Local-Expansion Model

November 4, 2015

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

eval : $ph\ ast\ \widehat{scp}\ \xi\ \widehat{\Sigma} \to \langle val, \widehat{\Sigma}\rangle$

$$\mathsf{eval}_{ph}[\![\textbf{App}(\texttt{lvalue}, ast_{id}), scp_i, \xi, \widehat{\Sigma}]\!] = \langle \xi(\mathsf{resolve}_{ph}[\![id_{result}, \Sigma_2]\!]), \widehat{\Sigma}_2\rangle$$
$$\text{subject to } \mathsf{eval}_{ph}[\![ast_{id}, scp_i, \xi, \widehat{\Sigma}]\!] = \langle id_{result}, \widehat{\Sigma}_2\rangle, \widehat{\Sigma}_2 = \langle \Sigma_2, \_, \_\rangle$$

$$\mathsf{eval}_{ph}[\![\textbf{App}(\texttt{lexpand}, ast, ast_{stops}), scp_i, \xi, \widehat{\Sigma}]\!] = \langle \mathsf{flip}_{ph}[\![stx_{exp}, scp_i]\!], \widehat{\Sigma}_4\rangle$$
$$\text{subject to } \mathsf{eval}_{ph}[\![ast, scp_i, \xi, \widehat{\Sigma}]\!] = \langle stx, \widehat{\Sigma}_2\rangle,$$
$$\mathsf{eval}_{ph}[\![ast_{stops}, scp_i, \xi, \widehat{\Sigma}_2]\!] = \langle \textbf{List}(id_{stop}, ...), \widehat{\Sigma}_3\rangle,$$
$$\{var \to \mathsf{unstop}[\![\xi(var)]\!] \mid var \in \mathrm{dom}(\xi)\} = \xi_{unstops}, \widehat{\Sigma}_3 = \langle \Sigma_3, \_, \_\rangle,$$
$$\mathsf{resolve}_{ph}[\![id_{stop}, \Sigma_3]\!], ... = name_{stop}, ...,$$
$$\xi_{unstops} + \{name_{stop} \to \textsc{Stop}(\xi_{unstops}(name_{stop}))\}\ ... = \xi_{stops},$$
$$\mathsf{expand}_{ph}[\![\mathsf{flip}_{ph}[\![stx, scp_i]\!], \xi_{stops}, \widehat{\Sigma}_3]\!] = \langle stx_{exp}, \widehat{\Sigma}_4\rangle$$

$$\mathsf{eval}_{ph}[\![\textbf{App}(\texttt{lbinder}, ast_{id}), scp_i, \xi, \widehat{\Sigma}]\!] = \langle \mathsf{prune}_{ph}[\![id_{result}, \overline{scp}_{u2}]\!], \widehat{\Sigma}_2\rangle$$
$$\text{subject to } \mathsf{eval}_{ph}[\![ast_{id}, scp_i, \xi, \widehat{\Sigma}]\!] = \langle id_{result}, \widehat{\Sigma}_2\rangle, \widehat{\Sigma}_2 = \langle \_, \_, \overline{scp}_{u2}\rangle$$

$$\mathsf{eval}_{ph}[\![\textbf{App}(ast_{fun}, ast_{arg}), \widehat{scp}, \xi, \widehat{\Sigma}]\!] = \mathsf{eval}_{ph}[\![ast_{body}[var \leftarrow val_{arg}], \widehat{scp}, \xi, \widehat{\Sigma}_3]\!]$$
$$\text{subject to } \mathsf{eval}_{ph}[\![ast_{fun}, \widehat{scp}, \xi, \widehat{\Sigma}]\!] = \langle \textbf{Fun}(var, ast_{body}), \widehat{\Sigma}_2\rangle,$$
$$\mathsf{eval}_{ph}[\![ast_{arg}, \widehat{scp}, \xi, \widehat{\Sigma}_2]\!] = \langle val_{arg}, \widehat{\Sigma}_3\rangle$$

$$\mathsf{eval}_{ph}[\![\textbf{App}(prim, ast_{arg}, ...), \widehat{scp}, \xi, \widehat{\Sigma}]\!] = \langle \delta(prim, val_{arg}, ...), \widehat{\Sigma}_2\rangle$$
$$\text{subject to } \mathsf{eval}^{*}[\![ph, (), (ast_{arg}\ ...), \widehat{scp}, \xi, \widehat{\Sigma}]\!] = \langle (val_{arg}\ ...), \widehat{\Sigma}_2\rangle$$

$$\mathsf{eval}_{ph}[\![val, \widehat{scp}, \xi, \widehat{\Sigma}]\!] = \langle val, \widehat{\Sigma}\rangle$$

unstop : $transform \to transform$

$$\mathsf{unstop}[\![\textsc{Stop}(transform)]\!] = transform$$
$$\mathsf{unstop}[\![transform]\!] = transform$$

