme CESK* With Flat Closures

Syntactic Domains

$$
\begin{aligned}
e \in \mathsf{Exp} &::= æ \\
&\mid (\texttt{if}\ e\ e\ e) \mid (\texttt{set!}\ x\ e) \\
&\mid (\texttt{call/cc}\ e) \\
&\mid \mathit{apply} \mid \mathit{let} \mid \mathit{call} \\
æ \in \mathsf{AExp} &::= x \mid \mathit{lam} \mid \mathit{op} \\
&\mid (\texttt{quote}\ e) \mid b \mid n \\
n &\in \mathbb{Z} \\
b \in \mathbb{B} &\triangleq \{\texttt{\#t}, \texttt{\#f}\} \\
x \in \mathsf{Var} &\triangleq \text{The set of identifiers} \\
\mathit{op} \in \mathsf{Prim} &\triangleq \text{The set of prims} \\
\mathit{apply} \in \mathsf{Apply} &::= (\texttt{apply}\ e\ e) \\
\mathit{call} \in \mathsf{Call} &::= (e\ e\ ...) \\
\mathit{let} \in \mathsf{Let} &::= (\texttt{let}\ ([x\ e]\ ...)\ e) \\
\mathit{lam} \in \mathsf{Lam} &::= (\lambda\ (x)\ e) \mid (\lambda\ x\ e)
\end{aligned}
$$

Semantic Domains

$$
\begin{aligned}
\varsigma \in \Sigma &\triangleq E\langle \mathit{Eval}\rangle + A\langle \mathit{Apply}\rangle \\
\mathit{Eval} &\triangleq \mathsf{Exp} \times \mathit{Env} \times \mathit{Store} \times \mathit{KAddr} \\
\mathit{Apply} &\triangleq \mathit{Val} \times \mathit{Env} \times \mathit{Store} \times \mathit{KAddr} \\
\rho \in \mathit{Env} &\triangleq \mathbb{N} \times \mathsf{Exp}^* \\
\sigma \in \mathit{Store} &\triangleq \mathit{BAddr} \rightharpoonup \mathit{Val} \\
&\quad \times \mathit{KAddr} \rightharpoonup \mathit{Kont} \\
a \in \mathit{BAddr} &\triangleq \mathsf{Var} \times \mathit{Env} \\
a_\kappa \in \mathit{KAddr} &\triangleq \mathbb{N} \\
v \in \mathit{Val} &\triangleq \mathit{Clo} + \mathbb{Z} + \mathbb{B} \\
&\quad + \mathsf{Prim} + \mathit{Kont} \\
&\quad + \{\mathbf{quote}(e), \mathbf{cons}(v, v), \mathit{Void}\} \\
\mathit{clo} \in \mathit{Clo} &\triangleq \mathsf{Lam} \times \mathit{Env} \\
\kappa \in \mathit{Kont} &::= \mathbf{mt} \\
&\mid \mathbf{ifk}(e, e, \rho, a_\kappa) \\
&\mid \mathbf{setk}(a, a_\kappa) \\
&\mid \mathbf{callcck}((\texttt{call/cc}\ e), a_\kappa) \\
&\mid \mathbf{applyk}(\mathit{apply}, v?, e, \rho, a_\kappa) \\
&\mid \mathbf{callk}(\mathit{call}, \mathit{done}, \mathit{todo}, \rho, a_\kappa) \\
\mathit{done} &\in \mathit{Val}^* \\
\mathit{todo} &\in \mathsf{Exp}*
\end{aligned}
$$

Useful Functions

Callable helper function

$$CALL : Val \times Val^* \times Env \times Store \times \mathsf{Exp} \times KAddr \to \Sigma$$

$$CALL(clo, \overrightarrow{v}, \rho, \sigma, e, a_\kappa) \triangleq E\langle e_b, \rho', \sigma', a_\kappa\rangle$$

where $clo = ((\lambda\ (x\ ...)\ e_b), \rho_\lambda)$

$$\rho' \triangleq new\rho(e, \rho)$$
$$a_{x_i} \triangleq (x_i, \rho')$$
$$x'_j \triangleq free((\lambda\ (x\ ...)\ e_b))$$
$$a_{x'_j} \triangleq (x'_j, \rho')$$
$$\sigma' \triangleq \sigma \sqcup [a_{x_i} \mapsto \overrightarrow{v_i}] \sqcup [a_{x'_j} \mapsto \sigma(x'_j, \rho_\lambda)]$$

$$CALL(clo, \overrightarrow{v}, \rho, \sigma, e, a_\kappa) \triangleq E\langle e_b, \rho', \sigma', a_\kappa\rangle$$

