Rules Reference for the Undecorated Gallifrey Type System

Matthew Milano Joshua Turcotti Andrew C. Myers

2021-04-14

This is the **Undecorated** version of the system.

1 Grammar

$$
\begin{array}{rrcl}
\text{(function)} & \mathit{fn} & \in & \mathit{FunctionNames} \\
\text{(variable)} & x & \in & \mathit{VariableNames} \\
\text{(class)} & \mathcal{C} & \in & \mathit{ClassNames} \\
\text{(region)} & r & \in & \mathit{RegionNames} \\
\text{(location)} & l & \in & \mathit{LocationNames} \\
\text{(field)} & f & \in & \mathit{FieldNames} \\
\text{(type)} & \tau & ::= & \mathcal{C} \mid \texttt{int} \mid \texttt{bool} \mid \texttt{unit} \mid (q_{\text{ARG}}\ \tau \rightarrow q_{\text{RET}}\ \tau)
\end{array}
$$

$$
\begin{array}{rrcl}
\text{(arg qualifier)} & q_{\text{ARG}} & ::= & \texttt{preserves} \mid \texttt{consumes} \\
\text{(return qualifier)} & q_{\text{RET}} & ::= & \texttt{iso} \mid \texttt{bnd} \\
\text{(function definition)} & \text{FDEF} & ::= & \texttt{def}\ q_{\text{RET}}\ \tau\ \mathit{fn}(q_{\text{ARG}}\ \tau\ x)\{e\} \\
\text{(program)} & p & ::= & \text{FDEF};\ p \mid e
\end{array}
$$

$$
\begin{array}{rrcl}
\text{(virtual command)} & \text{VIR} & ::= & \texttt{focus}\ x \mid \texttt{unfocus}\ x \mid \texttt{explore}\ x.f \mid \texttt{retract}\ x.f \mid \texttt{attach}\ \{e\}\ \texttt{to}\ \{e\} \\
 & & & \mid \texttt{swap}\ \{e\}\ \texttt{with}\ \{e\} \mid \texttt{drop-var}\ x \mid \texttt{drop-reg}\ \{e\} \mid \texttt{invalidate-var}\ x
\end{array}
$$

$$
\begin{array}{rrcl}
\text{(expression)} & e & ::= & l \mid x \mid e; e \mid e; \text{VIR} \mid e.f \mid e.f = e \mid x = e \mid \mathit{fn} \mid e(e) \mid e \oplus_r e \mid \texttt{new-}\tau \\
 & & & \mid \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\} \mid \texttt{if}(e)\{e\}\ \texttt{else}\ \{e\} \mid \texttt{while}(e)\{e\} \\
 & & & \mid \texttt{send-}\tau(e) \mid \texttt{recv-}\tau \mid \texttt{detach}\ x\ \texttt{in}\ \{e\}\ \texttt{else}\ \{e\} \\
\text{(evaluation context)} & E[] & ::= & []; e \mid []; \text{VIR} \mid [].f \mid e.f = [] \mid [].f = l \mid x = [] \mid [](e) \mid l([]) \mid [] \oplus_r e \mid l \oplus_r [] \\
 & & & \mid \texttt{if}([])\{e\}\ \texttt{else}\ \{e\} \mid \texttt{send-}\tau([]) \mid l; \texttt{drop-reg}\ \{[]\} \\
 & & & \mid l; \texttt{attach}\ \{[]\}\ \texttt{to}\ \{e\} \mid l; \texttt{attach}\ \{l\}\ \texttt{to}\ \{[]\} \\
 & & & \mid l; \texttt{swap}\ \{[]\}\ \texttt{with}\ \{e\} \mid l; \texttt{swap}\ \{l\}\ \texttt{with}\ \{[]\}
\end{array}
$$

$$
\begin{aligned}
\text{(dynamic reservation)} \quad & d ::= \; l,\; d \mid \cdot \\
\text{(heap)} \quad & h ::= \; l \mapsto (\tau, v),\; h \mid \cdot \\
\text{(stack)} \quad & s ::= \; x \mapsto l,\; s \mid \cdot \\
\text{(regionality)} \quad & \rho ::= \; \mathrm{P}
\end{aligned}
$$

2 Final Rules

$\boxed{\text{F1}}$ - Expression-Well-Typedness

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega' \qquad \vdash d;h;s : \mathcal{H};\Gamma;\Omega;\mathrm{P}\ \texttt{agree}}{\vdash\ (d;h;s;e : r\ \tau)\ \texttt{well-typed}}$$

$\boxed{\text{F2}}$ - Dynamic-Static-Agreement

$$\frac{\begin{array}{c}\mathrm{P}, s \vdash h/\mathcal{H}\ \texttt{graph-simple} \qquad \mathcal{H}, \mathrm{P}, h, s \vdash d\ \texttt{res-sufficient} \\ \mathrm{P}, h, s \vdash \mathcal{H}\ \texttt{convex} \qquad \vdash h\ \texttt{heap-closed} \qquad \vdash h, \mathrm{P}\ \texttt{heap-agree} \\ \vdash \mathcal{H}, \mathrm{P}, h, s\ \texttt{bnd-ref-sane} \qquad \vdash \mathcal{H}, \Gamma\ \texttt{binding-agree} \qquad \mathcal{H}, s, \mathrm{P} \vdash \Gamma\ \texttt{binding-sane} \\ s \vdash \mathcal{H}\ \texttt{non-aliasing} \qquad \mathrm{P}, h, s \vdash \mathcal{H}\ \texttt{target-accurate} \qquad \Omega \vdash \mathcal{H}, \Gamma, \mathrm{P}\ \texttt{well-bounded}\end{array}}{\vdash d, h, s : \mathcal{H};\Gamma;\Omega;\mathrm{P}\ \texttt{agree}}$$

$\boxed{\text{F3}}$ - Graph-Simplicity-Enforcement

$$\frac{G_S(\mathcal{H}, \mathrm{P}, h, s)\ \texttt{is a forest}}{\mathrm{P}, s \vdash h/\mathcal{H}\ \texttt{graph-simple}}$$

