Proof Appendix for the Type System

MATTHEW MILANO AND JOSHUA TURCOTTI

ACM Reference Format:
Matthew Milano and Joshua Turcotti. 2021. Proof Appendix for the Type System. 1, 1 (April 2021), 61 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 STRUCTURE AND ROLE OF THIS DOCUMENT

This document completely specifies the rules for the typing and evaluation of the system described in the accompanying paper, and provides proofs of progress and preservation for single-threaded and multi-threaded models of the system. The system as presented in the paper contains an entirely regions-free dynamic semantics, which proves that an implementation of the language can run without any tracking of the regions construct. This erasure comes with guarantees of performance and implementation simplicity, but makes proofwork difficult, so we introduce a *decorated* version of the system in which the typechecker communication information about the regions chosen during typechecking to the dynamic semantics through syntactic annotations. We will first provide an unabriged specification of the undecorated system in Section 2. We will state, but defer the proof, of Progress and Preservation for that undecorated system. We will then proceed to embed the undecorated system into a multi-threaded model in section 3, and give a strong thread-safety invariant on the concurrent configuration along with proofs that its Progress and Preservation follow from that of the single-threaded undecorated system. In section 4, we will introduce the unabriged specification of the decorated system. In section 4.8.1, we will state both Progress and Preservation for the decorated system, and projection lemmas proving that decorated Progress and Preservation imply undecorated Progress and Preservation. We will prove the former in section 5, and the latter in section 5. Once we have accomplished these goals, we will have proven that the system specified in the accompanying paper is sound, and provides a strong new option for safe concurrent development.

2 SPECIFICATION OF THE UNDECORATED SYSTEM

2.1 Preliminary Notions

2.1.1 The Contextual Graph. The invariants for this system rely on a graph of regions and isolated references formed from the information in the contexts $\mathcal{H}$, P, h, and s. We define two versions of this graph as follows:
$\mathbf{G}_{total}(\mathrm{P}, h)$

Given the contexts P and h, we will construct a directed graph $G_{total} = (N_{total}, E_{total})$. First, construct (add to N_{total}) a node r for each r in the range of $\mathrm{P} \restriction_r$ (call these region-nodes). Now, construct a node (l, f) (ref-nodes) for each reference `iso` $f\ \tau \in fields(h \restriction_\tau (l))$, for each location l in the domain of P, with constructed (added to E_{total}) edges

Author's address: Matthew Milano and Joshua Turcotti.

Manuscript submitted to ACM

$(\mathrm{P} \restriction_r (l), (l, q_{\textsc{ret}}, f))$ and $((l, q_{\textsc{ret}}, f), \mathrm{P} \restriction_r (\restriction_v (l).f)$.

$\mathbf{G}_S(\mathcal{H}, \mathrm{P}, h, s)$

In reality, we do not care about the entire graph $G_{total}(\mathrm{P}, h)$. With respect to enforcing simplicity, we care only about a subgraph. To obtain this subgraph, first we define the **semi-tracked** references $S_T(\mathcal{H}, \mathrm{P}, h, s)$ as the set of references $(l, \mathtt{iso}, f) \in N_{total}(\mathcal{H}, \mathrm{P}, h, s)$ such that $\mathrm{P} \restriction_r (l) \in dom(\mathcal{H})$ and $\nexists x : (s(x) = l) \wedge (x.f \in \textit{reg-refs}(\mathcal{H}))$. See section 2.5.1 for the definition of *reg-refs*. These are the references that originate in a tracked region but are not themselves tracked - and necessarily any isolated reference or region that could become tracked through a sequence of `focus` and `explore` commands will be reachable from a reference in S_T.

Now we consider the subset consisting of all nodes $\chi \in N_{total}(\mathrm{P}, h)$ such that $\mathcal{H}, \mathrm{P}, h, s \vdash \chi_s \hookrightarrow \chi$ (see M1A, this excludes `invalid` references) for some $\chi_s \in S_T(\mathcal{H}, \mathrm{P}, h, s)$. Call the induced subgraph of $G_{total}(\mathrm{P}, h)$ on this subset of nodes $G_S(\mathcal{H}, \mathrm{P}, h, s)$. We allow the derivation of the statement $\mathrm{P}, s \vdash h/\mathcal{H}$ `simple` iff $G_S(\mathcal{H}, \mathrm{P}, h, s)$ is a forest (see F4 below). Continuing with the intuition from out definition of S_T, we point out that G_S is exactly the graph of regions and isolated references in the current heap that are not currently usable but could become usable after a series of virtual commands. Guaranteeing that this graph is a forest guarantees that losing access to a tracked object through a `send` disallows us from accessing any objects that the thread gaining access to the object through `recv` would subsequently gain access to. This is a key property for thread safety, and justifies the inclusion of the judgment $\mathrm{P}, s \vdash h/\mathcal{H}$ `simple` in the well-typedness of a configuration.

2.1.2 The Type and Value Constructors. We define the relation:

$$\texttt{extracts-fresh-heap-regfree}(h_{old}, \tau_{root}; h_{new}, l_{root})$$

This encodes the information that the new heap h_{new} is fresh and well-formed, and rooted at the location l_{root} which has type τ_{root}. Its formal specification will be deferred to the decorated system's section 4.2.

2.1.3 Classes. As mentioned above, the only guarantee enforced here is that there are no cycles in the type graph containing isolated references.

2.1.4 Functions. The set of functions is computed statically in this pass. $\mathcal{F}$ is defined to be the set of function names, together with their types of the form $(q_{\textsc{arg}}\ \tau \longrightarrow q_{\textsc{ret}}\ \tau')$. After $\mathcal{F}$ is defined, rule T0 is used to check each function definition, possibly terminating type-checking with failure. If all checks are successful, then F_v will be a function mapping pairs $(fn, \tau) \in \mathcal{F}$ to values v_f, and F_d will be a function mapping values v_f to the annotated lambda expressions that will serve as the source of truth for function bodies. For each statement $\vdash q_{\textsc{ret}}\ \tau'\ fn(q_{\textsc{arg}}\ \tau\ x)e$ derived by T0, there exists some v_f such that $F_v(fn) = v_f$, and $F_d(v_f) = \lambda x.e$.

2.2 Grammar

We present the grammar for the language in the following figures, including all relevant contexts and namespaces.

2.2.1 *Reserved Namespaces.*

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

2.2.2 *Static Contexts.*

$$
\begin{aligned}
\text{(pinnedness metavariable) } \circ &::= \dagger \mid \cdot \\
\text{(heap tracking context) } \mathcal{H} &::= r^{\circ}\langle X\rangle,\ \mathcal{H} \mid \cdot \\
\text{(region tracking contents) } X &::= x[F],\ X \mid \cdot \\
\text{(variable tracking contents) } F &::= f \rightarrowtail r,\ F \mid \cdot \\
\text{(variable bindings context) } \Gamma &::= x : r\ \tau,\ \Gamma \mid \cdot \\
\text{(region names context) } \Omega &::= r,\ \Omega \mid \cdot \\
\text{(location bindings context) } \mathrm{P} &::= l : r\ \tau,\ \mathrm{P} \mid \cdot
\end{aligned}
$$

2.2.3 *Expressions and Programs.*

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

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

$$
\begin{aligned}
\text{(expression) } e ::=\ & l \mid x \mid e;e \mid e;\textsc{vir} \mid e.f \mid e.f = e \mid x = e \mid fn \mid e(e) \mid e \oplus e \mid \texttt{new-}\tau\\
& \mid \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\} \mid \texttt{if}(e)\{e\}\ \texttt{else}\ \{e\} \mid \texttt{while}(e)\{e\}\\
& \mid \texttt{send-}\tau(e) \mid \texttt{recv-}\tau \mid \texttt{detach}\ x\ \texttt{in}\ \{e\}\ \texttt{else}\ \{e\}
\end{aligned}
$$

$$
\begin{aligned}
\text{(evaluation context) } E[] ::=\ & [];e \mid [];\textsc{vir} \mid [].f \mid e.f = [] \mid [].f = l \mid x = [] \mid [](e) \mid l([]) \mid [] \oplus e \mid l \oplus []\\
& \mid \texttt{if}([])\{e\}\ \texttt{else}\ \{e\} \mid \texttt{send-}\tau([]) \mid l;\texttt{drop-reg}\ \{[]\}\\
& \mid l;\texttt{attach}\ \{[]\}\ \texttt{to}\ \{e\} \mid l;\texttt{attach}\ \{l\}\ \texttt{to}\ \{[]\}\\
& \mid l;\texttt{swap}\ \{[]\}\ \texttt{with}\ \{e\} \mid l;\texttt{swap}\ \{l\}\ \texttt{with}\ \{[]\}
\end{aligned}
$$

2.2.4 *Dynamic Contexts.*

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

2.3 Typing Rules

We present the typing rules for the language in the following figures, including programs, functions, and virtual commands.

2.3.1 *Program Typing.* $\boxed{\vdash p}$

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

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

2.3.2 *Function Definition Typing.* $\boxed{\vdash q_{\text{RET}}\ \tau\ fn(q_{\text{ARG}}\ \tau\ x)\{e\}}$

T1 - Function-Definition-Typing

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

2.3.3 *Expression Typing.* $\boxed{\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}; \Gamma; \Omega}$

T2 - Variable-Ref-Typing

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

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

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

T5 - Isolated-Field-Reference-Typing

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

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

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

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

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

T8 - Assign-Var-Typing

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

T9L - Function-Application-Typing–Left-Eval

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

T9R - Function-Application-Typing–Right-Eval

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

T10 - Function-Name-Typing

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

T11 - New-Loc-Typing

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

T12 - Declare-Var-Typing

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

T13L - OPlus-Typing–Left-Eval

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

T13R - OPlus-Typing–Right-Eval

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

T14 - If-Statement-Typing

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

T15 - While-Statement-Typing

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

T16 - Focus-Typing

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

T17 - Explore-Typing

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

T18 - Retract-Typing

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

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

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ_e}_e\langle X_e\rangle;\Gamma';\Omega' \qquad \mathcal{H}', r^{\dagger}_e\langle X_e\rangle;\Gamma';\Omega';\cdot \vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}'', r^{\dagger}_e\langle X'_e\rangle, r^{\cdot}_1\langle X_1\rangle;\Gamma'';\Omega'' \\ \mathcal{H}'', r^{\dagger}_e\langle X'_e\rangle, r^{\dagger}_1\langle X_1\rangle;\Gamma'';\Omega'';\cdot \vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}''', r^{\dagger}_e\langle X''_e\rangle, r^{\dagger}_1\langle X'_1\rangle, r^{\circ_2}_2\langle X_2\rangle;\Gamma''';\Omega''' \\ \mathcal{H}_{out} = \mathcal{H}'''[r_1 \mapsto r_2], r^{\circ_e}_e\langle X''_e[r_1 \mapsto r_2]\rangle, r^{\circ_2}_2\langle X'_1[r_1 \mapsto r_2], X_2[r_1 \mapsto r_2]\rangle \qquad r_e \neq r_1\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{attach}\ \{e_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'''[r_1 \mapsto r_2];\Omega'''}$$

T20M - Attach-Typing–Middle-Eval

$$\frac{\begin{array}{c}(l : r_e\ \tau_e) \in \mathrm{P} \qquad \mathcal{H}, r^{\dagger}_e\langle X_e\rangle;\Gamma;\Omega;\mathrm{P} \vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}', r^{\dagger}_e\langle X'_e\rangle, r^{\cdot}_1\langle X_1\rangle;\Gamma';\Omega' \\ \mathcal{H}', r^{\dagger}_e\langle X'_e\rangle, r^{\dagger}_1\langle X_1\rangle;\Gamma';\Omega';\cdot \vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}'', r^{\dagger}_e\langle X''_e\rangle, r^{\dagger}_1\langle X'_1\rangle, r^{\circ_2}_2\langle X_2\rangle;\Gamma'';\Omega'' \\ \mathcal{H}_{out} = \mathcal{H}''[r_1 \mapsto r_2], r^{\circ_e}_e\langle X''_e[r_1 \mapsto r_2]\rangle, r^{\circ_2}_2\langle X'_1[r_1 \mapsto r_2], X_2[r_1 \mapsto r_2]\rangle \qquad r_e \neq r_1\end{array}}{\mathcal{H}, r^{\circ_e}_e\langle X_e\rangle;\Gamma;\Omega;\mathrm{P} \vdash l;\texttt{attach}\ \{e_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma''[r_1 \mapsto r_2];\Omega''}$$

T20R - Attach-Typing–Right-Eval

$$\frac{\begin{array}{c}(l : r_e\ \tau_e) \in \mathrm{P} \qquad (l_1 : r_1\ \tau_1) \in \mathrm{P} \qquad \mathcal{H}, r^{\dagger}_e\langle X_e\rangle,\ r^{\dagger}_1\langle X_1\rangle;\Gamma;\Omega;\mathrm{P} \vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}', r^{\dagger}_e\langle X'_e\rangle, r^{\dagger}_1\langle X'_1\rangle, r^{\circ_2}_2\langle X_2\rangle;\Gamma';\Omega' \\ \mathcal{H}_{out} = \mathcal{H}'[r_1 \mapsto r_2], r^{\circ_e}_e\langle X'_e[r_1 \mapsto r_2]\rangle, r^{\circ_2}_2\langle X'_1[r_1 \mapsto r_2], X_2[r_1 \mapsto r_2]\rangle \qquad r_e \neq r_1\end{array}}{\mathcal{H}, r^{\circ_e}_e\langle X_e\rangle, r^{\cdot}_1\langle X_1\rangle;\Gamma;\Omega;\mathrm{P} \vdash l;\texttt{attach}\ \{l_1\}\ \texttt{to}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'[r_1 \mapsto r_2];\Omega'}$$

T21L - Swap-Typing–Left-Eval

$$\frac{\begin{array}{c}\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r_e\ \tau_e \dashv \mathcal{H}', r^{\circ}_e\langle X_e\rangle;\Gamma';\Omega' \qquad \mathcal{H}', r^{\dagger}_e\langle X_e\rangle;\Gamma';\Omega';\cdot \vdash e_1 : r_1\ \tau_1 \dashv \mathcal{H}'', r^{\dagger}_e\langle X'_e\rangle, r^{\cdot}_1\langle X_1\rangle;\Gamma'';\Omega'' \\ \mathcal{H}'', r^{\dagger}_e\langle X'_e\rangle, r^{\dagger}_1\langle X_1\rangle;\Gamma'';\Omega'';\cdot \vdash e_2 : r_2\ \tau_2 \dashv \mathcal{H}''', r^{\dagger}_e\langle X''_e\rangle, r^{\dagger}_1\langle X'_1\rangle, r^{\cdot}_2\langle X_2\rangle;\Gamma''';\Omega''' \\ \mathcal{H}_{out} = \mathcal{H}'''[r_1 \mapsto r_2, r_2 \mapsto r_1], r^{\circ}_e\langle X''_e[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r^{\cdot}_1\langle X_2[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r^{\cdot}_2\langle X'_1[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle \\ r_e \neq r_1 \qquad r_e \neq r_2\end{array}}{\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e;\texttt{swap}\ \{e_1\}\ \texttt{with}\ \{e_2\} : r_e\ \tau_e \dashv \mathcal{H}_{out};\Gamma'''[r_1 \mapsto r_2, r_2 \mapsto r_1];\Omega'''}$$

T21M - Swap-Typing–Middle-Eval

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

T21R - Swap-Typing–Right-Eval

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

T22 - Location-Ref-Typing

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

T23 - Send-Typing

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

T24 - Receive-Typing

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

T25 - Drop-Variable-Typing

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

T26L - Drop-Region-Typing–Left-Eval

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

T26R - Drop-Region-Typing–Right-Eval

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

T27 - Detach-Typing

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

T28 - Invalidate-Variable-Typing

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

2.3.4 Heap Rules. $\boxed{\vdash q_{\text{ARG}}\ r : \mathcal{H} \Rightarrow \mathcal{H}}$

H1 - Consumes-Heap-Effect

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

H2 - Preserves-Heap-Effect

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

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

H3 - Isolated-Func-Heap-Effect

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

H4 - Consumes-Bounded-Func-Heap-Effect

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

H5 - Preserves-Bounded-Func-Heap-Effect

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

2.4 Stepping Rules

2.4.1 Notation. We define two predicates that will be of use in the formulation of the following rules.

The first, *matches-field-access*(E), holds iff E is an evaluation context that was derived from either of the forms $[\,].f$ (field references) or $[\,].f = l$ (field assignment) in the grammar.

The second, *matches-*BND*-fld-access*(E, x, h, s), holds iff E is an evaluation context that was derived from either of the forms $[\,].f$ (field reference) or $[\,].f = l$ (field assignment) in the grammar, where the field of $[\,] = x$ being accessed is bounded. Specifically, where $f \in FieldNames$ was used in the rule deriving E, we have that $\texttt{bnd}\ f\ \tau' \in fields(h \upharpoonright_{\tau} (s(x)))$ for some τ' (noting that this predicate can still be computed with a single pass over the small set $fields(h \upharpoonright_{\tau} (s(x)))$).

Both of these will be used in the formulation of rule E1B.

Also of use in restricting the scope of Evaluation Context stepping rules, we use the term *detaching* to refer to all expressions whose *bottom-most base* expression n (see linked definitions) is a `detach`, and *non-detaching* to refer to all expressions whose *bottom-most base* expression is not a `detach`. This is because `detach`s are very context-sensitive, and cannot step in an arbitrary enclosing evaluation context. In fact, it will directly need to reference the set of locations in any enclosing nested sequence of zero or more evaluation contexts $E^*[\,]$. We define the term $locs(E^*[\,])$ to refer to the set of locations that syntactically occur in a sequence of evaluation contexts - i.e. those referenced in the grammar rules deriving any $E[\,]$ in the sequence.

To specify the stepping of `detach`, we need more terms. First, for any location l, given the heap h, let

$$min\text{-}reg(h, l) = \{l_{src} \in dom(h) : \exists l' : h \vdash l \xhookrightarrow{\text{BND}} l' \wedge h \vdash l_{src} \xhookrightarrow{\text{BND}} l'\}$$

Second, for stack s and sequence of evaluation contexts $E^*[]$, let

$$rem\text{-}root\text{-}set(h, s, E^*[], x) := range(s \restriction_{dom(s)-\{x\}}) \cup locs(E^*[])$$

Third, define the proposition

$$heap\text{-}separable(h, s, E^*[], x) := (min\text{-}reg(h, s(x)) \cap rem\text{-}root\text{-}set(h, s, E^*[], x) = \emptyset)$$

We also define the function $FV(e)$, which takes an expression e and returns the set of free variables in e, and the function $NR(e)$, which takes an expression e and returns the set of regions that syntactically appear anywhere in e

Finally, we will require in two instances (E8 and E11) bijective maps renaming regions. The notation $bijections(_1, _2)$ represents the set of bijections from $_1 \subseteq RegionNames$ to $_2 \subseteq RegionNames$ extended to be an involution on $_1 \cup _2$ and to be constant on $RegionNames - (_1 \cup _2)$. Some map $\phi \in bijections(_1, _2)$ can act on any symbol defined in this system, ignoring everything but region names upon which it acts as specified.

2.4.2 Rules. $\boxed{(d, h, s, e) \xrightarrow{\text{eval}} (d, h, s, e)}$

E1a - Common-Context-Step

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

E1b - Var-Resolve-Context-Step

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

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

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

E4 - Sequence-Step

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

E5 - OPlus-Step

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

$\boxed{\text{E6}}$ - 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

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

$\boxed{\text{E8}}$ - While-Step

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

$\boxed{\text{E9}}$ - 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

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

$\boxed{\text{E12}}$ - Bounded-Reference-Step

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

$\boxed{\text{E13}}$ - Isolated-Reference-Step

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

$\boxed{\text{E14}}$ - Bounded-Assignment-Step

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

$\boxed{\text{E15}}$ - Isolated-Assignment-Step

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

$\boxed{\text{E16}}$ - Virtual-Command-Step

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

E17 - Function-Name-Step

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

E18a - Detach-Step–Success

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

E18b - Detach-Step–Failure

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

2.5 Invariants

In this section we introduce the invariants describing soundness and other key properties of the type system as inference rules (sec 2.5.2), with verbal descriptions of their functionality (sec 2.5.3), and assorted supporting “meta” rules (sec 2.5.4) defining reachability with respect to our contexts. We begin with some necessary notation.

2.5.1 Predicates and Notation. We use the term $loc\text{-}refs(\mathcal{H}, s)$ to refer to the set of terms $l.f$ for which x is a tracked variable in $\mathcal{H}$ with tracked reference f, and $s(x) = l$.

We use the term $vars(\mathcal{H})$ to refer to the set of variables x that syntactically occur in $\mathcal{H}$. We use the term $reg\text{-}vars(\mathcal{H})$ to refer to the set of terms $x@r$ such that r is a tracked region in $\mathcal{H}$ and x is tracked in r.

We use the term $reg\text{-}refs(\mathcal{H})$ to refer to the set of terms $x.f@(r \rightarrowtail r_f)$ such that f is a tracked field under a tracked variable x in tracked region r whose target region is r_f (r_f itself not necessarily tracked).

We use the term $ref\text{-}valid(\mathcal{H}, s, l, f)$ to refer to the proposition that, for all variables $x \in vars(\mathcal{H})$, if $s(x) = l$ and $x.f@(r \rightarrowtail r_f) \in reg\text{-}refs(\mathcal{H})$, then $r_f \in dom(\mathcal{H})$. The particular state of $x.f@(r \rightarrowtail r_f)$ *without* $r_f \in dom(\mathcal{H})$ signifies that $x.f$ is **invalid** - the predicate *ref-valid* is meant to allow us to conclude that $x.f$ is not invalid.

We use the term $live(l; \mathcal{H}, \Gamma, h, s, e)$ to refer to the proposition that, there exists a location l_{root} such that $h \vdash l_{root} \hookrightarrow l$ (see M2a - M2c), and either l_{root} syntactically occurs as a subexpression of e or there exists x with $s(x) = l$ and $\Gamma \restriction_r (x) \in dom(\mathcal{H})$.

We define an important concept: the $live\text{-}set(\mathcal{H}, \mathrm{P}, h, s)$ of locations will be the set of locations l in a region $\mathrm{P} \restriction_r (l)$ that is reachable (M1a - M1d) from an $\mathcal{H}$-tracked region. A key property that is noteworthy if not directly proven is that the *live-set* never grows as the result of a step to include new locations that aren’t also fresh (not previously in $dom(h)$).

When discussing reachability forms (see M1a and M2a), we use the notation $\vdash a \hookrightarrow b \hookrightarrow c$ to denote the conjunction of the two judgments $\vdash a \hookrightarrow b$ and $\vdash b \hookrightarrow c$, with appropriate contexts to the left of the turnstile.

Finally, we enforce the behavior of each context $\mathcal{H}, \Gamma, \mathrm{P}, h, s$ as a partial function - namely, disallowing duplicate keys. $\mathcal{H}$ is a particular case in which the codomain consists itself of partial functions, and any element of the codomain’s codomain also consists of partial functions. Specifically, in $\mathcal{H}$, regions are mapped to tuples containing pinnedness information for that region, and a partial map from variables in that region, to a partial map from fields under that variable to their target regions.

