Definition-Contexts Model

November 4, 2015

$$
\begin{aligned}
\mathit{ast} &::= \mathit{var} \mid \textbf{\textsc{App}}(\mathit{ast}, \mathit{ast}, ...) \mid \mathit{val} \\
\mathit{var} &::= \textbf{\textsc{Var}}(\mathit{name}) \\
\mathit{val} &::= \textbf{\textsc{Fun}}(\mathit{var}, \mathit{ast}) \mid \mathit{atom} \mid \textbf{\textsc{List}}(\mathit{val}, ...) \mid \mathit{stx} \\
\mathit{stx} &::= \textbf{\textsc{Stx}}(\mathit{atom}, \mathit{ctx}) \mid \textbf{\textsc{Stx}}(\textbf{\textsc{List}}(\mathit{stx}, ...), \mathit{ctx}) \\
\mathit{id} &::= \textbf{\textsc{Stx}}(\mathit{sym}, \mathit{ctx}) \\
\mathit{ctx} &::= \text{a mapping from } \mathit{ph} \text{ to } \overline{\mathit{scp}} \\
\overline{\mathit{scp}} &::= \{\mathit{scp}, ...\} \\
\mathit{atom} &::= \mathit{sym} \mid \mathit{prim} \mid .... \\
\mathit{sym} &::= \text{'}\mathit{name} \\
\mathit{prim} &::= \texttt{\textbf{stx-e}} \mid \texttt{\textbf{mk-stx}} \mid .... \\
\xi &::= \text{a mapping from } \mathit{name} \text{ to } \mathit{transform} \\
\mathit{transform} &::= \texttt{lambda} \mid \texttt{let-syntax} \mid \texttt{quote} \mid \texttt{syntax} \mid \textsc{Var}(\mathit{id}) \mid \mathit{val} \\
\Sigma &::= \text{binding store, } \mathit{name} \rightarrow (\overline{\mathit{scp}} \rightarrow \mathit{name}) \\
\mathit{name} &::= \text{a token such as } \texttt{x}\text{, } \texttt{egg}\text{, or } \texttt{lambda} \\
\mathit{scp} &::= \text{a token that represents a scope} \\
\mathit{ph} &::= \mathit{integer} \\
\widehat{\mathit{scp}} &::= \mathit{scp} \mid \bullet \\
\hat{\Sigma} &::= \langle \Sigma, \overline{\mathit{scp}}, \overline{\mathit{scp}} \rangle
\end{aligned}
$$

