κ-VM : abstract machine

Input Language

$$
\begin{array}{llll}
P & ::= & \mathsf{0} \mid P[P] \mid (P \mid P) \mid E \mid (\nu\, r)\, P & \text{Process} \\
 & \mid & ([u = u]\, P\ ,\ P) \mid \langle D \rangle \mid \mathsf{pass}\, V & \\
V & ::= & r \mid \lambda\, x.P & \text{Value} \\
 & \mid & \mathsf{reify}_{\mathsf{control}}(\mathcal{D}\ ,\ \mathcal{H}\ ,\mathcal{R}) & \\
 & \mid & \mathsf{reify}_{\mathsf{content}}(\mathcal{E}\ ,\ \mathcal{L}) & \\
D & ::= & J = P & \text{Definition} \\
J & ::= & r \mid J \mid J & \text{Join-pattern} \\
u & ::= & x \mid r & \text{Name} \\
E & ::= & V \mid x \mid (E\ E) & \text{Expression}
\end{array}
$$

Syntactic Categories

$r, s, t \ldots$	$\in$	Ref	Reference
$x, y, z \ldots$	$\in$	Var	Variable
$u, v, w \ldots$	$\in$	$\mathsf{Name} = \mathsf{Ref} \cup \mathsf{Var}$	Names
$P, Q, R \ldots$	$\in$	Pr	Process
$A, A' \ldots$	$\in$	$\mathsf{Abst} = \mathsf{Var} \times \mathsf{Pr}$	Lambda-abstraction
$V, W, \ldots$	$\in$	Val	Value
$E, F, G \ldots$	$\in$	Expr	Expression
$J, J' \ldots$	$\in$	$\mathsf{Join} = \mathsf{Ref}^*$	Join-pattern
$D, D' \ldots$	$\in$	$\mathsf{Def} = \mathsf{Join} \times \mathsf{Pr}$	Definition

with $\mathsf{Abst} \cup \mathsf{Ref} \subset \mathsf{Val}$, $\mathsf{Val} \subset \mathsf{Expr} \subset \mathsf{Pr}$, $\mathsf{Def} \subset \mathsf{Pr}$

State of the Abstract Machine

- Execution Engine

$$
\begin{array}{rcll}
\mathcal{D}, \mathcal{D}' & \in & \mathsf{DefSpace} = Set(\mathsf{Env} \times \mathsf{Def}) & \text{Activable local definitions} \\
\mathcal{H}, \mathcal{H}' & \in & \mathsf{Heap} = \mathsf{Ref} \mapsto Queue(\mathsf{Val}^*) & \text{Heap of messages} \\
\mathcal{R}, \mathcal{R}' & \in & \mathsf{RQueue} = Queue(\mathsf{Env} \times \mathsf{Pr}) & \text{Run Queue} \\
\mathcal{E}, \mathcal{E}' & \in & \mathsf{Ether} = \mathsf{LocName} & \text{Internal Ether} \\
\mathcal{L}, \mathcal{L}' & \in & \mathsf{SubLoc} = Set(\mathsf{LocName}) & \text{Sub-locations} \\
\mathbb{E} & \in & \mathsf{EEState} = \mathsf{LocName} \times \mathsf{DefSpace} \times & \text{State of the Execution Engine} \\
 & & \mathsf{Heap} \times \mathsf{RQueue} \times \mathsf{Ether} \times \mathsf{SubLoc} &
\end{array}
$$

- Oracle

$$
O \quad \in \quad \mathsf{Oracle} = \mathsf{Ref} \mapsto \mathsf{LocName} \quad \text{Oracle}
$$

Reduction rules

$$\frac{\text{Pre-conditions}}{\text{State}_1 \longrightarrow \text{State}_2}$$

$$\mathbb{E}_{l_{i_1}}; \ldots ; \mathbb{E}_{l_{i_p}} \parallel O \longrightarrow \mathbb{E}_{l_{j_1}}; \ldots ; \mathbb{E}_{l_{j_q}} \parallel O'$$

- Sequential core :

NIL

$$\frac{}{l : (\mathcal{D}, \mathcal{H}, (\rho, \mathsf{0}) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \mathcal{E}, \mathcal{L})}$$

PRL

$$\frac{}{l : (\mathcal{D}, \mathcal{H}, (\rho, (P \mid Q)) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, (\rho, P) :: \mathcal{R} :: (\rho, Q), \mathcal{E}, \mathcal{L})}$$