2.5.2 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

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

$\boxed{\text{F3}}$ - 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

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

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

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

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

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

$\boxed{\text{F10}}$ - Variable-Binding-Sanity

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

$\boxed{\text{F11}}$ - 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

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

this rule does not exist in the undecorated system

2.5.3 F Invariant Explanations.

- $\boxed{\text{F1}}$ - This judgment indicates that an expression typechecks and is well-formed with respect to the reservation d, regionality map P, and dynamic contexts h, s.
- $\boxed{\text{F2}}$ - This judgment indicates that all of the provided static and dynamic contexts are in agreement with each other and are well-formed.
- $\boxed{\text{F3}}$ - This judgment indicates that the reachable region graph is a forest - indicating that all untracked objects are safe to send away without the possibility of ensuing races.
- $\boxed{\text{F4}}$ - This judgment indicates that the reservation d is sufficiently large to contain all locations reachable from the current context.
- $\boxed{\text{F5}}$ - This judgment indicates that the static model of the heap, $\mathcal{H}$, contains no gaps or other breaches of well-formedness in its tracking.
- $\boxed{\text{F6}}$ - This judgment indicates that the heap h is closed under following field references.
- $\boxed{\text{F7}}$ - This judgment indicates that the heap h and the regionality map P agree on the types of all locations.
- $\boxed{\text{F8}}$ - This judgment indicates that any bounded references from *live* locations stay within their region.
- $\boxed{\text{F9}}$ - This judgment indicates that $\mathcal{H}$ and Γ agree on their bindings of variables.
- $\boxed{\text{F10}}$ - This judgment indicates that the bindings of tracked variables have the correct regionality.
- $\boxed{\text{F11}}$ - This judgment indicates that no tracked variables alias each other.
- $\boxed{\text{F12}}$ - This judgment indicates that the targets of references described in $\mathcal{H}$ are accurate.
- $\boxed{\text{F13}}$ - This judgment indicates that only region names from Ω are present in any context.

2.5.4 Supporting (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

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

$\boxed{\text{M1C}}$ - 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

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

$\boxed{\text{M3B}}$ - 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

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

2.6 Statements of Progress and Preservation

2.7 Definitions

***base* expression:** An expression e that cannot be expressed as $E[\bar{e}]$ for evaluation contexts $E[]$ and expression $\bar{e} \notin \mathit{LocationNames}$.

***bottom-most* base expression:** Given e, the unique *base* expression $\bar{e}$ such that $e = E^*[\bar{e}]$ for a nested sequence $E^*[]$ of zero or more evaluation contexts.

***non-blocking* expression:** An expression e whose *bottom-most base* expression is not of the form `send-`$\tau(l)$ or `recv-`$\tau()$.

2.8 Statement of Main Theorems

Above, we provided typing rules for the multi-thread communication primitives `send` and `recv`. Unfortunately, their effects on the heap are only sane in a multi-threaded semantics in which the two can step in parallel. We thus only claim Progress and Preservation for *non-blocking* expressions:

Theorem 2.1 (Progress). *For any well-typed configuration* $\vdash\ (d, h, s, e : r\ \tau)$ *where* $e \notin$ *LocationNames is a non-blocking expression, there exists another dynamic configuration* (d', h', s', e') *such that* $(d, h, s, e)\xrightarrow{eval}(d', h', s', e')$.

Theorem 2.2 (Preservation). *For any well-typed configuration* $\vdash\ (d, h, s, e\ :\ r\ \tau)$ *that steps with the relation* $(d, h, s, e)\xrightarrow{eval}(d', h', s', e')$*, the configuration* $\vdash\ (d', h', s', e' : r\ \tau)$ *is also well-typed.*

3 CONCURRENCY

In this section we give an approach for embedding the single-thread semantics described above into a multi-thread model. We state Progress and Preservation for this multi-threaded system, and show that they are easily provable as an extension of Progress and Preservation of the single-threaded system. We begin with two small items of notation.

3.1 Notation

Grammar Addition:

$$\text{(Vector) } \forall A : \overline{\langle A \rangle} := A, \overline{\langle A \rangle} \mid \cdot$$

$$\text{(Evaluation Context Sequence) } E^*[] := E[E^*[]] \mid []$$

3.2 Rules

Below we give the typing and evaluation rules for the concurrent semantics. There is a single typing rule, which states that each thread wraps a well-typed expression with sane, agreeing contexts. Note that the contexts h, P are global to all threads, whereas $d, s, \mathcal{H}, \Gamma, \Omega, \mathcal{H}', \Gamma', \Omega'$ are separated out into local copies for each thread. There are two stepping rules that can derive a step between concurrent configurations. The first corresponds to the case in which a single thread steps a *non-blocking* expression, updating its local contexts and expression as well as the global contexts. The second illustrates the case in which two threads have independently reached a `send-`$\tau(l)$ and a `recv-`$\tau()$ as their *bottom-most base* expressions, and the heap reachable from l is transferred from the first thread to the second using the communication evaluation rule EC3. This is our communication primitive in the multi-threaded Gallifrey semantics.

3.2.1 *Typing (TC) Rules.* $\boxed{\vdash\ (h, \overline{\langle d, s, e \rangle})}$

TC1 - Concurrent-Well-Typedness

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

3.2.2 *Stepping (EC) Rules.* $\boxed{(d, s, e; d, s, e) \xrightarrow{\text{comm-eval}} (d, s, e; d, s, e)}$

$\boxed{\text{EC1}}$ - 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

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

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

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

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

3.3 Concurrent Progress and Preservation

We state and prove Progress and Preservation in this concurrency model. Since well-typedness of a concurrent configuration includes pairwise disjointedness of the respective reservations of each thread, and no thread is allowed to access locations outside its reservation, Progress and Preservation encode a strong form of thread-safety.

3.3.1 *Statements.*

Theorem 3.1 (Concurrent Progress). *For any well-typed concurrent configuration $(h, d_n, \vec{s_n}, e_n)$ there exists another dynamic configuration $(h', d'_n, \vec{s'_n}, e'_n)$ such that $(h, d_n, \vec{s_n}, e_n) \xrightarrow{\textit{concur-eval}} (h', d'_n, \vec{s'_n}, e'_n)$, as long as one of the following two conditions holds:*

i) for some j: $e_j \notin LocationNames$ is a non-blocking expression

ii) for some τ, a, b: e_a and e_b admit $\texttt{send-}\tau(l)$ and $\texttt{recv-}\tau()$, respectively as their bottom-most base expressions.

Theorem 3.2 (Concurrent Preservation). *For any well-typed configuration $(h, d_n, \vec{s_n}, e_n)$ that steps with the relation $(h, d_n, \vec{s_n}, e_n) \xrightarrow{\textit{concur-eval}} (h', d'_n, \vec{s'_n}, e'_n)$, the configuration $(h', d'_n, \vec{s'_n}, e'_n)$ is also well-typed.*

3.3.2 Useful Lemmas. The following lemmas are useful in our proof of Concurrent Progress and Preservation. They are provided here without proof, but will be proven and discussed in more detail in the decorated system's Progress and Preservation argument in section 5.

Lemma 3.3 (Inner Typing Dependence). *For any expression e of the form $E[\bar{e}]$ for some expression $\bar{e} \notin LocationNames$, if $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$ is a well-typed configuration with respect to some dynamic context, then there exist $\bar{r}, \bar{\tau}, \bar{\mathcal{H}}', \bar{\Gamma}', \bar{\Omega}'$ such that $\mathcal{H}; \Gamma; \Omega; P \vdash \bar{e} : \bar{r}\ \bar{\tau} \dashv \bar{\mathcal{H}}'; \bar{\Gamma}'; \bar{\Omega}'$ is also a well-typed configuration and $\bar{\Omega}' \subseteq \Omega'$.*

Lemma 3.4 (Evaluation Context Injectivity). *If $E_1[e_1] = E_2[e_2]$, where $E_1[], E_2[]$ are evaluation contexts and e_1, e_2 are expressions not in LocationNames, then $E_1 = E_2$ and $e_1 = e_2$.*

Lemma 3.5. *All expressions in the language e can be uniquely decomposed as $E^*[e]$ for some sequence E^* of zero or more evaluation contexts and a base expression e.*

The following lemmas will be of key importance to our Concurrent Progress and Preservation proofs, but are new and stated with proof.

Lemma 3.6 (Syntactic Location Region Tracking). *For any location l syntactically present in a well-typed expression $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$, $P \restriction_r (l)$ is a tracked region in $\mathcal{H}$.*

Proof. By induction on the unique structure $e = E^*[\bar{e}]$ for *base* expression $\bar{e}$ (see lemma 5.1). If e is a *base* expression, then either $e = l$ for some location l, in which case inversion of T22 trivially yields the claim, or e contains no syntactic locations, as can be seen by inspection of the remaining rules and ruling out choices of subexpressions that would allow further decomposition into evaluation contexts. To complete the induction, we let $e = E[e']$ and show that if $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e : r\tau \dashv \mathcal{H}'; \Gamma'; \Omega'$ and $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e' : r'\ \tau' \dashv \mathcal{H}''; \Gamma''; \Omega''$ (WLOG by 5.21) and all locations syntactically present in e' have their regions tracked in $\mathcal{H}$, then all of the locations that syntactically appear in $E[]$ are (exactly the difference between those in e and e') have their regions tracked in $\mathcal{H}$. Simple inspection of the rules for deriving $E[]$ in the grammar, and inversion of the corresponding typing rules in each case, yields our claim, and we are done. □

Lemma 3.7 (Heap Monotonicity Preservation Strengthening). *If $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$, $(d, h, s : \mathcal{H}; \Gamma; \Omega; P\ \texttt{agree})$, and $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$, then:*

i) h is a subfunction of h', $dom(h') - dom(h) \subseteq d' - d$

ii) we can choose $\bar{\mathrm{P}}$ such that $\exists \bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{\Omega}' : \bar{\mathcal{H}}'; \bar{\Gamma}'; \bar{\Omega}'; \bar{P} \vdash e' : r\ \tau \dashv \mathcal{H}'; \Gamma'; \bar{\Omega}'$, $(d', h', s' : \bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}; \bar{P}$, and P is a subfunction of $\bar{\mathrm{P}}$ up to possible changes of the mappings of keys in d.

Proof. This is a strengthening of Preservation (Theorem 2.2) in which we i) show a property of h' and ii) show an assumption about $\bar{\mathrm{P}}$ can be made without invalidating the Theorem. The former is the result of a trivial inspection of the E rules, seeing that all of them are deterministically nondecreasing on h, and when the grow h they grow d in parallel. The latter requires more careful justification. This case proceeds by induction on the structure of e. The inductive case follows trivially for all expressions by lemma 5.21 and the observation that E1a and E1b lift the effects of a step from inside an evaluation context without perturbation. We focus on which rules could have been used to derive the step, excluding E1a and E1b. For most, we can trivially pick $\bar{\mathrm{P}} = \mathrm{P}$. For E3, E5, and E19, $\bar{\mathrm{P}}$ can be chosen to be a strict expansion of P in parallel with the expansion of h to h' in these cases. The only cases in which $\bar{\mathrm{P}}$ must differ from P on existing keys are subcases of E18, when e_j's *bottom-most base* expression is an explore, a swap, or an attach. In all three of these cases, a region is renamed, changing the mappings in P of all locations in that region. In the case of attach and swap, the renamed region must be tracked in $\mathcal{H}$, and in the case of explore the region is reachable from a tracked region in $\mathcal{H}$. In either case, F4 guarantees that all locations in that region are in d_j, so we can conclude that $\bar{\mathrm{P}}$ can always be picked to agree with P on the complement of d, concluding our strengthening of preservation.

□

3.3.3 *Proofs.*

Assuming Progress (Theorem 2.1) and Preservation (Theorem 2.2) for the single-threaded system, we prove Theorems 3.1 and 3.2.

Theorem 3.6 (Concurrent Progress (Restated from 3.1)). *For any well-typed concurrent configuration $(h, d_n, \vec{s_n}, e_n)$ there exists another dynamic configuration $(h', d_n', \vec{s_n'}, e_n')$ such that $(h, d_n, \vec{s_n}, e_n) \xrightarrow{concur\text{-}eval} (h', d_n', \vec{s_n'}, e_n')$, as long as one of the following two conditions holds:*

i) for some j: $e_j \notin LocationNames$ is a non-blocking expression

ii) for some τ, a, b: e_a and e_b admit $\texttt{send-}\tau(l)$ and $\texttt{recv-}\tau()$, respectively as their bottom-most base expressions.

Proof of Theorem 3.1. Given the well-typed concurrent configuration $(h, d_n, \vec{s_n}, e_n)$, we derive a step in each of the cases under which progress holds:

i) For a *non-blocking* expression $e_j \notin LocationNames$, by Theorem 2.1 e_j steps with $(d_j, h, s_j, e_j) \xrightarrow{\text{eval}} (d_j', h, s_j', e_j')$. We conclude by application of step EC1.

ii) For a matching send/recv pair $\langle d_a, s_a, e_a \rangle$ and $\langle d_b, s_b, e_b \rangle$. To apply EC2, it suffices to show that $d_{sep} = \{l \in dom(h) : h \vdash l_{root} \hookrightarrow l\}$ is a subset of d_a, the reservation of the sending expression. To show this we must invert TC1 to obtain the $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e_a : r_a\ \tau_a \dashv \mathcal{H}'; \Gamma'; \Omega'$ and $d_a, h, s_a : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ $\texttt{agree}$. We observe by our case assumption that e_a takes the form $E_a^*[\texttt{send-}\tau(l)]$, so repeated application of lemma 5.21 tells us $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash \texttt{send-}\tau(l) : r_{out}\ \tau_{out} \dashv \bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}$ for some $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}$. We now invert T23 and H1 to determine that l_{root} is in a region $r^{\circ}_{root}\langle\rangle$ with no tracked references. Any region r reachable from r_{root} cannot be tracked, because the predicate derived from inversion of F2 on the judgment $d_a, h, s_a : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ $\texttt{agree}$ follows by inversion of F5 would then force some reference originating at r_{root} to be tracked. Any isolated reference originating in a region reachable from r_{root} thus cannot be **invalid**, as invalidity requires tracking of the source region. Thus all paths of references originating at l_{root} consist only of non-**invalid** references, which allows us to conclude through inversion of F4 that $d_{sep} \subseteq d_a$, and we can conclude this case by application of EC3 and subsequently EC2.

Having shown that a step is possible in all cases given the assumptions of Theorem 3.1, we have shown that the theorem holds. □

Theorem 3.7 (Concurrent Preservation (Restated from 3.2)). *For any well-typed configuration $(h, d_n, \vec{s_n}, e_n)$ that steps with the relation $(h, d_n, \vec{s_n}, e_n) \xrightarrow{\textit{concur-eval}} (h', d_n', \vec{s_n'}, e_n')$, the configuration $(h', d_n', \vec{s_n'}, e_n')$ is also well-typed.*

Proof of Theorem 3.2. We are given well-typed concurrent configuration $\vdash (h, d_n, \vec{s_n}, e_n)$, and wish to show that under any possible step $(h, d_n, \vec{s_n}, e_n) \xrightarrow{\text{concur-eval}} (h', d_n', \vec{s_n'}, e_n')$, the concurrent configuration $\vdash (h', d_n', \vec{s_n'}, e_n')$ is also well-typed. This assumed step could only have been derived by EC1 or EC2, and we show that our claim holds in either case:

EC1: For some j, $(d_j, h, s_j, e_j) \xrightarrow{\text{eval}} (d_j', h', s_j', e_j')$, and for all $i \neq j : (d_i', s_i', e_i') = (d_i, s_i, e_i)$. From the assumption of well-typedness for the original concurrent configuration, and inversion of TC1, we know that $\mathcal{H}_j; \Gamma_j; \Omega_j; \mathrm{P} \vdash e_j : r_j\ \tau_j \dashv \mathcal{H}_j'; \Gamma_j'; \Omega_j'$ and $d_j, h, s_j : \mathcal{H}_j; \Gamma_j; \Omega_j; \mathrm{P}$ $\texttt{agree}$. We can now apply Theorem 2.2 to determine that $\bar{\mathcal{H}}_j; \bar{\Gamma}_j; \bar{\Omega}_j; \bar{\mathrm{P}} \vdash e_j' : r_j\ \tau_j \dashv \mathcal{H}_j'; \Gamma_j'; \Omega_j''$ and $d_j', h', s_j' : \bar{\mathcal{H}}_j; \bar{\Gamma}_j; \bar{\Omega}_j; \bar{\mathrm{P}}$ $\texttt{agree}$. To apply TC1 and conclude the well-typedness of the new concurrent configuration $(h', d_n', \vec{s_n'}, e_n')$, it now suffices to show that for all $i \neq j$, $d_i \cap d_j' = \emptyset$, $\mathcal{H}_i; \Gamma_i; \Omega_i; \bar{\mathrm{P}} \vdash e_i : r_i\ \tau_i \dashv \mathcal{H}_i'; \Gamma_i'; \Omega_i'$ and $d_i, h', s_i : \mathcal{H}_i; \Gamma_i; \Omega_i; \bar{\mathrm{P}}$ $\texttt{agree}$.

We consider the rules that could have derived our step for thread j, and note that after inverting E1A and E1B as many times as necessary, the only ones under which $d_j' \neq d_j$ are E3 and E19. But under both of these, $d_j' - d_j \subseteq dom(h') - dom(h)$, so since $d_i \subseteq dom(h)$ for all i by inversion of F5 and F7 on the context agreement of the original configuration, $d_i \cap d_j = \emptyset \implies d_i \cap d_j' = \emptyset$. This takes care of preservation for the first property we need to show for all $i \neq j$.

We now focus on preservation of our next goal, well-typedness after updating P to $\bar{\mathrm{P}}$. To show that this property holds for thread i, we claim that it suffices to show that $\mathrm{P}(l) = \bar{\mathrm{P}}(l)$ for all locations $l \in d_i$. To justify this, we observe that typing depends on P only through T22, which typechecks static locations. By lemma 3.6, the regions of all such locations typechecked is contained in $\mathcal{H}_i$. By inversion of F4 on the well-typedness of thread i's initial configuration, all locations in tracked regions are in d_i, so we conclude that the well-typedness of e_i depends on P only through locations in d_i. Now we note that lemma 3.7 tells us that we can choose $\bar{\mathrm{P}}$ such that it differs from P only on locations in d_j, which suffices to show well-typedness for all threads $i \neq j$ because we have already shown $d_i \cap d_j = \emptyset$.

Now, to complete the proof of this Theorem, it suffices to show that for all $i \neq j$, $d_i, h', s_i : \mathcal{H}_i; \Gamma_i; \Omega_i; \bar{\mathrm{P}}$ agree, noting that we are given $d_i, h, s_i : \mathcal{H}_i; \Gamma_i; \Omega_i; \mathrm{P}$ agree. We do so by listing all of the invariants required in as premises of F2 to obtain context agreement. Specifically, we list those that could have been invalidated by replacing h, P with $h', \bar{\mathrm{P}}$, and argue that they are not:

F3, F4, F5, F8, F10, F12**:** All of these depend on h and P only on locations in regions reachable from a region tracked in $\mathcal{H}_i$, i.e. on locations in the *live-set*. It thus suffices to argue that for all locations in the original configuration's *live-set*, h agrees with h' and $\bar{\mathrm{P}}$ agrees with P, and that the new configuration's *live-set* does not exceed that of the old. To establish the former, we note that the well-typedness of the original configuration implies all locations reachable from a tracked region are contained in d_i which is disjoint from d_j, and our choice by lemma 3.7 guarantees that d_j is an upper bound for the set of locations on which $\bar{\mathrm{P}}$ and P can disagree. To establish the latter, we note that any new locations, or locations that change their region, will be mapped in the new configuration to regions reachable from a region tracked by thread j (this is from inspection of the evaluation rules). Well-typedness of thread j's new configuration implies all such regions force all of their locations to be in d_j, which is disjoint from d_i, so we can conclude the latter as well, concluding our argument for preservation of these invariants.

F6, F7**:** The judgments from these rules do not depend on any context local to thread i, and thus are already given from context agreement for thread j.

F13**:** Since this is the only invariant that depends on Ω_i, we are free to choose Ω such that this invariant holds, choosing Ω'_i to match.

We have shown that, under thread j's step, all of the other threads $i \neq j$ preserve their well-typedness and invariants, and the reservation d_j does not expand to conflict with any other d_i, so we can conclude that in this case, Concurrent Preservation holds.

EC2**:** We note that this case uses the same h and P in both the original and the new configuration, so for all $i \notin \{a, b\}$ the necessary properties to apply TC1 to the new configuration come directly from inversion of TC1 on original configuration. Here, it suffices to show well-typedness and context agreement for threads a and b. By lemma 5.21, $\mathcal{H}_a; \Gamma_a; \Omega_a; \mathrm{P} \vdash e_a : r_a\ \tau_a \dashv \mathcal{H}'_a; \Gamma'_a; \Omega'_a$ and $\mathcal{H}_b; \Gamma_b; \Omega_b; \mathrm{P} \vdash e_b : r_b\ \tau_b \dashv \mathcal{H}'_b; \Gamma'_b; \Omega'_b$ imply $\mathcal{H}_a; \Gamma_a; \Omega_a; \mathrm{P} \vdash \texttt{send-}\tau(l_{root}) : r_u\ \texttt{unit} \dashv \mathcal{H}''_a; \Gamma''_a; \Omega''_a$ and $\mathcal{H}_b; \Gamma_b; \Omega_b; \mathrm{P} \vdash \texttt{recv-}\tau() : r_{new}\ \tau \dashv \mathcal{H}''_b; \Gamma''_b; \Omega''_b$. We also note that by the structure of typing rules for evaluation contexts, it suffices to show $\bar{\mathcal{H}}_a; \Gamma_a; \Omega_a; \bar{\mathrm{P}} \vdash \texttt{new-unit} : r_u\ \texttt{unit} \dashv \mathcal{H}''_a; \Gamma''_a; \Omega''_a$ and $\mathcal{H}_b, \dot{r}_{new}\langle\rangle; \Gamma_b; \Omega_b; \bar{\mathrm{P}} \vdash l_{root} : r_{new}\ \tau \dashv \mathcal{H}''_b; \Gamma''_b; \Omega'''_b$, where $\mathcal{H}_a = r^{\circ}\langle\rangle, \bar{\mathcal{H}}_a$. With respect to these static contexts, by the well-typedness of our original configuration, we have $(d_a \uplus d_{sep}, h, s_a : \mathcal{H}_a; \Gamma_a; \Omega_a; \mathrm{P}\ \texttt{agree})$ and $(d_b, h, s_b : \mathcal{H}_b; \Gamma_b; \Omega_b; \mathrm{P}\ \texttt{agree})$, from which we seek to prove $(d_a, h, s_a : \bar{\mathcal{H}}_a; \Gamma_a, \Omega_a; \bar{\mathrm{P}}\ \texttt{agree})$ and $(d_b \uplus d_{sep}, h, s_b : \mathcal{H}_b, \dot{r}_{new}\langle\rangle; \Gamma_b; \Omega_b; \bar{\mathrm{P}}\ \texttt{agree})$. We now have exactly 4 subgoals, the proof of each of which will conclude our proof of this Preservation case. Next, we will give these proofs, but first we will specify our choice of $\bar{\mathrm{P}}$, which is still totally generalized. Noting that because, in the original configuration $d_{sep} \subseteq d_a \uplus d_{sep} \subseteq dom(h) = dom(\mathrm{P})$ by inversion of F4 and F7, we let r_{new} (as used above in the typing of thread b after stepping) be a fresh region name along with sufficiently many others (we call the entire set Ω_{new} to bijectively rename the entire range of $\mathrm{P} \upharpoonright_r$ as restricted to d_{sep}, with the region r of l_{root} in the old configuration being renamed to r_{new}. Let P_{sep} be P restricted to d_{sep}, and let $\bar{\mathrm{P}}_{sep}$ be P_{sep} after renaming all of its regions as described above. $\bar{\mathrm{P}}$ will be exactly P, with P_{sep} replaced by $\bar{\mathrm{P}}_{sep}$. We now prove our 4 sufficient subgoals:

$\boxed{\bar{\mathcal{H}}_a; \Gamma_a; \Omega_a; \bar{\mathbf{P}} \vdash \texttt{new-unit} : r_u\ \texttt{unit} \dashv \mathcal{H}''_a; \Gamma''_a; \Omega''_a}$**:** Sufficiently many of these contexts remain fully generalized (namely all of the outputs) to trivially conclude that this typing judgment holds.

$\boxed{\mathcal{H}_b, r^{\cdot}_{new}\langle\rangle; \Gamma_b; \Omega_b; \bar{\mathbf{P}} \vdash l_{\boldsymbol{root}} : r_{new}\ \tau \dashv \mathcal{H}''_b; \Gamma''_b; \Omega'''_b}$**:** Since d_{sep} is defined as exactly the set of locations reachable (including reflexively, see M2c) from l_{root}, $l_{root} \in d_{sep}$. That $\bar{\mathrm{P}} \upharpoonright_r (l_{root}) = r_{new}$ is explicitly laid out above in the definition of $\bar{\mathrm{P}}$, and that $\bar{\mathrm{P}} \upharpoonright_\tau (l_{root}) = \tau$ follows from inversion of T23 on thread a's original expression. We can see by inspection that r_{new} is tracked, so we conclude this subgoal holds by application of T22.