where $clo = ((\lambda\ x\ e_b), \rho_\lambda)$

$$\rho' \triangleq new\rho(e, \rho)$$
$$a_x \triangleq (x, \rho')$$
$$x'_j \triangleq free((\lambda\ x\ e_b))$$
$$a_{x'_j} \triangleq (x'_j, \rho')$$
$$\sigma' \triangleq \sigma \sqcup [a_x \mapsto \overrightarrow{v}] \sqcup [a_{x'_j} \mapsto \sigma(x'_j, \rho_\lambda)]$$

$$CALL(\kappa, [v], \rho, \sigma, e, \_) \triangleq A\langle v, \rho, \sigma', a_\kappa\rangle$$

where $a_\kappa \triangleq |\sigma|$

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

$$CALL(op, \overrightarrow{v}, \rho, \sigma, \_, a_\kappa) \triangleq A\langle v, \rho, \sigma, a_\kappa\rangle$$

where $v \triangleq op$ applied to $\overrightarrow{v}$

Injection

$$inj : \mathsf{Exp} \to \Sigma$$
$$inj(e) \triangleq (e, (0, \epsilon), \{0 : \mathbf{mt}\}, 0)$$

Allocation

$$new\rho : \mathsf{Exp} \times Env \to Env$$
$$new\rho(e, (n, \overrightarrow{e})) \triangleq (n + 1, e :: \overrightarrow{e})$$

Atomic Evaluation

$$\mathcal{A} : Eval \to Val$$
$$\mathcal{A}(E\langle n, \_, \_, \_\rangle) \triangleq n$$
$$\mathcal{A}(E\langle b, \_, \_, \_\rangle) \triangleq b$$
$$\mathcal{A}(E\langle (\texttt{quote}\ e), \_, \_, \_\rangle) \triangleq \mathbf{quote}(e)$$
$$\mathcal{A}(E\langle op, \rho, \sigma, \_\rangle) \triangleq op \text{ when } (op, \rho) \notin \sigma$$
$$\mathcal{A}(E\langle lam, \rho, \_, \_\rangle) \triangleq (lam, \rho)$$
$$\mathcal{A}(E\langle x, \rho, \sigma, \_\rangle) \triangleq \sigma(x, \rho)$$

Store Joining

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

Eval Semantics

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

$$E\langle(\texttt{if}\ e_c\ e_t\ e_f), \rho, \sigma, a_\kappa\rangle \leadsto E\langle e_c, \rho, \sigma', a'_\kappa\rangle$$
$$\begin{aligned}\text{where } a'_\kappa &\triangleq |\sigma| \\ \kappa &\triangleq \mathbf{ifk}(e_t, e_f, \rho, a_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]\end{aligned}$$

$$E\langle let, \rho, \sigma, a_\kappa\rangle \leadsto E\langle call, \rho, \sigma, a_\kappa\rangle$$
$$\begin{aligned}\text{where } let &= (\texttt{let}\ ([x_s\ e_s]\ ...)\ e_b) \\ call &= ((\lambda\ (x_s\ ...)\ e_b)\ e_s)\end{aligned}$$

$$E\langle(\texttt{set!}\ x\ e), \rho, \sigma, a_\kappa\rangle \leadsto E\langle e, \rho, \sigma', a'_\kappa\rangle$$
$$\begin{aligned}\text{where } a &\triangleq (x, \rho) \\ a'_\kappa &\triangleq |\sigma| \\ \kappa &\triangleq \mathbf{setk}(a, a_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]\end{aligned}$$

$$E\langle(\texttt{call/cc}\ e), \rho, \sigma, a_\kappa\rangle \leadsto E\langle e, \rho, \sigma', a'_\kappa\rangle$$
$$\begin{aligned}\text{where } a'_\kappa &\triangleq |\sigma| \\ \kappa &\triangleq \mathbf{callcck}((\texttt{call/cc}\ e), a_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]\end{aligned}$$

$$E\langle(\texttt{apply}\ e_f\ e), \rho, \sigma, a_\kappa\rangle \leadsto E\langle e_f, \rho, \sigma', a'_\kappa\rangle$$
$$\begin{aligned}\text{where } a'_\kappa &\triangleq |\sigma| \\ \kappa &\triangleq \mathbf{applyk}((\texttt{apply}\ e_f\ e), \varnothing, e, \rho, a_\kappa) \\ \sigma' &\triangleq \sigma \sqcup [a'_\kappa \mapsto \kappa]\end{aligned}$$