IF.THEN

$$\frac{}{l : (\mathcal{D}, \mathcal{H}, (\rho, ([u \ = \ u]\ P\ ,\ Q)) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, (\rho, P) :: \mathcal{R}, \mathcal{E}, \mathcal{L})}$$

IF.ELSE

$$\frac{u \neq v}{l : (\mathcal{D}, \mathcal{H}, (\rho, ([u \ = \ v]\ P\ ,\ Q)) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, (\rho, Q) :: \mathcal{R}, \mathcal{E}, \mathcal{L})}$$

EVAL

$$\frac{\mathbb{S}_l = [\![\ E\ ]\!]_{begin} \quad \mathbb{S}_l \overset{*}{\longrightarrow}_\beta \mathbb{S}'_l \quad \mathbb{S}'_l \not\longrightarrow_\beta \quad P = [\![\ \mathbb{S}'_l\ ]\!]_{end}}{l : (\mathcal{D}, \mathcal{H}, (\rho, E) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, \mathcal{R} :: (\rho, P), \mathcal{E}, \mathcal{L})}$$

NEW

$$\frac{\rho' = \rho\{r \mapsto i\} \quad i = freshName()}{l : (\mathcal{D}, \mathcal{H}, (\rho, (\nu\ r)\ P :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, (\rho', P) :: \mathcal{R}, \mathcal{E}, \mathcal{L})}$$

- Definitions :

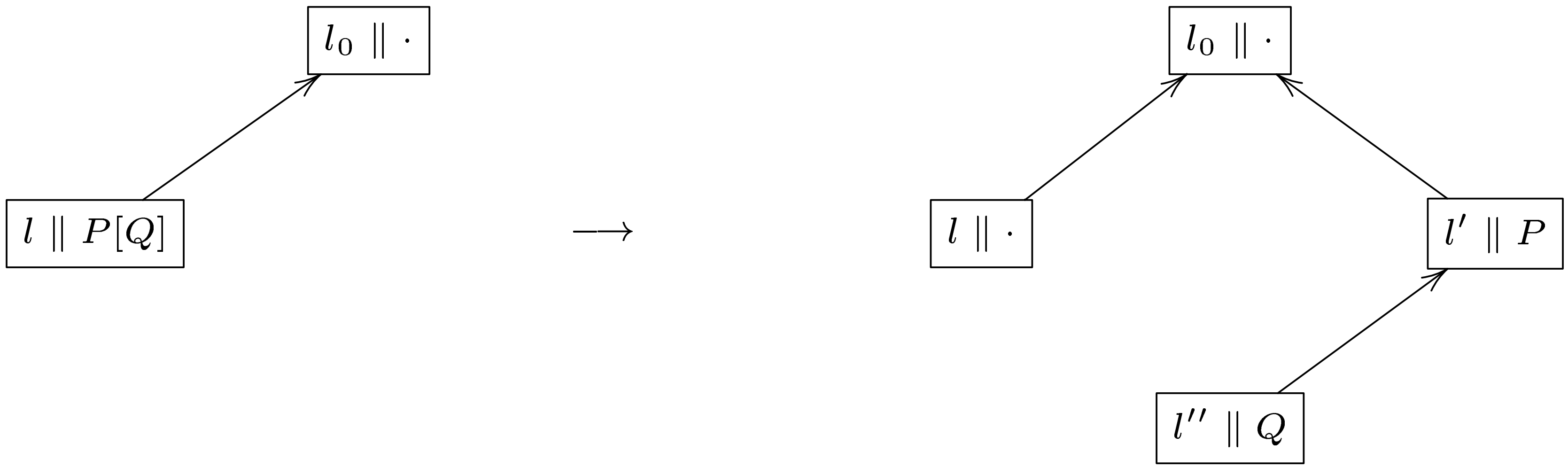
DEF
$$\frac{\langle D\rangle = \langle J = P\rangle \quad \rho' = closure(\rho, \langle D\rangle) \quad \mathcal{D}' = \mathcal{D} \cup \{(\rho', D)\} \qquad \mathcal{H}' = \mathcal{H}\ \{\rho(r) \mapsto \bullet\}_{r \in \mathsf{dn}(J)} \quad O' = O\ \{r \mapsto l\}_{r \in \mathsf{dn}(J)}}{l : (\mathcal{D}, \mathcal{H}, (\rho, \langle D\rangle) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \parallel O \longrightarrow l : (\mathcal{D}', \mathcal{H}', \mathcal{R}, \mathcal{E}, \mathcal{L}) \parallel O'}$$