$\boxed{\text{F4}}$ - Reservation-Sufficiency

$$\frac{\textit{live-set}(\mathcal{H}, \mathrm{P}, h, s) \subseteq d \subseteq \textit{dom}(h)}{\mathcal{H}, \mathrm{P}, h, s \vdash d\ \texttt{res-sufficient}}$$

$\boxed{\text{F5}}$ - H-Convex

$$\frac{\forall (r, r', \chi) : [((r \in \textit{dom}(\mathcal{H})) \wedge (r' \in \textit{dom}(\mathcal{H})) \wedge (\mathcal{H}, \mathrm{P}, h, s \vdash r \hookrightarrow \chi \hookrightarrow r')) \implies (\chi \in \textit{dom}(\mathcal{H}) \cup \textit{loc-refs}(\mathcal{H}))]}{\mathrm{P}, h, s \vdash \mathcal{H}\ \texttt{convex}}$$

$\boxed{\text{F6}}$ - Heap-Closure

$$\frac{\forall (l \in \textit{dom}(h), \tau, v, f, l') : [((h(l) = (\tau, v) \wedge (v.f = l')) \implies (\exists q_{\text{RET}}, \tau_f, v_f : (q_{\text{RET}}\ f\ \tau_f \in \textit{fields}(\tau) \wedge h(l') = (\tau_f, v_f)))]}{\vdash h\ \texttt{heap-closed}}$$

$\boxed{\text{F7}}$ - Heap-Rho-Agreement

$$\frac{\textit{dom}(h) = \textit{dom}(\mathrm{P}) \qquad \forall (l \in \textit{dom}(h)) : [h \upharpoonright_\tau (l) = \mathrm{P} \upharpoonright_\tau (l)]}{\vdash h, \mathrm{P}\ \texttt{heap-agree}}$$

$\boxed{\text{F8}}$ - Bounded-Ref-Sanity

$$\frac{\forall (l, l', f) : [(l \in \textit{live-set}(\mathcal{H}, \mathrm{P}, h, s) \wedge (h \upharpoonright_v (l).f = l') \wedge (\mathrm{P} \upharpoonright_r (l) \neq \mathrm{P} \upharpoonright_r (l'))) \implies (\texttt{iso}\ f\ \tau' \in \textit{fields}(h \upharpoonright_\tau (l)))]}{\vdash \mathcal{H}, \mathrm{P}, h, s\ \texttt{bnd-ref-sane}}$$

$\boxed{\text{F9}}$ - H-Gamma-Agreement

$$\frac{\forall (x, r) : [(x@r \in \textit{reg-vars}(\mathcal{H})) \implies ((x \in \textit{dom}(\Gamma)) \wedge (\Gamma \upharpoonright_r (x) = r))]}{\vdash \mathcal{H}, \Gamma\ \texttt{binding-agree}}$$

$\boxed{\text{F10}}$ - VARIABLE-BINDING-SANITY

$$\frac{\forall(x, r, \tau) : [(\Gamma \vdash x : r\ \tau) \implies ((x \in \mathit{dom}(s)) \land ((r \in \mathit{dom}(\mathcal{H})) \implies (\mathrm{P}(s(x)) = (r, \tau))))]}{\mathcal{H}, \mathrm{P}, s \vdash \Gamma\ \texttt{binding-sane}}$$

$\boxed{\text{F11}}$ - H-NON-ALIASING

$$\frac{\forall(x, x') : [(x, x' \in \mathit{vars}(\mathcal{H})) \implies ((x = x') \lor (s(x) \neq s(x')))]}{s \vdash \mathcal{H}\ \texttt{non-aliasing}}$$

$\boxed{\text{F12}}$ - H-TARGET-ACCURACY

$$\frac{\forall(x, f, r, r_f) : [((x.f@(r \rightarrowtail r_f) \in \mathit{reg\text{-}refs}(\mathcal{H})) \land (r_f \in \mathit{dom}(\mathcal{H}))) \implies (\mathrm{P}_r(h \restriction_v (s(x)).f) = r_f))]}{\mathrm{P}, h, s \vdash \mathcal{H}\ \texttt{target-accurate}}$$

$\boxed{\text{F13}}$ - OMEGA-BOUNDING

$$\frac{\mathit{dom}(\mathcal{H}) \cup \tau\mathit{targets}(\mathcal{H}) \cup \mathit{range}(\mathrm{P} \restriction_r) \subseteq \Omega \qquad \mathit{range}(\Gamma \restriction_r) \subseteq \Omega \cup \{\bot\}}{\Omega \vdash \mathcal{H}, \Gamma, \mathrm{P}\ \texttt{well-bounded}}$$

this rule does not exist in the undecorated system

3 Meta Rules

$\boxed{\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi}$

$\boxed{\text{M1A}}$ - FORWARD-REGION-REACHABILITY

$$\frac{\mathrm{P}(l) = (r, \tau) \qquad \texttt{iso}\ f\ \tau_f \in \mathit{fields}(\tau) \qquad \chi = r \qquad \chi' = l.f}{\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi'}$$

$\boxed{\text{M1B}}$ - BACKWARD-REGION-REACHABILITY

$$\frac{\texttt{iso}\ f\ \tau_f \in \mathit{fields}(h \restriction_\tau (l)) \qquad \mathrm{P} \restriction_r (l') = r \qquad \begin{array}{c} h \restriction_v (l).f = l' \\ \mathit{ref\text{-}valid}(\mathcal{H}, s, l, f) \end{array} \qquad \chi = l.f \qquad \chi' = r}{\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi'}$$

$\boxed{\text{M1C}}$ - TRANSITIVE-REGION-REACHABILITY

$$\frac{\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi' \hookrightarrow \chi''}{\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi''}$$

$\boxed{\text{M1D}}$ - REFLEXIVE-REGION-REACHABILITY

$$\mathcal{H}, \mathrm{P}, h, s \vdash \chi \hookrightarrow \chi$$

$\boxed{h \vdash l \hookrightarrow l}$

$\boxed{\text{M2A}}$ - FORWARD-LOCATION-REACHABILITY

$$\frac{q_{\text{RET}}\ f\ \tau' \in \mathit{fields}(\tau) \qquad h(l) = (\tau, v) \qquad v.f = l'}{h \vdash l \hookrightarrow l'}$$