$$\delta(\texttt{stx-e}, \textbf{Stx}(val, ctx)) = val$$
$$\delta(\texttt{mk-stx}, atom, \textbf{Stx}(val, ctx)) = \textbf{Stx}(atom, ctx)$$
$$\delta(\texttt{mk-stx}, \textbf{List}(stx, ...), \textbf{Stx}(val, ctx)) = \textbf{Stx}(\textbf{List}(stx, ...), ctx)$$

parse : $ph\ stx\ \Sigma \to ast$

$$\mathsf{parse}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{lambda}, id_{arg}, stx_{body}), ctx), \Sigma]\!] = \textbf{Fun}(\textbf{Var}(\mathsf{resolve}_{ph}[\![id_{arg}, \Sigma]\!]), \mathsf{parse}_{ph}[\![stx_{body}, \Sigma]\!])$$
$$\text{subject to } \mathsf{resolve}_{ph}[\![id_{lambda}, \Sigma]\!] = \texttt{lambda}$$

$$\mathsf{parse}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{quote}, stx), ctx), \Sigma]\!] = \mathsf{strip}[\![stx]\!]$$
$$\text{subject to } \mathsf{resolve}_{ph}[\![id_{quote}, \Sigma]\!] = \texttt{quote}$$

$$\mathsf{parse}_{ph}[\![\textbf{Stx}(\textbf{List}(id_{syntax}, stx), ctx), \Sigma]\!] = stx$$
$$\text{subject to } \mathsf{resolve}_{ph}[\![id_{syntax}, \Sigma]\!] = \texttt{syntax}$$

$$\mathsf{parse}_{ph}[\![\textbf{Stx}(\textbf{List}(stx_{rator}, stx_{rand}, ...), ctx), \Sigma]\!] = \textbf{App}(\mathsf{parse}_{ph}[\![stx_{rator}, \Sigma]\!], \mathsf{parse}_{ph}[\![stx_{rand}, \Sigma]\!], ...)$$

$$\mathsf{parse}_{ph}[\![id, \Sigma]\!] = \textbf{Var}(\mathsf{resolve}_{ph}[\![id, \Sigma]\!])$$

resolve : $ph\ id\ \Sigma \to name$

$$\text{resolve}_{ph}[\![\textbf{Stx}(\text{'}name, ctx), \Sigma]\!] = name_{biggest}$$
$$\text{subject to } \Sigma(name) = \{\overline{scp}_{bind} \leftarrow name_{bind}, ...\},$$
$$\text{biggest-subset}[\![ctx(ph), \{\overline{scp}_{bind}, ...\}]\!] = \overline{scp}_{biggest},$$
$$\{\overline{scp}_{bind} \leftarrow name_{bind}, ...\}(\overline{scp}_{biggest}) = name_{biggest}$$

$$\text{resolve}_{ph}[\![\textbf{Stx}(\text{'}name, ctx), \Sigma]\!] = name$$

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

$$\text{biggest-subset}[\![\overline{scp}_{ref}, \{\overline{scp}_{bind}, ...\}]\!] = \overline{scp}_{biggest}$$
$$\text{subject to } \overline{scp}_{biggest} \subseteq \overline{scp}_{ref}, \overline{scp}_{biggest} \in \{\overline{scp}_{bind}, ...\},$$
$$\overline{scp}_{bind} \subseteq \overline{scp}_{ref} \Rightarrow \overline{scp}_{bind} \subseteq \overline{scp}_{biggest}$$