MSG
$$\frac{\mathcal{H}(\rho(r)) = m \quad \mathcal{H}' = \mathcal{H}\ \{\rho(r) \mapsto m :: (\rho, \widetilde{V})\}}{l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}', \mathcal{R}, \mathcal{E}, \mathcal{L})}$$

$$\frac{\rho(r) \notin \mathsf{dom}\,\mathcal{H}}{l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}, \mathcal{R} :: (\rho, r\widetilde{V}), \mathcal{E}, \mathcal{L})}$$

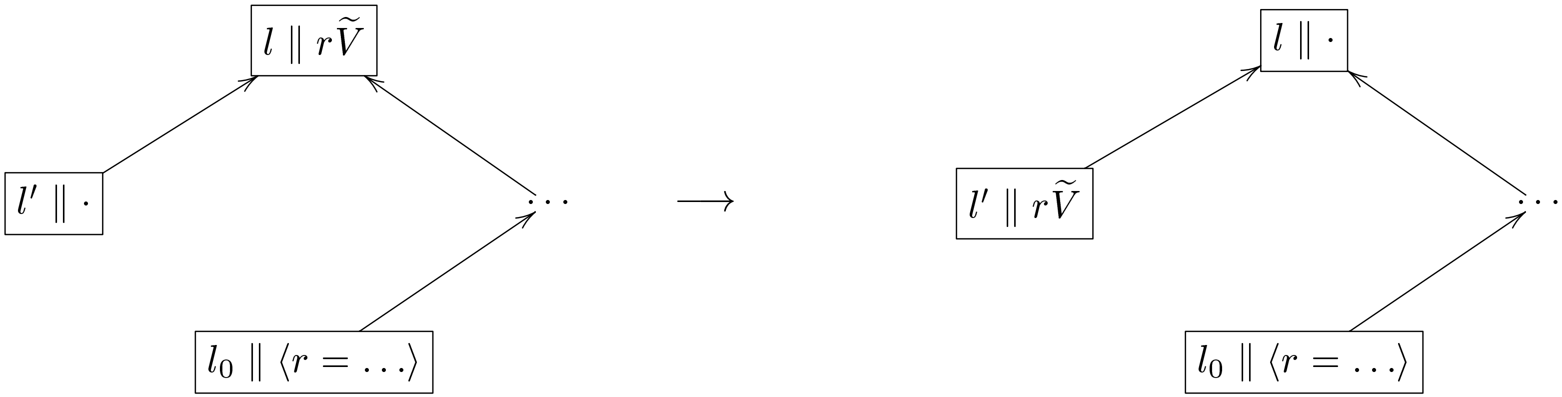
COM
$$\frac{(\rho, J = P) \in \mathcal{D} \quad J = r_1 \mid \ldots \mid r_q \quad \forall r \in \mathsf{dn}(J) \quad \mathcal{H}(\rho(r)) = (\rho_r, \widetilde{V_r}) :: m_r \qquad \rho' = \rho\ \cup \bigcup_{r \in \mathsf{dn}(J)} \rho_r \quad \mathcal{H}' = \mathcal{H}\ \{\rho(r) \mapsto m_r\}_{r \in \mathsf{dn}(J)}}{l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \mathcal{E}, \mathcal{L}) \longrightarrow l : (\mathcal{D}, \mathcal{H}', \mathcal{R} :: (\rho', P\ \widetilde{V}_1 \ldots \widetilde{V}_q), \mathcal{E}, \mathcal{L})}$$

- Domains :



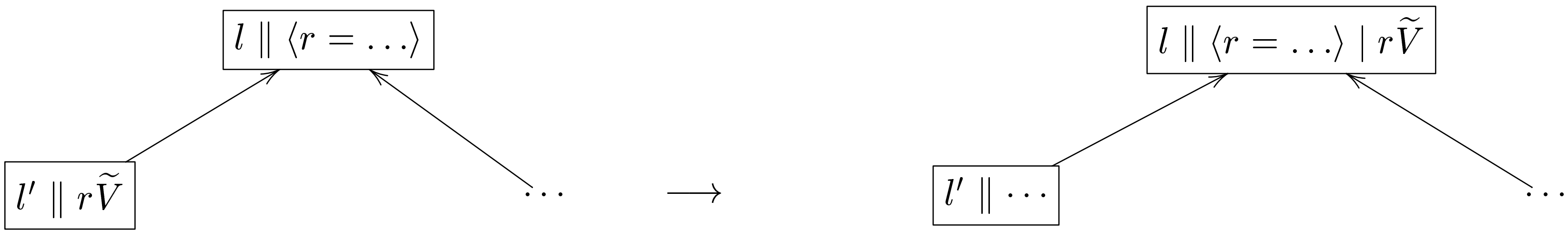
$$\frac{\begin{array}{c} l' = freshLocName() \qquad l'' = freshLocName() \\ \rho' = closure(\rho, P) \quad \rho'' = closure(\rho, Q) \end{array}}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0) \ ; \ l : (\mathcal{D}, \mathcal{H}, (\rho, P[Q]) :: \mathcal{R}, \sharp, \emptyset) \longrightarrow \\ l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0 \cup \{l'\}) \ ; \ l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \sharp, \emptyset) \ ; \\ l' : (\emptyset, \emptyset, (\rho', P), l'', \emptyset) \ ; \ l'' : (\emptyset, \emptyset, (\rho'', Q), \sharp, \emptyset) \end{array}}$$