$\mathsf{eval} : ph\ ast\ s\hat{c}p\ \xi\ \widehat{\Sigma} \rightarrow \langle val, \widehat{\Sigma}\rangle$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{new-defs}}), scp_i, \xi, \langle \Sigma, \overline{scp}_p, \overline{scp}_u\rangle\rrbracket && = \langle \textbf{\textsc{Defs}}(scp_{defs}, addr), \widehat{\Sigma}_3\rangle\\
&\text{subject to } \mathsf{alloc\text{-}scope}\llbracket\Sigma\rrbracket = \langle scp_{defs}, \Sigma_2\rangle, \mathsf{alloc\text{-}def\text{-}env}\llbracket\Sigma_2\rrbracket = \langle addr, \Sigma_3\rangle,\\
&\qquad\qquad \langle \Sigma_2 + \{addr \rightarrow \xi\}, \{scp_{defs}\} \cup \overline{scp}_p, \overline{scp}_u\rangle = \widehat{\Sigma}_3
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{def-bind}}, ast_{defs}, ast_{id}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle 0, \langle \Sigma_6, \overline{scp}_{p3}, \overline{scp}_{u3}\rangle\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{defs}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle \textbf{\textsc{Defs}}(scp_{defs}, addr), \widehat{\Sigma}_2\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{id}, scp_i, \xi, \widehat{\Sigma}_2\rrbracket = \langle id_{arg}, \widehat{\Sigma}_3\rangle, \widehat{\Sigma}_3 = \langle \Sigma_3, \overline{scp}_{p3}, \overline{scp}_{u3}\rangle,\\
&\qquad\qquad \mathsf{add}_{ph}\llbracket \mathsf{prune}_{ph}\llbracket \mathsf{flip}_{ph}\llbracket id_{arg}, scp_i\rrbracket, \overline{scp}_{u3}\rrbracket, scp_{defs}\rrbracket = id_{defs},\\
&\qquad\qquad \mathsf{alloc\text{-}name}\llbracket\Sigma_3\rrbracket = \langle name_{new}, \Sigma_4\rangle, \Sigma_4 + \{id_{defs} \rightarrow name_{new}\} = \Sigma_5,\\
&\qquad\qquad \Sigma_5(addr) = \xi_{defs},\\
&\qquad\qquad \Sigma_5 + \{addr \rightarrow \xi_{defs} + \{name_{new} \rightarrow \textsc{Var}(id_{defs})\}\} = \Sigma_6
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{def-bind}}, ast_{defs}, ast_{id}, ast_{stx}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle 0, \widehat{\Sigma}_9\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{defs}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle \textbf{\textsc{Defs}}(scp_{defs}, addr), \widehat{\Sigma}_2\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{id}, scp_i, \xi, \widehat{\Sigma}_2\rrbracket = \langle id_{arg}, \widehat{\Sigma}_3\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{stx}, scp_i, \xi, \widehat{\Sigma}_3\rrbracket = \langle stx_{arg}, \widehat{\Sigma}_4\rangle, \widehat{\Sigma}_4 = \langle \Sigma_4, \overline{scp}_{p4}, \overline{scp}_{u4}\rangle,\\
&\qquad\qquad \mathsf{add}_{ph}\llbracket \mathsf{flip}_{ph}\llbracket stx_{arg}, scp_i\rrbracket, scp_{defs}\rrbracket = stx_{arg2},\\
&\qquad\qquad \mathsf{expand}_{ph+1}\llbracket stx_{arg2}, \xi_{primitives}, \langle \Sigma_4, \varnothing, \varnothing\rangle\rrbracket = \langle stx_{exp}, \langle \Sigma_5, \_, \_\rangle\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket \mathsf{parse}_{ph+1}\llbracket stx_{exp}, \Sigma_5\rrbracket, \bullet, \xi, \langle \Sigma_5, \overline{scp}_{p4}, \varnothing\rangle\rrbracket = \langle val_{exp}, \widehat{\Sigma}_6\rangle,\\
&\qquad\qquad \widehat{\Sigma}_6 = \langle \Sigma_6, \_, \_\rangle, \Sigma_6(addr) = \xi_{defs},\\
&\qquad\qquad \mathsf{add}_{ph}\llbracket \mathsf{prune}_{ph}\llbracket \mathsf{flip}_{ph}\llbracket id_{arg}, scp_i\rrbracket, \overline{scp}_{u4}\rrbracket, scp_{defs}\rrbracket = id_{defs},\\