$\boxed{d_a, h, s_a : \bar{\mathcal{H}}_a; \Gamma_a; \Omega_a; \bar{\mathbf{P}}\ \textbf{agree}}$**:** As above, we list all the invariants serving as premises of F2 that could have been invalidated by replacing $d_a \uplus d_{sep}, \mathcal{H}_a, \mathrm{P}$ with $d_a, \bar{\mathcal{H}}_a, \bar{\mathrm{P}}$:

- F3**:** We note again that $\bar{\mathcal{H}}_a$ is identical to $\mathcal{H}_a$ except with $r^{\circ}\langle\rangle$ dropped. All of the regions in $regs(\mathrm{P}_{sep})$ were untracked and reachable, in the original configuration, from some **semi-tracked**reference originating in region r. Those regions cannot have been reachable from another **semi-tracked** reference in the original configuration or they would have necessarily had an in-degree 2 ancestor in G_S and F3 would have been violated on that configuration. By F5 r itself could not have been reachable from a **semi-tracked** ref, so after its removal from tracking none of the regions in $regs(\mathrm{P}_{sep})$ (including r) are semi-tracked-reachable. Thus F3 holds on the new configuration as a weakening of its application to the old configuration, and it is totally unobservant of the changes made to regionality by replacing P_{sep} with $\bar{\mathrm{P}}_{sep}$.
- F4**:** None of the locations in d_{sep}, i.e. those removed from the reservation across the step, can be in a region that was reachable from a tracked region besides r, so the paired removal of r from tracking and d_{sep} from the dynamic reservation does not violate this invariant.
- F5**:** Assume there are two tracked regions r_{src} and r_{dest} in $dom(\bar{\mathcal{H}}_a)$ in the new configuration with an untracked object χ reachable from the first that reaches the second. Since $\bar{\mathcal{H}}_a$ is contained in $\mathcal{H}_a$, both of these regions were tracked in the original configuration too. If χ were tracked in the original configuration, then the fact that χ is tracked before the step but after implies it was reachable from r. But since reachability is transitive (see M2b) and r_{dest} is reachable from χ, r_{dest} must not be tracked after the step either. It is, so χ cannot have been tracked in the original configuration, which violates F5. F5 is known to hold by the well-typedness of the original configuration, so we can conclude that no such triple r_{src}, χ, r_{dest} exists, which is equivalent to concluding that F5 holds on the new configuration.
- F7**:** Our construction of $\bar{\mathrm{P}}$ changed neither domain nor typing, so this invariant is trivially preserved.
- F8**:** In the new configuration, a bounded reference whose target location but not source location is reachable from l_{root} will now cross regions - as its source location will be in the same region but its target will be in a fresh one from Ω_{new}. To see that all such source locations are not in the *live-set*, preserving this invariant, we note that all of the regions reachable from r were uniquely reachable from r, as seen above, so they are not tracked, and no longer reachable from a tracked region in the new configuration. Thus all locations that

were in a region reachable from r, but not themselves reachable from l_{root} will be dropped from the *live-set*, and any bounded refs that cross regions will not break this invariant. We can conclude it is preserved as this is the only way it could have been violated given it held on the old configuration.

- F9, F11, F12: Lack of changes to tracked variables prevent these invariants from being invalidated.
- F10: This is only possible weakened by the loss of tracked regions
- F13: Since Ω_a is still fully generalized, we can conclude it contains exactly what it needs to contain for this invariant to hold.

$\boxed{d_b \uplus d_{\mathbf{sep}}, h, s_b : \mathcal{H}_b, r^{\cdot}\langle\rangle; \Gamma_b; \Omega_b; \bar{\mathbf{P}}\ \mathsf{agree}}$: As above, we list all the invariants serving as premises of F2 that could have been invalidated by replacing $d_b, \mathcal{H}_b, \mathrm{P}$ with $d_b \uplus d_{sep}, (\mathcal{H}_b, r^{\cdot}_{new}\langle\rangle), \bar{\mathrm{P}}$:

- F3: Since P_{sep} is equivalent to $\bar{\mathrm{P}}_{sep}$ up to bijective renaming of regions, we can conclude that the reachable region graph from r_{new} in $h, \bar{\mathrm{P}}_{sep}$ is identical to the reachable region graph from r in h, P_{sep}, and thus the former is a tree just as F3 on thread a's original configuration tells us the latter is. It is also fully disjoint from the existing reachable region graph from the tracked regions in $\mathcal{H}_b$, which is a forest by F3 on thread b's original configuration. Any non-**invalid** references crossing between the two would have violated F4 on the original configuration of one of thread a or b, depending on which direction they pointed, so there are none, and thus the union between the old forest and the new tree really is disjoint, so replacing $\mathcal{H}_b$ with $\mathcal{H}_b, r^{\cdot}_{new}\langle\rangle$ along with replacing P with $\bar{\mathrm{P}}$ will preserve F3.
- F4: Adding a region to tracking, along with adding all reachable locations from it to the reservation is guaranteed to preserve this invariant as long as no locations outside that reachable location graph share regions with it, which was guaranteed by our choice of Ω_{new}, so we can conclude that this invariant is preserved.
- F5: r_{new} is the only region tracked in the new configuration but not the original, and no tracked regions are reachable from r_{new} in the new configuration, so any 3-tuples r_{src}, χ, r_{dest} that could violate this invariant would need $r_{dest} = r_{new}$. If r_{new} were targeted by any reference in $\mathcal{H}_b$ it would have to be **invalid**, or it would have violated F4 in the original configuration because parts of thread a's reservation d_a, disjoint from d_b, would have been in thread b's *live-set*. But all **invalid** references mention their target region, which would have violated F13 on the original configuration because r_{new} is a fresh name. Thus no such 3-tuples can exist.
- F7: Preserved as preserved in case above.
- F9, F11, F12: Lack of changes to tracked variables prevent these invariants from being invalidated.
- F10: No variables exist in r_{new} as it is fresh, and $\bar{\mathrm{P}}$ does not differ from P on any locations in tracked regions, so this invariant is unaffected.
- F13: Since Ω_b is still fully generalized, we can conclude it contains exactly what it needs to contain for this invariant to hold.

We have now shown all 4 subgoals that were required to apply TC1 to the new configuration, so we can conclude that preservation holds in this case.

There were only two possible rules that could have derived the concurrent step, and we showed that in either case the result was a well-typed concurrent configuration. This establishes Concurrent Preservation for the multi-threaded Gallifrey Type System. □

4 SPECIFICATION OF THE DECORATED SYSTEM

We now proceed to specify the decorated system. All rules will be restated, with varying degrees of difference from the undecorated originals which we outline here. The grammar, section 4.3, will illustrate the key expressions now bear annotations of the form $@r$ or $@\Omega$. These record fresh region names chosen arbitrarily by the type system, to ensure that when the program steps, it can do so with awareness of the choice of fresh region names the static system expected. This can be seen to be necessary by examination of the new stepping rules, section 4.7, which shows the greatest extent of difference from the undecorated system. The rules now include a dynamic context ρ which serves the function that P did in the undecorated system of serving as a global source of truth for the regionality of all locations in the heap. The difference is that ρ here, unlike P in the undecorated system, is contained in dynamic configurations and manipulated deterministically. The invariants, section 4.7, are identical to those in the undecorated system except that they use ρ instead of P wherever a source of regionality truth is needed, and that one additional invariant is added, F14$^{(\text{D})}$, stating that a, possibly very small, static P still exists in this decorated system as strict submap of ρ. It will be used exclusively to typecheck locations when they arise in the program text. The typing rules, section 4.5 are nearly identical between the decorated system and the undecorated, but inspection will reveal the exact sites at which syntactic annotations are introduced. Finally, in section 4.8.1, we state Progress and Preservation for this decorated system, which will notably contain much heavier configurations and be more fully deterministic than the undecorated counterparts to these theorems. In the same section, we will state lemmas that suffice to project Progress and Preservation down from the decorated to the undecorated systems, deferring their proofs, and proof of their sufficiency, to section 5. Immediately after the statement of the decorated system concludes, we will proceed to the weightiest proof in this appendix: in section 4.8.1, the proofs of decorated Progress and Preservation.

4.1 Contextual Graph Construction

The contextual graph $\mathbf{G}_S(\mathcal{H}, \rho, h, s)$ is specified exactly as in the decorated system's section 2.1.1, except relying on ρ as a source of truth for regionality instead of P.

4.2 The Type and Value Constructors

We define the relation $\texttt{extracts-fresh-heap}(\Omega; \rho_{old}, r_{root}, \tau_{root}; \rho_{new}, h_{new}, l_{root})$ to be the following list of properties:

1. **Rootedness**: $\rho_{new}(l_{root}) = (r_{root}, \tau_{root})$
2. **Disjointness**: $dom(\rho_{old}) \uplus dom(\rho_{new})$
3. **Minimality**: $\forall l \in dom(h_{new}) : h_{new} \vdash l_{root} \hookrightarrow l$
4. **Region-Freshness**: $regs(\rho_{new}) \cap \Omega \subseteq \{r_{root}\}$
5. **Heap-Sanity**: $(\vdash h_{new}\ \texttt{heap-closed}) \land (\vdash h_{new}, \rho_{new}\ \texttt{agree}) \land (\cdot, \cdot, \cdot, l_{root} \vdash h_{new}, \rho_{new}\ \texttt{live-bnd-ref-sane})$
6. **Simplicity**: $G_{total}(\rho_{new}, h_{new})$ is a forest

We note that there exists a restricted, regions-free form of this relation:

$$\texttt{extracts-fresh-heap-regfree}(h_{old}, \tau_{root}; h_{new}, l_{root})$$

defined as

$$\exists r_{root}, \rho_{old}, \rho_{new} : (\rho_{old}) \restriction_{\tau} \equiv (h_{old}) \restriction_{\tau} \land \texttt{extracts-fresh-heap}(\cdot; \rho_{old}, r_{root}, \tau_{root}; \rho_{new}, h_{new}, l_{root})$$

This regions-free restriction was referred to above, in the undecorated system's section 2.1.2, and used in its stepping rules because regionality information was not dynamically available.

4.2.1 Functions. The set of functions is computed statically in this pass. $\mathcal{F}$ is defined to be the set of unique function names, together with their types of the form $(q_{\text{ARG}}\ \tau \longrightarrow q_{\text{RET}}\ \tau)$. After $\mathcal{F}$ is defined, rule $\boxed{\mathrm{T0}^{(\mathrm{D})}}$ is used to check each function definition, possibly terminating type-checking with failure if $\boxed{\mathrm{T0}^{(\mathrm{D})}}$ is underivable or if a function name is repeated. If all checks are successful, then F_v will be a function mapping pairs $(\mathit{fn}, \tau \in \mathcal{F}$ to values v_f, and F_d will be a function mapping values v_f to the annotated lambda expressions that will serve as the source of truth for function bodies. For each statement $\vdash q_{\text{RET}}\ \tau\ \mathit{fn}(q_{\text{ARG}}\ \tau x)e@\Omega_{out}$ derived by $\boxed{\mathrm{T0}^{(\mathrm{D})}}$, there exists some v_f such that $F_v(\mathit{fn}) = v_f$, and $F_d(v_f) = \text{ł}x.e@\Omega_{out}$.

4.3 Grammar

4.3.1 Reserved Namespaces.

$$\text{(function) } \mathit{fn} \in \mathit{FunctionNames}$$

$$\text{(variable) } x \in \mathit{VariableNames}$$

$$\text{(class) } C \in \mathit{ClassNames}$$

$$\text{(region) } r \in \mathit{RegionNames}$$

$$\text{(location) } l \in \mathit{LocationNames}$$

$$\text{(field) } f \in \mathit{FieldNames}$$

$$\text{(type) } \tau ::= C \mid \texttt{int} \mid \texttt{bool} \mid \texttt{unit} \mid (q_{\text{ARG}}\ \tau \longrightarrow q_{\text{RET}}\ \tau)$$

4.3.2 Static Contexts.

$$\text{(pinnedness metavariable) } \circ ::= \dagger \mid \cdot$$

$$\text{(heap tracking context) } \mathcal{H} ::= r^{\circ}\langle X \rangle,\ \mathcal{H} \mid \cdot$$

$$\text{(region tracking contents) } X ::= x[F],\ X \mid \cdot$$

$$\text{(variable tracking contents) } F ::= f \rightarrowtail r,\ F \mid \cdot$$

$$\text{(variable bindings context) } \Gamma ::= x : r\ \tau,\ \Gamma \mid \cdot$$

$$\text{(region names context) } \Omega ::= r,\ \Omega \mid \cdot$$

$$\text{(location bindings context) } \mathrm{P} ::= l : r\ \tau,\ \mathrm{P} \mid \cdot$$

4.3.3 *Expressions and Programs.*

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

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

$$
\begin{array}{rcl}
\text{(expression) } e & ::= & l \mid x \mid e; e \mid e; \textsc{vir} \mid e.f \mid e.f = e \mid x = e \mid fn@r \mid e(e)@\Omega \mid e \oplus_r e \mid \texttt{new-}\tau@r \\
& & \mid \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\} \mid \texttt{if}(e)\{e\}\ \texttt{else}\ \{e\} \mid \texttt{while}(e)\{e\}@r \\
& & \mid \texttt{send-}\tau(e)@r \mid \texttt{recv-}\tau@r \mid \texttt{detach}\ x@r\ \texttt{in}\ \{e\}\ \texttt{else}\ \{e\}
\end{array}
$$

$$
\begin{array}{rcl}
\text{(evaluation context) } E[] & ::= & []; e \mid []; \textsc{vir} \mid [].f \mid e.f = [] \mid [].f = l \mid x = [] \mid [](e)@\Omega \mid l([])@\Omega \mid [] \oplus_r e \mid l \oplus_r [] \\
& & \mid \texttt{if}([])\{e\}\ \texttt{else}\ \{e\} \mid \texttt{send-}\tau([])@r \mid l; \texttt{drop-reg}\ \{[]\} \\
& & \mid l; \texttt{attach}\ \{[]\}\ \texttt{to}\ \{e\} \mid l; \texttt{attach}\ \{l\}\ \texttt{to}\ \{[]\} \\
& & \mid l; \texttt{swap}\ \{[]\}\ \texttt{with}\ \{e\} \mid l; \texttt{swap}\ \{l\}\ \texttt{with}\ \{[]\}
\end{array}
$$

4.3.4 *Dynamic Contexts.*

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

4.4 Typing Rules

4.5 Typing Rules

4.5.1 *Program Typing.* $\boxed{\vdash p}$

$\boxed{\mathrm{T0}^{(\mathrm{D})}}$ - Program Typing

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

4.5.2 *Function Definition Typing.* $\boxed{\vdash q_{\text{RET}}\ \tau\ \mathit{fn}(q_{\text{ARG}}\ \tau\ x)\{e\}@\Omega}$

$\boxed{\mathrm{T1}^{(\mathrm{D})}}$ - Function-Definition-Typing

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

4.5.3 *Expression Typing.* $\boxed{\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}; \Gamma; \Omega}$

$\boxed{\mathrm{T2}^{(\mathrm{D})}}$ - Variable-Ref-Typing

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

$\boxed{\mathrm{T3}^{(\mathrm{D})}}$ - Sequence-Typing

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

$\boxed{\mathrm{T4}^{(\mathrm{D})}}$ - Bounded-Field-Reference-Typing

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

$\boxed{\mathrm{T5}^{(\mathrm{D})}}$ - Isolated-Field-Reference-Typing

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

$\boxed{\mathrm{T6L}^{(\mathrm{D})}}$ - Bounded-Field-Assignment-Typing–Left-Eval

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

$\boxed{\mathrm{T6R}^{(\mathrm{D})}}$ - Bounded-Field-Assignment-Typing–Right-Eval

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

$\boxed{\mathrm{T7}^{(\mathrm{D})}}$ - Isolated-Field-Assignment-Typing

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

$\boxed{\mathrm{T8}^{(\mathrm{D})}}$ - Assign-Var-Typing

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

$\boxed{\text{T9L}^{(\text{D})}}$ - Function-Application-Typing–Left-Eval

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

$\boxed{\text{T9R}^{(\text{D})}}$ - Function-Application-Typing–Right-Eval

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

$\boxed{\text{T10}^{(\text{D})}}$ - Function-Name-Typing

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

$\boxed{\text{T11}^{(\text{D})}}$ - New-Loc-Typing

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

$\boxed{\text{T12}^{(\text{D})}}$ - Declare-Var-Typing

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

$\boxed{\text{T13L}^{(\text{D})}}$ - OPlus-Typing–Left-Eval

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

$\boxed{\text{T13R}^{(\text{D})}}$ - OPlus-Typing–Right-Eval

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

$\boxed{\text{T14}^{(\text{D})}}$ - If-Statement-Typing

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

$\boxed{\mathrm{T15}^{(\mathrm{D})}}$ - While-Statement-Typing

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

$\boxed{\mathrm{T16}^{(\mathrm{D})}}$ - Focus-Typing

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

$\boxed{\mathrm{T17}^{(\mathrm{D})}}$ - Explore-Typing

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

$\boxed{\mathrm{T18}^{(\mathrm{D})}}$ - Retract-Typing

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

$\boxed{\mathrm{T19}^{(\mathrm{D})}}$ - Unfocus-Typing

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

$\boxed{\mathrm{T20L}^{(\mathrm{D})}}$ - Attach-Typing–Left-Eval

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

$\boxed{\mathrm{T20M}^{(\mathrm{D})}}$ - Attach-Typing–Middle-Eval

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

$\boxed{\mathrm{T20R}^{(\mathrm{D})}}$ - Attach-Typing–Right-Eval

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

T21L$^{(\text{D})}$ - Swap-Typing–Left-Eval

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

T21M$^{(\text{D})}$ - 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$^{(\text{D})}$ - 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$^{(\text{D})}$ - Location-Ref-Typing

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

T23$^{(\text{D})}$ - Send-Typing

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

T24$^{(\text{D})}$ - Receive-Typing

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

T25$^{(\text{D})}$ - Drop-Variable-Typing

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

T26L$^{(\text{D})}$ - Drop-Region-Typing–Left-Eval

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

$\boxed{\text{T26R}^{(\text{D})}}$ - Drop-Region-Typing–Right-Eval

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

$\boxed{\text{T27}^{(\text{D})}}$ - Detach-Typing

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

$\boxed{\text{T28}^{(\text{D})}}$ - Invalidate-Variable-Typing

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

4.5.4 Heap Rules. $\boxed{\vdash q_{\text{ARG}}\ r : \mathcal{H} \Rightarrow \mathcal{H}}$

$\boxed{\text{H1}^{(\text{D})}}$ - Consumes-Heap-Effect

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

$\boxed{\text{H2}^{(\text{D})}}$ - Preserves-Heap-Effect

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

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

$\boxed{\text{H3}^{(\text{D})}}$ - Isolated-Func-Heap-Effect

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

$\boxed{\text{H4}^{(\text{D})}}$ - Consumes-Bounded-Func-Heap-Effect

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

$\boxed{\text{H5}^{(\text{D})}}$ - Preserves-Bounded-Func-Heap-Effect

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

4.6 Stepping Rules

$$\boxed{(d, h, s, \rho, \Omega, e) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, e)}$$

$\boxed{\text{E1A}^{(\text{D})}}$ - Common-Context-Step

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

$\boxed{\text{E1B}^{(\text{D})}}$ - Var-Resolve-Context-Step

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

$\boxed{\text{E2}^{(\text{D})}}$ - Variable-Ref-Step

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

$\boxed{\text{E3}^{(\text{D})}}$ - New-Loc-Step

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

$\boxed{\text{E4}^{(\text{D})}}$ - Sequence-Step

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

$\boxed{\text{E5}^{(\text{D})}}$ - OPlus-Step

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

$\boxed{\text{E6}^{(\text{D})}}$ - If-True-Step

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

$\boxed{\text{E7}^{(\text{D})}}$ - If-False-Step

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

$\boxed{\text{E8}^{(\text{D})}}$ - While-Step

$$\frac{\begin{array}{c}\Omega_{new} \cap \Omega = \emptyset \qquad \phi \in \mathit{bijections}(\mathit{NR}(e_{body}) \uplus \mathit{NR}(e_{bool}), \Omega_{new}) \\ e = \texttt{while}(e_{bool})\{e_{body}\}\texttt{@}r_u \qquad e' = \texttt{if}(e_{bool})\{e_{body}; \phi(e)\} \texttt{ else } \{\texttt{new-unit@}r_u\}\end{array}}{(d, h, s, \rho, \Omega, e) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega \uplus \Omega_{new}, e')}$$

$\boxed{\text{E9}^{(\textsc{d})}}$ - Declare-Var-Step

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

$\boxed{\text{E10}^{(\textsc{d})}}$ - Assign-Var-Step

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

$\boxed{\text{E11}^{(\textsc{d})}}$ - Function-Application-Step

$$\frac{\begin{array}{c}\Omega_{new} \cap \Omega \subseteq \Omega'_{out} \qquad \phi \in \mathit{bijections}(\mathit{NR}(e), \Omega_{new}) \qquad \phi(\Omega_{out}) = \Omega'_{out} \qquad l_f, l \in d \\ h(l_f) = ((q_{\textsc{arg}}\ \tau \rightarrow q_{\textsc{ret}}\ \tau'), v_f) \qquad F_d(v_f) = \lambda x.e@\Omega_{out} \qquad e \equiv_\alpha e' \qquad \mathit{FV}(e') = \{x\} \qquad \mathit{vars}(e') \cap \mathit{dom}(s) = \emptyset\end{array}}{(d, h, s, \rho, \Omega, l_f(l)@\Omega'_{out}) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega \uplus (\Omega_{new} - \Omega'_{out}), \texttt{declare}\ x : \tau\ \texttt{in}\ \{x = l; \phi(e')\})}$$

$\boxed{\text{E12}^{(\textsc{d})}}$ - Bounded-Reference-Step

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

$\boxed{\text{E13}^{(\textsc{d})}}$ - Isolated-Reference-Step

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

$\boxed{\text{E14}^{(\textsc{d})}}$ - Bounded-Assignment-Step

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

$\boxed{\text{E15}^{(\textsc{d})}}$ - Isolated-Assignment-Step

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

$\boxed{\text{E16a}^{(\textsc{d})}}$ - Focus-Step

$$(d, h, s, \rho, \Omega, l; \texttt{focus}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E18b}^{(\textsc{d})}}$ - Unfocus-Step

$$(d, h, s, \rho, \Omega, l; \texttt{unfocus}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E16c}^{(\textsc{d})}}$ - Explore-Step

$$\frac{r_{old} = \rho \upharpoonright_r (h \upharpoonright_v (s(x)).f)}{(d, h, s, \rho, \Omega, l; \texttt{explore}\ x.f@r_{new}) \xrightarrow{\text{eval}} (d, h, s, \rho[r_{old} \mapsto r_{new}], \Omega, l)}$$

$\boxed{\text{E16D}^{(\text{D})}}$ - Retract-Step

$$(d, h, s, \rho, \Omega, l; \texttt{retract}\ x.f) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E16E}^{(\text{D})}}$ - Attach-Step

$$\frac{r_1 = \rho \upharpoonright_r (l_1) \qquad r_2 = \rho \upharpoonright_r (l_2)}{(d, h, s, \rho, \Omega, l; \texttt{attach}\ l_1\ \texttt{to}\ l_2) \xrightarrow{\text{eval}} (d, h, s, \rho[r_1 \mapsto r_2], l)}$$

$\boxed{\text{E16F}^{(\text{D})}}$ - Drop-Variable-Step

$$(d, h, s, \rho, \Omega, l; \texttt{drop-var}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E18G}^{(\text{D})}}$ - Drop-Region-Step

$$(d, h, s, \rho, \Omega, l; \texttt{drop-reg}\ l_d) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E16H}^{(\text{D})}}$ - Swap-Step

$$\frac{r_1 = \rho \upharpoonright_r (l_1) \qquad r_2 = \rho \upharpoonright_r (l_2)}{(d, h, s, \rho, \Omega, l; \texttt{swap}\ l_1\ \texttt{with}\ l_2) \xrightarrow{\text{eval}} (d, h, s, \rho[r_1 \mapsto r_2, r_2 \mapsto r_1], \Omega, l)}$$