$\boxed{\text{M2B}}$ - Transitive-Location-Reachability

$$\frac{h \vdash l \hookrightarrow l' \hookrightarrow l''}{h \vdash l \hookrightarrow l''}$$

$\boxed{\text{M2C}}$ - Reflexive-Location-Reachability

$$h \vdash l \hookrightarrow l$$

$$\boxed{h \vdash l \xhookrightarrow{\text{BND}} l}$$

$\boxed{\text{M3A}}$ - Forward-Location-Reachability

$$\frac{\texttt{bnd}\ f\ \tau' \in \mathit{fields}(\tau) \qquad h(l) = (\tau, v) \qquad v.f = l'}{h \vdash l \hookrightarrow l'}$$

$\boxed{\text{M3B}}$ - Transitive-Location-Reachability

$$\frac{h \vdash l \xhookrightarrow{\text{BND}} l' \xhookrightarrow{\text{BND}} l''}{h \vdash l \xhookrightarrow{\text{BND}} l''}$$

$\boxed{\text{M3C}}$ - Reflexive-Location-Reachability

$$h \vdash l \xhookrightarrow{\text{BND}} l$$

4 Typing Rules

$$\boxed{\vdash p}$$

$\boxed{\text{T0}}$ - Program Typing

$$\frac{\vdash \text{FDEF}_1 \ \ldots \ \vdash \text{FDEF}_n \qquad \cdot;\cdot;\cdot;\cdot \vdash e : r\ \tau \dashv \mathcal{H};\Gamma;\Omega}{\vdash \text{FDEF}_1;\ldots;\text{FDEF}_n;e}$$

$$\boxed{\vdash q_{\text{RET}}\ \tau\ \mathit{fn}(q_{\text{ARG}}\ \tau\ x)\{e\}}$$

$\boxed{\text{T1}}$ - Function-Definition-Typing

$$\frac{\begin{array}{c}(\mathit{fn}, (q_{\text{ARG}}\ \tau \to q_{\text{RET}}\ \tau')) \in \mathcal{F} \\ (r^{\dagger}\langle\rangle; x : r\ \tau; \{r\}; \cdot) \vdash e : r'\ \tau' \dashv (\mathcal{H}; x : r_{\mathit{final}}\ \tau; \{r\} \uplus \Omega_{\mathit{out}} \uplus \Omega_{\mathit{extra}}) \\ \vdash (q_{\text{ARG}}\ r \to q_{\text{RET}}\ r') : (r^{\circ}\langle\rangle; \{r\}) \Rightarrow (\mathcal{H}; \{r\} \uplus \Omega_{\mathit{out}})\end{array}}{\vdash \texttt{def}\ q_{\text{RET}}\ \tau'\ \mathit{fn}(q_{\text{ARG}}\ \tau\ x)\{e\}}$$

$$\boxed{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H};\Gamma;\Omega}$$

$\boxed{\text{T2}}$ - Variable-Ref-Typing

$$\frac{r \in \mathit{dom}(\mathcal{H}) \qquad x : r\ \tau \in \Gamma}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash x : r\ \tau \dashv \mathcal{H};\Gamma;\Omega}$$

T3 - Sequence-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega' \qquad \mathcal{H}';\Gamma';\Omega';\cdot \vdash e' : r'\ \tau' \dashv \mathcal{H}'';\Gamma'';\Omega''}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;e' : r'\ \tau' \dashv \mathcal{H}'';\Gamma'';\Omega''}$$

T4 - Bounded-Field-Reference-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega' \qquad \texttt{bnd}\ f\ \tau_f \in \mathit{fields}(\tau)}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e.f : r\ \tau_f \dashv \mathcal{H}';\Gamma;\Omega'}$$

T5 - Isolated-Field-Reference-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\cdot \vdash x : r\ \tau \dashv \mathcal{H};\Gamma;\Omega \qquad \texttt{iso}\ f\ \tau_f \in \mathit{fields}(\tau) \qquad \mathcal{H} = \mathcal{H}', r^{\circ}\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ'}\langle X'\rangle}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash x.f : r_f\ \tau_f \dashv \mathcal{H};\Gamma;\Omega}$$

T6L - Bounded-Field-Assignment-Typing–Left-Eval

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_f : r\ \tau_f \dashv \mathcal{H}', r^{\circ}\langle X\rangle;\Gamma';\Omega' \\ \mathcal{H}', r^{\dagger}\langle X\rangle;\Gamma';\Omega';\cdot \vdash e : r\ \tau \dashv \mathcal{H}'', r^{\dagger}\langle X'\rangle;\Gamma'';\Omega'' \qquad \texttt{bnd}\ f\ \tau_f \in \mathit{fields}(\tau)\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e.f = e_f : r\ \tau_f \dashv \mathcal{H}'', r^{\circ}\langle X'\rangle;\Gamma'';\Omega''}$$

T6R - Bounded-Field-Assignment-Typing–Right-Eval

$$\frac{(l : r\ \tau_f) \in \mathrm{P}, r^{\dagger}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}', r^{\dagger}\langle X'\rangle;\Gamma';\Omega'}{\mathcal{H}, r^{\circ}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash e.f = l : r\ \tau_f \dashv \mathcal{H}', r^{\circ}\langle X'\rangle;\Gamma';\Omega'}$$

T7 - Isolated-Field-Assignment-Typing

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_f : r_f\ \tau_f \dashv \mathcal{H}', r^{\circ}\langle x[f \rightarrowtail r_{old}, F], X\rangle, r_f^{\circ_f}\langle X_f\rangle;\Gamma';\Omega' \\ (x : r\ \tau) \in \Gamma' \qquad \texttt{iso}\ f\ \tau_f \in \mathit{fields}(\tau)\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash x.f = e_f : r_f\ \tau_f \dashv \mathcal{H}', r^{\circ}\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ_f}\langle X_f\rangle;\Gamma';\Omega'}$$

T8 - Assign-Var-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma', x : r_{old}\ \tau;\Omega' \qquad x \notin \mathit{vars}(\mathcal{H}')}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash x = e : r\ \tau \dashv \mathcal{H}';\Gamma', x : r\ \tau;\Omega'}$$