INTRUDE : $(r\widetilde{V} \mid P)[Q] \longrightarrow P[(r\widetilde{V} \mid Q)] \quad r \in \mathsf{df}(Q) \ \wedge \ r \notin \{\mathbf{i}, \mathbf{o}\}$



$$\frac{O(r) = l_0 \quad l_0.isDesc(l) = 1 \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, l', \mathcal{L}) \ ; \ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \sharp, \emptyset) \parallel O \longrightarrow \\ l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, l', \mathcal{L}) \ ; \ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}' :: (\rho, r\widetilde{V}), \sharp, \emptyset) \parallel O \end{array}}$$

$$\textbf{EXTRUDE :}\ P[(r\widetilde{V} \mid Q)] \longrightarrow (r\widetilde{V} \mid P)[Q] \quad r \in \mathsf{df}(P)\ \wedge\ r \notin \{\mathbf{i}, \mathbf{o}\}$$



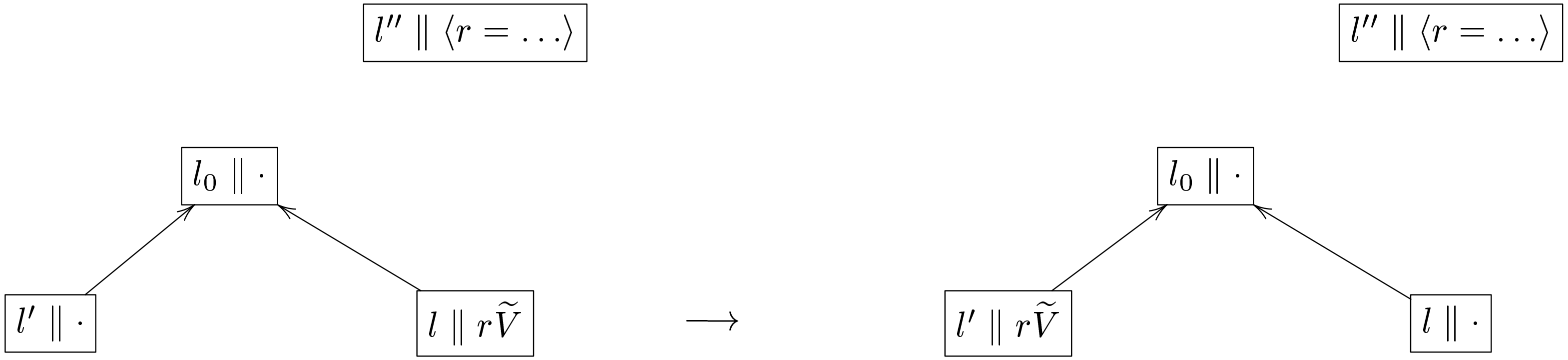
$$\frac{O(r) = l \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, l', \mathcal{L})\ ;\ l' : (\mathcal{D}', \mathcal{H}', (\rho, r\widetilde{V}) :: \mathcal{R}', \sharp, \emptyset) \parallel O \longrightarrow \\ l : (\mathcal{D}, \mathcal{H}, \mathcal{R} :: (\rho, r\widetilde{V}), l', \mathcal{L})\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \sharp, \emptyset) \parallel O \end{array}}$$

$$\mathbf{IN:}\ (r\widetilde{V} \mid P[Q]) \longrightarrow (r\widetilde{V} \mid P)[Q] \quad r \in \mathsf{df}(P)\ \wedge\ r \notin \{\mathbf{i}, \mathbf{o}\}$$

$$\begin{tikzcd}
 & \boxed{l_0 \parallel \cdot} & & & & \boxed{l_0 \parallel \cdot} & \\
\boxed{l \parallel r\widetilde{V}} \arrow[ur] & & \boxed{l' \parallel \langle r = \ldots \rangle} \arrow[ul] & \longrightarrow & \boxed{l \parallel \cdot} \arrow[ur] & & \boxed{l' \parallel \langle r = \ldots \rangle \mid r\widetilde{V}} \arrow[ul]
\end{tikzcd}$$