$\boxed{\text{E16I}^{(\text{D})}}$ - Invalidate-Variable-Step

$$(d, h, s, \rho, \Omega, l; \texttt{invalidate-var}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l)$$

$\boxed{\text{E17}^{(\text{D})}}$ - Function-Name-Step

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

$\boxed{\text{E18A}^{(\text{D})}}$ - Detach-Step–Success

$$\frac{\begin{array}{c} \textit{heap-separable}(h, s, E^*[], x) \qquad \rho = \bar{\rho} \uplus \rho_{sep} \qquad dom(\rho_{sep}) = \textit{min-reg}(h, s(x)) \cup \{l \in dom(h) \mid h \vdash s(x) \hookrightarrow l\} \\ \phi \in \textit{bijections}(\textit{range}(\rho_{sep} \upharpoonright_r), r_{new} \uplus \Omega_{new}) \qquad \phi(\rho_{sep} \upharpoonright_r (s(x))) = r_{new} \qquad \rho' = \bar{\rho} \uplus \phi(\rho_{sep}) \end{array}}{(d, h, s, \rho, \Omega, E^*[\texttt{detach}\ x@r_{new}\ \texttt{in}\ \{e_{succ}\}\ \texttt{else}\ \{e_{fail}\}]) \xrightarrow{\text{eval}} (d, h, s, \rho', \Omega \uplus \Omega_{new}, E^*[e_{succ}; \texttt{invalidate-var}\ x])}$$

$\boxed{\text{E18B}^{(\text{D})}}$ - Detach-Step–Failure

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

4.7 Invariants

$\boxed{\text{F1}^{(\text{D})}}$ - Expression-Well-Typedness

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

$\boxed{\text{F2}^{(\text{D})}}$ - Dynamic-Static-Agreement

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

$\boxed{\text{F3}^{(\text{D})}}$ - Graph-Simplicity-Enforcement

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

$\boxed{\text{F4}^{(\text{D})}}$ - Reservation-Sufficiency

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

$\boxed{\text{F5}^{(\text{D})}}$ - H-Convex

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

$\boxed{\text{F6}^{(\text{D})}}$ - Heap-Closure

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

$\boxed{\text{F7}^{(\text{D})}}$ - Heap-Rho-Agreement

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

$\boxed{\text{F8}^{(\text{D})}}$ - Bounded-Ref-Sanity

$$\frac{\forall(l, l', f) : [(l \in \textit{live-set}(\mathcal{H}, \rho, h, s) \land (h \restriction_v (l).f = l') \land (\rho \restriction_r (l) \neq \rho \restriction_r (l'))) \implies (\texttt{iso}\ f\ \tau' \in \textit{fields}(h \restriction_\tau (l)))]}{\vdash \mathcal{H}, \rho, h, s\ \texttt{bnd-ref-sane}}$$

$\boxed{\text{F9}^{(\text{D})}}$ - H-Gamma-Agreement

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

$\boxed{\text{F10}^{(\text{D})}}$ - Variable-Binding-Sanity

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

$\boxed{\text{F11}^{(\text{D})}}$ - H-Non-Aliasing

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

$\boxed{\text{F12}^{(\text{D})}}$ - H-Target-Accuracy

$$\frac{\forall (x, f, r, r_f) : [((x.f@(r \rightarrowtail r_f) \in \textit{reg-refs}(\mathcal{H})) \wedge (r_f \in dom(\mathcal{H}))) \implies (\rho \restriction_r (h \restriction_v (s(x)).f) = r_f))]}{\rho, h, s \vdash \mathcal{H}\ \texttt{target-accurate}}$$

$\boxed{\text{F13}^{(\text{D})}}$ - Omega-Bounding

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

$\boxed{\text{F14}^{(\text{D})}}$ - Rho-Subsumption

$$\frac{\forall l \in dom(\mathrm{P}) : l \in dom(\rho) \wedge \rho(l) = \mathrm{P}(l)}{\rho \vdash \mathrm{P}\ \texttt{subsumed}}$$

4.8 Statements of Progress, Preservation, and Projection

4.8.1 Progress and Preservation. We now state Progress and Preservation for this decorated system. Above, we provided typing rules for the multi-thread communication primitives send and recv. Unfortunately, their effects on the heap are only sane in a multi-threaded semantics in which the two can step in parallel. We thus only claim Progress and Preservation for *non-blocking* expressions:

Theorem 4.1 (Progress). *For any well-typed configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ where $e \notin$ LocationNames is a non-blocking expression, there exists another dynamic configuration (d', h', s', ρ'), static Ω'', and expression e' such that $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$.*

Theorem 4.2 (Preservation). *For any well-typed configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}_{out}, \Gamma_{out}, \Omega_{out}$ that steps with the relation $(d, h, s, \rho, \Omega_{out}, e) \xrightarrow{eval} (d', h', s', \rho', \Omega'_{out}, e')$, there exists $\mathcal{H}', \Gamma', \Omega', P'$ such that the configuration $\mathcal{H}', \Gamma', \Omega', P' \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}_{out}, \Gamma_{out}, \Omega'_{out}$ is also well-typed.*

4.8.2 Projection. Structurally, these theorems serve as strengthenings of their undecorated counterparts, Theorems 2.1 and 2.2, that will be inductively provable. Our final goal is to have a system with a regions-free dynamic semantics, so we necessarily prove that Theorems 4.1 and 4.2 indeed are such strengthenings. We claim the following 4 lemmas suffice.

Lemma 4.3 (Configuration Lifting). *For any well-typed undecorated configuration $\vdash (d, h, s, e : r\ \tau)$, there exist $\mathcal{H}, \Gamma, \Omega, P, \mathcal{H}', \Gamma', \Omega', \mathfrak{e}$ such that $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ is a well-typed decorated configuration and $\textsc{strip}(\mathfrak{e}) = e$.*

Lemma 4.4 (Configuration Projection). *For any well-typed decorated configuration $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega',\ \vdash (d, h, s, e : r\ \tau)$ is a well-typed undecorated configuration, where $e = \textsc{strip}(\mathfrak{e})$.*

Lemma 4.5 (Step Lifting). *For any undecorated step $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$, if $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ is a well-typed decorated configuration and $\textsc{strip}(\mathfrak{e}) = e$, then there exist $P', \Omega'', \mathfrak{e}'$ such that $(d, h, s, P, \Omega', \mathfrak{e}) \xrightarrow{eval} (d', h', s', P', \Omega'', \mathfrak{e}')$ is a decorated step and $\textsc{strip}(\mathfrak{e}') = e'$.*

Lemma 4.6 (Step Projection). *For any decorated step $(d, h, s, \rho, \Omega, \mathfrak{e}) \xrightarrow{eval} (d', h', s', \rho', \Omega', \mathfrak{e}')$, $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$ is an undecorated step, where $e = \textsc{strip}(\mathfrak{e})$ and $e' = \textsc{strip}(\mathfrak{e})$.*

We will defer the proof that these lemmas suffice to project Progress and Preservation to section 4.4, and we will defer the proofs that they hold to section . For now, we will proceed to prove decorated Progress and Preservation.

5 PROOFS OF DECORATED PROGRESS AND PRESERVATION

5.1 Structure of this Proof

In section 5.2, we state and prove a series of lemmas that will be essential to the proofs of Progress and Preservation for the Gallifrey type system. We observe that both Progress and Preservation are naturally inductive over the Evaluation context structure of the grammar (see lemmas 5.1 and 5.11 below for a formalization of this notion), and so we begin with the *base* expressions as base cases (see *link* for reminder of definition of this term), proving progress for *base* expressions in 5.3 and preservation for *base* expressions in 5.4. We then give inductive proofs of the full Progress and Preservation Theorems 4.1 and 4.2 in section 5.5, which will conclude this section.

5.2 Lemmas

Lemma 5.1. *All expressions in the language e can be uniquely decomposed as $E^*[e]$ for some sequence E^* of zero or more evaluation contexts and a base expression e.*

Proof. Inspection of rules for expressions and evaluation contexts shows that such a decomposition always exists, and lemma 5.11 below says that it is unique. □

Lemma 5.2 (Base Expression E Rule Restriction). *For each base expression e, detaching expression e, or isolated field reference or assignment e, there exists a step derived from some rule* E2$^{(D)}$-E18B$^{(D)}$ *whose LHS admits e as its expression.*

Proof. Argue that every possible derivation of e in the grammar with locations in any possible evaluation context hole corresponds to the LHS of at least one evaluation rule E2$^{(D)}$-E18B$^{(D)}$ □

Lemma 5.3 (P Typing Expansion). $(\mathcal{H};\Gamma;\Omega;P \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega') \wedge (P \subset \bar{P}) \implies (\mathcal{H};\Gamma;\Omega;\bar{P} \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega')$

Proof. Argue that all typing rules except T22$^{(D)}$ do nothing but pass their outer P into an inner typing expression - inductively requiring us only to consider T22$^{(D)}$, which clearly places no upper bound only a lower bound on P. □

Lemma 5.4 (Ω Horiztonal Typing Growth). $(\mathcal{H};\Gamma;\Omega;P \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega') \implies \Omega \subseteq \Omega'$

Proof. Induction. □

Lemma 5.5 (Ω Disjoint Symmetric Expansion). $(\mathcal{H};\Gamma;\Omega;P \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega') \wedge (\Omega_{new} \cap \Omega = \emptyset) \wedge (\Omega_{new} \cap \Omega' = \emptyset) \implies (\mathcal{H};\Gamma;\Omega \uplus \Omega_{new};P \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega' \uplus \Omega_{new})$

Proof. Can be seen to be inductive in most T$^{(D)}$ cases that could derive e by applying lemma 5.4 to subexpressions to obtain disjointness of Ω_{new} from all needed intermediate contexts. Cases T11$^{(D)}$, T17$^{(D)}$, and T24$^{(D)}$ require us to observe that the new region name chosen appears in the output Ω, and thus does not appear in Ω_{new}, so addition of Ω_{new} to the input Ω does not invalidate the choice of new region name. Cases T9L$^{(D)}$, T9R$^{(D)}$, and T23$^{(D)}$ require us to make the same observation but with respect to new region names chosen through H3$^{(D)}$, noting that symmetric expansion in the case of H5$^{(\mathrm{D})}$ is trivial. Finally, we note that cases T9L$^{(\mathrm{D})}$, T9R$^{(\mathrm{D})}$, T14$^{(\mathrm{D})}$, and T15$^{(\mathrm{D})}$ expose choice of Ω in the syntax of the expression, but only as a function of the differences between input and output Ω for subexpressions, so the expression is unaffected by disjoint symmetric expansion of Ω. □

Lemma 5.6 (Ω Annotation Accumulative). *If* $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$*, then* $\Omega' - \Omega = NR(e)$.

Proof. Induction. □

Lemma 5.7 (Bijective Region Renaming). *If* $\phi : RegionNames \longrightarrow RegionNames$ *is a bijection, then for any* $\mathcal{H}, \Gamma, \Omega, P, d, h, s, \rho, e, r, \tau, \mathcal{H}', \Omega'$*, we have that*

$$\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma', \Omega'$$

if and only if

$$\phi(\mathcal{H}); \phi(\Gamma); \phi(\Omega); \phi(P) \vdash (d, h, s, \phi(\rho), \phi(e) : \phi(r)\ \tau) \dashv \phi(\mathcal{H}'); \phi(\Gamma'), \phi(\Omega')$$

Proof. Induction. □

Lemma 5.8 (Dynamic Region Monotonicity). $(d, h, s, \rho, \Omega, e) \xrightarrow{eval} (d', h', s', \rho', \Omega', e') \implies (\Omega \subseteq \Omega')$

Proof. Induction. □

Lemma 5.9 (Dynamic Region Symmetric Shrinking). $(d, h, s, \rho, \Omega \uplus \Omega_{extra}, e) \xrightarrow{eval} (d', h', s', \rho', \Omega' \uplus \Omega_{extra}) \implies (d, h, s, \rho, \Omega, e) \xrightarrow{eval} (d', h', s', \rho', \Omega')$

Proof. Induction. □

Lemma 5.10 (Dynamic Region Arbitrarity). *If* $(d, h, s, \rho, \Omega, e) \xrightarrow{eval} (d', h', s', \rho', \Omega', e')$*, then for any* $\bar{\Omega}$ *containing* Ω *there exists* $\bar{\Omega}'$ *such that* $(d, h, s, \rho, \bar{\Omega}, e) \xrightarrow{eval} (d', h', s', \rho', \bar{\Omega}', e')$.

Proof. Induction. Only nonvacuous cases are E3$^{(\mathrm{D})}$, E11$^{(\mathrm{D})}$, and E16$^{(\mathrm{D})}$, in each of which we note that the choice of Ω_{new} is arbitrary except for disjointness from Ω and cardinality, so a new $\bar{\Omega}_{new}$ could be chosen to be disjoint from any $\bar{\Omega}$. □

Lemma 5.11 (Evaluation Context Injectivity). *If* $E_1[e_1] = E_2[e_2]$*, where* $E_1[], E_2[]$ *are evaluation contexts and* e_1, e_2 *are expressions not in LocationNames, then* $E_1 = E_2$ *and* $e_1 = e_2$.

Proof. Inspection. □

Lemma 5.12 (Context Agreement Pinnedness Independence). *If* $\mathcal{H}$ *and* $\mathcal{H}'$ *are identical up to pinnedness, then* $\vdash d, h, s, \rho : \mathcal{H}; \Gamma; \Omega; P$ `agree` *iff* $\vdash d, h, s, \rho, : \mathcal{H}'; \mathcal{H}; \Omega; P$ `agree`.

Proof. Inspection of F3$^{(\mathrm{D})}$-F14$^{(\mathrm{D})}$ shows no observance of pinnedness. □

Lemma 5.13 (Tracking Growth Ω-Bounding). *If* $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$*, then* $dom(\mathcal{H}') - dom(\mathcal{H})$ *is upper bounded by* $\Omega' - \Omega$

Proof. Induction, noting that the only cases in which $dom(\mathcal{H}')$ exceeds $dom(\mathcal{H})$ are the result of obtaining a fresh region r from outside Ω, and then adding it as $r^{\cdot}\langle\rangle$ to $\mathcal{H}$. □

Lemma 5.14 (Pinned Typing Symmetry). *If $r^{\circ_l}\langle X\rangle, \mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv r^{\circ_r}\langle X'\rangle, \mathcal{H}'; \Gamma'; \Omega'$, then $\circ_l = \circ_r$.*

Proof. Induction. □

Lemma 5.15 (Pinned Weakening). $r^{\dagger}\langle X\rangle, \mathcal{H}; \Gamma; \Omega; R \vdash (d, h, s, \rho, e : r\ \tau) \dashv r^{\dagger}\langle X'\rangle, \mathcal{H}'; \Gamma'; \Omega' \implies r^{\cdot}\langle X\rangle, \mathcal{H}; \Gamma; \Omega; R \vdash (d, h, s, \rho, e : r\ \tau) \dashv r^{\cdot}\langle X'\rangle, \mathcal{H}'; \Gamma'; \Omega'$

Proof. Invert $\boxed{\text{F1}^{(\text{D})}}$. The proof tree for $\boxed{\text{F2}^{(\text{D})}}$ does not observe pinnedness, so we are free to re-use it to conclude agreement of the $\mathcal{H}$ with unpinned r. We note that all typing rules either do no observe pinnedness or match it using either an explicit unpin $(\cdot)$, as in $\boxed{\text{T20L}^{(\text{D})}}$-$\boxed{\text{T20R}^{(\text{D})}}$ and $\boxed{\text{T21L}^{(\text{D})}}$-$\boxed{\text{T21R}^{(\text{D})}}$, or using the metavariable $\circ$, as in all others. In both cases replacing a † with an · does not invalidate the application of the rule, so we can conclude the unpinned version is well-typed, re-unifying with $\boxed{\text{F1}^{(\text{D})}}$ to obtain the well-typedness of the unpinned configuration. □

Lemma 5.16 (ρ-preservation of Pinned Regions). *If $\mathcal{H}, r^{\dagger}\langle X\rangle; \Gamma; \Omega; P \vdash (d, h, s, \rho, e : r_{outer}\ \tau_{outer}) \dashv \mathcal{H}', r^{\dagger}\langle X'\rangle; \Gamma'; \Omega'$, then for any step $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$ and location l such that $\rho(l) = (r, \tau)$, we have $\rho'(l) = \rho(l)$.*

Proof. Iterative application of lemma 5.17 allows us to reduce e to a *base* expression $\bar{e}$ such that $\mathcal{H}, r^{\dagger}\langle X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, \bar{e} : r_{inner}\ \tau_{inner}) \dashv \bar{\mathcal{H}}, r^{\dagger}\langle \bar{X}\rangle; \bar{\Gamma}; \bar{\Omega}$, noting that $\boxed{\text{F1}^{(\text{D})}}$ is dependent on the output contexts only through typing. Once in this form, we conclude by application of lemma 5.18 □

Lemma 5.17. *If $\mathcal{H}, r^{\dagger}\langle X\rangle; \Gamma; \Omega; P \vdash E[e] : r_{outer}\ \tau_{outer} \dashv \mathcal{H}', r^{\dagger}\langle X'\rangle; \Gamma';'$, then there exist sufficient contexts such that $\mathcal{H}, r^{\dagger}\langle X\rangle; \Gamma; \Omega; P \vdash e : r_{inner}\ \tau_{inner} \dashv \bar{\mathcal{H}}, r^{\dagger}\langle \bar{X}\rangle; \bar{\Gamma}; \bar{P}$.*

Proof. We make a few brief notes about the structure of typing rules for evaluation contexts, from which the lemma is true by inspection for all evaluation contexts and their corresponding $\boxed{\text{T}^{(\text{D})}}$ rules. First, whenever a subexpression in a typing rule syntactically coincides with the hole in an evaluation context, then it is typechecked with the same input contexts as the outer expression. Next, if the region r is not present in the output contexts used to typecheck a subexpression, then in no rules is there a mechanism to introduce that region to the output contexts used to typecheck the outer expression. In most evaluation context typing rules, for example $\boxed{\text{T3}^{(\text{D})}}$ with the hole at e, the relationships between the output contexts of the subexpression and the output contexts of the outer expression is as the LHS and RHS, respectively, of a separate typing judgment on another subexpression, in which case our desired property is equivalent to lemma 5.13. Finally, the pinnedness status of a region is always constant from the LHS to RHS contexts. TODO: Should anything more be written here? □

Lemma 5.18 (Base ρ-preservation of Pinned Regions). *If e is a base expression, and $\mathcal{H}, r^{\dagger}\langle X\rangle; \Gamma; \Omega; P \vdash (d, h, s, \rho, e : r_{outer}\ \tau_{outer}) \dashv \mathcal{H}', r^{\dagger}\langle X'\rangle; \Gamma'; \Omega'$, then for any step $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$ and location l such that $\rho(l) = (r, \tau)$, we have $\rho'(l) = \rho(l)$.*

Proof. Since e is a *base* expression, some rule $\boxed{\text{E2}^{(\text{D})}}$-$\boxed{\text{E17}^{(\text{D})}}$ must have derived our assumed step. $\boxed{\text{E16C}^{(\text{D})}}$, $\boxed{\text{E16E}^{(\text{D})}}$, and $\boxed{\text{E16H}^{(\text{D})}}$ are the only such steps that modify existing the mappings of existing locations in ρ to generate ρ' (others strictly append), so it suffices to assume that one of them derived our step:

$\boxed{\text{E16C}^{(\text{D})}}$**:** Here just as in the previous case, ρ and ρ' differ exactly at the locations in a region r_{old} that we will show cannot be tracked, and thus cannot be r. Consider the inversion of rule $\boxed{\text{T17}^{(\text{D})}}$ (which is the only that could typecheck the LHS of an $\boxed{\text{E16C}^{(\text{D})}}$ step). $x.f$ is a **semi-tracked** reference here, which means that its target region in the pre-step configuration cannot be tracked or F5$^{(\text{D})}$ would be violated. Since r_{old} is exactly $x.f$'s target region in the pre-step configuration, it cannot be r, and thus if E16C$^{(\text{D})}$ derived our step, $\rho'(l) = \rho(l)$.

E16E$^{(\text{D})}$, E16H$^{(\text{D})}$: These cases are extremely explicit, their ρ and ρ' differ only at locations in unpinned regions, and r is a pinned region so if either of these rules derived our step, $\rho'(l) = \rho(l)$. Note: this is exactly the needed logic for which pinnedness was introduced into the system!

□

Lemma 5.19 (Total Context Symmetric Expansion). *Given a base typing judgment $\mathcal{H}; \Gamma; \Omega; P \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$, and expansions $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}$ such that $vars(e) \cap (vars(\bar{\mathcal{H}}) \cup dom(\bar{\Gamma})) = \emptyset$, $srcs(\bar{\mathcal{H}}) \cup targets(\bar{\mathcal{H}}) \cup range(\bar{\Gamma} \restriction_r) \subseteq \bar{\Omega} \cup \{\bar{r}\}$, $\bar{\Omega} \cup \Omega' = \emptyset$, and $\bar{r}$ is pinned in $\mathcal{H}$, we can symmetrically expand to conclude:*

$$\mathcal{H} \uplus \bar{\mathcal{H}}; \Gamma \uplus \bar{\Gamma}; \Omega \uplus \bar{\Omega}; P \vdash e : r\ \tau \dashv \mathcal{H}' \uplus \bar{\mathcal{H}}; \Gamma' \uplus \bar{\Gamma}; \Omega' \uplus \bar{\Omega}$$

Proof. In most T$^{(\text{D})}$ cases that could derive e, it can be trivially seen that this lemma is either inductive over subexpressions or follows from the monotonicity of the typing conditions with respect to the contexts. Some cases have have conditions that could potentially be invalidated by context growth: T8$^{(\text{D})}$ requires $x \notin vars(\mathcal{H}')$, T11$^{(\text{D})}$ and T24$^{(\text{D})}$ requires $r \notin \Omega$, and T12$^{(\text{D})}$ requires x to be outside a large union, but such conditions are all subsumed by the separation assumptions on variables and regions in this lemma. The only other cases of note are T20L$^{(\text{D})}$-T20R$^{(\text{D})}$ and T21L$^{(\text{D})}$-T21R$^{(\text{D})}$, which perform global mappings on region names between their LHS and RHS contexts. Non-interference here is a result of the restriction that the support of the the induced difference map of such mappings is exclusively *unpinned* regions in Ω', whereas the only region in common between our base typing judgment and our expansion contexts is guaranteed to be pinned. TODO: Make sure this is convincing enough - I have personally done a non-generic case-by-case argument to become convinced. □

Lemma 5.20 (Inner Typing Ω Non-Interference). *If $\mathcal{H}; \Gamma; \Omega; P \vdash E[\bar{e}] : r\ \tau \dashv \mathcal{H}_{out}; \Gamma_{out}; \Omega_{out}$ was derived by a typing rule whose application included $\mathcal{H}; \Gamma; \Omega; P \vdash \bar{e} : \bar{r}\ \bar{\tau} \dashv \bar{\mathcal{H}}_{out}; \bar{\Gamma}_{out}; \bar{\Omega}_{out}$ as a premise, then there exists a set Ω_{sep} such that the same typing rule could be applied with Ω and $\bar{\Omega}_{out}$ symmetrically expanded by any set Ω_{new} that's disjoint from Ω_{sep}, and the result will exhibit no changes but the same symmetric expansion of Ω, Ω_{out}. In particular, $\Omega_{sep} = \Omega_{out} - \bar{\Omega}_{out}$ is such a set.*

Proof. This is a property that we claim holds for all T$^{(\text{D})}$ rules that could have derived the well-typedness of $E[\bar{e}]$. We can only conclude that, for a given rule, a candidate Ω_{sep} actually has the property specified by the lemma by noting that the rule relates the output $\bar{\Omega}_{out}$ used to typecheck the inner expression to the output Ω_{out} used to typecheck the outer expansion by disjoint expansion by $\Omega_{sep} = \Omega_{out} - \bar{\Omega}_{out}$ and no other conditions. For each evaluation context, we identify the unique relevant rule, and we perform this inspection, noting the value of Ω_{sep} observed:

TODO: Is this case-by-case analysis overkill?