T9L - Function-Application-Typing–Left-Eval

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_f : r_f\ (q_{\text{ARG}}\ \tau \rightarrow q_{\text{RET}}\ \tau') \dashv \mathcal{H}', r_f^{\circ}\langle X\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_f^{\dagger}\langle X\rangle;\Gamma';\Omega';\cdot \vdash e : r\ \tau \dashv \mathcal{H}'', r_f^{\dagger}\langle X'\rangle;\Gamma'';\Omega'' \\ \vdash (q_{\text{ARG}}\ r \rightarrow q_{\text{RET}}\ r') : (\mathcal{H}'', r_f^{\circ}\langle X'\rangle;\Omega'') \Rightarrow (\mathcal{H}''';\Omega'' \uplus \Omega_{out})\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_f(e) : r'\ \tau' \dashv \mathcal{H}''';\Gamma'';\Omega'' \uplus \Omega_{out}}$$

T9R - Function-Application-Typing–Right-Eval

$$\frac{\begin{array}{c}(l_f : r_f\ (q_{\text{ARG}}\ \tau \rightarrow q_{\text{RET}}\ \tau')) \in \mathrm{P} \qquad \mathcal{H}, r_f^{\dagger}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}', r_f^{\dagger}\langle X'\rangle;\Gamma';\Omega' \\ \vdash (q_{\text{ARG}}\ r \rightarrow q_{\text{RET}}\ r') : (\mathcal{H}', r_f^{\circ}\langle X'\rangle;\Omega') \Rightarrow (\mathcal{H}'';\Omega' \uplus \Omega_{out})\end{array}}{\mathcal{H}, r_f^{\circ}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash l_f(e) : r'\ \tau' \dashv \mathcal{H}'';\Gamma';\Omega' \uplus \Omega_{out}}$$

T10 - Function-Name-Typing

$$\frac{(\mathit{fn}, \tau) \in \mathcal{F} \qquad \mathcal{H};\Gamma;\Omega;\cdot \vdash \texttt{new-}\tau : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \mathit{fn} : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'}$$

$\boxed{\text{T11}}$ - New-Loc-Typing

$$\frac{r \notin \Omega}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{new-}\tau : r\ \tau \dashv \mathcal{H}, r^{\cdot}\langle\rangle;\Gamma;\Omega \uplus \{r\}}$$

$\boxed{\text{T12}}$ - Declare-Var-Typing

$$\frac{\begin{array}{c}\mathcal{H};\Gamma, x : \bot\ \tau;\Omega;\cdot \vdash e : r\ \tau' \dashv \mathcal{H}';\Gamma', x : r_{final}\ \tau;\Omega' \\ x \notin vars(\Gamma) \cup vars(\Gamma') \cup vars(\mathcal{H}) \cup vars(\mathcal{H}')\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\} : r\ \tau' \dashv \mathcal{H}';\Gamma';\Omega'}$$

$\boxed{\text{T13L}}$ - OPlus-Typing–Left-Eval

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_1 : r_1\ \tau \dashv \mathcal{H}', r_1^{\circ}\langle X\rangle;\Gamma';\Omega' \qquad \mathcal{H}', r_1^{\dagger}\langle X\rangle;\Gamma';\Omega';\cdot \vdash e_2 : r_2\ \tau \dashv \mathcal{H}'', r_1^{\dagger}\langle X'\rangle;\Gamma'';\Omega'' \\ \mathcal{H}'', r_1^{\circ}\langle X'\rangle;\Gamma'';\Omega'';\cdot \vdash \texttt{new-}\tau' : r\ \tau' \dashv \mathcal{H}''';\Gamma''';\Omega''' \qquad \vdash \tau \oplus \tau : \tau'\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_1 \oplus e_2 : r\ \tau' \dashv \mathcal{H}''';\Gamma''';\Omega'''}$$

$\boxed{\text{T13R}}$ - OPlus-Typing–Right-Eval

$$\frac{\begin{array}{c}(l_1 : r_1\ \tau) \in \mathrm{P} \qquad \mathcal{H}, r_1^{\dagger}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash e_2 : r_2\ \tau \dashv \mathcal{H}', r_1^{\dagger}\langle X'\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_1^{\circ}\langle X'\rangle;\Gamma';\Omega';\cdot \vdash \texttt{new-}\tau' : r\ \tau' \dashv \mathcal{H}'';\Gamma'';\Omega'' \qquad \vdash \tau \oplus \tau : \tau'\end{array}}{\mathcal{H}, r_1^{\circ}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash l_1 \oplus e_2 : r\ \tau' \dashv \mathcal{H}'';\Gamma'';\Omega''}$$

$\boxed{\text{T14}}$ - If-Statement-Typing

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_b : r_b\ \texttt{bool} \dashv \mathcal{H}';\Gamma';\Omega' \\ \mathcal{H}';\Gamma';\Omega';\cdot \vdash e_t : r\ \tau \dashv \mathcal{H}'';\Gamma'';\Omega_t \qquad \mathcal{H}';\Gamma';\Omega';\cdot \vdash e_f : r\ \tau \dashv \mathcal{H}'';\Gamma'';\Omega_f\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{if}(e_b)\{e_t\}\ \texttt{else}\ \{e_f\} : r\ \tau \dashv \mathcal{H}'';\Gamma'';\Omega_t \cup \Omega_f}$$

$\boxed{\text{T15}}$ - While-Statement-Typing

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e_b : r_b\ \texttt{bool} \dashv \mathcal{H};\Gamma;\Omega' \\ \mathcal{H};\Gamma;\Omega';\cdot \vdash e : r\ \tau \dashv \mathcal{H};\Gamma;\Omega'' \qquad \mathcal{H};\Gamma;\Omega'';\cdot \vdash \texttt{new-unit} : r_u\ \texttt{unit} \dashv \mathcal{H}';\Gamma';\Omega'''\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{while}(e_b)\{e\} : r_u\ \texttt{unit} \dashv \mathcal{H}';\Gamma';\Omega'''}$$

$\boxed{\text{T16}}$ - Focus-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle\rangle;\Gamma';\Omega' \qquad (x : r\ \tau \in \Gamma')}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{focus}\ x : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle x[]\rangle;\Gamma';\Omega'}$$