strip : $stx \to val$

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

$\mathsf{expand} : ph\ stx\ \xi\ \widehat{\Sigma} \to \langle stx, \widehat{\Sigma} \rangle$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(id, stx, ...), ctx), \xi, \widehat{\Sigma}]\!] = \langle \mathbf{Stx}(\mathbf{List}(id, stx, ...), ctx), \widehat{\Sigma} \rangle$
subject to $\widehat{\Sigma} = \langle \Sigma, \_, \_ \rangle$, $\xi(\mathsf{resolve}_{ph}[\![id, \Sigma]\!]) = \textsc{Stop}(\_)$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(id_{lam}, id_{arg}, stx_{bdy}), ctx), \xi, \langle \Sigma, \overline{scp}_p, \overline{scp}_u \rangle]\!]$
$= \langle \mathbf{Stx}(\mathbf{List}(id_{lam}, id_{new}, stx_{bdy2}), ctx), \langle \Sigma_4, \overline{scp}_p, \overline{scp}_u \rangle \rangle$
subject to $\mathsf{resolve}_{ph}[\![id_{lam}, \Sigma]\!] = \texttt{lambda}$, $\mathsf{alloc\text{-}name}[\![\Sigma]\!] = \langle name_{new}, \Sigma_1 \rangle$,
$\mathsf{alloc\text{-}scope}[\![\Sigma_1]\!] = \langle scp_{new}, \Sigma_2 \rangle$, $\mathsf{add}_{ph}[\![id_{arg}, scp_{new}]\!] = id_{new}$,
$\Sigma_2 + \{id_{new} \to name_{new}\} = \Sigma_3$, $\xi + \{name_{new} \to \textsc{Var}(id_{new})\} = \xi_{new}$,
$\langle \Sigma_3, \{scp_{new}\} \cup \overline{scp}_p, \varnothing \rangle = \widehat{\Sigma}_3$,
$\mathsf{expand}_{ph}[\![\mathsf{add}_{ph}[\![stx_{bdy}, scp_{new}]\!], \xi_{new}, \widehat{\Sigma}_3]\!] = \langle stx_{bdy2}, \langle \Sigma_4, \_, \_ \rangle \rangle$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(id_{quote}, stx), ctx), \xi, \widehat{\Sigma}]\!] = \langle \mathbf{Stx}(\mathbf{List}(id_{quote}, stx), ctx), \widehat{\Sigma} \rangle$
subject to $\widehat{\Sigma} = \langle \Sigma, \_, \_ \rangle$, $\mathsf{resolve}_{ph}[\![id_{quote}, \Sigma]\!] = \texttt{quote}$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(id_{syntax}, stx), ctx), \xi, \widehat{\Sigma}]\!]$
$= \langle \mathbf{Stx}(\mathbf{List}(id_{syntax}, stx_{pruned}), ctx), \widehat{\Sigma} \rangle$
subject to $\widehat{\Sigma} = \langle \Sigma, \overline{scp}_p, \overline{scp}_u \rangle$, $\mathsf{resolve}_{ph}[\![id_{syntax}, \Sigma]\!] = \texttt{syntax}$, $\mathsf{prune}_{ph}[\![stx, \overline{scp}_p]\!] = stx_{pruned}$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(id_{ls}, id, stx_{rhs}, stx_{body}), ctx), \xi, \langle \Sigma, \overline{scp}_p, \overline{scp}_u \rangle]\!] = \langle stx_{result}, \langle \Sigma_6, \overline{scp}_p, \overline{scp}_u \rangle \rangle$
subject to $\mathsf{resolve}_{ph}[\![id_{ls}, \Sigma]\!] = \texttt{let-syntax}$, $\mathsf{alloc\text{-}name}[\![\Sigma]\!] = \langle name_{new}, \Sigma_1 \rangle$,
$\mathsf{alloc\text{-}scope}[\![\Sigma_1]\!] = \langle scp_{new}, \Sigma_2 \rangle$, $\mathsf{add}_{ph}[\![id, scp_{new}]\!] = id_{new}$,
$\Sigma_2 + \{id_{new} \to name_{new}\} = \Sigma_3$,
$\mathsf{expand}_{ph+1}[\![stx_{rhs}, \xi_{primitives}, \langle \Sigma_3, \varnothing, \varnothing \rangle]\!] = \langle stx_{exp}, \langle \Sigma_4, \_, \_ \rangle \rangle$,
$\mathsf{eval}_{ph}[\![\mathsf{parse}_{ph+1}[\![stx_{exp}, \Sigma_4]\!], \bullet, \xi, \langle \Sigma_4, \overline{scp}_p, \varnothing \rangle]\!] = \langle val_{exp}, \langle \Sigma_5, \_, \_ \rangle \rangle$,
$\xi + \{name_{new} \to val_{exp}\} = \xi_{new}$, $\mathsf{add}_{ph}[\![stx_{body}, scp_{new}]\!] = stx_{body2}$,
$\mathsf{expand}_{ph}[\![stx_{body2}, \xi_{new}, \langle \Sigma_5, \{scp_{new}\} \cup \overline{scp}_p, \varnothing \rangle]\!] = \langle stx_{result}, \langle \Sigma_6, \_, \_ \rangle \rangle$

$\mathsf{expand}_{ph}[\![stx_{macapp}, \xi, \langle \Sigma, \overline{scp}_p, \overline{scp}_u \rangle]\!] = \langle stx_{result}, \widehat{\Sigma}_5 \rangle$
subject to $stx_{macapp} = \mathbf{Stx}(\mathbf{List}(id_{mac}, stx_{arg}, ...), ctx)$, $\xi(\mathsf{resolve}_{ph}[\![id_{mac}, \Sigma]\!]) = val$,
$\mathsf{alloc\text{-}scope}[\![\Sigma]\!] = \langle scp_u, \Sigma_2 \rangle$, $\mathsf{alloc\text{-}scope}[\![\Sigma_2]\!] = \langle scp_i, \Sigma_3 \rangle$,
$\langle \Sigma_3, \{scp_u\} \cup \overline{scp}_p, \{scp_u\} \cup \overline{scp}_u \rangle = \widehat{\Sigma}_3$,
$\mathsf{flip}_{ph}[\![\mathsf{add}_{ph}[\![stx_{macapp}, scp_u]\!], scp_i]\!] = stx_{macapp2}$,
$\mathsf{eval}_{ph}[\![\mathbf{App}(val, stx_{macapp2}), scp_i, \xi, \widehat{\Sigma}_3]\!] = \langle stx_{exp}, \widehat{\Sigma}_4 \rangle$,
$\mathsf{expand}_{ph}[\![\mathsf{flip}_{ph}[\![stx_{exp}, scp_i]\!], \xi, \widehat{\Sigma}_4]\!] = \langle stx_{result}, \widehat{\Sigma}_5 \rangle$