&\qquad\qquad \mathsf{alloc\text{-}name}\llbracket\Sigma_6\rrbracket = \langle name_{new}, \Sigma_7\rangle, \Sigma_7 + \{id_{defs} \rightarrow name_{new}\} = \Sigma_8,\\
&\qquad\qquad \langle \Sigma_8 + \{addr \rightarrow \xi_{defs} + \{name_{new} \rightarrow val_{exp}\}\}, \overline{scp}_{p4}, \overline{scp}_{u4}\rangle = \widehat{\Sigma}_9
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{lexpand}}, ast_{expr}, ast_{stops}, ast_{defs}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle stx_{exp2}, \widehat{\Sigma}_5\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{expr}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle stx, \widehat{\Sigma}_2\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{stops}, scp_i, \xi, \widehat{\Sigma}_2\rrbracket = \langle \textbf{\textsc{List}}(id_{stop}, ...), \widehat{\Sigma}_3\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{defs}, scp_i, \xi, \widehat{\Sigma}_3\rrbracket = \langle \textbf{\textsc{Defs}}(scp_{defs}, addr), \widehat{\Sigma}_4\rangle,\\
&\qquad\qquad \widehat{\Sigma}_4 = \langle \Sigma_4, \_, \_\rangle, \Sigma_4(addr) = \xi_{defs},\\
&\qquad\qquad \{var \rightarrow \mathsf{unstop}\llbracket \xi_{defs}(var)\rrbracket \mid var \in \mathrm{dom}(\xi_{defs})\} = \xi_{unstops},\\
&\qquad\qquad \mathsf{resolve}_{ph}\llbracket id_{stop}, \Sigma_4\rrbracket, ... = name_{stop}, ...,\\
&\qquad\qquad \xi_{unstops} + \{name_{stop} \rightarrow \textsc{Stop}(\xi_{unstops}(name_{stop}))\}\ ... = \xi_{stops},\\
&\qquad\qquad \mathsf{expand}_{ph}\llbracket \mathsf{add}_{ph}\llbracket \mathsf{flip}_{ph}\llbracket stx, scp_i\rrbracket, scp_{defs}\rrbracket, \xi_{stops}, \widehat{\Sigma}_4\rrbracket = \langle stx_{exp}, \widehat{\Sigma}_5\rangle,\\
&\qquad\qquad \mathsf{flip}_{ph}\llbracket stx_{exp}, scp_i\rrbracket = stx_{exp2}
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{lvalue}}, ast_{id}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle \xi(\mathsf{resolve}_{ph}\llbracket id_{result}, \Sigma_2\rrbracket), \widehat{\Sigma}_2\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{id}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle id_{result}, \widehat{\Sigma}_2\rangle, \widehat{\Sigma}_2 = \langle \Sigma_2, \_, \_\rangle
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{lexpand}}, ast_{expr}, ast_{stops}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle \mathsf{flip}_{ph}\llbracket stx_{exp}, scp_i\rrbracket, \widehat{\Sigma}_4\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{expr}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle stx, \widehat{\Sigma}_2\rangle,\\
&\qquad\qquad \mathsf{eval}_{ph}\llbracket ast_{stops}, scp_i, \xi, \widehat{\Sigma}_2\rrbracket = \langle \textbf{\textsc{List}}(id_{stop}, ...), \widehat{\Sigma}_3\rangle,\\
&\qquad\qquad \{var \rightarrow \mathsf{unstop}\llbracket \xi(var)\rrbracket \mid var \in \mathrm{dom}(\xi)\} = \xi_{unstops}, \widehat{\Sigma}_3 = \langle \Sigma_3, \_, \_\rangle,\\
&\qquad\qquad \xi_{unstops} + \{\mathsf{resolve}_{ph}\llbracket id_{stop}, \Sigma_3\rrbracket \rightarrow \textsc{Stop}(\xi(\mathsf{resolve}_{ph}\llbracket id_{stop}, \Sigma_3\rrbracket))\}\ ... = \xi_{stops},\\
&\qquad\qquad \mathsf{expand}_{ph}\llbracket \mathsf{flip}_{ph}\llbracket stx, scp_i\rrbracket, \xi_{stops}, \widehat{\Sigma}_3\rrbracket = \langle stx_{exp}, \widehat{\Sigma}_4\rangle
\end{aligned}$$