$\boxed{\text{T17}}$ - Explore-Typing

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle x[F], X\rangle;\Gamma';\Omega' \\ (x : r\ \tau) \in \Gamma' \qquad \texttt{iso}\ f\ \tau' \in \mathit{fields}(\tau) \qquad r_{new} \notin \Omega'\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{explore}\ x.f : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle x[f \rightarrowtail r_{new}, F], X\rangle, r^{\cdot}_{new}\langle\rangle;\Gamma';\Omega' \uplus \{r_{new}\}}$$

$\boxed{\text{T18}}$ - Retract-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle x[f \rightarrowtail r_{old}, F], X\rangle, r_{old}^{\circ_{old}}\langle\rangle;\Gamma';\Omega' \qquad r_e \neq r_{old}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{retract}\ x.f : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}\langle x[F], X\rangle;\Gamma';\Omega'}$$

$\boxed{\text{T19}}$ - Unfocus-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r\langle x[], X\rangle;\Gamma';\Omega' \qquad (x : r\ \tau) \in \Gamma'}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{unfocus}\ x : r_e\ \tau_e \dashv \mathcal{H}', r\langle X\rangle;\Gamma';\Omega'}$$

T20L - Attach-Typing–Left-Eval

$$\frac{\begin{array}{c} \mathcal{H};\Gamma;\Omega;\mathrm{P}\vdash e : r_e\ \tau_e \dashv \mathcal{H}', r_e^{\circ_e}\langle X_e\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_e^{\dagger}\langle X_e\rangle;\Gamma';\Omega';\cdot\vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}'', r_e^{\dagger}\langle X_e'\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma'';\Omega'' \\ \mathcal{H}'', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1\rangle;\Gamma'';\Omega'';\cdot\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}''', r_e^{\dagger}\langle X_e''\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\circ_2}\langle X_2\rangle;\Gamma''';\Omega''' \\ \mathcal{H}_{out} = \mathcal{H}'''[r_1\mapsto r_2], r_e^{\circ_e}\langle X_e''[r_1\mapsto r_2]\rangle, r_2^{\circ_2}\langle X_1'[r_1\mapsto r_2], X_2[r_1\mapsto r_2]\rangle \qquad r_e\neq r_1 \end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P}\vdash e;\texttt{attach}\ \{e_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'''[r_1\mapsto r_2];\Omega'''}$$

T20M - Attach-Typing–Middle-Eval

$$\frac{\begin{array}{c} (l : r_e\ \tau_e)\in\mathrm{P} \qquad \mathcal{H}, r_e^{\dagger}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P}\vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1\rangle;\Gamma';\Omega';\cdot\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}'', r_e^{\dagger}\langle X_e''\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\circ_2}\langle X_2\rangle;\Gamma'';\Omega'' \\ \mathcal{H}_{out} = \mathcal{H}''[r_1\mapsto r_2], r_e^{\circ_e}\langle X_e''[r_1\mapsto r_2]\rangle, r_2^{\circ_2}\langle X_1'[r_1\mapsto r_2], X_2[r_1\mapsto r_2]\rangle \qquad r_e\neq r_1 \end{array}}{\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P}\vdash l;\texttt{attach}\ \{e_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma''[r_1\mapsto r_2];\Omega''}$$

T20R - Attach-Typing–Right-Eval

$$\frac{\begin{array}{c} (l : r_e\ \tau_e)\in\mathrm{P} \qquad (l_1 : r_1\ \tau_1)\in\mathrm{P} \qquad \mathcal{H}, r_e^{\dagger}\langle X_e\rangle,\ r_1^{\dagger}\langle X_1\rangle;\Gamma;\Omega;\mathrm{P}\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\circ_2}\langle X_2\rangle;\Gamma';\Omega' \\ \mathcal{H}_{out} = \mathcal{H}'[r_1\mapsto r_2], r_e^{\circ_e}\langle X_e'[r_1\mapsto r_2]\rangle, r_2^{\circ_2}\langle X_1'[r_1\mapsto r_2], X_2[r_1\mapsto r_2]\rangle \qquad r_e\neq r_1 \end{array}}{\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma;\Omega;\mathrm{P}\vdash l;\texttt{attach}\ \{l_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'[r_1\mapsto r_2];\Omega'}$$

T21L - Swap-Typing–Left-Eval

$$\frac{\begin{array}{c} \mathcal{H};\Gamma;\Omega;\mathrm{P}\vdash e : r_e\ \tau_e \dashv \mathcal{H}', r_e^{\circ}\langle X_e\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_e^{\dagger}\langle X_e\rangle;\Gamma';\Omega';\cdot\vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}'', r_e^{\dagger}\langle X_e'\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma'';\Omega'' \\ \mathcal{H}'', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1\rangle;\Gamma'';\Omega'';\cdot\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}''', r_e^{\dagger}\langle X_e''\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\cdot}\langle X_2\rangle;\Gamma''';\Omega''' \\ \mathcal{H}_{out} = \mathcal{H}'''[r_1\mapsto r_2, r_2\mapsto r_1], r_e^{\circ}\langle X_e''[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_2^{\cdot}\langle X_1'[r_1\mapsto r_2, r_2\mapsto r_1]\rangle \\ r_e\neq r_1 \qquad r_e\neq r_2 \end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P}\vdash e;\texttt{swap}\ \{e_1\}\ \texttt{with}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'''[r_1\mapsto r_2, r_2\mapsto r_1];\Omega'''}$$

T21M - Swap-Typing–Middle-Eval

$$\frac{\begin{array}{c} (l : r_e\ \tau_e)\in\mathrm{P} \qquad \mathcal{H}, r_e^{\dagger}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P}\vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1\rangle;\Gamma';\Omega';\cdot\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}'', r_e^{\dagger}\langle X_e''\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\cdot}\langle X_2\rangle;\Gamma'';\Omega'' \\ \mathcal{H}_{out} = \mathcal{H}''[r_1\mapsto r_2, r_2\mapsto r_1], r_e^{\circ}\langle X_e''[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_2^{\cdot}\langle X_1'[r_1\mapsto r_2, r_2\mapsto r_1]\rangle \\ r_e\neq r_1 \qquad r_e\neq r_2 \end{array}}{\mathcal{H}, r_e^{\circ}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P}\vdash l;\texttt{swap}\ \{l_1\}\ \texttt{with}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma''[r_1\mapsto r_2, r_2\mapsto r_1];\Omega''}$$