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

Apply Semantics

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

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

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

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

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow \varsigma'$$
$$\text{where}$$
$$\sigma(a_\kappa) = \mathbf{callcck}(e, a'_\kappa)$$
$$\varsigma' \triangleq \mathit{CALL}(v, [\sigma(a'_\kappa)], \rho, \sigma, e, a'_\kappa)$$

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow E\langle e, \rho_\kappa, \sigma', a''_\kappa\rangle$$
$$\text{where } \sigma(a_\kappa) = \mathbf{applyk}(\mathit{apply}, \varnothing, e, \rho_\kappa, a'_\kappa)$$
$$a''_\kappa \triangleq |\sigma|$$
$$\kappa \triangleq \mathbf{applyk}(\mathit{apply}, v, e, \rho_\kappa, a'_\kappa)$$
$$\sigma' \triangleq \sigma \sqcup [a''_\kappa \mapsto \kappa]$$

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow \varsigma'$$
$$\text{where } \sigma(a_\kappa) = \mathbf{applyk}(\mathit{apply}, v_f, \_, \_, a'_\kappa)$$
$$\varsigma' \triangleq \mathit{CALL}(v_f, v, \rho, \sigma, \mathit{apply}, a'_\kappa)$$

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

$$A\langle v, \rho, \sigma, a_\kappa\rangle \rightsquigarrow \varsigma'$$
$$\text{where}$$
$$\sigma(a_\kappa) = \mathbf{callk}(\mathit{call}, \mathit{done}, \epsilon, \_, a'_\kappa)$$
$$\mathit{done} +\!\!+ [v] = v_h :: \overrightarrow{v}$$
$$\varsigma' \triangleq \mathit{CALL}(v_h, \overrightarrow{v} +\!\!+ [v], \rho, \sigma, \mathit{call}, a'_\kappa)$$
$$\blacksquare$$

1 Abstract Semantics of Scheme

Abstract Semantic Domains:

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

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

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 $\sigma[a]$

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

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 $e_{\hat{\varsigma}}$

$$\text{æ} \rightsquigarrow A\langle \hat{v}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle$$

where $\hat{v} \triangleq \hat{\mathcal{A}}(\hat{\varsigma})$

$$(\texttt{if}\ e_c\ e_t\ e_f) \rightsquigarrow E\langle e_c, \hat{\rho}, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{ifk}(e_t, e_f, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

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

$$(\texttt{let}\ bnds\ e_b) \rightsquigarrow E\langle e_0, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $bnds = ([x_0\ e_0]\ [x_s\ e_s]\ ...)$

$\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{letk}(e_{\hat{\varsigma}}, x_0 :: x_s, [\,],$
$e_s, e_b, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

$$(\texttt{call/cc}\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{callcck}(e_{\hat{\varsigma}}, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

$$(\texttt{set!}\ x\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{a} \triangleq \hat{\rho}(x)$

$\hat{\kappa} \triangleq \mathbf{setk}(\hat{a}, \hat{a}_{\hat{\kappa}})$

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

$$(\texttt{prim}\ op) \rightsquigarrow A\langle \hat{v}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}, \rangle$$

where $\hat{v} = op$ applied to 0 arguments

$$(\texttt{prim}\ op\ e_0\ e_s\ ...) \rightsquigarrow E\langle e_0, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{primk}(op, [\,], e_s, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

$$(\texttt{apply-prim}\ op\ e) \rightsquigarrow E\langle e, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{appprimk}(op, \hat{a}_{\hat{\kappa}})$

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

$$(\texttt{apply}\ e_f\ e_x) \rightsquigarrow E\langle e_f, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{appappk}(e_{\hat{\varsigma}}, \varnothing, e_x, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

$$(e_f\ e_s\ ...) \rightsquigarrow E\langle e_f, \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}}\rangle$$

where $\hat{a}'_{\hat{\kappa}} \triangleq \widehat{kalloc}(e_{\hat{\varsigma}}, \hat{\rho})$

$\hat{\kappa} \triangleq \mathbf{appk}(e_{\hat{\varsigma}}, [\,], e_s, \hat{\rho}, \hat{a}_{\hat{\kappa}})$

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

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 $\hat{v} = (\textit{not}\ \texttt{\#f}\ \textit{nor}\ \top, \_)$

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

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

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

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

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

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

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

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

$$\mathbf{setk}(\hat{a}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{A\langle (\textit{Void}, \emptyset), \hat{\sigma}', \hat{a}_{\hat{\kappa}}, \rangle\}$$
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 $\hat{v} \ni (\_, ((\lambda\ (x)\ e_b), \hat{\rho}_\lambda))$

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

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

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

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

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

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

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

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

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

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

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

More Apply Rules

Rules for when the control is a value

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

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

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

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

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

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

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