$$\begin{aligned}
&\mathsf{eval}_{ph}\llbracket \textbf{\textsc{App}}(\texttt{\textbf{lbinder}}, ast_{id}), scp_i, \xi, \widehat{\Sigma}\rrbracket && = \langle \mathsf{prune}_{ph}\llbracket id_{result}, \overline{scp}_{u2}\rrbracket, \widehat{\Sigma}_2\rangle\\
&\text{subject to } \mathsf{eval}_{ph}\llbracket ast_{id}, scp_i, \xi, \widehat{\Sigma}\rrbracket = \langle id_{result}, \widehat{\Sigma}_2\rangle, \widehat{\Sigma}_2 = \langle \_, \_, \overline{scp}_{u2}\rangle
\end{aligned}$$

$$\begin{aligned}
&\text{eval}_{ph}\llbracket \mathbf{App}(ast_{fun}, ast_{arg}), \widehat{scp}, \xi, \hat{\Sigma}\rrbracket = \text{eval}_{ph}\llbracket ast_{body}[var \leftarrow val_{arg}], \widehat{scp}, \xi, \hat{\Sigma}_3\rrbracket\\
&\quad \text{subject to } \text{eval}_{ph}\llbracket ast_{fun}, \widehat{scp}, \xi, \hat{\Sigma}\rrbracket = \langle \mathbf{Fun}(var, ast_{body}), \hat{\Sigma}_2\rangle,\\
&\qquad\qquad \text{eval}_{ph}\llbracket ast_{arg}, \widehat{scp}, \xi, \hat{\Sigma}_2\rrbracket = \langle val_{arg}, \hat{\Sigma}_3\rangle
\end{aligned}$$

$$\begin{aligned}
&\text{eval}_{ph}\llbracket \mathbf{App}(prim, ast_{arg}, ...), \widehat{scp}, \xi, \hat{\Sigma}\rrbracket = \langle \delta(prim, val_{arg}, ...), \hat{\Sigma}_2\rangle\\
&\quad \text{subject to } \text{eval*}\llbracket ph, (), (ast_{arg}\ ...), \widehat{scp}, \xi, \hat{\Sigma}\rrbracket = \langle (val_{arg}\ ...), \hat{\Sigma}_2\rangle
\end{aligned}$$

$$\text{eval}_{ph}\llbracket val, \widehat{scp}, \xi, \hat{\Sigma}\rrbracket = \langle val, \hat{\Sigma}\rangle$$

$\text{unstop} : transform \rightarrow transform$

$$\begin{aligned}
\text{unstop}\llbracket \mathrm{Stop}(transform)\rrbracket &= transform\\
\text{unstop}\llbracket transform\rrbracket &= transform
\end{aligned}$$

$$\begin{aligned}
\delta(\texttt{stx-e}, \mathbf{Stx}(val, ctx)) &= val\\
\delta(\texttt{mk-stx}, atom, \mathbf{Stx}(val, ctx)) &= \mathbf{Stx}(atom, ctx)\\
\delta(\texttt{mk-stx}, \mathbf{List}(stx, ...), \mathbf{Stx}(val, ctx)) &= \mathbf{Stx}(\mathbf{List}(stx, ...), ctx)
\end{aligned}$$

$\text{parse} : ph\ stx\ \Sigma \rightarrow ast$

$$\begin{aligned}
&\text{parse}_{ph}\llbracket \mathbf{Stx}(\mathbf{List}(id_{lambda}, id_{arg}, stx_{body}), ctx), \Sigma\rrbracket = \mathbf{Fun}(\mathbf{Var}(\text{resolve}_{ph}\llbracket id_{arg}, \Sigma\rrbracket), \text{parse}_{ph}\llbracket stx_{body}, \Sigma\rrbracket)\\
&\quad \text{subject to } \text{resolve}_{ph}\llbracket id_{lambda}, \Sigma\rrbracket = \texttt{lambda}
\end{aligned}$$

$$\begin{aligned}
&\text{parse}_{ph}\llbracket \mathbf{Stx}(\mathbf{List}(id_{quote}, stx), ctx), \Sigma\rrbracket = \text{strip}\llbracket stx\rrbracket\\
&\quad \text{subject to } \text{resolve}_{ph}\llbracket id_{quote}, \Sigma\rrbracket = \texttt{quote}
\end{aligned}$$

