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

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

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

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

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

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

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

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

Apply Semantics

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

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

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

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

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

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

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

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

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

Scheme CESK* m-CFA with P4F

Semantic Domains

$$\hat{\varsigma} \in \hat{\Sigma} \triangleq E\langle\widehat{Eval}\rangle + A\langle\widehat{Apply}\rangle$$

$$\widehat{Eval} \triangleq \mathsf{Exp} \times \widehat{Env} \times \widehat{Store} \times \widehat{KAddr}$$

$$\widehat{Apply} \triangleq \widehat{Val} \times \widehat{Env} \times \widehat{Store} \times \widehat{KAddr}$$

$$\hat{\rho} \in \widehat{Env} \triangleq \mathsf{Exp}^m$$

$$\hat{\sigma} \in \widehat{Store} \triangleq \widehat{BAddr} \rightharpoonup \widehat{Val} \times \widehat{KAddr} \rightharpoonup \widehat{Kont}$$

$$\hat{a} \in \widehat{BAddr} \triangleq \mathsf{Var} \times \widehat{Env}$$

$$\hat{a}_{\hat{\kappa}} \in \widehat{KAddr} \triangleq \mathsf{Exp} \times \widehat{Env}$$

$$\hat{v} \in \widehat{Val} \triangleq (\widehat{InnerVal} + \top + \bot) \times \mathcal{P}(\widehat{Clo}) \times \mathcal{P}(\widehat{Kont}) \times \mathcal{P}(\mathsf{Prim})$$

$$\hat{iv} \in \widehat{InnerVal} \triangleq \mathbb{Z} + \mathbb{B} + \mathsf{Prim} + \widehat{Kont} + \{\mathbf{quote}(e), \mathbf{cons}(\hat{v}, \hat{v}), \mathit{Null}, \mathit{Void}\}$$

$$\widehat{clo} \in \widehat{Clo} \triangleq \mathsf{Lam} \times \widehat{Env}$$

$$\hat{\kappa} \in \widehat{Kont} ::= \widehat{\mathbf{mtk}}$$
$$\mid \widehat{\mathbf{ifk}}(e, e, \hat{\rho}, \hat{a}_{\hat{\kappa}})$$
$$\mid \widehat{\mathbf{setk}}(\hat{a}, \hat{a}_{\hat{\kappa}})$$
$$\mid \widehat{\mathbf{callcck}}((\texttt{call/cc}\ e), \hat{a}_{\hat{\kappa}})$$
$$\mid \widehat{\mathbf{applyk}}(\mathit{apply}, \hat{v}?, e, \hat{\rho}, \hat{a}_{\hat{\kappa}})$$
$$\mid \widehat{\mathbf{callk}}(\mathit{call}, \widehat{\mathit{done}}, \mathit{todo}, \hat{\rho}, \hat{a}_{\hat{\kappa}})$$

$$\widehat{done} \in \widehat{Val}^*$$

Call Helper

$$\widehat{CALL}_{\widehat{clo}} : \widehat{Clo} \times \widehat{Val}^* \times \widehat{Env} \times \widehat{Store} \times \mathsf{Exp} \times \widehat{KAddr} \to \hat{\Sigma}$$

$$\widehat{CALL}_{\widehat{clo}}((\mathit{lam}, \hat{\rho}_\lambda), \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, e, \hat{a}_{\hat{\kappa}}) \triangleq A\langle e_b, \hat{\rho}', \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle$$

where $\mathit{lam} = (\lambda\ (x\ ...)\ e_b)$

$$\hat{\rho}' \triangleq \widehat{new\rho}(\hat{\rho})$$
$$\hat{a}_{x_i} \triangleq (x_i, \hat{\rho}')$$
$$x'_j \triangleq \mathit{free}(\mathit{lam})$$
$$a_{x'_j} \triangleq (x'_j, \hat{\rho}')$$
$$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [a_{x_i} \mapsto \overrightarrow{\hat{v}}_i] \sqcup [a_{x'_j} \mapsto \hat{\sigma}(x'_j, \rho_\lambda)]$$

$$\widehat{CALL}_{\widehat{clo}}((\mathit{lam}, \hat{\rho}_\lambda), \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, e, \_) \triangleq A\langle \hat{v}, \hat{\rho}, \hat{\sigma}', \hat{a}_{\hat{\kappa}}\rangle$$

where $\mathit{lam} = (\lambda\ (x\ ...)\ e_b)$

$$\hat{\rho}' \triangleq \widehat{new\rho}(\hat{\rho})$$
$$\hat{a}_{x} \triangleq (x, \hat{\rho}')$$
$$x'_j \triangleq \mathit{free}(\mathit{lam})$$
$$a_{x'_j} \triangleq (x'_j, \hat{\rho}')$$
$$\hat{\sigma}' \triangleq \hat{\sigma} \sqcup [a_{x} \mapsto \overrightarrow{\hat{v}}] \sqcup [a_{x'_j} \mapsto \hat{\sigma}(x'_j, \rho_\lambda)]$$

$$\widehat{CALL}_{\hat{\kappa}} : \widehat{Kont} \times \widehat{Val}^* \times \widehat{Env} \times \widehat{Store} \times \mathsf{Exp} \to \hat{\Sigma}$$