$\boxed{[\,]; e}$: Invert T3$^{(\text{D})}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{[\,]; \textsc{vir}}$: Possible to invert T16$^{(\text{D})}$, T18$^{(\text{D})}$, T19$^{(\text{D})}$, or T25$^{(\text{D})}$ and obtain $\Omega_{sep} = \emptyset$. If instead must invert T17$^{(\text{D})}$, $\Omega_{sep} = \{r_{new}\}$. If T20L$^{(\text{D})}$ or T21L$^{(\text{D})}$, then $\Omega_{sep} = NR(e_1) \cup NR(e_2)$. Last option is T26L$^{(\text{D})}$, in which case $\Omega_{sep} = NR(e_d)$.

$\boxed{[\,].f}$: Invert T4$^{(\text{D})}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{e.f = []}$: Invert $\boxed{\text{T6L}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e)$, or invert $\boxed{\text{T7}^{(\text{D})}}$ and $\Omega_{sep} = \emptyset$.

$\boxed{[].f = l}$: Invert $\boxed{\text{T6R}^{(\text{D})}}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{x = []}$: Invert $\boxed{\text{T8}^{(\text{D})}}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{[](e)@\Omega}$: Invert $\boxed{\text{T9L}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e) \cup \Omega_{out}$ (new temporary binding of Ω_{out} from proof tree).

$\boxed{l([])@\Omega}$: Invert $\boxed{\text{T9R}^{(\text{D})}}$, and observe $\Omega_{sep} = \Omega_{out}$ (new temporary binding of Ω_{out} from proof tree).

$\boxed{[] \oplus_r e}$: Invert $\boxed{\text{T13L}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e_2) \cup \{r\}$.

$\boxed{l \oplus_r []}$: Invert $\boxed{\text{T13R}^{(\text{D})}}$, and observe $\Omega_{sep} = \{r\}$.

$\boxed{\texttt{if}([])\{e\}\ \texttt{else}\ \{e\}}$: Invert $\boxed{\text{T14}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e_t) \cup NR(e_f)$.

$\boxed{\texttt{send-}\ \tau(e)@r}$: Invert $\boxed{\text{T23}^{(\text{D})}}$, and observe $\Omega_{sep} = \{r'\}$.

$\boxed{l; \texttt{drop-reg}\ \{[]\}}$: Invert $\boxed{\text{T26R}^{(\text{D})}}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{l; \texttt{attach}\ \{[]\}\ \texttt{to}\ \{e\}}$: Invert $\boxed{\text{T20M}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e_2)$.

$\boxed{l; \texttt{attach}\ \{l\}\ \texttt{to}\ \{[]\}}$: Invert $\boxed{\text{T20R}^{(\text{D})}}$, and observe $\Omega_{sep} = \emptyset$.

$\boxed{l; \texttt{swap}\ \{[]\}\ \texttt{with}\ \{e\}}$: Invert $\boxed{\text{T21M}^{(\text{D})}}$, and observe $\Omega_{sep} = NR(e_2)$.

$\boxed{l; \texttt{swap}\ \{l\}\ \texttt{with}\ \{[]\}}$: Invert $\boxed{\text{T21R}^{(\text{D})}}$, and observe $\Omega_{sep} = \emptyset$.

□

Lemma 5.21 (Inner Typing Dependence). *For any expression e of the form $E[\bar{e}]$ for some expression $\bar{e} \notin LocationNames$, if $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ is a well-typed configuration then there exist $\bar{r}, \bar{\tau}, \bar{\mathcal{H}}', \bar{\Gamma}', \bar{\Omega}'$ such that $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, \bar{e} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}'; \bar{\Gamma}'; \bar{\Omega}'$ is also a well-typed configuration and $\bar{\Omega}' \subseteq \Omega'$.*

Proof. It suffices to show that $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (\bar{e} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}'; \bar{\Gamma}'; \bar{\Omega}'$. For each evaluation context, exactly one typing rule can be used to invert it. In most cases, one of the premises post-inversion will be exactly the typing for $\bar{e}$ we need to conclude the lemma holds. In the cases of a secondary or tertiary subexpression, such as $\boxed{\text{T6R}^{(\text{D})}}$, $\boxed{\text{T20M}^{(\text{D})}}$, or $\boxed{\text{T20R}^{(\text{D})}}$ we may be given a typing for $\bar{e}$ with some regions pinned that were not pinned in $\mathcal{H}$. Here, we can still conclude that our exact desired typing judgment holds by application of lemma 5.15. Trivial inspection in each case also tells us that the desired containment $\bar{\Omega}' \subseteq \Omega'$ holds. □

5.3 Base Progress

Our goal in this section is to show:

Lemma 5.22 (Base Progress). *Let $e \notin LocationNames$ be a non-blocking base expression, a detaching expression, an isolated field reference of the form $x.f$, or an isolated field assignment of the form $x.f = l$. For any well-typed configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ where $e \notin LocationNames$, there exists another dynamic configuration (d', h', s', ρ'), static Ω'', and expression e' such that $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$.*

Proof. For a well-typed *base* expression $e \notin LocationNames$, we consider each $\boxed{\text{T}^{(\text{D})}}$ rule able to derive the well-typedness of e and show that there exists an expression e' with dynamic configuration $(d', h', s', \rho', \Omega'')$ such that

$(d, h, s, \rho, \Omega') \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'')$ is derivable by some $\boxed{\text{E}^{(\text{D})}}$ rule. Notably, we need consider neither $\boxed{\text{T22}^{(\text{D})}}$ nor $\boxed{\text{T24}^{(\text{D})}}$.

Here, our proof goals are all of the form "given some $\boxed{\text{TX}^{(\text{D})}}$ via which e can be proven well-typed, there is some $\boxed{\text{EX}^{(\text{D})}}$ which allows e (and its configuration) to step." In stating these proof goals we perform some clear inlinings and simplifications from the typing rule used to derive the assumptions. Once we have proven these goals for each rule $\boxed{\text{T2}^{(\text{D})}}$ - $\boxed{\text{T26R}^{(\text{D})}}$, we have proven progress for *base* expressions.

Case $\boxed{\text{T2}^{(\text{D})}}$ $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, x : r\ \tau) \dashv \mathcal{H}; \Gamma; \Omega \implies (d, h, s, \rho, \Omega, x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, s(x) : r\ \tau)$

Proof. Inverting $\boxed{\text{T2}^{(\text{D})}}$ on the assumed well-typedness of $x : r\ \tau$ tells us $r \in dom(\mathcal{H})$ and $(x : r\ \tau) \in \Gamma$. Inverting $\boxed{\text{F10}^{(\text{D})}}$ on the assumed agreement of the contexts $\mathcal{H}, \rho, s, \Gamma$ allows us to conclude $x \in dom(s)$ and $\rho(s(x)) = (r, \tau)$. Noting that r is tracked and reflexively reachable from itself ($\boxed{\text{M1D}^{(\text{D})}}$), inverting $\boxed{\text{F4}^{(\text{D})}}$ on the assumed agreement of $\mathcal{H}, \rho, h, s$ allows us to conclude $s(x) \in d$. This is now sufficient information to apply $\boxed{\text{E2}^{(\text{D})}}$ and conclude that the desired step is derivable. □

Case $\boxed{\text{T3}^{(\text{D})}}$ $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash l; e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega' \implies (d, h, s, \rho, \Omega', l; e) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega', e)$

Proof. Note that since $e_1; e_2 = E[e_1]$ for an evaluation context E, we can conclude $e_1 = l$ for some $l \in$ *LocationNames*. Once we have observed that our expression has this form, we can conclude the desired step by trivial application of $\boxed{\text{E4}^{(\text{D})}}$. □

Case $\boxed{\text{T4}^{(\text{D})}}$ $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash l.f : r\ \tau_f \dashv \mathcal{H}; \Gamma; \Omega$ implies that for some l_f: $(d, h, s, \rho, \Omega, l.f) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega', l_f)$.

Proof. Since $e.f = E[e]$ for an evaluation context E, we can conclude $e = l$ for some $l \in$ *LocationNames*. Now, letting $l_f = h \upharpoonright_v (l).f$, we seek to derive our desired step through application of $\boxed{\text{E12}^{(\text{D})}}$, requiring us to first prove $l, l_f \in d$ and $\mathsf{bnd}\ f\ \tau_f \in \mathit{fields}(h \upharpoonright_\tau (l))$, which will become our two new proof goals. Through inversion of $\boxed{\text{T4}^{(\text{D})}}$ and subsequently $\boxed{\text{T22}^{(\text{D})}}$ and $\boxed{\text{F14}^{(\text{D})}}$ we can conclude $\rho(l) = (r, \tau)$, $r \in dom(\mathcal{H})$, and $\mathsf{bnd}\ f\ \tau_f \in \mathit{fields}(\tau)$ for some τ. From $\rho \upharpoonright_\tau (l) = \tau$, inversion of $\boxed{\text{F7}^{(\text{D})}}$ tells us $h \upharpoonright_\tau (l) = \tau$, our second proof goal. Inversion of $\boxed{\text{F6}^{(\text{D})}}$ tells us $l_f \in dom(h)$, and inversion of $\boxed{\text{F7}^{(\text{D})}}$ tells us now that $\rho \upharpoonright_r (l_f) = \rho \upharpoonright_r (l) = r$. Recalling $r \in dom(\mathcal{H})$, this is sufficient information now (observing again $\boxed{\text{M1D}^{(\text{D})}}$) to conclude via inversion of $\boxed{\text{F4}^{(\text{D})}}$ that $l, l_f \in d$, our first proof goal. □

Case $\boxed{\text{T5}^{(\text{D})}}$ $\mathcal{H}, r^\circ\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, x.f : r_f\ \tau_f) \dashv \mathcal{H}, r^\circ\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega$ implies that for some l_f: $(d, h, s, \rho, \Omega, x.f) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l_f)$.

Proof. Inversion of $\boxed{\text{T5}^{(\text{D})}}$ and subsequently $\boxed{\text{T2}^{(\text{D})}}$ tells us $(x : r\ \tau) \in \Gamma$ and $\mathtt{iso}\ f\ \tau_f \in \mathit{fields}(\tau)$. $\boxed{\text{F10}^{(\text{D})}}$ now allows us to conclude that, for some $l = s(x)$, $\rho(l) = (r, \tau)$. $\boxed{\text{F7}^{(\text{D})}}$ tells us $l \in dom(h)$ with $h \upharpoonright_\tau (l) = \tau$, so let $l_f = h \upharpoonright_v (l).f$. We seek to derive our desired step through application of $\boxed{\text{E13}^{(\text{D})}}$, requiring us to introduce proof goals $l, l_f \in d$ and $\mathtt{iso}\ f\ \tau_f \in \mathit{fields}(h \upharpoonright_\tau (l)$. Noting that we have already shown the latter, we note that, under inversion of $\boxed{\text{F4}^{(\text{D})}}$, proving the former reduces to proving $\rho \upharpoonright_r (l_f) = r_f$ for some r_f reachable from r. The existence of such an r_f follows from $\boxed{\text{F6}^{(\text{D})}}$ and $\boxed{\text{F7}^{(\text{D})}}$, and its reachability from r is a trivial application of $\boxed{\text{M1A}^{(\text{D})}}$, $\boxed{\text{M1B}^{(\text{D})}}$, and $\boxed{\text{M1C}^{(\text{D})}}$ with intermediary $l.f$ and already known facts about tracking and *fields*. □

Case $\boxed{\text{T6L}^{(\text{D})}}$ We note that in $\boxed{\text{T6L}^{(\text{D})}}$, if $e_f \notin \mathit{LocationNames}$ then we do not have a *base* expression, and if $e_f \in \mathit{LocationNames}$ then we defer to case 5.3, concluding this case by exhaustion. Note that from here onward, we omit the reasoning for substitution of a location into a subexpression when necessary to ensure that the top level expression is base, including omitting all but the rightmost cases of typing rules.

Case $\boxed{\text{T6R}^{(\text{D})}}$ $\mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (l \mapsto (\tau, v)), s, \rho, l.f = l_f : r\ \tau_f) \dashv \mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega \implies (d, h \uplus (l \mapsto (\tau, v)), s, \rho, \Omega, l.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, \rho, \Omega, l_f)$. Additionally, the assumption that our heap takes the form $h \uplus (l \mapsto (\tau, v))$ is WLOG.

Proof. Inversion of $\boxed{\text{T6R}^{(\text{D})}}$ and subsequently $\boxed{\text{T22}^{(\text{D})}}$ tells us that, for some τ', we have that $(l_f : r\ \tau_f), (l : r\ \tau') \in \mathrm{P}$, and $\texttt{bnd}\ f\ \tau_f \in \mathit{fields}(\tau')$. First we must establish that this τ' is the same as that supplied in the proof goal, which can be done through inversion of $\boxed{\text{F14}^{(\text{D})}}$ to see $\rho(l) = (r, \tau')$ and then inversion of $\boxed{\text{F7}^{(\text{D})}}$ to see $l \in \mathit{dom}(\mathcal{H})$ (proving the WLOG component of our goal) and $\tau' = \tau$. To apply $\boxed{\text{E14}^{(\text{D})}}$ and conclude the desired step, it now suffices to show $l, l_f \in d$. But $\boxed{\text{F14}^{(\text{D})}}$ and $\boxed{\text{F5}^{(\text{D})}}$ as we can conclude that $\rho \upharpoonright_r (l) = \rho \upharpoonright_r (l_f) = r$, where r is clearly tracked, allowing reflexive reachability to complete the premises of $\boxed{\text{F5}^{(\text{D})}}$. □

Case $\boxed{\text{T7}^{(\text{D})}}$ $\mathcal{H}, r^\circ\langle x[f \rightarrowtail r_{old}, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (s(x) \mapsto (\tau, v)), s, \rho, x.f = l_f : r_f\ \tau_f) \dashv \mathcal{H}, r^\circ\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega \implies (d, h \uplus (s(x) \mapsto (\tau, v)), s, \rho, x.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (s(x) \mapsto (\tau, v[f \mapsto l_f])), s, \rho, l_f)$. Additionally, the assumption that our heap takes the form $h \uplus (s(x) \mapsto (\tau, v))$, is WLOG.

Proof. Inversion of $\boxed{\text{T7}^{(\text{D})}}$ and $\boxed{\text{T22}^{(\text{D})}}$ tells us that, for some τ', we have that $(l_f : r_f \tau_f) \in \mathrm{P}$, $(x : r\tau') \in \Gamma$, and $\texttt{iso}\ f\ \tau_f \in \mathit{fields}(\tau')$. From $(x : r\tau') \in \Gamma$, $\boxed{\text{F10}^{(\text{D})}}$ tells us that for some l, $s(x) = l$ and $\rho(l) = (r, \tau)$. $\boxed{\text{F7}^{(\text{D})}}$ then allows us to conclude that l is in the domain of the dynamic heap, so our WLOG goal is justified, and that $\tau' = \tau$.It now suffices to show $l, l_f \in d$. $\boxed{\text{F14}^{(\text{D})}}$ allows us to conclude $\rho(l_f) = (r_f, \tau_f)$, and as both r and r_f are tracked this is now sufficient information to conclude through $\boxed{\text{F4}^{(\text{D})}}$ that $l, l_f \in d$. □

Case $\boxed{\text{T8}^{(\text{D})}}$ $\mathcal{H}; \Gamma, x : r_{old}\ \tau; \Omega; \mathrm{P} \vdash (d, h, s \uplus (x \mapsto l_{old}), \rho, x = l : r\ \tau) \vdash \mathcal{H}; \Gamma, x : r, \tau; \Omega \implies (d, h, s \uplus (x \mapsto l_{old}), \rho, \Omega, x = l) \xrightarrow{\text{eval}} (d, h, s \uplus (x \mapsto l), \rho, \Omega, l)$. Additionally, the assumption that our stack takes the form $s \uplus (x \mapsto l_{old})$ is WLOG.

Proof. We briefly note that the formal assumption made is WLOG because $\boxed{\text{F10}^{(\text{D})}}$ guarantees $x \in \mathit{dom}(s)$. Inversion of $\boxed{\text{T8}^{(\text{D})}}$ and $\boxed{\text{T22}^{(\text{D})}}$ tells us that $(l : r\ \tau) \in \mathrm{P}$ and $r \in \mathit{dom}(\mathcal{H})$, which is also sufficient information to establish by $\boxed{\text{F14}^{(\text{D})}}$ that $\rho(l) = (r, \tau)$, and then by $\boxed{\text{F4}^{(\text{D})}}$ that $l \in d$. Our desired step now follows directly through application of $\boxed{\text{E10}^{(\text{D})}}$. □

Case $\boxed{\text{T9R}^{(\text{D})}}$ $\mathcal{H}, r_f^\circ\langle X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l_f(l)@\Omega'_{out} : r'\ \tau') \dashv \mathcal{H}'; \Gamma'; \Omega \uplus \Omega'_{out}$ implies that for some Ω_{new}, ϕ, e': $(d, h, s, \rho, \Omega \uplus \Omega'_{out}, l_f(l)@\Omega'_{out}) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega \uplus (\Omega_{new} - \Omega'_{out}), \texttt{declare}\ x : \tau\ \texttt{in}\ \{x = l; \phi(e')\})$.

Proof. Inversion of $\boxed{\text{T9R}^{(\text{D})}}$ and $\boxed{\text{T22}^{(\text{D})}}$ tells us $(l : r\ \tau), (l_f : r_f(q_{\text{ARG}}\ \tau \longrightarrow q_{\text{RET}}\ \tau')) \in \mathrm{P}$ and $r \in \mathit{dom}(\mathcal{H}) \uplus \{r_f\}$. By $\boxed{\text{F14}^{(\text{D})}}$ and $\boxed{\text{F4}^{(\text{D})}}$, this is already sufficient information to conclude $l, l_f \in d$, as seen in many cases above. $\boxed{\text{F7}^{(\text{D})}}$ now tells us $h(l_f) = ((q_{\text{ARG}}\ \tau \longrightarrow q_{\text{RET}}\ \tau'), v_f$ for some functional value v_f, which must be mapped by F_d to some $\lambda x.e@\Omega_{out}$. We now choose $\Omega_{new} \subset \mathit{RegionNames} - (\Omega - \Omega'_{out})$ to have equal cardinality to $\mathit{NR}(e)$. We choose $\phi \in \mathit{bijections}(\mathit{NR}(e), \Omega_{new})$ such that it maps Ω_{out}, if nonempty, to Ω'_{out}. Examining rules

H3$^{(\mathrm{D})}$-H5$^{(\mathrm{D})}$, we see that both Ω_{out} and Ω'_{out} are empty iff $(q_{\text{ARG}}, q_{\text{RET}}) = (\texttt{preserves}, \texttt{bnd})$, through rules T1$^{(\mathrm{D})}$ and T9R$^{(\mathrm{D})}$ respectively. In the case that both are empty, such a ϕ trivially exists. If they are nonempty, such a ϕ exists because $\Omega_{out} \in NR(e)$ by applying lemma 5.6 to the typing judgment result of inverting T1$^{(\mathrm{D})}$ (see section 4.2.1 for more detail on how we know we have a judgment on which to invert T1$^{(\mathrm{D})}$ to begin with). We have now concluded all the premises needed to apply E11$^{(\mathrm{D})}$ except those regarding α-renaming of e to e', which is standard and formalized no further here. □

Case T10$^{(\mathrm{D})}$ $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, fn@r : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega'$ implies that for some l, v_f, τ: $(d, h, s, \rho, \Omega, fn@r) \xrightarrow{\text{eval}} (d \uplus \{l\}, h \uplus (l \mapsto (\tau, v_f)), s, \rho \uplus (r, \tau)), \Omega, l)$.

Proof. Choose any $l \notin dom(h)$, choose the unique τ such that $(fn, \tau) \in \mathcal{F}$, and let $v_f = F_v(fn)$ (see section 4.2.1 for any questions about these constructs). E17$^{(\mathrm{D})}$ may now be trivially applied to conclude the desired step. □

Case T11$^{(\mathrm{D})}$ $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \Omega \uplus \{r\}, \texttt{new-}\tau@r : r\ \tau) \dashv \mathcal{H}, r^{\cdot}\langle\rangle;\Gamma;\Omega \uplus \{r\}$ implies that for some h_{new}, ρ_{new}, l: $(d, h, s, \rho, \Omega \uplus \{r\}, \texttt{new-}\tau@r) \xrightarrow{\text{eval}} (d \uplus dom(h_{new}), h \uplus h_{new}, s, \rho \uplus \rho_{new}, l)$.

Proof. Inversion of T11$^{(\mathrm{D})}$ tells us only that $r \notin \Omega$. We must choose h_{new}, ρ_{new}, l such that all 6 conditions of $\texttt{extracts-fresh-heap}(\rho, r, \tau; \rho_{new}, h_{new}, l)$ are satisfied. The details here are exactly the details of implementing a constructor for arbitrary types, and thus are largely deferred to an implementation of the language. We will summarize by saying that fresh locations are chosen from $LocationNames - dom(H)$ and fresh region names are chosen from $RegionNames - (\Omega \uplus \{r\})$, with references instantiated to form a tree rooted at l, and each new location in the tree being given a new region name iff its parent reference was isolated. □

Case T12$^{(\mathrm{D})}$ $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \Omega, \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\} : r\ \tau') \dashv \mathcal{H}';\Gamma';\Omega' \implies (d, h, s, \rho, \Omega', \texttt{declare}\ x : \tau\ \texttt{in}\ \{e\}) \xrightarrow{\text{eval}} (d, h, s\uplus, \rho, \Omega', e; \texttt{drop-var}\ x)$.

Proof. Trivial application of E9$^{(\mathrm{D})}$ suffices to conclude the desired step. □

Case T13R$^{(\mathrm{D})}$ $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \Omega, l_1 \oplus_r l_2 : r\ \tau') \dashv \mathcal{H}';\Gamma';\Omega'$ implies that for some l_3, v_3, τ: $(d, h, s, \rho, \Omega', l_1 \oplus_r l_2) \xrightarrow{\text{eval}} (d \uplus \{l_3\}, h \uplus (l_3 \mapsto (\tau', v_3)), s, \rho \uplus (l_3 \mapsto (r, \tau')), \Omega', l_3)$.

Proof. Inversion of T13R$^{(\mathrm{D})}$ and T22$^{(\mathrm{D})}$ tells us that, for some τ, r_1, r_2: $\vdash \tau \oplus \tau : \tau'$, $(l_1 : r_1\ \tau), (l_2 : r_2\ \tau) \in \mathrm{P}$ and $r_1, r_2 \in dom(\mathcal{H})$. With this and inversions of F14$^{(\mathrm{D})}$ and F4$^{(\mathrm{D})}$, we can conclude $l_1, l_2 \in d$. We also observe that by F7$^{(\mathrm{D})}$, there exist v_1, v_2 satisfying $(l_1 : \tau\ v_1), (l_2 : \tau\ v_2) \in h$. This immediately gives us $\vdash h \restriction_\tau (l_1) \oplus h \restriction_\tau (l_2) : \tau'$, and we can choose $v_3 = [[\oplus]](h \restriction_v (l_1), h \restriction_v (l_2))$ to apply E5$^{(\mathrm{D})}$ and conclude our desired step. □

Case T14$^{(\mathrm{D})}$ **and** T15$^{(\mathrm{D})}$ Trivial because no evaluation is performed by these cases.

Case T16$^{(\mathrm{D})}$, T17$^{(\mathrm{D})}$, T18$^{(\mathrm{D})}$, T19$^{(\mathrm{D})}$, T20R$^{(\mathrm{D})}$, T21R$^{(\mathrm{D})}$, T25$^{(\mathrm{D})}$, T26R$^{(\mathrm{D})}$, **and** T28$^{(\mathrm{D})}$ Trivial because the virtual commands have no dynamic requirements to step.