$$\begin{aligned}
&\text{parse}_{ph}\llbracket \mathbf{Stx}(\mathbf{List}(id_{syntax}, stx), ctx), \Sigma\rrbracket = stx\\
&\quad \text{subject to } \text{resolve}_{ph}\llbracket id_{syntax}, \Sigma\rrbracket = \texttt{syntax}
\end{aligned}$$

$$\text{parse}_{ph}\llbracket \mathbf{Stx}(\mathbf{List}(stx_{rator}, stx_{rand}, ...), ctx), \Sigma\rrbracket = \mathbf{App}(\text{parse}_{ph}\llbracket stx_{rator}, \Sigma\rrbracket, \text{parse}_{ph}\llbracket stx_{rand}, \Sigma\rrbracket, ...)$$

$$\text{parse}_{ph}\llbracket id, \Sigma\rrbracket = \mathbf{Var}(\text{resolve}_{ph}\llbracket id, \Sigma\rrbracket)$$

$\text{resolve} : ph\ id\ \Sigma \rightarrow name$

$$\begin{aligned}
&\text{resolve}_{ph}\llbracket \mathbf{Stx}(\text{'}name, ctx), \Sigma\rrbracket = name_{biggest}\\
&\quad \text{subject to } \Sigma(name) = \{\overline{scp}_{bind} \leftarrow name_{bind}, ...\},\\
&\qquad\qquad \text{biggest-subset}\llbracket ctx(ph), \{\overline{scp}_{bind}, ...\}\rrbracket = \overline{scp}_{biggest},\\
&\qquad\qquad \{\overline{scp}_{bind} \leftarrow name_{bind}, ...\}(\overline{scp}_{biggest}) = name_{biggest}
\end{aligned}$$

$$\text{resolve}_{ph}\llbracket \mathbf{Stx}(\text{'}name, ctx), \Sigma\rrbracket = name$$

$\text{biggest-subset} : \overline{scp}\ \{\overline{scp}, ...\} \rightarrow \overline{scp}$

$$\begin{aligned}
&\text{biggest-subset}\llbracket \overline{scp}_{ref}, \{\overline{scp}_{bind}, ...\}\rrbracket = \overline{scp}_{biggest}\\
&\quad \text{subject to } \overline{scp}_{biggest} \subseteq \overline{scp}_{ref}, \overline{scp}_{biggest} \in \{\overline{scp}_{bind}, ...\},\\
&\qquad\qquad \overline{scp}_{bind} \subseteq \overline{scp}_{ref} \Rightarrow \overline{scp}_{bind} \subseteq \overline{scp}_{biggest}
\end{aligned}$$

$\text{strip} : stx \rightarrow val$

$\text{strip}[\![\textbf{Stx}(atom, ctx)]\!] \quad = atom$

$\text{strip}[\![\textbf{Stx}(\textbf{List}(stx, ...), ctx)]\!] = \textbf{List}(\text{strip}[\![stx]\!], ...)$

$\text{expand} : ph\ stx\ \xi\ \hat{\Sigma} \rightarrow \langle stx, \hat{\Sigma} \rangle$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{stop}, stx, ...), ctx), \xi, \hat{\Sigma}]\!] \quad = \langle \textbf{Stx}(\textbf{List}(id_{stop}, stx, ...), ctx), \hat{\Sigma} \rangle$

subject to $\hat{\Sigma} = \langle \Sigma, \_, \_ \rangle$, $\xi(\text{resolve}_{ph}[\![id_{stop}, \Sigma]\!]) = \textsc{Stop}(\_)$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{lam}, id_{arg}, stx_{body}), ctx), \xi, \langle \Sigma, \overline{scp_p}, \overline{scp_u} \rangle]\!]$