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

where $\hat{a}_{\hat{\kappa}} \triangleq (e, \hat{\rho})$

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

$$\widehat{CALL}_{op} : \mathsf{Prim} \times \widehat{Val}^* \times \widehat{Env} \times \widehat{Store} \times \widehat{KAddr} \to \hat{\Sigma}$$

$$\widehat{CALL}_{op}(\mathit{op}, \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}) \triangleq A\langle \hat{v}, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}\rangle$$

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

Helper Functions

Truthy / Falsy

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

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

Call Callable

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

$$\begin{aligned}
&\widehat{CALL}(\hat{v}, \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, e, \hat{a}_{\hat{\kappa}}) \triangleq \hat{\varsigma}' \\
&\text{where } \hat{v} = (_, \widehat{clo}_s, \hat{\kappa}_s, op_s) \\
&\widehat{clo} \in \widehat{clo}_s \\
&\hat{\kappa} \in \hat{\kappa}_s \\
&op \in op_s \\
&\hat{\varsigma}_{\widehat{clo}} \triangleq \widehat{CALL}_{\widehat{clo}}(\widehat{clo}, \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, e, \hat{a}_{\hat{\kappa}}) \\
&\hat{\varsigma}_{\hat{\kappa}} \triangleq \widehat{CALL}_{\hat{\kappa}}(\hat{\kappa}, \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, e) \\
&\hat{\varsigma}_{op} \triangleq \widehat{CALL}_{op}(op, \overrightarrow{\hat{v}}, \hat{\rho}, \hat{\sigma}, \hat{a}_{\hat{\kappa}}) \\
&\hat{\varsigma}' \triangleq \{\hat{\varsigma}_{\widehat{clo}}, \hat{\varsigma}_{\hat{\kappa}}, \hat{\varsigma}_{op}\}
\end{aligned}$$

Inner Value Join

$$\hat{iv}_o \sqcup \hat{iv}_n \triangleq \begin{cases} \hat{iv}_n & \text{if } \hat{iv}_o = \bot \\ \hat{iv}_o & \text{if } \hat{iv}_n = \bot \\ \hat{iv}_o & \text{if } \hat{iv}_o = \hat{iv}_n \\ \top & \text{otherwise.} \end{cases}$$

Injection

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

Allocation

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

Atomic Evaluation

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

Store Joining:

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

$$\hat{\sigma} \sqcup [\hat{a} \mapsto \hat{\kappa}] \triangleq \hat{\sigma}[\hat{a} \mapsto \hat{\kappa}]$$
$$\text{When } \hat{a} \notin \hat{\sigma}$$

$$\begin{aligned}
&\hat{\sigma} \sqcup [\hat{a} \mapsto \hat{v}] \triangleq \hat{\sigma}' \\
&\text{when } \hat{a} \in \hat{\sigma} \\
&\text{where } \hat{\sigma}(\hat{a}) = (\hat{iv}_o, \widehat{clo}_o, \hat{\kappa}_o, op_o) \\
&\hat{v} = (\hat{iv}_n, \widehat{clo}_n, \hat{\kappa}_n, op_n) \\
&\widehat{clo}' \triangleq \widehat{clo}_o \cup \widehat{clo}_n \\
&\hat{\kappa}' \triangleq \hat{\kappa}_o \cup \hat{\kappa}_n \\
&op' \triangleq op_o \cup op_n \\
&\hat{iv}' \triangleq \hat{iv}_o \sqcup \hat{iv}_n \\
&\hat{\sigma}' \triangleq \hat{\sigma}[\hat{a} \mapsto (\hat{iv}', \widehat{clo}', \hat{\kappa}', op')]
\end{aligned}$$

Abstract Eval Semantics

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

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

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

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

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

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

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

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 $\hat{\kappa}_{\hat{\varsigma}}$

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

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

$$\widehat{\mathbf{setk}}(\hat{a}, \hat{a}'_{\hat{\kappa}}) \rightsquigarrow \{A\langle (\mathit{Void}, \varnothing, \varnothing, \varnothing), \hat{\rho}, \hat{\sigma}', \hat{a}'_{\hat{\kappa}} \rangle\}$$
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

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

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