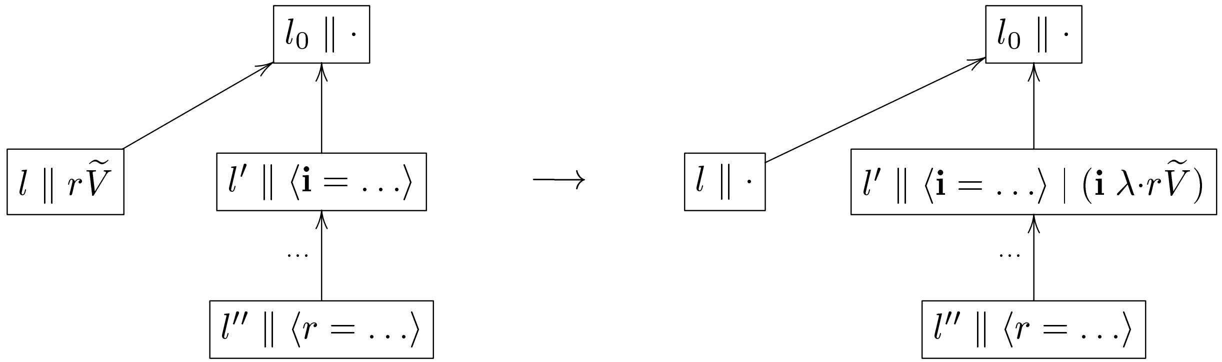
$$\frac{l' \in \mathcal{L}_0 \quad O(r) = l' \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0)\ ;\ l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, \sharp, \emptyset)\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \mathcal{E}', \mathcal{L}') \parallel O \longrightarrow \\ l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0)\ ;\ l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \sharp, \emptyset)\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}' :: (\rho, r\widetilde{V}), \mathcal{E}', \mathcal{L}') \parallel O \end{array}}$$

$$\textbf{OUT :} \ (r\widetilde{V} \mid P)[Q] \longrightarrow (r\widetilde{V} \mid P[Q]) \quad r \notin \mathsf{df}(P[Q]) \ \wedge \ r \notin \{\mathbf{i}, \mathbf{o}\}$$



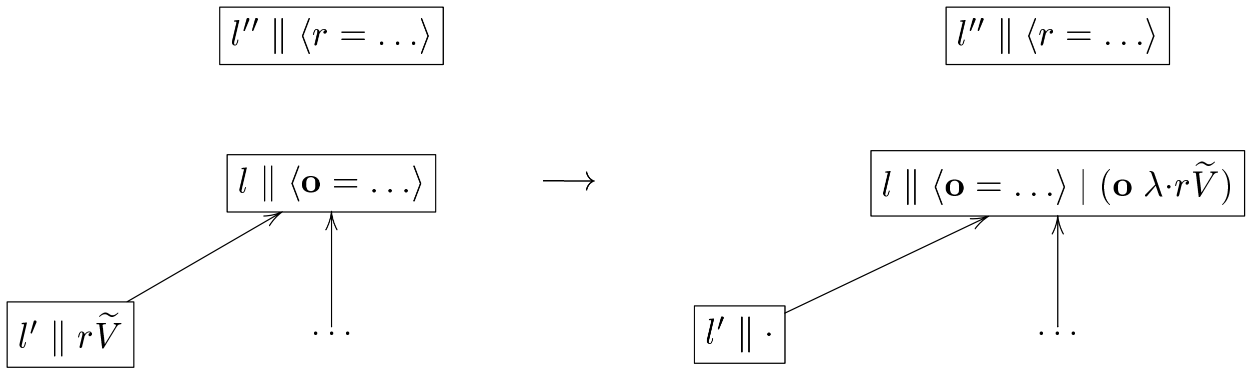
$$\frac{l \in \mathcal{L}_0 \quad O(r) = l'' \quad l'' \neq l \quad l''.isDesc(l) = 0 \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l', \mathcal{L}_0) \ ; \ l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \ ; \ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \sharp, \emptyset) \parallel O \longrightarrow \\ l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l', \mathcal{L}_0) \ ; \ l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \mathcal{E}, \mathcal{L}) \ ; \ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}' :: (\rho, r\widetilde{V}), \sharp, \emptyset) \parallel O \end{array}}$$