$= \langle \textbf{Stx}(\textbf{List}(id_{lam}, id_{new}, stx_{body2}), ctx), \langle \Sigma_4, \overline{scp_p}, \overline{scp_u} \rangle \rangle$

subject to $\text{resolve}_{ph}[\![id_{lam}, \Sigma]\!] = \texttt{lambda}$, $\text{alloc-name}[\![\Sigma]\!] = \langle name_{new}, \Sigma_1 \rangle$,

$\text{alloc-scope}[\![\Sigma_1]\!] = \langle scp_{new}, \Sigma_2 \rangle$, $\text{add}_{ph}[\![id_{arg}, scp_{new}]\!] = id_{new}$,

$\Sigma_2 + \{id_{new} \rightarrow name_{new}\} = \Sigma_3$, $\xi + \{name_{new} \rightarrow \textsc{Var}(id_{new})\} = \xi_{new}$,

$\text{expand}_{ph}[\![\text{add}_{ph}[\![stx_{body}, scp_{new}]\!], \xi_{new}, \langle \Sigma_3, \{scp_{new}\} \cup \overline{scp_p}, \emptyset \rangle]\!] = \langle stx_{body2}, \langle \Sigma_4, \_, \_ \rangle \rangle$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{quote}, stx), ctx), \xi, \hat{\Sigma}]\!] \quad = \langle \textbf{Stx}(\textbf{List}(id_{quote}, stx), ctx), \hat{\Sigma} \rangle$

subject to $\hat{\Sigma} = \langle \Sigma, \_, \_ \rangle$, $\text{resolve}_{ph}[\![id_{quote}, \Sigma]\!] = \texttt{quote}$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{syntax}, stx), ctx), \xi, \hat{\Sigma}]\!]$

$= \langle \textbf{Stx}(\textbf{List}(id_{syntax}, stx_{pruned}), ctx), \hat{\Sigma} \rangle$

subject to $\hat{\Sigma} = \langle \Sigma, \overline{scp_p}, \overline{scp_u} \rangle$, $\text{resolve}_{ph}[\![id_{syntax}, \Sigma]\!] = \texttt{syntax}$, $\text{prune}_{ph}[\![stx, \overline{scp_p}]\!] = stx_{pruned}$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{ls}, id, stx_{rhs}, stx_{body}), ctx), \xi, \langle \Sigma, \overline{scp_p}, \overline{scp_u} \rangle]\!] = \langle stx_{result}, \langle \Sigma_6, \overline{scp_p}, \overline{scp_u} \rangle \rangle$

subject to $\text{resolve}_{ph}[\![id_{ls}, \Sigma]\!] = \texttt{let-syntax}$, $\text{alloc-name}[\![\Sigma]\!] = \langle name_{new}, \Sigma_1 \rangle$,

$\text{alloc-scope}[\![\Sigma_1]\!] = \langle scp_{new}, \Sigma_2 \rangle$, $\text{add}_{ph}[\![id, scp_{new}]\!] = id_{new}$,

$\Sigma_2 + \{id_{new} \rightarrow name_{new}\} = \Sigma_3$,

$\text{expand}_{ph+1}[\![stx_{rhs}, \xi_{primitives}, \langle \Sigma_3, \emptyset, \emptyset \rangle]\!] = \langle stx_{exp}, \langle \Sigma_4, \_, \_ \rangle \rangle$,

$\text{eval}_{ph}[\![\text{parse}_{ph+1}[\![stx_{exp}, \Sigma_4]\!], \bullet, \xi, \langle \Sigma_4, \overline{scp_p}, \emptyset \rangle]\!] = \langle val_{exp}, \langle \Sigma_5, \_, \_ \rangle \rangle$,

$\xi + \{name_{new} \rightarrow val_{exp}\} = \xi_{new}$, $\text{add}_{ph}[\![stx_{body}, scp_{new}]\!] = stx_{body2}$,