T21R - Swap-Typing–Right-Eval

$$\frac{\begin{array}{c} (l : r_e\ \tau_e)\in\mathrm{P} \qquad (l_1 : r_1\ \tau_1)\in\mathrm{P} \qquad \mathcal{H}, r_e^{\dagger}\langle X_e\rangle, r_1^{\dagger}\langle X_1\rangle;\Gamma;\Omega;\mathrm{P}\vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r_1^{\dagger}\langle X_1'\rangle, r_2^{\cdot}\langle X_2\rangle;\Gamma';\Omega' \\ \mathcal{H}_{out} = \mathcal{H}'[r_1\mapsto r_2, r_2\mapsto r_1], r_e^{\circ}\langle X_e'[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1\mapsto r_2, r_2\mapsto r_1]\rangle, r_2^{\cdot}\langle X_1'[r_1\mapsto r_2, r_2\mapsto r_1]\rangle \\ r_e\neq r_1 \qquad r_e\neq r_2 \end{array}}{\mathcal{H}, r_e^{\circ}\langle X_e\rangle, r_1^{\cdot}\langle X_1\rangle;\Gamma;\Omega;\mathrm{P}\vdash l;\texttt{swap}\ \{l_1\}\ \texttt{with}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'[r_1\mapsto r_2, r_2\mapsto r_1];\Omega'}$$

T22 - Location-Ref-Typing

$$\frac{(l : r\ \tau)\in\mathrm{P} \qquad r\in dom(\mathcal{H})}{\mathcal{H};\Gamma;\Omega;\mathrm{P}\vdash l : r\ \tau \dashv \mathcal{H};\Gamma;\Omega}$$

$\boxed{\text{T23}}$ - Send-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega' \qquad \vdash (\texttt{consumes}\ r \rightarrow \texttt{iso}\ r') : (\mathcal{H}';\Omega') \Rightarrow (\mathcal{H}'';\Omega'')}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{send-}\tau(e) : r'\ \texttt{unit} \dashv \mathcal{H}'';\Gamma';\Omega''}$$

$\boxed{\text{T24}}$ - Receive-Typing

$$\frac{r \notin \Omega}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \texttt{recv-}\tau() : r\ \tau \dashv \mathcal{H}, r^{\cdot}\langle\rangle;\Gamma;\Omega \uplus \{r\}}$$

$\boxed{\text{T25}}$ - Drop-Variable-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}';\Gamma', x : r\ \tau;\Omega' \qquad x \notin \mathit{vars}(\mathcal{H}')}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e; \texttt{drop-var}\ x : r_e\ \tau_e \dashv \mathcal{H}';\Gamma';\Omega'}$$

$\boxed{\text{T26L}}$ - Drop-Region-Typing–Left-Eval

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r_e^{\circ_e}\langle X_e\rangle;\Gamma';\Omega' \\ \mathcal{H}', r_e^{\dagger}\langle X_e\rangle;\Gamma';\Omega';\cdot \vdash e_d : r\ \tau \dashv \mathcal{H}'', r_e^{\dagger}\langle X_e'\rangle, r^{\circ}\langle X'\rangle;\Gamma'';\Omega'' \qquad r \neq r_e\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e; \texttt{drop-reg}\ \{e_d\} : r_e\ \tau_e \dashv \mathcal{H}'', r_e^{\circ_e}\langle X_e'\rangle;\Gamma'';\Omega''}$$

$\boxed{\text{T26R}}$ - Drop-Region-Typing–Right-Eval

$$\frac{(l : r_e\ \tau_e) \in \mathrm{P} \qquad \mathcal{H}, r_e^{\dagger}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P} \vdash e_d : r\ \tau \dashv \mathcal{H}', r_e^{\dagger}\langle X_e'\rangle, r^{\circ}\langle X'\rangle;\Gamma';\Omega' \qquad r \neq r_e}{\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P} \vdash l; \texttt{drop-reg}\ \{e_d\} : r_e\ \tau_e \dashv \mathcal{H}', r_e^{\circ_e}\langle X_e'\rangle;\Gamma';\Omega'}$$

$\boxed{\text{T27}}$ - Detach-Typing

$$\frac{\begin{array}{c}x \notin \mathit{vars}(X) \qquad \mathcal{H}, r^{\circ}\langle X\rangle, r_{new}^{\cdot}\langle\rangle;\Gamma, x : r_{new}\ \tau;\Omega;\cdot \vdash e_{succ} : r_{out}\ \tau_{out} \dashv \mathcal{H}';\Gamma', x : r_{\mathit{final}}\ \tau;\Omega_{succ} \\ \mathcal{H}, r^{\circ}\langle X\rangle;\Gamma, x : r\ \tau;\Omega;\cdot \vdash e_{\mathit{fail}} : r_{out}\ \tau_{out} \dashv \mathcal{H}';\Gamma', x : r_{\mathit{final}}';\Omega_{\mathit{fail}}\end{array}}{\mathcal{H}, r^{\circ}\langle X\rangle;\Gamma, x : r\ \tau;\Omega;\mathrm{P} \vdash \texttt{detach}\ x\ \texttt{in}\ \{e_{succ}\}\ \texttt{else}\ \{e_{\mathit{fail}}\} : r_{out}\ \tau_{out} \dashv \mathcal{H}';\Gamma', x : \bot\ \tau;\Omega_{succ} \cup \Omega_{\mathit{fail}}}$$

$\boxed{\text{T28}}$ - Invalidate-Variable-Typing

$$\frac{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_{out}\ \tau_{out} \dashv \mathcal{H}';\Gamma', x : r\ \tau;\Omega'}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e; \texttt{invalidate-var}\ x : r_{out}\ \tau_{out} \dashv \mathcal{H}';\Gamma', x : \bot\ \tau;\Omega'}$$