$$\textbf{FILTER.IN :}\ (r\widetilde{V} \mid P[Q]) \longrightarrow ((\mathbf{i}\ \lambda{\cdot}r\widetilde{V}) \mid P)[Q] \quad r \in \mathsf{df}(Q)\ \wedge\ \mathbf{i} \in \mathsf{df}(P)\ \wedge\ r \notin \{\mathbf{i}, \mathbf{o}\}$$



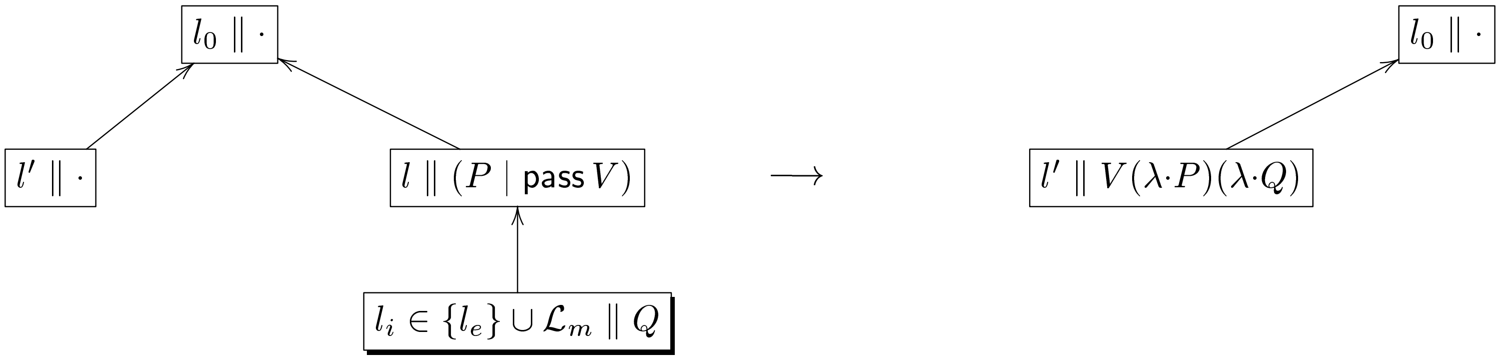
$$\frac{l' \in \mathcal{L}_0 \quad O(r) = l'' \quad l''.isDesc(l') = 1 \quad O(\mathbf{i}) = l' \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0)\ ;\ l : (\mathcal{D}, \mathcal{H}, (\rho, r\widetilde{V}) :: \mathcal{R}, \sharp, \emptyset)\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \mathcal{E}', \mathcal{L}') \parallel O \longrightarrow \\ l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0)\ ;\ l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, \sharp, \emptyset)\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}' :: (\rho, (\mathbf{i}\ \lambda{\cdot}r\widetilde{V})), \mathcal{E}', \mathcal{L}') \parallel O \end{array}}$$

$$\textbf{FILTER.OUT :}\ P[(r\widetilde{V} \mid Q)] \longrightarrow ((\mathbf{o}\ \lambda{\cdot}r\widetilde{V}) \mid P)[Q] \quad r \notin \mathsf{df}(P[Q]) \ \wedge\ \mathbf{o} \in \mathsf{df}(P) \ldots$$



$$\frac{O(r) = l'' \quad l \neq l'' \quad l''.isDesc(l) = 0 \quad O(\mathbf{o}) = l \quad r \notin \{\mathbf{i}, \mathbf{o}\}}{\begin{array}{c} l : (\mathcal{D}, \mathcal{H}, \mathcal{R}, l', \mathcal{L})\ ;\ l' : (\mathcal{D}', \mathcal{H}', (\rho, r\widetilde{V}) :: \mathcal{R}', \sharp, \emptyset) \parallel O \longrightarrow \\ l : (\mathcal{D}, \mathcal{H}, \mathcal{R} :: (\rho, (\mathbf{o}\ \lambda{\cdot}r\widetilde{V})), l', \mathcal{L})\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \sharp, \emptyset) \parallel O \end{array}}$$

$$\textbf{PASS :}\ (P \mid \mathsf{pass}\,V)[Q] \longrightarrow V(\lambda{\cdot}P)(\lambda{\cdot}Q)$$