$\text{expand}_{ph}[\![stx_{body2}, \xi_{new}, \langle \Sigma_5, \{scp_{new}\} \cup \overline{scp_p}, \emptyset \rangle]\!] = \langle stx_{result}, \langle \Sigma_6, \_, \_ \rangle \rangle$

$\text{expand}_{ph}[\![stx_{macapp}, \xi, \langle \Sigma, \overline{scp_p}, \overline{scp_u} \rangle]\!] \quad = \langle stx_{result}, \hat{\Sigma}_5 \rangle$

subject to $stx_{macapp} = \textbf{Stx}(\textbf{List}(id_{mac}, stx_{arg}, ...), ctx)$, $\xi(\text{resolve}_{ph}[\![id_{mac}, \Sigma]\!]) = val$,

$\text{alloc-scope}[\![\Sigma]\!] = \langle scp_u, \Sigma_2 \rangle$, $\text{alloc-scope}[\![\Sigma_2]\!] = \langle scp_i, \Sigma_3 \rangle$,

$\langle \Sigma_3, \{scp_u\} \cup \overline{scp_p}, \{scp_u\} \cup \overline{scp_u} \rangle = \hat{\Sigma}_3$,

$\text{eval}_{ph}[\![\textbf{App}(val, \text{flip}_{ph}[\![\text{add}_{ph}[\![stx_{macapp}, scp_u]\!], scp_i]\!]), scp_i, \xi, \hat{\Sigma}_3]\!] = \langle stx_{exp}, \hat{\Sigma}_4 \rangle$,

$\text{expand}_{ph}[\![\text{flip}_{ph}[\![stx_{exp}, scp_i]\!], \xi, \hat{\Sigma}_4]\!] = \langle stx_{result}, \hat{\Sigma}_5 \rangle$

$\text{expand}_{ph}[\![\textbf{Stx}(\textbf{List}(stx_{rtor}, stx_{rnd}, ...), ctx), \xi, \langle \Sigma, \overline{scp_p}, \overline{scp_u} \rangle]\!]$

$= \langle \textbf{Stx}(\textbf{List}(stx_{exprtor}, stx_{exprnd}, ...), ctx), \langle \Sigma_1, \overline{scp_p}, \overline{scp_u} \rangle \rangle$

subject to $\text{expand*}_{ph}[\![(), (stx_{rtor}\ stx_{rnd}\ ...), \xi, \langle \Sigma, \overline{scp_p}, \emptyset \rangle]\!] = \langle (stx_{exprtor}\ stx_{exprnd}\ ...), \Sigma_1 \rangle$

$\text{expand}_{ph}[\![id, \xi, \hat{\Sigma}]\!] \quad = \langle id_{new}, \hat{\Sigma} \rangle$

subject to $\hat{\Sigma} = \langle \Sigma, \_, \_ \rangle$, $\xi(\text{resolve}_{ph}[\![id, \Sigma]\!]) = \textsc{Var}(id_{new})$

$\mathsf{expand*} : \mathit{ph}\ (\mathit{stx}\ ...)\ (\mathit{stx}\ ...)\ \xi\ \hat{\Sigma} \rightarrow \langle(\mathit{stx}\ ...), \Sigma\rangle$

$\mathsf{expand*}_{\mathit{ph}}[\![(\mathit{stx}_{\mathit{done}}\ ...), (), \xi, \langle\Sigma, \_, \_\rangle]\!] = \langle(\mathit{stx}_{\mathit{done}}\ ...), \Sigma\rangle$