5 Heap Rules

$\boxed{\vdash q_{\text{ARG}}\ r : \mathcal{H} \Rightarrow \mathcal{H}}$

$\boxed{\text{H1}}$ - Consumes-Heap-Effect

$$\vdash \texttt{consumes}\ r : \mathcal{H}, r^{\circ}\langle\rangle \Rightarrow \mathcal{H}$$

$\boxed{\text{H2}}$ - Preserves-Heap-Effect

$$\vdash \texttt{preserves}\ r : \mathcal{H}, r^{\circ}\langle\rangle \Rightarrow \mathcal{H}, r^{\circ}\langle\rangle$$

$\boxed{\vdash (q_{\text{ARG}}\ r \rightarrow q_{\text{RET}}\ r) : (\mathcal{H};\Omega) \Rightarrow (\mathcal{H};\Omega)}$

H3 - ISOLATED-FUNC-HEAP-EFFECT

$$\frac{\vdash q_{\text{ARG}}\ r : \mathcal{H} \Rightarrow \mathcal{H}' \qquad r_{new} \notin \Omega}{\vdash (q_{\text{ARG}}\ r \rightarrow \texttt{iso}\ r_{new}) : (\mathcal{H};\Omega) \Rightarrow (\mathcal{H}', r_{new}\langle\rangle; \Omega \uplus \{r_{new}\})}$$

H4 - CONSUMES-BOUNDED-FUNC-HEAP-EFFECT

$$\frac{\vdash (\texttt{consumes}\ r \rightarrow \texttt{iso}\ r') : (\mathcal{H};\Omega) \Rightarrow (\mathcal{H}';\Omega')}{\vdash (\texttt{consumes}\ r \rightarrow \texttt{bnd}\ r') : (\mathcal{H};\Omega) \Rightarrow (\mathcal{H}';\Omega')}$$

H5 - PRESERVES-BOUNDED-FUNC-HEAP-EFFECT

$$\frac{\vdash \texttt{preserves}\ r : \mathcal{H} \Rightarrow \mathcal{H}}{\vdash (\texttt{preserves}\ r \rightarrow \texttt{bnd}\ r) : (\mathcal{H};\Omega) \Rightarrow (\mathcal{H};\Omega)}$$

6 Evaluation Rules

E1A - COMMON-CONTEXT-STEP

$$\frac{(d,h,s,e) \xrightarrow{\text{eval}} (d',h',s',e') \qquad e \notin \mathit{VariableNames}}{(d,h,s,E[e]) \xrightarrow{\text{eval}} (d',h',s',E[e'])}$$

E1B - VAR-RESOLVE-CONTEXT-STEP

$$\frac{(d,h,s,x) \xrightarrow{\text{eval}} (d,h,s,l) \qquad \mathit{matches\text{-}field\text{-}access}(E) \implies \mathit{matches\text{-}}\textsc{bnd}\mathit{\text{-}fld\text{-}access}(E,x,h,s)}{(d,h,s,E[x]) \xrightarrow{\text{eval}} (d,h,s,E[l])}$$

E2 - VARIABLE-REF-STEP

$$\frac{s(x) = l \qquad l \in d}{(d,h,s,x) \xrightarrow{\text{eval}} (d,h,s,l)}$$

E3 - NEW-LOC-STEP

$$\frac{\texttt{extracts-fresh-heap-regfree}(h,\tau;h_{new},l) \qquad d_{new} = \mathit{dom}(h_{new})}{(d,h,s,\texttt{new-}\tau) \xrightarrow{\text{eval}} (d \uplus d_{new}, h \uplus h_{new}, s, l)}$$

E4 - SEQUENCE-STEP

$$(d,h,s,l;e) \xrightarrow{\text{eval}} (d,h,s,e)$$

E5 - OPLUS-STEP

$$\frac{l_1,l_2 \in d \qquad l_3 \notin \mathit{dom}(h) \qquad [[\oplus]](h \restriction_v (l_1), h \restriction_v (l_2)) = v_3 \qquad \vdash h \restriction_\tau (l_1) \oplus h \restriction_\tau (l_2) : \tau'}{(d,h,s,l_1 \oplus l_2) \xrightarrow{\text{eval}} (d \uplus \{l_3\}, h \uplus (l_3 \mapsto (\tau', v_3)), s, l_3)}$$

E6 - IF-TRUE-STEP

$$\frac{h \restriction_v (l) = \texttt{true} \qquad l \in d}{(d,h,s,\texttt{if}(l)\{e_t\}\ \texttt{else}\ \{e_f\}) \xrightarrow{\text{eval}} (d,h,s,e_t)}$$

$\boxed{\text{E7}}$ - IF-FALSE-STEP

$$\frac{h\restriction_v(l) = \texttt{false} \qquad l \in d}{(d, h, s, \texttt{if}(l)\{e_t\}\ \texttt{else}\ \{e_f\}) \xrightarrow{\text{eval}} (d, h, s, e_f)}$$

$$(d, h, s, \texttt{while}(e_{bool})\{e_{body}\}) \xrightarrow{\text{eval}} (d, h, s, \texttt{if}(e_{bool})\{e_{body}; \texttt{while}(e_{bool})\{e_{body}\}\}\ \texttt{else}\ \{\texttt{new-unit}\})$$

$\boxed{\text{E9}}$ - DECLARE-VAR-STEP

$$(d, h, s, \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\}) \xrightarrow{\text{eval}} (d, h, s[x \mapsto \bot], e; \texttt{drop-var}\ x)$$

$\boxed{\text{E10}}$ - ASSIGN-VAR-STEP

$$\frac{l \in d}{(d, h, s \uplus (x \mapsto l_{old}), x = l) \xrightarrow{\text{eval}} (d, h, s \uplus (x \mapsto l), l)}$$

$\boxed{\text{E11}}$ - FUNCTION-APPLICATION-STEP

$$\frac{\begin{array}{c} l_f, l \in d \qquad h(l_f) = ((q_{\text{ARG}}\ \tau \to q_{\text{RET}}\ \tau'), v_f) \\ F_d(v_f) = \lambda x.e \qquad e \equiv_\alpha e' \qquad FV(e') = \{x\} \qquad vars(e') \uplus dom(s) \end{array}}{(d, h, s, l_f(l)) \xrightarrow{\text{eval}} (d, h, s, \texttt{declare}\ x : \tau\ \texttt{in}\ \{x = l; e'\})}$$