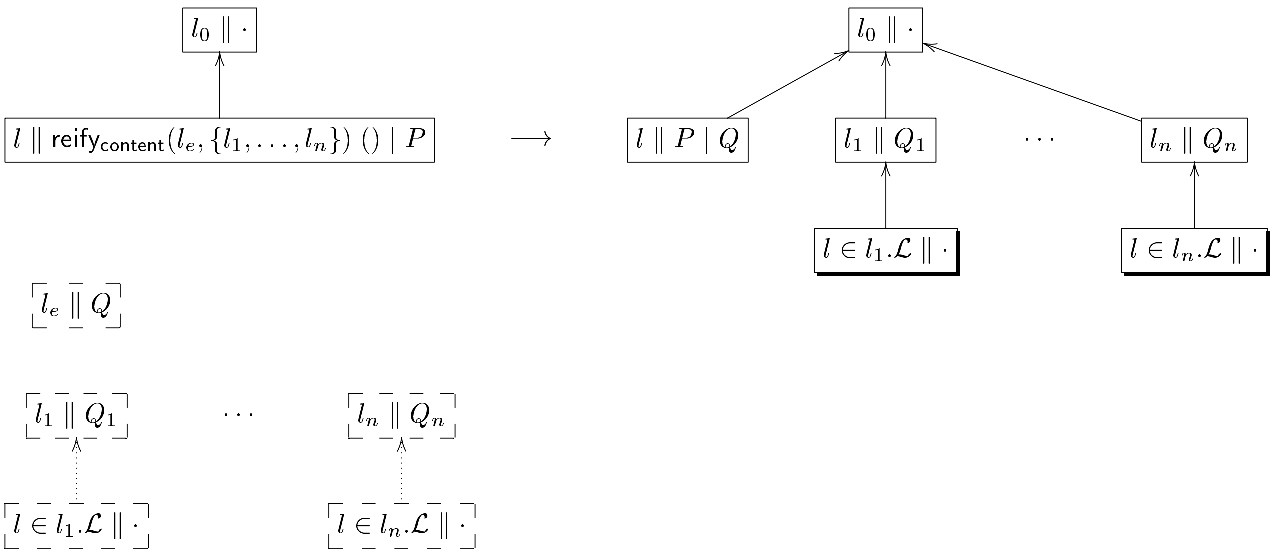


$$\frac{l \in \mathcal{L}_0 \quad \mathcal{L}_m = \bigcup_{l_i \in \mathcal{L}} desc(l_i) \quad O' = unbind(O, l)}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l', \mathcal{L}_0)\ ;\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}', \sharp, \emptyset)\ ; \\ l : (\mathcal{D}, \mathcal{H}, (\rho, \mathsf{pass}\,V) :: \mathcal{R}, l_e, \mathcal{L})\ ;\ l_e\ ;\ \mathcal{L}_m \parallel O \longrightarrow l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l', \mathcal{L}_0 \setminus \{l\})\ ; \\ l' : (\mathcal{D}', \mathcal{H}', \mathcal{R}' :: (\rho, V\,\mathsf{reify}_{\mathsf{control}}(\mathcal{D}, \mathcal{H}, \mathcal{R})\ \mathsf{reify}_{\mathsf{content}}(l_e, \mathcal{L})), \sharp, \emptyset) \parallel O' \end{array}}$$

REIFY.CONTROL

$$\frac{O' = O\{r \mapsto l\}_{r \in \mathsf{dn}(D), D \in \mathcal{D}_m}}{l : (\mathcal{D}, \mathcal{H}, (\rho, (\mathsf{reify}_{\mathsf{control}}(\mathcal{D}_m, \mathcal{H}_m, \mathcal{R}_m)\ ()) :: \mathcal{R}, \mathcal{E}, \mathcal{L}) \parallel O \longrightarrow \\ l : (\mathcal{D} \cup \mathcal{D}_m, \mathcal{H} \cup \mathcal{H}_m, \mathcal{R} :: \mathcal{R}_m, \mathcal{E}, \mathcal{L}) \parallel O'}$$