$\mathsf{expand*}_{\mathit{ph}}[\![(\mathit{stx}_{\mathit{done}}\ ...), (\mathit{stx}_{0}\ \mathit{stx}_{1}\ ...), \xi, \langle\Sigma, \overline{\mathit{scp}}_{p}, \emptyset\rangle]\!]$
$= \mathsf{expand*}_{\mathit{ph}}[\![(\mathit{stx}_{\mathit{done}}\ ...\ \mathit{stx}_{\mathit{done0}}), (\mathit{stx}_{1}\ ...), \xi, \langle\Sigma_{2}, \overline{\mathit{scp}}_{p}, \emptyset\rangle]\!]$
subject to $\mathsf{expand}_{\mathit{ph}}[\![\mathit{stx}_{0}, \xi, \langle\Sigma, \overline{\mathit{scp}}_{p}, \emptyset\rangle]\!] = \langle\mathit{stx}_{\mathit{done0}}, \langle\Sigma_{2}, \_, \_\rangle\rangle$

$\mathsf{prune} : \mathit{ph}\ \mathit{stx}\ \overline{\mathit{scp}} \rightarrow \mathit{stx}$

$\mathsf{prune}_{\mathit{ph}}[\![\mathbf{STX}(\mathit{atom}, \mathit{ctx}), \overline{\mathit{scp}}_{p}]\!] = \mathbf{STX}(\mathit{atom}, \mathit{ctx}+\{\mathit{ph} \rightarrow \mathit{ctx}(\mathit{ph}) \setminus \overline{\mathit{scp}}_{p}\})$
$\mathsf{prune}_{\mathit{ph}}[\![\mathbf{STX}(\mathbf{LIST}(\mathit{stx}, ...), \mathit{ctx}), \overline{\mathit{scp}}_{p}]\!] = \mathbf{STX}(\mathbf{LIST}(\mathit{stx}_{\mathit{pruned}}, ...), \mathit{ctx}+\{\mathit{ph} \rightarrow \mathit{ctx}(\mathit{ph}) \setminus \overline{\mathit{scp}}_{p}\})$
subject to $\mathsf{prune}_{\mathit{ph}}[\![\mathit{stx}, \overline{\mathit{scp}}_{p}]\!], ... = \mathit{stx}_{\mathit{pruned}}, ...$

$\mathsf{add} : \mathit{ph}\ \mathit{stx}\ \mathit{scp} \rightarrow \mathit{stx}$

$\mathsf{add}_{\mathit{ph}}[\![\mathbf{STX}(\mathit{atom}, \mathit{ctx}), \mathit{scp}]\!] = \mathbf{STX}(\mathit{atom}, \mathit{ctx}+\{\mathit{ph} \rightarrow \{\mathit{scp}\} \cup \mathit{ctx}(\mathit{ph})\})$
$\mathsf{add}_{\mathit{ph}}[\![\mathbf{STX}(\mathbf{LIST}(\mathit{stx}, ...), \mathit{ctx}), \mathit{scp}]\!] = \mathbf{STX}(\mathbf{LIST}(\mathsf{add}_{\mathit{ph}}[\![\mathit{stx}, \mathit{scp}]\!], ...), \mathit{ctx}+\{\mathit{ph} \rightarrow \{\mathit{scp}\} \cup \mathit{ctx}(\mathit{ph})\})$

$\mathsf{flip} : \mathit{ph}\ \mathit{stx}\ \mathit{scp} \rightarrow \mathit{stx}$

$\mathsf{flip}_{\mathit{ph}}[\![\mathbf{STX}(\mathit{atom}, \mathit{ctx}), \mathit{scp}]\!] = \mathbf{STX}(\mathit{atom}, \mathit{ctx}+\{\mathit{ph} \rightarrow \mathit{scp} \oplus \mathit{ctx}(\mathit{ph})\})$
$\mathsf{flip}_{\mathit{ph}}[\![\mathbf{STX}(\mathbf{LIST}(\mathit{stx}, ...), \mathit{ctx}), \mathit{scp}]\!] = \mathbf{STX}(\mathbf{LIST}(\mathsf{flip}_{\mathit{ph}}[\![\mathit{stx}, \mathit{scp}]\!], ...), \mathit{ctx}+\{\mathit{ph} \rightarrow \mathit{scp} \oplus \mathit{ctx}(\mathit{ph})\})$