Case T27$^{(\mathrm{D})}$ We need not observe any properties of the typing rule T27$^{(\mathrm{D})}$ to conclude this case. Any *detaching* expression steps with any dynamic contexts by either E18A$^{(\mathrm{D})}$ or E18B$^{(\mathrm{D})}$, depending on whether $heap\text{-}separable(h, s, E^*[], x)$ is true. The remaining premises are satisfied by choice of fresh Ω_{new}.

□

5.4 Base Preservation

Our goal in this section is to show:

Lemma 5.23 (Base Preservation). *Let e be a base expression, a detaching expression, an isolated field reference of the form $x.f$, or an isolated field assignment of the form $x.f = l$. For any well-typed configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ that steps with the relation $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$, there exists $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{P}$ such that the configuration $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{P} \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega''$ is also well-typed.*

Proof. Let $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ be a well-typed configuration for an expression e as qualified above. We argue that for any possible step $(d, h, s, \rho, \Omega', e) \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'', e')$ there exists $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{\mathrm{P}}$ such that $\bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}; \bar{\mathrm{P}} \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega''$. By lemma 5.2, some $\boxed{\mathrm{E}^{(\mathrm{D})}}$ rule besides $\boxed{\mathrm{E1A}^{(\mathrm{D})}}$ or $\boxed{\mathrm{E1B}^{(\mathrm{D})}}$ derived our step, so to prove our goal it suffices to consider all other cases and prove the following subgoals:

i) The invariants $\boxed{\mathrm{F3}^{(\mathrm{D})}}$-$\boxed{\mathrm{F14}^{(\mathrm{D})}}$ are preserved under stepping: assuming $\vdash e; d, h, s, \rho : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ `agree` we show $\vdash e'; d', h', s', \rho' : \bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}; \bar{\mathrm{P}}$ `agree`.

ii) The well-typedness of the expression e is preserved under stepping: assuming $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$ we show $\bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}; \bar{\mathrm{P}} \vdash e' : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega''$.

When phrasing the proof goal in each case below we perform some obvious simplifications derived from the restricted form of the conclusions of relevant typing and evaluation rules, and we instantiate a values of $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{\mathrm{P}}$. Here is the general form of the proofs that follow in cases $\boxed{\mathrm{E2}^{(\mathrm{D})}}$-$\boxed{\mathrm{E17}^{(\mathrm{D})}}$:

Case $\boxed{\mathrm{E[X]}^{(\mathrm{D})}}$**:** $(\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega')$

$\wedge\ ((d, h, s, \rho, \Omega', e) \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'', e'))$

$\implies (\bar{\mathcal{H}}; \bar{\Gamma}; \bar{\Omega}; \bar{\mathrm{P}} \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega')$

We use these two assumptions, well-typedness of the original configuration and stepping of the original configuration to the new configuration, to prove our two subgoals:

i) This subgoal holds if each invariant is preserved. Each invariant relies only a subset of the contexts, and only a subset of the contexts differ between the original and the new configuration. To determine which invariants require nontrivial arguments for satisfaction in the new configuration, we first list the contexts that differ: e.g. $\bar{\mathcal{H}}$ differs from $\mathcal{H}$, $\bar{\Gamma}$ differs from Γ, etc. We then list the invariants reliant on the differing contexts:

$\boxed{\mathrm{F[X]}^{(\mathrm{D})}}$**:** We show each such invariant holds for $e', \bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{\mathrm{P}}, d', h', s', \rho'$

ii) We show well-typedness of e'.

For brevity, we will frequently use overbarred terms $(\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \ldots)$ to represent the contexts under which e' is type-checked without explicitly binding them.

Case $\boxed{\mathrm{E2}^{(\mathrm{D})}}$**:** $(\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, x : r\ \tau) \dashv \mathcal{H}; \Gamma; \Omega)$

$\wedge\ ((d, h, s, \rho, \Omega', x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega', l))$

$\implies (\mathcal{H}; \Gamma; \Omega; l : r\ \tau \vdash (d, h, s, \rho, l : r\ \tau) \dashv \mathcal{H}; \Gamma; \Omega)$

i) $\bar{\mathrm{P}} = (l : r\ \tau)$ differs from P, forcing us to argue that the following invariants are preserved:

$\boxed{\mathrm{F13}^{(\mathrm{D})}}$**:** No new regions names are introduced, so this invariant is preserved.

$\boxed{\text{F14}^{(\text{D})}}$: Inverting $\boxed{\text{E2}^{(\text{D})}}$ tells us $s(x) = l$. Inverting $\boxed{\text{T2}^{(\text{D})}}$, and then inverting $\boxed{\text{F10}^{(\text{D})}}$ on the assumed context agreement of the original configuration tells us $\rho(s(x)) = (r, \tau)$ and $r \in dom(\mathcal{H})$. $\rho(l) = (r, \tau)$ is exactly what we need to establish that $\boxed{\text{F14}^{(\text{D})}}$ holds on the new configuration containing $\bar{\text{P}} = (l : r\ \tau)$.

ii) $r \in dom(\mathcal{H})$, as shown above, is exactly the additional information required to apply $\boxed{\text{T22}^{(\text{D})}}$ and conclude that the desired typing judgment holds.

Case $\boxed{\text{E3}^{(\text{D})}}$: $(\mathcal{H}; \Gamma; \Omega; \text{P} \vdash (d, h, s, \rho, \texttt{new-}\tau\texttt{@}r : r\ \tau) \dashv \mathcal{H}, r^{\cdot}\langle\rangle; \Gamma; \Omega \uplus \{r\})$

$\wedge\ ((d, h, s, \rho, \Omega \uplus \{r\}, \texttt{new-}\tau\texttt{@}r) \xrightarrow{\text{eval}} (d \uplus d_{new}, h \uplus h_{new}, s, \rho \uplus \rho_{new}, \Omega \uplus regs(\rho_{new}), l)$

$\implies\ (\mathcal{H}, r^{\cdot}\langle\rangle; \Gamma; \Omega \uplus regs(\rho_{new}); l : r\ \tau \vdash (d \uplus d_{new}, h \uplus h_{new}, s, \rho \uplus \rho_{new}, l : r\ \tau) \dashv \mathcal{H}, r^{\cdot}\langle\rangle; \Gamma; \Omega \uplus regs(\rho_{new}))$

i) $\bar{\mathcal{H}}, \bar{\Omega}, \bar{\text{P}}$ differ from $\mathcal{H}, \Omega, \text{P}$, as seen in the goal statement. Additionally, d', h', ρ' differ from d, h, ρ.

$\boxed{\text{F3}^{(\text{D})}}$: From the premise `extracts-fresh-heap` $(\Omega \uplus \{r\}; \rho, r, \tau; \rho_{new}, h_{new}, l)$ of $\boxed{\text{E3}^{(\text{D})}}$, we can conclude by **Simplicity** that $G_{total}(\rho_{new}, h_{new})$ is a forest. Thus $G_S(\mathcal{H}, \rho', h', s)$ is just $G_S(\mathcal{H}, \rho, h, s)$ with an appended tree that is disjoint over locations by **Disjointness**, regions by **Region-Freshness**, incoming references by $\boxed{\text{F6}^{(\text{D})}}$ on the original configuration, and outgoing references by **Heap-Sanity**, so it is a forest. Now, to obtain $G_S(\mathcal{H}, r^{\cdot}\langle\rangle, \rho', h', s)$, we must simply delete the root of that appended tree, which clearly will yield another forest, so we can conclude that $\boxed{\text{F3}^{(\text{D})}}$ is preserved.

$\boxed{\text{F4}^{(\text{D})}}$: All locations that are added to $dom(\rho)$ are also added to d (by **Heap-Sanity** of `extracts-fresh-heap`), so $\boxed{\text{F4}^{(\text{D})}}$ cannot be violated.

$\boxed{\text{F5}^{(\text{D})}}$: The only region added to $dom(\mathcal{H})$ is r, and there exist no other regions r' such that $(\mathcal{H}, r^{\cdot}\langle\rangle), \rho', h', s \vdash r \hookrightarrow r'$ or $(\mathcal{H}, r^{\cdot}\langle\rangle), \rho', h', s \vdash r' \hookrightarrow r$, so there exists no additional χ that could invalidate this invariant.

$\boxed{\text{F6}^{(\text{D})}}$: This invariant is maintained under disjoint unions, and so $\vdash h$ `heap-closed` (by assumption) and $\vdash h_{new}$ `heap-closed` (from **Heap-Sanity** of `extracts-fresh-heap`) together imply $\vdash h'$ `heap-closed`.

$\boxed{\text{F7}^{(\text{D})}}$: This invariant is also maintained under disjoint unions, so preservation follows from the same argument as $\boxed{\text{F6}^{(\text{D})}}$.

$\boxed{\text{F8}^{(\text{D})}}$: The *live-set* grows by adding all of d_{new}, but **Heap-Sanity** tells us that $\boxed{\text{F8}^{(\text{D})}}$ already holds over d_{new}.

$\boxed{\text{F9}^{(\text{D})}}$: No tracked variables are added to $\mathcal{H}$, so this invariant is preserved..

$\boxed{\text{F10}^{(\text{D})}}$: No Γ-bindings exist in the new region r by $\boxed{\text{F13}^{(\text{D})}}$ (noting $r \notin \Omega$), and ρ does not change on any existing locations, so this invariant is preserved.

$\boxed{\text{F11}^{(\text{D})}}$: The set of tracked variables in $\mathcal{H}$ is unchanged, so this invariant is unaffected.

$\boxed{\text{F12}^{(\text{D})}}$: No tracked references are added to $\mathcal{H}$, and ρ does not change on any existing locations, so this invariant is preserved.

$\boxed{\text{F13}^{(\text{D})}}$: The exact set of regions in the new configuration but not the old configuration is $regs(\rho_{new})$, which is added to Ω in parallel, so this invariant is preserved.

$\boxed{\text{F14}^{(\text{D})}}$: Follows from **Rootedness** of the mapping for l added to ρ.

ii) **Rootedness** of `extracts-fresh-heap` gives us $(l : r, \tau \in \rho')$, and $r \in dom(\mathcal{H}, r^{\cdot}\langle\rangle)$ is trivial, so we can conclude by application of $\boxed{\text{T22}^{(\text{D})}}$.

Case $\boxed{\text{E4}^{(\text{D})}}$**:** $(\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d,h,s,\rho,l;e : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega')$

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

$\implies\ (\mathcal{H};\Gamma;\Omega;\cdot \vdash (d,h,s,\rho,e : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega')$

i) $\bar{\mathrm{P}} = \cdot$ differs from P.

$\boxed{\text{F8}^{(\text{D})}}$**:** weakened by eliminating l as a possible root of the *live-set*.

$\boxed{\text{F13}^{(\text{D})}}$, $\boxed{\text{F14}^{(\text{D})}}$**:** both weakened by replacing P with $\cdot$.

ii) Inverting $\boxed{\text{T3}^{(\text{D})}}$, together with observing the symmetric input and output contexts in the conclusion of $\boxed{\text{T22}^{(\text{D})}}$, allows us to conclude that $\mathcal{H};\Gamma;\Omega;\cdot \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'$.

Case $\boxed{\text{E5}^{(\text{D})}}$**:** This is just a special case of $\boxed{\text{E3}^{(\text{D})}}$; the updates to all dynamic contexts here satisfy `extracts-fresh-heap`$(\Omega,\rho,r,\tau;(l_3 \mapsto (r,\tau')),(l \mapsto (\tau',v_3)),l_3)$ so we dispatch to preservation for $\boxed{\text{E3}^{(\text{D})}}$. TODO: is this formal enough?

Case $\boxed{\text{E6}^{(\text{D})}}$**:** $(\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d,h,s,\rho,\texttt{if}(l)\{e_t\}\ \texttt{else}\ \{e_f\} : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega_t \cup \Omega_f)$

$\wedge\ ((d,h,s,\rho,\Omega_t \cup \Omega_f,\texttt{if}(l)\{e_t\}\ \texttt{else}\ \{e_f\}) \xrightarrow{\text{eval}} (d,h,s,\rho,\Omega_t \cup \Omega_f,e_t))$

$\implies\ (\mathcal{H};\Gamma;\Omega \uplus (NR(e_f) - NR(e_t));\cdot \vdash (d,h,s,\rho,e_t : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega_t \cup \Omega_f)$

i) Ω,P differ.

$\boxed{\text{F8}^{(\text{D})}}$**:** Since the expression is replaced with a subexpression of itself, the *live-set* can only shrink, so this invariant is weakened.

$\boxed{\text{F13}^{(\text{D})}}$**:** Only states a lower bound for Ω, which is thus preserved by growth of Ω, and $\Omega \subseteq \Omega \uplus (NR(e_f) - NR(e_t))$.

$\boxed{\text{F14}^{(\text{D})}}$**:** Trivial since $\cdot$ used for P.

ii) Inverting $\boxed{\text{T14}^{(\text{D})}}$ we obtain the typing judgment $\mathcal{H};\Gamma;\Omega;\cdot \vdash e_t : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega_t$. By lemma 5.4, $\Omega \subseteq \Omega_t$, so $\Omega_f - \Omega_t$ is disjoint from both the LHS and RHS in this judgment, and lemma 5.5 allows us to conclude $\mathcal{H};\Gamma;\Omega \uplus (\Omega_f - \Omega_t);\cdot \vdash e_t : r\ \tau :\dashv \mathcal{H}';\Gamma';\Omega_t \cup \Omega_f$. We finally note that by lemma 5.6 $NR(e_f) - NR(e_t) = (\Omega_f - \Omega) - (\Omega_t - \Omega) = (\Omega_f - \Omega_t)$, and we are done.

Case $\boxed{\text{E7}^{(\text{D})}}$**:** Symmetric to case for $\boxed{\text{E6}^{(\text{D})}}$.

Case $\boxed{\text{E8}^{(\text{D})}}$**:** Where $e = \texttt{while}(e_{bool})\{e_{body}\}@r_u$, $e' = \texttt{if}(e_{bool})\{e_{body};\phi(e)\}\ \texttt{else}\ \{\texttt{new-unit}@r_u\}$, and ϕ is a bijection from $NR(e_{body}) \uplus NR(e_{bool})$ to Ω_{new}:

$(\mathcal{H};\Gamma;\Omega;\cdot \vdash (d,h,s,\rho,e : r_u\ \texttt{unit}) \dashv \mathcal{H}';\Gamma';\Omega''')$

$\wedge\ ((d,h,s,\rho,\Omega''',e) \xrightarrow{\text{eval}} (d,h,s,\rho,\Omega''' \uplus \Omega_{new},e'))$

$\implies\ (\mathcal{H};\Gamma;\Omega;\cdot \vdash (d,h,s,\rho,e' : r_u\ \texttt{unit}) \dashv \mathcal{H}';\Gamma';\Omega''' \uplus \Omega_{new})$

i) Only the expression differs.

TODO: Is the following typing argument impenetrable?

ii) On the top level, we conclude typing for e' by applying $\boxed{\text{T14}^{(\text{D})}}$ with $\Omega_t = \Omega''' \uplus \Omega_{new}$ and $\Omega_f = \Omega' \uplus \{r_u\}$. Incorporating all typing judgments derivable from inverting $\boxed{\text{T15}^{(\text{D})}}$ on the given typing judgment for e, this reduces our goal to proving:

$$\mathcal{H};\Gamma;\Omega';\cdot \vdash e_{body};\phi(e) : r_u\ \texttt{unit} \dashv \mathcal{H}';\Gamma';\Omega''' \uplus \Omega_{new} \tag{1}$$

$$\mathcal{H};\Gamma;\Omega';\cdot \vdash \texttt{new-unit}@r_u : r_u\ \texttt{unit} \dashv \mathcal{H}';\Gamma';\Omega' \uplus \{r_u\} \tag{2}$$

Goal 2 follows from a trivial application of T11$^{(\text{D})}$ on a premise from the above inversion of T15$^{(\text{D})}$, so we turn our attention to goal 1. Under application of T3$^{(\text{D})}$ with the typing for e_{body} in mind, we can see that its proof reduces to proving:

$$\mathcal{H};\Gamma;\Omega'';\cdot\vdash\phi(e):r_u\ \texttt{unit}\dashv\mathcal{H}';\Gamma';\Omega'''\uplus\Omega_{new}\tag{3}$$

Noting that by lemma 5.6, $\Omega''-\Omega=NR(e_{bool})\uplus NR(e_{body})=(\Omega'''\uplus\Omega_{new})-(\Omega\uplus\Omega_{new}\uplus\{r_u\})$, we apply lemma 5.5 to reduce goal 3 under symmetric expansion, and then observe $\phi(NR(e_{bool})\uplus NR(e_{body}))=\Omega_{new}$ (by assumption), $\Omega'''=\Omega\uplus NR(e_{bool})\uplus NR(e_{body})\uplus\{r_u\}$ (from lemma 5.6 as used above), and ϕ is constant on Ω and $\{r_u\}$ to conclude that $\phi(\Omega''')=\Omega\oplus\Omega_{new}\uplus\{r_u\}$. These two reductions admit the new goal:

$$\mathcal{H};\Gamma;\Omega;\cdot\vdash\phi(e):r_u\ \texttt{unit}\dashv\mathcal{H}';\Gamma';\phi(\Omega''')\tag{4}$$

We know ϕ is constant on Ω, and the set of regions mentioned in $\mathcal{H}$ and Γ is contained in Ω by F13$^{(\text{D})}$, so $\mathcal{H}$ and Γ are constant under ϕ. The set of regions mentioned in $\mathcal{H}',\Gamma'$ exceeds that mentioned in $\mathcal{H},\Gamma$ by only r_u, and thus exceeds Ω by only r_u, on which ϕ is also constant. Thus $\mathcal{H}'$ and Γ' are constant under ϕ. By lemma 5.7 we are done, as goal 4 is directly a bijective renaming under ϕ of the assumption $\mathcal{H};\Gamma;\Omega;\cdot\vdash e:r_u\ \texttt{unit}\dashv\mathcal{H}';\Gamma';\Omega'''$.

Case E9$^{(\text{D})}$**:** $(\mathcal{H};\Gamma;\Omega;\cdot\vdash(d,h,s,\rho,\texttt{declare}\ x:\tau\ \texttt{in}\ \{e\}:r\ \tau')\dashv\mathcal{H}';\Gamma';\Omega')$

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

$\implies\ (\mathcal{H};\Gamma,x:\bot\ \tau;\Omega;\cdot\vdash(d,h,s[x\mapsto\bot],\rho,e;\texttt{drop-var}\ x:r\ \tau')\dashv\mathcal{H}';\Gamma';\Omega')$

i) Γ,s differ.

F3$^{(\text{D})}$, F4$^{(\text{D})}$, F5$^{(\text{D})}$, F8$^{(\text{D})}$, F9$^{(\text{D})}$, F11$^{(\text{D})}$, F12$^{(\text{D})}$**:** All not affected because x does not appear in $\mathcal{H}$

F10$^{(\text{D})}$**:** Preserved because $\bot\notin dom(\mathcal{H})$

F13$^{(\text{D})}$**:** Preserved because $\bot$ trivially allowed for in $range(\Gamma\upharpoonright_r)$

ii) Inverting T12$^{(\text{D})}$ on the given typing judgment for $\texttt{declare}\ x:\tau\ \texttt{in}\ \{e\}$ tells us $\mathcal{H};\Gamma,x:\bot\ \tau;\Omega;\cdot\vdash e:r\ \tau'\dashv\mathcal{H}';\Gamma',x:r_{final}\ \tau;\Omega'$ and that x is not present in $\mathcal{H}$ or $\mathcal{H}'$. This is strictly stronger than the premise for T25$^{(\text{D})}$ to derive $\mathcal{H};\Gamma,x:\bot\ \tau;\Omega;\cdot\vdash e;\texttt{drop-var}\ x:r\ \tau'\dashv\mathcal{H}';\Gamma';\Omega'$, so we are done.

Case E10$^{(\text{D})}$**:** $(\mathcal{H};\Gamma,x:r_{old}\ \tau;\Omega;\mathrm{P}\vdash(d,h,s\uplus(x\mapsto l_{old}),\rho,x=l:r\ \tau)\dashv\mathcal{H};\Gamma,x:r\ \tau;\Omega)$

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

$\implies\ \mathcal{H};\Gamma,x:(\rho\upharpoonright_r(l))\ \tau;\Omega;\mathrm{P}\vdash(d,h,s\uplus(x\mapsto l),\rho,l:r\ \tau)\dashv\mathcal{H};\Gamma,x:r\ \tau;\Omega)$

As a preliminary, we show that $\rho(l)=r\ \tau$. Inverting T8$^{(\text{D})}$ on the typing assumption for $x=l$ yields $\mathcal{H};\Gamma,x:r_{old}\ \tau;\Omega;\mathrm{P}\vdash l:r\ \tau\dashv\mathcal{H};\Gamma,x:r_{old}\ \tau;\Omega$, and a subsequent inversion of T22$^{(\text{D})}$ yields $(l:r\ \tau)\in\mathrm{P}$ and $r\in dom(\mathcal{H})$. Inverting F14$^{(\text{D})}$ on the assumption gives us $(l:r\ \tau)\in\rho$, which concludes this subgoal.

i) Γ,s altered.

F3$^{(\text{D})}$**:** Inverting T8$^{(\text{D})}$ tells us $x\notin vars(\mathcal{H})$, so changes to x's s-binding have no effect on G_S.

F4$^{(\text{D})}$, F5$^{(\text{D})}$, F8$^{(\text{D})}$**:** These depends on s only through calls to *ref-valid*. Since $x\notin vars(\mathcal{H})$, changes to $s(x)$ do not affect *ref-valid* for any references.

F9$^{(\text{D})}$**:** Γ does not change on any $x\in vars(\mathcal{H})$.

$\boxed{\text{F10}^{(\text{D})}}$: s is updated exactly to map x to l, and our preliminary above tells us that the new Γ agrees with ρ w.r.t. their mappings on x and l, respectively.

$\boxed{\text{F11}^{(\text{D})}}$, $\boxed{\text{F12}^{(\text{D})}}$: s does not change on any $x \in vars(\mathcal{H})$.

$\boxed{\text{F13}^{(\text{D})}}$: No new regions are introduced to Γ that were not already present in P.

ii) Inverting $\boxed{\text{T8}^{(\text{D})}}$ gives us exactly the typing judgment on l we need up to changes in Γ, and, noting that $\boxed{\text{T22}^{(\text{D})}}$ does not depend on Γ, we are free to conclude that our desired typing judgment on l holds.