$\mathsf{expand}_{ph}[\![\mathbf{Stx}(\mathbf{List}(stx_{rtor}, stx_{rnd}, ...), ctx), \xi, \langle \Sigma, \overline{scp}_p, \overline{scp}_u \rangle]\!]$
$= \langle \mathbf{Stx}(\mathbf{List}(stx_{exprtor}, stx_{exprnd}, ...), ctx), \langle \Sigma_1, \overline{scp}_p, \overline{scp}_u \rangle \rangle$
subject to $\mathsf{expand}^*_{ph}[\![(), (stx_{rtor}\ stx_{rnd}\ ...), \xi, \langle \Sigma, \overline{scp}_p, \varnothing \rangle]\!] = \langle (stx_{exprtor}\ stx_{exprnd}\ ...), \Sigma_1 \rangle$

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

$\mathsf{expand}^\star : ph\ (stx\ ...)\ (stx\ ...)\ \xi\ \widehat{\Sigma} \to \langle (stx\ ...), \Sigma \rangle$

$\mathsf{expand}^\star_{ph}[\![(stx_{done}\ ...), (), \xi, \langle \Sigma, \_, \_ \rangle]\!] = \langle (stx_{done}\ ...), \Sigma \rangle$

$\mathsf{expand}^\star_{ph}[\![(stx_{done}\ ...), (stx_0\ stx_1\ ...), \xi, \langle \Sigma, \overline{scp}_p, \varnothing \rangle]\!]$
$= \mathsf{expand}^\star_{ph}[\![(stx_{done}\ ...\ stx_{done0}), (stx_1\ ...), \xi, \langle \Sigma_2, \overline{scp}_p, \varnothing \rangle]\!]$
subject to $\mathsf{expand}_{ph}[\![stx_0, \xi, \langle \Sigma, \overline{scp}_p, \varnothing \rangle]\!] = \langle stx_{done0}, \langle \Sigma_2, \_, \_ \rangle \rangle$

$\mathsf{prune} : ph\ stx\ \overline{scp} \to stx$

$\mathsf{prune}_{ph}[\![\mathbf{Stx}(atom, ctx), \overline{scp}_p]\!] = \mathbf{Stx}(atom, ctx{+}\{ph \to ctx(ph) \setminus \overline{scp}_p\})$

$\mathsf{prune}_{ph}[\![\mathbf{Stx}(\mathbf{List}(stx, ...), ctx), \overline{scp}_p]\!] = \mathbf{Stx}(\mathbf{List}(stx_{pruned}, ...), ctx{+}\{ph \to ctx(ph) \setminus \overline{scp}_p\})$
subject to $\mathsf{prune}_{ph}[\![stx, \overline{scp}_p]\!], ... = stx_{pruned}, ...$

$\mathsf{add} : ph\ stx\ scp \to stx$

$\mathsf{add}_{ph}[\![\mathbf{Stx}(atom, ctx), scp]\!] = \mathbf{Stx}(atom, ctx{+}\{ph \to \{scp\} \cup ctx(ph)\})$
$\mathsf{add}_{ph}[\![\mathbf{Stx}(\mathbf{List}(stx, ...), ctx), scp]\!] = \mathbf{Stx}(\mathbf{List}(\mathsf{add}_{ph}[\![stx, scp]\!], ...), ctx{+}\{ph \to \{scp\} \cup ctx(ph)\})$

$\mathsf{flip} : ph\ stx\ scp \to stx$

$\mathsf{flip}_{ph}[\![\mathbf{Stx}(atom, ctx), scp]\!] = \mathbf{Stx}(atom, ctx{+}\{ph \to scp \oplus ctx(ph)\})$
$\mathsf{flip}_{ph}[\![\mathbf{Stx}(\mathbf{List}(stx, ...), ctx), scp]\!] = \mathbf{Stx}(\mathbf{List}(\mathsf{flip}_{ph}[\![stx, scp]\!], ...), ctx{+}\{ph \to scp \oplus ctx(ph)\})$