REIFY.CONTENT



$$\frac{l_e : (\mathcal{D}_e, \mathcal{H}_e, \mathcal{R}_e, \sharp, \emptyset) \qquad O' = [\bigotimes_{l_i \in \mathcal{L}_m} rebind(O, l_i, l_i)] \otimes rebind(O, l_e, l)}{\begin{array}{c} l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0)\ ;\ l : (\mathcal{D}, \mathcal{H}, (\rho, (\mathsf{reify}_{\mathsf{content}}(l_e, \mathcal{L}_m)\ ())) :: \mathcal{R}, \sharp, \emptyset) \parallel O \longrightarrow \\ l_0 : (\mathcal{D}_0, \mathcal{H}_0, \mathcal{R}_0, l, \mathcal{L}_0 \cup \mathcal{L}_m)\ ;\ l : (\mathcal{D} \cup \mathcal{D}_e, \mathcal{H} \cup \mathcal{H}_e, \mathcal{R} :: \mathcal{R}_e, \sharp, \emptyset)\ ;\ \bigcup_{l_i \in \mathcal{L}_m} desc(l_i) \parallel O' \end{array}}$$

A Call-By-Name Evaluator

- State : $\mathbb{S}_l = (\mathcal{P}, \zeta, \mathcal{T})$
  - ⋆ *Stack* $\mathcal{P}$: pointers towards heap nodes.
  - ⋆ *Global Environment* ζ :
    - ∘ Maps a reference r to an address in the heap. Allows to keep track of free references inside the expression being evaluated.
    - ∘ Element of $\mathsf{EvalEnv} = \mathsf{Name} \mapsto \mathsf{Adr}$.
  - ⋆ *Heap* $\mathcal{T}$:
    - ∘ Set of nodes representing the expressions to evaluate. Indexed by addresses $\alpha \in \mathsf{Adr}$.
    - ∘ Element of $\mathsf{EvalHeap} = \mathsf{Adr} \mapsto \mathsf{Nodes}$.

Heap Nodes

e, f, g	$::=$	$e_p \mid e_i$	Heap nodes
e_p^α	$::=$	$\langle x \rangle^\alpha \mid \langle r \rangle^\alpha \mid \lambda^\alpha\ f.g \mid f @^\alpha g \mid \nabla^\alpha\ f$	Functional nodes
	$\mid$	$\mathsf{reify}_{\mathsf{control}}{}^\alpha(\mathcal{D}\ ,\ \mathcal{H}\ ,\mathcal{R}) \mid \mathsf{reify}_{\mathsf{content}}{}^\alpha(\mathcal{E}\ ,\ \mathcal{L})$	
e_i^α	$::=$	$\mathsf{0}^\alpha \mid f[g]^\alpha \mid (f \mid^\alpha g) \mid (\nu^\alpha\ f)\ g$	Other nodes
	$\mid$	$([e_1\ =\ e_2]^\alpha\ f_1\ ,\ f_2) \mid \langle e_1 \mid \ldots \mid e_n = f \rangle^\alpha \mid \mathsf{pass}^\alpha f$	

Reduction

UNWIND

$$\frac{\mathcal{T}(\alpha) = e_1^{\alpha_1} @^{\alpha} e_2^{\alpha_2} \quad \mathcal{T}(\alpha_1) \neq \langle r \rangle}{\mathbb{S} = (\alpha :: \mathcal{P}, \zeta, \mathcal{T}) \longrightarrow_\beta \mathbb{S}' = (\alpha_1 :: \alpha :: \mathcal{P}, \zeta, \mathcal{T})}$$

BETA

$$\frac{\begin{array}{c}\mathcal{T}(\alpha_0) = e_{lambda}^{\alpha_0} \quad \mathcal{T}(\alpha_1) = e_{lambda}^{\alpha_0} @^{\alpha_1} e_{arg}^{\alpha_a} \quad e_{lambda}^{\alpha_0} = \lambda^{\alpha_0} \ \langle x \rangle . e_{body} \\ \epsilon = \{x \mapsto \alpha_a\} \quad (\alpha_r, \mathcal{T}', \zeta') = \mathcal{I}(e_{body}, \epsilon, \zeta, \mathcal{T}) \quad \mathcal{T}'(\alpha_1) = \nabla^{\alpha_1} \ \mathcal{T}(\alpha_r)\end{array}}{\mathbb{S} = (\alpha_0 :: \alpha_1 :: \mathcal{P}, \zeta, \mathcal{T}) \longrightarrow_\beta \mathbb{S}' = (\alpha_1 :: \mathcal{P}, \zeta', \mathcal{T}')}$$

INDIR

$$\frac{\mathcal{T}(\alpha) = \nabla^{\alpha} \ e^{\beta}}{\mathbb{S} = (\alpha :: \mathcal{P}, \zeta, \mathcal{T}) \longrightarrow_\beta \mathbb{S}' = (\beta :: \mathcal{P}, \zeta, \mathcal{T})}$$