Case $\boxed{\text{E11}^{(\text{D})}}$: $(\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l_f(l)@\Omega'_{out} : r'\ \tau') \dashv \mathcal{H}'; \Gamma; \Omega \uplus \Omega'_{out})$

$\wedge\ ((d, h, s, \rho, \Omega', l_f(l)@\Omega'_{out}) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega \uplus \Omega_{new}, \texttt{declare}\ x : \tau\ \texttt{in}\ \{x = l; \phi(e')\}))$

$\implies (\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, \texttt{declare}\ x : \tau\ \texttt{in}\ \{x = l; \phi(e')\} : r'\ \tau') \dashv \mathcal{H}'; \Gamma; \Omega \uplus \Omega_{new})$

i) Only the expression differs. TODO: Is the following typing argument impenetrable?

ii) By application of $\boxed{\text{T12}^{(\text{D})}}$, $\boxed{\text{T3}^{(\text{D})}}$, and $\boxed{\text{T8}^{(\text{D})}}$, it can be seen that our proof goal reduces to:

$$\mathcal{H}; \Gamma, x : r\ \tau; \Omega; \cdot \vdash \phi(e') : r'\ \tau' \dashv \mathcal{H}'; \Gamma; \Omega \uplus \Omega_{new} \tag{5}$$

We note that inversion of $\boxed{\text{E11}^{(\text{D})}}$ on the stepping relation gives us $h(l_f) = ((q_{\text{ARG}}\ \tau \longrightarrow q_{\text{RET}}\ \tau'), v_f)$ and $F_d(v_f) = \lambda x.e@\Omega_{out}$, and consulting section 4.2.1 allows us to assume the derivation of $\vdash \texttt{def}\ q_{\text{RET}}\ \tau'\ fn(q_{\text{ARG}}\ \tau\ x)\{e\}@\Omega_{out}$ for some function name fn, heap context $\mathcal{H}_{out}$, and region names $r_{sym}, r'_{sym}, r_{final}$. Inversion of $\boxed{\text{T1}^{(\text{D})}}$ on this `def` judgment, combined with the α-renaming given as a premise through inversion of $\boxed{\text{E11}^{(\text{D})}}$, gives us:

$$r^{\dagger}_{sym}\langle\rangle; x : r_{sym}\ \tau; \{r_{sym}\}; \cdot \vdash e' : r'_{sym}\ \tau' \dashv \mathcal{H}_{out}; x : r_{final}\ \tau; \{r_{sym}\} \uplus \Omega_{out} \uplus \Omega_{extra} \tag{6}$$

We note two things regarding this new inversion. $NR(e) = \Omega_{out} \uplus \Omega_{extra}$ by lemma 5.6, and $\vdash (q_{\text{ARG}}\ r_{sym} \longrightarrow q_{\text{RET}}\ r'_{sym}) : (r^{\circ}_{sym}\langle\rangle; \{r_{sym}\}) \Rightarrow (\mathcal{H}_{out}; \{r_{sym}\} \uplus \Omega_{out})$. We note that $r_{sym} \notin NR(e)$ as another consequence of lemma 5.6, so we are free to perform a renaming of it to r (lemma 5.7). We also note that under ϕ, r'_{sym} maps to r' because either $r'_{sym} = r_{sym}$ and $r' = r$ (in the case $q_{\text{ARG}}, q_{\text{RET}} = \texttt{preserves}, \texttt{bnd}$) or $\{r'_{sym}\} = \Omega_{out}$ and $\{r'\} = \Omega'_{out}$ (otherwise). Thus, under another application of lemma 5.7:

$$r^{\dagger}\langle\rangle; x : r\ \tau; \{r\}; \cdot \vdash \phi(e') : r'\ \tau' \dashv \mathcal{H}'_{out}; x : r'_{final}\ \tau; \{r\} \uplus \Omega_{new} \tag{7}$$

Finally, we observe that $\mathcal{H} - r^{\circ}\langle\rangle + \mathcal{H}'_{out} = \mathcal{H}'$ by the symmetry between the two judgments $\vdash (q_{\text{ARG}}\ r \longrightarrow q_{\text{RET}}\ r') : (r^{\circ}\langle\rangle; \{r\}) \Rightarrow (\mathcal{H}'_{out}; \{r\} \uplus \Omega'_{out})$ (stated in the preceding paragraph then mapped by ϕ) and $\vdash (q_{\text{ARG}}\ r \longrightarrow q_{\text{RET}}\ r') : (\mathcal{H}; \Omega) \Rightarrow (\mathcal{H}'; \Omega \uplus \Omega'_{out})$ (from inversion of $\boxed{\text{T9R}^{(\text{D})}}$). This allows us to see that goal 5 is just a symmetric context expansion of fact 7 by $(\mathcal{H} - r^{\circ}\langle\rangle, \Gamma, \Omega - \{r\})$, and since we know that the only region mentioned in both fact 7 and $(\mathcal{H} - r^{\circ}\langle\rangle, \Gamma, \Omega - \{r\})$ is r, which is pinned in 7, and by $\boxed{\text{F9}^{(\text{D})}}$ and $\boxed{\text{F10}^{(\text{D})}}$ $vars(\phi(e')) \cap dom(s) = \emptyset \implies vars(\phi(e')) \cap (vars(\Gamma) \cup vars(\mathcal{H})) = \emptyset$, we can apply lemma 5.19 to conclude our goal 5, and we are done.

Case $\boxed{\text{E12}^{(\text{D})}}$: $(\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l.f : r\ \tau_f) \dashv \mathcal{H}; \Gamma; \Omega)$

$\wedge\ ((d, h, s, \rho, \Omega, l.f) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l_f))$

$\implies (\mathcal{H}; \Gamma; \Omega; l_f : r\ \tau_f \vdash (d, h, s, \rho, l_f : r\ \tau_f) \dashv \mathcal{H}; \Gamma; \Omega)$

i) P altered.

$\boxed{\text{F13}^{(\text{D})}}$: Trivially preserved; region name r was already present in P.

$\boxed{\text{F14}^{(\text{D})}}$: We must show $\rho(l_f) = (r, \tau_f)$. Observing that the latter two premises after inverting $\boxed{\text{E12}^{(\text{D})}}$ on the assumed step, with **Non-Shadowing**, give exactly the information needed to apply $\boxed{\text{F6}^{(\text{D})}}$ and conclude $h(l_f) = (\tau_f, v_f)$. $\boxed{\text{F7}^{(\text{D})}}$ then allows us to conclude $\rho(l_f) = (r, \tau_f)$, where regionality comes from boundedness of f.

ii) Inverting $\boxed{\text{T4}^{(\text{D})}}$ tells us $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash l : r\ \tau \dashv \mathcal{H}; \Gamma; \Omega$, so $\mathcal{H}$ tracks r, which is sufficient information to apply $\boxed{\text{T22}^{(\text{D})}}$ and conclude that our desired typing judgment holds.

Case $\boxed{\text{E13}^{(\text{D})}}$**:** $(\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, x.f : r_f\ \tau_f) \dashv \mathcal{H}; \Gamma; \Omega)$

$\wedge\ ((d, h, s, \rho, \Omega, x.f) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l_f))$

$\Longrightarrow\ (\mathcal{H}; \Gamma; \Omega; l_f : r_f\ \tau_f \vdash (d, h, s, \rho, l_f : r_f\ \tau_f) \dashv \mathcal{H}; \Gamma; \Omega)$

i) P altered.

$\boxed{\text{F13}^{(\text{D})}}$: Inversion of $\boxed{\text{T5}^{(\text{D})}}$ shows us that r_f was already present in $\mathcal{H}$ so no new regions names are added to a context and this invariant is preserved.

$\boxed{\text{F14}^{(\text{D})}}$: We must show $\rho(l_f) = (r_f, \tau_f)$. That $\rho \restriction_\tau (l_f) = \tau_f$ follows, as above, from $\boxed{\text{F6}^{(\text{D})}}$ and $\boxed{\text{F7}^{(\text{D})}}$. That $\rho \restriction_r (l_f) = r_f$ follows directly from inverting $\boxed{\text{F12}^{(\text{D})}}$ and instantiating its implication with the premises derived from inversion of $\boxed{\text{E13}^{(\text{D})}}$ on the assumed step.

ii) Inversion of $\boxed{\text{T5}^{(\text{D})}}$ gave us $r_f \in dom(\mathcal{H})$, so application of $\boxed{\text{T22}^{(\text{D})}}$ is clearly possible to conclude that our desired typing judgment holds.

Case $\boxed{\text{E14}^{(\text{D})}}$**:** $(\mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (l \mapsto (\tau, v)), s, \rho, l.f = l_f : r\ \tau_f) \dashv \mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega)$

$\wedge\ ((d, h \uplus (l \mapsto (\tau, v)), s, \rho, \Omega, l.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (l \mapsto \tau, v[f \mapsto l_f])), s, \rho, \Omega, l_f)$

$\Longrightarrow\ (\mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, \rho, l_f : r\ \tau_f) \dashv \mathcal{H}, r^\circ\langle X\rangle; \Gamma; \Omega)$

i) h altered.

$\boxed{\text{F3}^{(\text{D})}}$: G_S does not read h over bounded references - and we know from inversion of $\boxed{\text{E14}^{(\text{D})}}$ on the assumed step that $\mathsf{bnd}\ f\ \tau_f \in \mathit{fields}(\tau)$, so **Non-Shadowing** prevents G_S from changing under our updates to h. With no changes to G_S, this invariant is trivially preserved.

$\boxed{\text{F4}^{(\text{D})}}$, $\boxed{\text{F5}^{(\text{D})}}$, $\boxed{\text{F8}^{(\text{D})}}$: These depend on h only through reachability, which, as in the case of $\boxed{\text{F3}^{(\text{D})}}$, is not affected by updates to bounded references

$\boxed{\text{F6}^{(\text{D})}}$: Inversion of $\boxed{\text{T6R}^{(\text{D})}}$ tells us $(l_f : r\ \tau_f) \in \mathrm{P}$ and thus by $\boxed{\text{F14}^{(\text{D})}}$ $(l_f : r\ \tau_f) \in \rho$. $\boxed{\text{F7}^{(\text{D})}}$ on the original configuration then gives us $h \restriction_\tau (l_f) = \tau_f$, which is exactly what $\boxed{\text{F6}^{(\text{D})}}$ demands to be preserved under this update.

$\boxed{\text{F7}^{(\text{D})}}$: We concluded above that $\rho \restriction_r (l_f) = r$ and inversion of $\boxed{\text{T6R}^{(\text{D})}}$, followed by inversion of $\boxed{\text{T22}^{(\text{D})}}$, followed by $\boxed{\text{F14}^{(\text{D})}}$ allows us to conclude that $\rho \restriction_r (l) = r$ as well, so the implication in $\boxed{\text{F7}^{(\text{D})}}$ can not be invalidated by this update as the only reference update does not cross regions.

$\boxed{\text{F12}^{(\text{D})}}$: This cannot be affected by changes to $h \restriction_v$ on bounded fields.

ii) As noted in the $\boxed{\text{F6}^{(\text{D})}}$ case above, $(l_f : r\ \tau_f) \in \mathrm{P}$, so we conclude the desired typing judgment by trivial application of $\boxed{\text{T22}^{(\text{D})}}$.

Case $\boxed{\text{E15}^{(\text{D})}}$**:** $(\mathcal{H}, r^\circ\langle x[f \rightarrowtail r_{old}, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (l \mapsto (\tau, v)), s, \rho, \Omega, x.f = l_f : r_f\ \tau_f) \dashv \mathcal{H}, r^\circ\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega)$

$\wedge\ ((d, h \uplus (l \mapsto (\tau, v)), s, \rho, \Omega, x.f = l_f) \xrightarrow{\text{eval}} (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, \rho, \Omega, l_f))$
$\implies\ (\mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h \uplus (l \mapsto (\tau, v[f \mapsto l_f])), s, \rho, \Omega, l_f : r_f\ \tau_f) \dashv \mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_f, F], X\rangle, r_f^{\circ f}\langle X_f\rangle; \Gamma; \Omega)$

i) $\mathcal{H}, h$ altered.

F3$^{(\mathrm{D})}$: Unlike in the bounded assignment case, G_{total} is indeed altered by this update. However, the set of **semi-tracked** references is not altered, and no χ reachable from a **semi-tracked** reference is altered, as it is easy to see that F5$^{(\mathrm{D})}$ implies no tracked χ can be reachable from a **semi-tracked** reference. Thus G_S is not affected by this update, so this invariant is preserved.

F4$^{(\mathrm{D})}$, F8$^{(\mathrm{D})}$: To show preservation of these invariants, it suffices to show that the *live-set* is the same in the new and old configurations. Clearly, r_f and all reachable regions from it are in both *live-set*s. If r_{old} is not tracked then f is **invalid** in the old configuration, so retargeting f from r_{old} would not drop it from the *live-set*. If r_{old} is tracked then it and all reachable regions from it are in both *live-set*s. In either case, the *live-set* is constant and we have shown preservation.

F5$^{(\mathrm{D})}$: No χ become untracked (equivalently, leave $dom(\mathcal{H}) \cup loc\text{-}refs(\mathcal{H})$) as a result of this update, any χ that is reachable from a tracked region after the update had to be reachable from a tracked region before the update, and any χ that reaches a tracked region after the update similarly had to reach a tracked region before the update. Both of the latter two properties follow from the fact that both endpoint regions of f in the new configuration are tracked, and now we can see there is no way for the implication premise to be invalidated by the updates, so F5$^{(\mathrm{D})}$ is preserved.

F6$^{(\mathrm{D})}$: Preservation here follows from inversion of T7$^{(\mathrm{D})}$, T22$^{(\mathrm{D})}$, and finally F14$^{(\mathrm{D})}$ on the assumed typing judgment.

F7$^{(\mathrm{D})}$: Since the field being updated is isolated, the last premise here cannot be invalidated and the former do not depend on $h \restriction_v$, so this is preserved.

F9$^{(\mathrm{D})}$: This invariant depends only on the set of tracked variables and their regionality, which is not changed under this update.

F10$^{(\mathrm{D})}$: This invariant depends only on the set of tracked regions, which is not changed under this update.

F11$^{(\mathrm{D})}$: This invariant depends only on the set of tracked variables, which is not changed under this update.

F12$^{(\mathrm{D})}$: Preservation here follows from $\rho \restriction_r (h \restriction_v (s(x)).f) = r_f$, which in the new configuration reduces to $\rho \restriction_r (l_f) = r_f$, which follows from inversion of T7$^{(\mathrm{D})}$, T22$^{(\mathrm{D})}$, and finally F14$^{(\mathrm{D})}$ on the assumed typing judgment.

F13$^{(\mathrm{D})}$: No new region names appear, so this is trivially preserved.

ii) The above mentioned inversions of T7$^{(\mathrm{D})}$ and T22$^{(\mathrm{D})}$ on the assumed typing judgment are sufficient to conclude $(l_f : r_f\ \tau_f) \in \mathrm{P}$, so we conclude the desired typing judgment by trivial application of T22$^{(\mathrm{D})}$.

Case E16A$^{(\mathrm{D})}$: $(\mathcal{H}, r^{\circ}\langle\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{focus}\ x : r_e\ \tau_e) \dashv \mathcal{H}, r^{\circ}\langle x[]\rangle; \Gamma; \Omega)$
$\wedge\ ((d, h, s, \rho, \Omega, l; \texttt{focus}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l))$
$\implies\ (\mathcal{H}, r^{\circ}\langle x[]\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l) \dashv \mathcal{H}, r^{\circ}\langle x[]\rangle; \Gamma; \Omega)$

i) $\mathcal{H}$ altered.

$\boxed{\text{F3}^{(\text{D})}}$, $\boxed{\text{F4}^{(\text{D})}}$, $\boxed{\text{F5}^{(\text{D})}}$, $\boxed{\text{F8}^{(\text{D})}}$, $\boxed{\text{F10}^{(\text{D})}}$, $\boxed{\text{F12}^{(\text{D})}}$, $\boxed{\text{F13}^{(\text{D})}}$: These depends on $\mathcal{H}$ only through $dom(\mathcal{H})$, $targets(\mathcal{H})$, $reg\text{-}refs(\mathcal{H})$, $loc\text{-}refs(\mathcal{H})$, and reachability (which in turn depends on $\mathcal{H}$ only through *ref-valid*). None of these are affected by the `focus` update, so this invariant is trivially preserved.

$\boxed{\text{F9}^{(\text{D})}}$: After the `focus` update, $x@r \in reg\text{-}vars(\mathcal{H})$, but the premise $(x : r\ \tau) \in \Gamma$ gleaned from inversion of $\boxed{\text{T16}^{(\text{D})}}$ on the assumed typing judgment is sufficient to conclude that this invariant is preserved.

$\boxed{\text{F11}^{(\text{D})}}$: If any $x' \in vars(\mathcal{H})$ were to exist with $x' \neq x$ and $s(x') = s(x)$ then by $\boxed{\text{F9}^{(\text{D})}}$ and $\boxed{\text{F10}^{(\text{D})}}$ on the original configuration it would have to be in the same region as x, but by inspection there are no other variables in region r, so no such x' can exist and thus this invariant is preserved.

ii) Inversion of $\boxed{\text{T16}^{(\text{D})}}$ and $\boxed{\text{T22}^{(\text{D})}}$ on the assumed typing judgment gives us $(l : r_e\ \tau_e) \in \mathrm{P}$ and $r_e \in dom(\mathcal{H}) \uplus \{r\}$, which is exactly the information we need to apply $\boxed{\text{T22}^{(\text{D})}}$ and conclude the desired typing judgment.

Case $\boxed{\text{E16B}^{(\text{D})}}$: $(\mathcal{H}, r\langle x[], X\rangle; \Gamma; O; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{unfocus}\ x : r_e\ \tau_e) \dashv \mathcal{H}, r\langle X\rangle; \Gamma; \Omega$

$\wedge\ ((d, h, s, \rho, \Omega, l; \texttt{unfocus}\ x) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, l))$

$\implies (\mathcal{H}, r\langle X\rangle; \Gamma; O; \mathrm{P} \vdash (d, h, s, \rho, l : r_e\ \tau_e) \dashv \mathcal{H}, r\langle X\rangle; \Gamma; \Omega$

i) $\mathcal{H}$ altered.

$\boxed{\text{F3}^{(\text{D})}}$, $\boxed{\text{F4}^{(\text{D})}}$, $\boxed{\text{F5}^{(\text{D})}}$, $\boxed{\text{F8}^{(\text{D})}}$, $\boxed{\text{F10}^{(\text{D})}}$, $\boxed{\text{F12}^{(\text{D})}}$, $\boxed{\text{F13}^{(\text{D})}}$: These are preserved by same reasoning as case $\boxed{\text{E16A}^{(\text{D})}}$.

$\boxed{\text{F9}^{(\text{D})}}$: Weakened.

$\boxed{\text{F11}^{(\text{D})}}$: Weakened.

ii) Inversion of $\boxed{\text{T19}^{(\text{D})}}$ and $\boxed{\text{T22}^{(\text{D})}}$ on the assumed typing judgment gives us $(l : r_e\ \tau_e) \in \mathrm{P}$ and $r_e \in dom(\mathcal{H}) \uplus \{r\}$, which is exactly the information we need to apply $\boxed{\text{T22}^{(\text{D})}}$ and conclude the desired typing judgment.

Case $\boxed{\text{E16C}^{(\text{D})}}$: $(\mathcal{H}, r^{\circ}\langle x[F], X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{explore}\ x.f@r_{new} : r_e\ \tau_e) \dashv \mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_{new}, F], X\rangle, \dot{r}_{new}\langle\rangle; \Gamma; \Omega \uplus \{r_{new}\})$

$\wedge\ ((d, h, s, \rho, \Omega \uplus \{r_{new}\}, l; \texttt{explore}\ x.f@\rho_{new}) \xrightarrow{\text{eval}} (d, h, s, \rho[r_{old} \mapsto r_{new}], \Omega \uplus \{r_{new}\}, l))$

$\implies (\mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_{new}, F], X\rangle, \dot{r}_{new}\langle\rangle; \Gamma; \Omega; l : r_e\ \tau_e \vdash (d, h, s, \rho[r_{old} \mapsto r_{new}], l : r_e\ \tau_e) \dashv \mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_{new}, F], X\rangle, \dot{r}_{new}\langle\rangle; \Gamma; \Omega \uplus \{r_{new}\})$

where $r_{old} = \rho \restriction_r (h \restriction_v (s(x)).f)$.

i) $\mathcal{H}, \Omega, \mathrm{P}, \rho$ altered. Before proceeding, we briefly note that $r_{new} \notin dom(\mathcal{H})$, as this would violate $\boxed{\text{F5}^{(\text{D})}}$ in the original configuration.

$\boxed{\text{F3}^{(\text{D})}}$: G_{total} is unaffected by the update $r_{old} \mapsto r_{new}$ as it is just a renaming. G_S is unaffected by that renaming as well because it occurred on an untracked region, making $x.f$ a **semi-tracked** reference in the original configuration. We note that in the new configuration, the only new **semi-tracked** references are those originating at ρ_{new}, which were reachable from $x.f$ in the original configuration (up to renaming), so any reference semi-tracked-reachable in the new configuration was also semi-tracked-reachable in the old configuration. This implies the new G_S is just a subgraph of the old G_S, and since any subgraph of a forest is a forest, $\boxed{\text{F3}^{(\text{D})}}$ is preserved.

F4$^{(\text{D})}$, F8$^{(\text{D})}$: The *live-set* is constant under injective renaming of regions, as is done here with $r_{old} \mapsto r_{new}$. Even in the old configuration, r_{old} was reachable from r, so the set of regions reachable from a tracked region is unchanged by adding r_{old} (or r_{new}) to the set of tracked regions. Since no locations change their regionality, except $r_{old} \mapsto r_{new}$, the *live-set* is constant and these invariants are preserved.

F5$^{(\text{D})}$: Since r_{new} is the only new tracked region, and anything reachable from r_{new} was already reachable from r, the only possibly violations of F5$^{(\text{D})}$ on the new configuration are χ from which r_{new} is reachable. But if such a χ were reachable from a tracked region besides r in the new configuration then it was also reachable from that region in the original contradiction, and as r_{new} itself was untracked this would produce an in-degree 2 contradiction via F3$^{(\text{D})}$ and F5$^{(\text{D})}$ in the old configuration. But $r \hookrightarrow \chi \hookrightarrow \rho_{new}$ implies $\chi = x.f$ by F3$^{(\text{D})}$ on the old configuration, and $x.f$ is tracked in the new configuration, so we can conclude there is no way that F5$^{(\text{D})}$ can be violated from this update.

F7$^{(\text{D})}$: The regionality equivalence relation on locations is not affected by this update by the freshness of r_{new}, so this invariant is unaffected.

F9$^{(\text{D})}$, F11$^{(\text{D})}$: These depend only on *reg-vars*$(\mathcal{H})$ and *vars*$(\mathcal{H})$, which are not affected by this update.

F10$^{(\text{D})}$: Since r_{new} is fresh it cannot be contained in Γ by F13$^{(\text{D})}$ on the old configuration, so this invariant is not affected by the update.

F12$^{(\text{D})}$: The definition of r_{old} and ρ-update $r_{old} \mapsto r_{new}$ trivially guarantees preservation of this invariant after adding $x.f@(r \rightarrowtail r_{new})$ to *reg-refs*$(\mathcal{H})$.

F13$^{(\text{D})}$: The only new region, r_{new}, to appear in any context is also added to Ω, preserving this invariant.

F14$^{(\text{D})}$: Inversion of T17$^{(\text{D})}$ and T22$^{(\text{D})}$, followed by observation of F14$^{(\text{D})}$ on the old configuration, gives us $\rho(l) = (r_e, \tau_e)$ and $r_e \in dom(\mathcal{H})$. The latter allows us to conclude $r_e \neq r_{old}$, so $\rho[r_{old} \mapsto r_{new}](l) = (r_e, \tau_e)$ as well and this invariant holds for the new configuration.

ii) $r_e \in dom(\mathcal{H})$ as noted immediately above, is exactly the information we need to apply T22$^{(\text{D})}$ and conclude the desired typing judgment.

Case E16D$^{(\text{D})}$: $(\mathcal{H}, r^{\circ}\langle x[f \rightarrowtail r_{old}, F], X\rangle, r^{\circ_{old}}_{old}\langle\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{retract } x.f : r_e\ \tau_e) \dashv \mathcal{H}, r^{\circ}\langle x[F], X\rangle; \Gamma; \Omega)$

$\wedge\ ((d, h, s, \rho, l; \texttt{retract } x.f) \xrightarrow{\text{eval}} (d, h, s, \rho, l))$

$\implies\ (\mathcal{H}, r^{\circ}\langle x[F], X\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l : r_e\ \tau_e) \dashv \mathcal{H}, r^{\circ}\langle x[F], X\rangle; \Gamma; \Omega)$

i) $\mathcal{H}$ altered.

F3$^{(\text{D})}$: $G_{total}(\rho, h)$ is clearly not affected, and the set of **semi-tracked**references loses all references originating at r_{old} and gains $x.f$, which adds exactly two nodes to G_S, one as a root unifying the previously disjoint trees rooted at each of r_{old}'s **semi-tracked**refs, and the other with a single outgoing edge targeting that root and no incoming edges. This update preserves tree structure, preserving F3$^{(\text{D})}$.

F4$^{(\text{D})}$, F8$^{(\text{D})}$: The set of regions reachable from a tracked region does not change and no location regionalities change, so the *live-set* is constant and these invariants are preserved.