$\boxed{\text{E14}}$ - BOUNDED-REFERENCE-STEP

$$\frac{l, l_f \in d \qquad h\restriction_v(l).f = l_f \qquad \texttt{bnd}\ f\ \tau \in \mathit{fields}(h\restriction_\tau(l))}{(d, h, s, l.f) \xrightarrow{\text{eval}} (d, h, s, l_f)}$$

$\boxed{\text{E15}}$ - ISOLATED-REFERENCE-STEP

$$\frac{l, l_f \in d \qquad s(x) = l \qquad h\restriction_v(l).f = l_f \qquad \texttt{iso}\ f\ \tau \in \mathit{fields}(h\restriction_\tau(l))}{(d, h, s, x.f) \xrightarrow{\text{eval}} (d, h, s, l_f)}$$

$\boxed{\text{E16}}$ - BOUNDED-ASSIGNMENT-STEP

$$\frac{l, l_f \in d \qquad \texttt{bnd}\ f\ \tau_f \in \mathit{fields}(\tau)}{(d, h \uplus (l \mapsto (\tau, v)), s, l.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, l_f)}$$

$\boxed{\text{E17}}$ - ISOLATED-ASSIGNMENT-STEP

$$\frac{s(x) = l \qquad l, l_f \in d \qquad \texttt{iso}\ f\ \tau_f \in \mathit{fields}(\tau)}{(d, h \uplus (l \mapsto (\tau, v)), s, x.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, l_f)}$$

$\boxed{\text{E18}}$ - Virtual-Command-Step

$$\frac{\mathit{LocationNames} \cap \mathit{subexprs}(\textsc{vir}) = \emptyset}{(d, h, s, l; \textsc{vir}) \xrightarrow{\text{eval}} (d, h, s, l)}$$

$\boxed{\text{E19}}$ - Function-Name-Step

$$\frac{(\mathit{fn}, \tau) \in \mathcal{F} \qquad v_f = F_v(\mathit{fn}) \qquad l \notin \mathit{dom}(h)}{(d, h, s, \mathit{fn}) \xrightarrow{\text{eval}} (d \uplus \{l\}, h \uplus (l \mapsto (\tau, v_f)), s, l)}$$

$\boxed{\text{E20a}}$ - Detach-Step–Success

$$\frac{\mathit{heap\text{-}separable}(h, s, E^*[], x)}{(d, h, s, E^*[\texttt{detach}\ x\ \texttt{in}\ \{e_{succ}\}\ \texttt{else}\ \{e_{fail}\}]) \xrightarrow{\text{eval}} (d, h, s, E^*[e_{succ}; \texttt{invalidate-var}\ x])}$$

$\boxed{\text{E20b}}$ - Detach-Step–Failure

$$\frac{\neg \mathit{heap\text{-}separable}(h, s, E^*[], x)}{(d, h, s, E^*[\texttt{detach}\ x\ \texttt{in}\ \{e_{succ}\}\ \texttt{else}\ \{e_{fail}\}]) \xrightarrow{\text{eval}} (d, h, s, E^*[e_{fail}; \texttt{invalidate-var}\ x])}$$

7 Concurrency Rules

$\boxed{\vdash\ (h, \overline{\langle d, s, e \rangle})}$

$\boxed{\text{TC1}}$ - Concurrent-Well-Typedness

$$\frac{\begin{array}{c}\forall i \in \{1..n\} : (\mathcal{H}_i; \Gamma_i; \Omega_i; \mathrm{P} \vdash e_i : r_i\ \tau_i \dashv \mathcal{H}'_i; \Gamma'_i; \Omega'_i) \wedge (d_i, h, s_i : \mathcal{H}_i; \Gamma_i; \Omega_i; \mathrm{P}\ \texttt{agree}) \\ \forall i, j \in \{1..n\} : (d_i \cap d_j \neq \emptyset \implies i = j)\end{array}}{\vdash\ (h, \overline{\langle d_n, s_n, e_n \rangle})\ \texttt{well-typed}}$$

$$\boxed{(d, s, e; d, s, e) \xrightarrow{\text{comm-eval}} (d, s, e; d, s, e)}$$

$\boxed{\text{EC1}}$ - Concurrent-Single-Step

$$\frac{j \in \{1..n\} \qquad (d_j, h, s_j, e_j) \xrightarrow{\text{eval}} (d'_j, h', s'_j, e'_j) \qquad \forall i \in \{1..n\} - \{j\} : (d'_i, s'_i, e'_i) = (d_i, s_i, e_i)}{(h, \overline{\langle d_n, s_n, e_n \rangle}) \xrightarrow{\text{concur-eval}} (h', \overline{\langle d'_n, s'_n, e'_n \rangle})}$$

$\boxed{\text{EC2}}$ - Concurrent-Paired-Step

$$\frac{a, b \in \{1..n\} \qquad h \vdash (d_a, s_a, e_a; d_b, s_b, e_b) \xrightarrow{\text{comm-eval}} (d'_a, s'_a, e'_a; d'_b, s'_b, e'_b) \\ \forall n \in \{1..n\} - \{a, b\} : (d'_n, s'_n, e'_n) = (d_n, s_n, e_n)}{(h, \overline{\langle d_n, s_n, e_n \rangle}) \xrightarrow{\text{concur-eval}} (h, \overline{\langle d'_n, s'_n, e'_n \rangle})}$$

$$\boxed{(d, s, e; d, s, e) \xrightarrow{\text{comm-eval}} (d, s, e; d, s, e)}$$

$\boxed{\text{EC3}}$ - Communication-Paired-Step

$$\frac{d_{sep} = \{l \in dom(h) : h \vdash l_{root} \hookrightarrow l\}}{h \vdash (d_a \uplus d_{sep}, s_a, E^*_a[\texttt{send-}\tau(l_{root})]; d_b, s_b, E^*_b[\texttt{recv-}\tau()]) \xrightarrow{\text{comm-eval}} (d_a, s_a, E^*_a[\texttt{new-unit}]; d_b \uplus d_{sep}, s_b, E^*_b[l_{root}])}$$