Concrete Scheme CESK* With Flat Closures

Syntactic Domains

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

Semantic Domains

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

Helper 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

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

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

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

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

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

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

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

Apply Semantics

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

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

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

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

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

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

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

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

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

Scheme CESK* m-CFA with P4F

Semantic Domains

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

Helper Functions

Call Callable

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

Truthy / Falsy

$$\begin{gathered}\widehat{TRUTHY} : \widehat{Val} \to Bool \\ \widehat{TRUTHY}(\bot, \_) \triangleq \texttt{true} \\ \widehat{TRUTHY}(\hat{iv}, \_) \triangleq \texttt{true} \text{ when } \hat{iv} \neq \texttt{\#f} \\ \widehat{TRUTHY}(\_) \triangleq \texttt{false} \text{ otherwise.}\end{gathered}$$

$$\begin{gathered}\widehat{FALSY} : \widehat{Val} \to Bool \\ \widehat{FALSY}(\texttt{\#f}, \varnothing) \triangleq \texttt{true} \\ \widehat{FALSY}(\_) \triangleq \texttt{false} \text{ otherwise.}\end{gathered}$$

Injection

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

Allocation

$$\begin{gathered}\widehat{new\rho} : \mathsf{Exp} \times \widehat{Env} \to \widehat{Env} \\ \widehat{new\rho}(e, \overrightarrow{e}) \triangleq first_m(e :: \overrightarrow{e})\end{gathered}$$

Atomic Evaluation

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

Store Joining:

$$\hat{\sigma} \sqcup [\hat{a}_{\hat{\kappa}} \mapsto \hat{\kappa}] \triangleq \hat{\sigma}[\hat{a}_{\hat{\kappa}} \mapsto \sigma(\hat{a}_{\hat{\kappa}}) \cup \{\hat{\kappa}\}]$$

$$\begin{gathered}\hat{\sigma} \sqcup [\hat{a} \mapsto (\hat{iv}, \widehat{clo})] \triangleq \\ \text{match on } \hat{\sigma}[a]\end{gathered}$$

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

Abstract Eval Semantics

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

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

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

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

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

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

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

Abstract Apply Semantics

$$\hat{\varsigma} = A\langle \hat{v}, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle$$
$$\hat{\kappa}_{\hat{\varsigma}} \in \hat{\sigma}(\hat{a}_{\hat{\kappa}})$$
Proceed by matching on $\hat{\kappa}_{\hat{\varsigma}}$

$$\widehat{\mathbf{mtk}} \rightsquigarrow \varnothing$$
where

$$\widehat{\mathbf{ifk}}(e_t, \_, \hat{\rho}_{\hat{\kappa}}, \hat{a}'_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_t, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}'_{\hat{\kappa}}\rangle\}$$
where $\widehat{TRUTHY}(\hat{v})$

$$\widehat{\mathbf{ifk}}(\_, e_f, \hat{\rho}_{\hat{\kappa}}, \hat{a}'_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_f, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}'_{\hat{\kappa}}\rangle\}$$
where $\widehat{FALSY}(\hat{v})$

$$\widehat{\mathbf{ifk}}(e_t, e_f, \hat{\rho}_{\hat{\kappa}}, \hat{a}'_{\hat{\kappa}})$$
$$\rightsquigarrow \{E\langle e_t, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}'_{\hat{\kappa}}\rangle, E\langle e_f, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}'_{\hat{\kappa}}\rangle\}$$
where $\neg(\widehat{TRUTHY}(\hat{v}) \vee \widehat{FALSY}(\hat{v}))$

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

$$\widehat{\mathbf{callcck}}(e, \hat{a}'_{\hat{\kappa}},) \rightsquigarrow \hat{\varsigma}'$$
where $\hat{\kappa}' \in \hat{\sigma}(\hat{a}'_{\hat{\kappa}})$

$\hat{\varsigma}' \triangleq \widehat{CALL}(\hat{v}, [\hat{\kappa}'], \hat{\rho}, \hat{\sigma}, e, \hat{a}'_{\hat{\kappa}})$

$$\widehat{\mathbf{applyk}}(\mathit{apply}, \varnothing, e, \hat{\rho}_{\hat{\kappa}}, \hat{a}'_{\hat{\kappa}})$$
$$\rightsquigarrow \{E\langle e, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}', \hat{a}''_{\hat{\kappa}}\rangle\}$$
where $\hat{a}''_{\hat{\kappa}} \triangleq (e, \hat{\rho}_{\hat{\kappa}})$

$\hat{\kappa}' \triangleq \widehat{\mathbf{applyk}}(\mathit{apply}, \hat{v}, e, \hat{\rho}_{\hat{\kappa}}, \hat{a}'_{\hat{\kappa}})$

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

$$\widehat{\mathbf{applyk}}(\mathit{apply}, \hat{v}_f, \_, \_, \hat{a}'_{\hat{\kappa}}) \rightsquigarrow \hat{\varsigma}'$$
where $\hat{\varsigma}' \triangleq \widehat{CALL}(\hat{v}_f, \hat{v}, \hat{\rho}, \hat{\sigma}, \mathit{apply}, \hat{a}'_{\hat{\kappa}})$

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

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

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

$$\widehat{\mathbf{callk}}(\mathit{call}, \widehat{\mathit{done}}, \epsilon, \_, \hat{a}'_{\hat{\kappa}}) \rightsquigarrow \hat{\varsigma}'$$
where

$\widehat{\mathit{done}} +\!\!+ [\hat{v}] = \hat{v}_h :: \hat{v}_t$

$\hat{\varsigma}' \triangleq \widehat{CALL}(\hat{v}_h, \hat{v}_t, \hat{\rho}, \hat{\sigma}, \mathit{call}, \hat{a}'_{\hat{\kappa}})$

1 Abstract Semantics of Scheme

Abstract Semantic Domains:

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

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

Abstract Atomic Evaluation:

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

Allocation:

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

Store Joining:

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

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

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 $e_{\hat{\varsigma}}$

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

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

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

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

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

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

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

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

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

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

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

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

$$\mathbf{ifk}(e_t, e_f, \hat{\rho}_{\hat{\kappa}}, \hat{a}_{\hat{\kappa}}) \rightsquigarrow \{E\langle e_t, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}} \rangle, E\langle e_f, \hat{\rho}_{\hat{\kappa}}, \hat{\sigma}, \hat{a}_{\hat{\kappa}} \rangle\}$$
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

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

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

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

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

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

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

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