F5$^{(\text{D})}$: Any tuple $r_{src} \hookrightarrow \chi \hookrightarrow r_{dest}$ with r_{src}, r_{dest} tracked and χ untracked (equiv. $\chi \notin dom(\mathcal{H}) \cup loc\text{-}refs(\mathcal{H})$) in the new configuration would have also been such a tuple in the old configuration because no regions are tracked in the new configuration that were not in the old, and the only χ untracked in the new but not the old are $x.f$ and r_{old} themselves, from which no tracked regions are reachable by F5$^{(\text{D})}$ on the old configuration combined with the observation that all outgoing refs from r_{old} are untracked. Thus we can conclude that $\boxed{\text{F5}^{(\mathrm{D})}}$ on the old configuration implies $\boxed{\text{F5}^{(\mathrm{D})}}$ on the new configuration.

$\boxed{\text{F9}^{(\mathrm{D})}}$, $\boxed{\text{F11}^{(\mathrm{D})}}$: These depend only on *reg-vars*$(\mathcal{H})$ and *vars*$(\mathcal{H})$, which are not affected by this update.

$\boxed{\text{F10}^{(\mathrm{D})}}$, $\boxed{\text{F12}^{(\mathrm{D})}}$: : Both weakened since the sets of tracked regions ($dom(\mathcal{H})$) and tracked references (*reg-refs*$(\mathcal{H})$) only decreases under this update.

$\boxed{\text{F13}^{(\mathrm{D})}}$: No new regions appear in any static context, so this invariant is trivially preserved.

ii) Inversion of $\boxed{\text{T18}^{(\mathrm{D})}}$ and $\boxed{\text{T22}^{(\mathrm{D})}}$ on the assumed typing judgment gives us $(l : r_e\ \tau_e) \in \mathrm{P}$ and $r_e \in dom(\mathcal{H}) \uplus \{r, r_{old}\}$ and $r_e \neq r_{old}$. The last two statements can be simplified to $r_e \in dom(\mathcal{H}) \uplus \{r\}$, which is exactly the additional information we need to apply $\boxed{\text{T22}^{(\mathrm{D})}}$ and conclude the desired typing judgment.

Case $\boxed{\text{E16E}^{(\mathrm{D})}}$: $(\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle, r_1^{\cdot}\langle X_1\rangle, r_2^{\circ_2}\langle X_2\rangle; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{attach}\ \{l_1\}\ \texttt{to}\ \{l_2\} : r_e\ \tau_e) \dashv \mathcal{H}[r_1 \mapsto r_2], r_e^{\circ_e}\langle X_e[r_1 \mapsto r_2]\rangle, r_2^{\circ_2}\langle (X_1, X_2)[r_1 \mapsto r_2]\rangle; \Gamma[r_1 \mapsto r_2]; \Omega)$
$\wedge\ ((d, h, s, \rho, \Omega, l; \texttt{attach}\ \{l_1\}\ \texttt{to}\ \{l_2\}) \xrightarrow{\text{eval}} (d, h, s, \rho[r_1 \mapsto r_2], l))$
$\implies\ (\mathcal{H}[r_1 \mapsto r_2], r_e^{\circ_e}\langle X_e[r_1 \mapsto r_2]\rangle, r_2^{\circ_2}\langle (X_1, X_2)[r_1 \mapsto r_2]\rangle); \Gamma; \Omega; l : r_e\ \tau_e \vdash (d, h, s, \rho[r_1 \mapsto r_2], l : r_e\ \tau_e) \dashv \mathcal{H}[r_1 \mapsto r_2], r_e^{\circ_e}\langle X_e[r_1 \mapsto r_2]\rangle, r_2^{\circ_2}\langle (X_1, X_2)[r_1 \mapsto r_2]\rangle; \Gamma[r_1 \mapsto r_2]; \Omega)$

In giving the form of the goal above, we assume $r_1 = \rho \restriction_r (l_1)$ and $r_2 = \rho \restriction_r (l_2)$. This follows from $r_1 = \mathrm{P} \restriction_r (l_1)$ and $r_2 = \mathrm{P} \restriction_r (l_2)$ by $\boxed{\text{F14}^{(\mathrm{D})}}$ on the original configuration. Inversion of $\boxed{\text{T20R}^{(\mathrm{D})}}$ on the assumed typing judgment directly yields the former, and a further inversion of $\boxed{\text{T22}^{(\mathrm{D})}}$ yields the latter.

i) $\mathcal{H}, \Gamma, \rho$ altered.

$\boxed{\text{F3}^{(\mathrm{D})}}$: We explicitly note the property that has been useful several times already: any references or regions that are tracked in $\mathcal{H}$ do not appear in G_S. We note that the only nodes to be modified under this update are tracked, and so G_S will not be affected. Slightly more formally, the set of **semi-tracked**references is not affected by this update, nor is the set of χ reachable from a **semi-tracked**reference.

$\boxed{\text{F4}^{(\mathrm{D})}}$, $\boxed{\text{F8}^{(\mathrm{D})}}$: Trivially, the *live-set* does not change so these invariants are preserved.

$\boxed{\text{F5}^{(\mathrm{D})}}$: Any 3-tuple $r_{src} \hookrightarrow \chi \hookrightarrow r_{dest}$ with r_{src}, r_{dest} tracked and χ untracked in the new configuration can be mapped to such a tuple in the old configuration by replacing r_2 with r_1 or r_2 as appropriate if it occurs as either r_{src} or r_{dest}. Thus $\boxed{\text{F5}^{(\mathrm{D})}}$ for the old configuration implies $\boxed{\text{F5}^{(\mathrm{D})}}$ for the new configuration.

$\boxed{\text{F7}^{(\mathrm{D})}}$: Since the regionality equivalence relation on locations only becomes coarser as a result of this update, this invariant is weakened.

$\boxed{\text{F9}^{(\mathrm{D})}}$, **rrF10,** $\boxed{\text{F12}^{(\mathrm{D})}}$: The same renaming, $r_1 \mapsto r_2$ is applied symmetrically to each of $\mathcal{H}, \Gamma$ and ρ, so these invariants, that treat the regionality of individual objects independently, are preserved.

$\boxed{\text{F11}^{(\mathrm{D})}}$: This depends only on *vars*$(\mathcal{H})$, which is not affected by this update, so this invariant is trivially preserved.

$\boxed{\text{F13}^{(\mathrm{D})}}$: No new region names are introduced, so this invariant is trivially preserved.

$\boxed{\text{F14}^{(\mathrm{D})}}$: Inversion of $\boxed{\text{T19R}^{(\mathrm{D})}}$ on the assumed typing judgment followed by observation of $\boxed{\text{F14}^{(\mathrm{D})}}$ on the original configuration gives us $(l : r_e\ \tau_e) \in \rho$ and $r_e \neq r_1$, which we can combine to conclude $(l : r_e\ \tau_e) \in \rho[r_1 \mapsto r_2]$, establishing that this invariant holds on the new configuration.

ii) $\boxed{\text{T22}^{(\mathrm{D})}}$ can be applied directly to conclude the desired typing judgment.

Case E16F$^{(\text{D})}$**:** $(\mathcal{H};\Gamma, x : r\ \tau;\Omega;\mathrm{P} \vdash (d, h, s, \rho, l;\texttt{drop-var}\ x : r_e\ \tau_e) \dashv \mathcal{H};\Gamma;\Omega)$

$\wedge\ ((d, h, s, \rho, \Omega, l;\texttt{drop-var}\ x)\xrightarrow{\text{eval}}(d, h, s, \rho, \Omega, l))$

$\implies\ (\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, l : r_e\ \tau_e) \dashv \mathcal{H};\Gamma;\Omega$

i) Γ altered.

F9$^{(\text{D})}$**:** Inverting T25$^{(\text{D})}$ tells us $x \notin \mathit{vars}(\mathcal{H})$, so this invariant is not affected.

F10$^{(\text{D})}$**:** This invariant is only weakened by the removal of x from $\mathit{dom}(\Gamma)$.

F13$^{(\text{D})}$**:** No new regions are added to a context, so this invariant is trivially preserved.

ii) Inverting T25$^{(\text{D})}$ gives us $\mathcal{H};\Gamma, x : r\ \tau;\Omega;\mathrm{P} \vdash l : r_e\ \tau_e \dashv \mathcal{H};\Gamma, x : r\ \tau;\Omega$, and noting that T22$^{(\text{D})}$'s derivation does not rely on Γ, we are free to use the same derivation to conclude the desired typing judgment on l.

Case E16G$^{(\text{D})}$**:** $(\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle, r^{\circ}\langle X\rangle;\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, l;\texttt{drop-reg}\ \{l_d\} : r_e\ \tau_e) \dashv \mathcal{H}, r_e^{\circ_e}\langle X_e\rangle;\Gamma;\Omega)$

$\wedge\ ((d, h, s, \rho, \Omega, l;\texttt{drop-reg}\ \{l_d\})\xrightarrow{\text{eval}}(d, h, s, \rho, \Omega, l))$

$\implies\ (\mathcal{H}, r_e^{\circ_e}\langle X_e\rangle;\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, l : r_e\ \tau_e) \dashv \mathcal{H}, r_e^{\circ_e}\langle X_e\rangle;\Gamma;\Omega)$

i) $\mathcal{H}$ altered.

F3$^{(\text{D})}$**:** G_{total} is not affected by this update. r could not have been reachable from a **semi-tracked**reference in the old configuration, so the effect of this update on G_S is exactly to remove every tree rooted at a **semi-tracked**reference originating from r, which preserves forestry and thus preserves this invariant.

F4$^{(\text{D})}$, F8$^{(\text{D})}$**:** The set of regions reachable from a tracked region strictly shrinks under this update and no locations change their regionality, so the *live-set* strictly shrinks and both of these invariants are strictly weakened.

F5$^{(\text{D})}$**:** The only possible 3-tuples $r_{src} \hookrightarrow \chi \hookrightarrow \rho_{dest}$ with r_{src}, r_{dest} tracked and χ untracked in the new configuration that were not also such a tuple in the old configuration set χ equal to r or a reference originating at r, as these are the only χ dropped from $\mathcal{H}$ when updating from the old to the new configuration. However such a tuple cannot exist, as any reference targeting r is either tracked, making it **invalid**, or untracked, invalidating F5$^{(\text{D})}$ in the original configuration, so necessarily r and all references originating at r are not reachable from any tracked region r_{src} and cannot be values of χ in a violating 3-tuple, so none can exist, and this invariant holds for the new configuration.

F9$^{(\text{D})}$, F10$^{(\text{D})}$, F11$^{(\text{D})}$, F12$^{(\text{D})}$, F13$^{(\text{D})}$**:** These invariants are all only weakened by the dropping of r from $\mathcal{H}$.

ii) Inverting T26R$^{(\text{D})}$ and then T22$^{(\text{D})}$ gives us $(l;r_e\ \tau_e) \in \mathrm{P}$, which is exactly the information we need to apply T22$^{(\text{D})}$ and conclude the desired typing judgment.

Case E16H$^{(\text{D})}$**:** $(\mathcal{H}, r_e^{\circ}\langle X_e\rangle, r_1^{\cdot}\langle X_1\rangle, r_2^{\cdot}\langle X_2\rangle;\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, l;\texttt{swap}\ l_1\ \texttt{with}\ l_2 : r_e\ \tau_e) \dashv \mathcal{H}[r_1 \mapsto r_2, r_2 \mapsto r_1], r_e^{\circ}\langle X_e[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_2^{\cdot}\langle X_1[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle;\Gamma[r_1 \mapsto r_2, r_2 \mapsto r_1];\Omega)$

$\wedge\ ((d, h, s, \rho, \Omega, \texttt{swap}\ l_1\ \texttt{with}\ l_2)\xrightarrow{\text{eval}}(d, h, s, \rho[r_1 \mapsto r_2, r_2 \mapsto r_1], \Omega, l))$

$\implies\ (\mathcal{H}[r_1 \mapsto r_2, r_2 \mapsto r_1], r_e^{\circ}\langle X_e[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_2^{\cdot}\langle X_1[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle;\Gamma[r_1 \mapsto r_2, r_2 \mapsto r_1];\Omega;\mathrm{P}[r_1 \mapsto r_2, r_2 \mapsto r_1] \vdash (d, h, s, \rho[r_1 \mapsto r_2, r_2 \mapsto r_1], l : r_e\ \tau_e) \dashv \mathcal{H}[r_1 \mapsto r_2, r_2 \mapsto r_1], r_e^{\circ}\langle X_e[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_1^{\cdot}\langle X_2[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle, r_2^{\cdot}\langle X_1[r_1 \mapsto r_2, r_2 \mapsto r_1]\rangle;\Gamma[r_1 \mapsto r_2, r_2 \mapsto r_1];\Omega)$

i) $\mathcal{H}, \Gamma, \mathrm{P}, \rho$ altered, but all just by a bijective region renaming within the existing Ω, which cannot invalidate the derivation of any invariant $\boxed{\text{F3}^{(\text{D})}}$ - $\boxed{\text{F14}^{(\text{D})}}$.

ii) Inversion of $\boxed{\text{T21R}^{(\text{D})}}$ gives us $(l : r_e\ \tau_e) \in \mathrm{P}, r_e \neq r_1$, and $r_e \neq r_2$, which can be combined to establish $(l : r_e\ \tau_e) \in \mathrm{P}[r_1 \mapsto r_2, r_2 \mapsto r_1]$, which is exactly the additional information needed to apply $\boxed{\text{T22}^{(\text{D})}}$ and conclude the desired typing judgment.

Case $\boxed{\text{E16I}^{(\text{D})}}$**:** $(\mathcal{H}; \Gamma, x : r\ \tau; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l; \texttt{invalidate-var}\ x : r_{out}\ \tau_{out}) \dashv \mathcal{H}; \Gamma, x : \bot\ \tau; \Omega)$

$\implies (\mathcal{H}; \Gamma, x : \bot\ \tau; \Omega; \mathrm{P} \vdash (d, h, s, \rho, l : r_{out}\ \tau_{out}) \dashv \mathcal{H}; \Gamma, x : \bot\ \tau; \Omega)$

i) Γ differs.

$\boxed{\text{F9}^{(\text{D})}}$**:**

$\boxed{\text{F10}^{(\text{D})}}$**:**

$\boxed{\text{F13}^{(\text{D})}}$**:**

ii) Inversion of $\boxed{\text{T28}^{(\text{D})}}$, and $\boxed{\text{T22}^{(\text{D})}}$, allows us to conclude $(l : rt) \in \mathrm{P}$ and $r \in dom(\mathcal{H})$, which allows us to re-apply $\boxed{\text{T22}^{(\text{D})}}$ on the new configuration and obtain exactly the desired typing judgment.

Case $\boxed{\text{E17}^{(\text{D})}}$**:** This is just a special case of $\boxed{\text{E3}^{(\text{D})}}$; the updates to all dynamic contexts here satisfy $\texttt{extracts-fresh-heap}(\rho, r, \tau; (l \mapsto (r, \tau)), (l \mapsto (\tau, v_f)), l)$ so we can dispatch to preservation for $\boxed{\text{E3}^{(\text{D})}}$. TODO: (same as $\boxed{\text{E5}^{(\text{D})}}$) is this formal enough?

Case $\boxed{\text{E18A}^{(\text{D})}}$**:** We begin with the special case in which $E^*[\,] = [\,]$ - i.e. we are dealing with a base expression: $(\mathcal{H}, r^\circ\langle X\rangle; \Gamma, x : r\ \tau; \Omega; \mathrm{P} \vdash (d, h, s, \rho, \texttt{detach}\ x\ \texttt{in}\ \{e_>\}\ \texttt{else}\ \{e_{fail}\} : r_{out}\ \tau_{out}) \dashv \mathcal{H}'; \Gamma'; \Omega_> \cup \Omega_{fail})$

$\wedge\ ((d, h, s, \rho, \Omega, \texttt{detach}\ x\ \texttt{in}\ \{e_>\}\ \texttt{else}\ \{e_{fail}\}) \xrightarrow{\text{eval}} (d, h, s, \rho, \Omega, e_>))$

$\implies (\mathcal{H}, r^\circ\langle X\rangle, r^{\cdot}_{new}\langle\rangle$

Case $\boxed{\text{E18B}^{(\text{D})}}$**:**

□

5.5 Full Progress and Preservation

We note that in both sections 5.3 and 5.4 above, we ignored expressions of the form $e = E[\bar{e}]$ for $\bar{e} \notin LocationNames$. To show Progress and Preservation for the complete decorated Gallifrey type system, Theorems 4.1 and 4.2 require an induction on the structure of expressions that we glean from lemma 5.1. Lemmas 5.3 and 5.4 will be our base cases for that induction, and we will proceed to use them to prove the entire induction:

Theorem 4.1 (Progress (Restated from Section 4.8.1)). *For any well-typed configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ where $e \notin LocationNames$ is a non-blocking expression, there exists another dynamic configuration (d', h', s', ρ'), static Ω'', and expression e' such that $(d, h, s, \rho, \Omega', e) \xrightarrow{eval} (d', h', s', \rho', \Omega'', e')$.*

Proof of Theorem 4.1. Let $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ be a well-typed configuration. If there exists no $E[\,], \bar{e}$ such that $e = E[\bar{e}]$ and $\bar{e} \notin LocationNames$, or if e is *detaching*, or if e is an isolated field reference of the form $x.f$, or isolated assignment of the form $x.f = l$, then apply Lemma 5.3 and we are done. Otherwise, perform induction on the structure of the evaluation contexts, and consider the general inductive case $E[\bar{e}]$. We apply lemma 5.21 to obtain the well-typed configuration $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \bar{e} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}';\bar{\Gamma}';\bar{\Omega}'$, with $\bar{\Omega}' \subseteq \Omega'$. The inductive hypothesis of Progress now gives us the step $(d, h, s, \rho, \bar{\Omega}', \bar{e}) \xrightarrow{\text{eval}} (d', h', s', \rho', \bar{\Omega}'', \bar{e}')$. We can apply lemma 5.10 to conclude that, for some Ω'', $(d, h, s, \rho, \Omega', \bar{e}) \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'', \bar{e}')$. We can now conclude by application of $\boxed{\text{E1A}^{(\text{D})}}$ or $\boxed{\text{E1B}^{(\text{D})}}$: $(d, h, s, \rho, \Omega', e) \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'', E[\bar{e}'])$. □

Theorem 4.2 (Preservation (Restated from Section 4.8.1)). *For any well-typed configuration* $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}_{out}, \Gamma_{out}, \Omega_{out}$ *that steps with the relation* $(d, h, s, \rho, \Omega_{out}, e) \xrightarrow{eval} (d', h', s', \rho', \Omega'_{out}, e')$*, there exists* $\mathcal{H}', \Gamma', \Omega', P'$ *such that the configuration* $\mathcal{H}', \Gamma', \Omega', P' \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}_{out}, \Gamma_{out}, \Omega'_{out}$ *is also well-typed.*

Proof of Theorem 4.2. Let $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}_{out};\Gamma_{out};\Omega_{out}$ be a well-typed configuration, and $(d, h, s, \rho, \Omega_{out}, e) \xrightarrow{\text{eval}} (d', h', s', \rho', \Omega'_{out}, e')$ be a step. If there exists no $E[], \bar{e}$ such that $e = E[\bar{e}]$ and $\bar{e} \notin LocationNames$, or if e is *detaching*, or if e is an isolated field reference of the form $x.f$, or isolated assignment of the form $x.f = l$, then apply Lemma 5.4 and we are done. Otherwise, perform induction over the structure of evaluation contexts. Consider the general inductive case $e = E[\bar{e}]$. Similarly to the application of lemma 5.21, we can note that all of the typing rules that could have derived the well-typedness of $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}_{out};\Gamma_{out};\Omega_{out}$ take as a premise $\mathcal{H}_p;\Gamma;\Omega;\mathrm{P} \vdash \bar{e} : \bar{r}\ \bar{\tau} \dashv \bar{\mathcal{H}}_{out};\bar{\Gamma}_{out};\bar{\Omega}_{out}$, where $\mathcal{H}_p$ is identical to $\mathcal{H}$ except 0, 1, or 2 regions that were tracked are forced to be pinned, and $\bar{\Omega}_{out} \subseteq \Omega_{out}$. Noting as we did in the proof of lemma 5.21 that proof trees for $\boxed{\text{F2}^{(\text{D})}}$ do not observe pinnedness, we can strengthen our observation with $d, h, s, \rho : \mathcal{H}_p;\Gamma;\Omega;\mathrm{P}$ `agree` to obtain $\mathcal{H}_p;\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \bar{e} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}_{out};\bar{\Gamma}_{out};\bar{\Omega}_{out}$.

We now wish to apply the same typing rule for $E[]$ that we inverted above to conclude that well-typedness of the inner expression $\bar{e}'$ after stepping lifts to well-typedness of the outer expression $E[\bar{e}']$ after stepping. In some cases, such as $\boxed{\text{T3}^{(\text{D})}}$ or $\boxed{\text{T4}^{(\text{D})}}$, this is trivial, however in others we note that there are additional predicates in the premises of the typing rule describing conditions on the the contexts used to typecheck $\bar{e}$. It is not immediately clear that these predicates still hold over the contexts used to typecheck $\bar{e}'$. We claim that there are exactly 3 ways by which the premises of the typing rule used to typecheck $E[\bar{e}]$, the rule inverted above to obtain a typing for $\bar{e}$, depends on the contexts used to typecheck its inner expression. We exclude dependence on anything that does not differ over stepping the inner expression, such as $\bar{r}, \bar{\tau}, \bar{\mathcal{H}}_{out}, \bar{\Gamma}_{out}$, which leaves exactly the following dependences:

i) There exists some set Ω_{sep} such that the outer expression will typecheck only if the output context $\bar{\Omega}'$ used to typecheck the inner expression is disjoint from Ω_{sep}, and the outer expression will typecheck with output context equal to Ω_{sep} appended as a disjoint union to the inner context's output $\bar{\Omega}'$.

ii) There exist 0, 1, or 2 location bindings $l_i : r_i\ \tau_i$ such that the outer expression will typecheck only if the context P' used to typecheck the inner expression contains those bindings, and:

iii) For the each of the r_i above, the input $\mathcal{H}_p$ used to typecheck the inner expression tracks and pins the region r_i.

In fact, in obtaining the original well-typedness judgment $\mathcal{H}_p;\Gamma;\Omega;\mathrm{P} \vdash \bar{e} : \bar{r}\ \bar{\tau} \dashv \bar{\mathcal{H}}_{out};\bar{\Gamma}_{out};\bar{\Omega}_{out}$ on the inner expression, values of Ω_{sep} and the bindings $l_i : r_i\ \tau_i$ are fixed by the proof tree. We will now point out their exact methods of extraction, and argue that they have the properties mentioned above:

i) Ω_{sep} exists and will be exactly the difference $\Omega_{extra} = \bar{\Omega}_{out} - \Omega_{out}$ determined through inversion above. Lemma 5.20 tells us that it has the desired dependence properties.

ii) The bindings $l_i : r_i\ \tau_i$ will be exactly the set of locations that syntactically appear in $E[]$ (which does not include any locations in the inner expression), and their bindings in the pre-step P. For example if $E[] = l_{left} \oplus_r []$, then

l_{left} is the unique location l_i the rule depends on. Inspection of all rules of the form $\boxed{\mathrm{T[X]}_{\mathrm{M}}{}^{(\mathrm{D})}}$ or $\boxed{\mathrm{T[X]}_{\mathrm{R}}{}^{(\mathrm{D})}}$ will show that this observation holds for all relevant typing rules. Since each region r_i was necessarily pinned and tracked in the original configuration, and because inversion of $\boxed{\mathrm{F14}^{(\mathrm{D})}}$ allows us to conclude that all bindings $l_i : r_i\ \tau_i$ exist in ρ, we can apply lemma 5.16 to conclude that they also exist in ρ'. $\boxed{\mathrm{F14}^{(\mathrm{D})}}$ inverted on the post-step configuration tells us that P' is subsumed by ρ', so if any of the locations l_i are bound in P' then they must already have the correct bindings $l_i : r_i\ \tau_i$. Any required bindings that do not exist may also be added to form P'' that satisfies property ii) above and allows application of $\boxed{\mathrm{F14}^{(\mathrm{D})}}$ and $\boxed{\mathrm{F2}^{(\mathrm{D})}}$ to argue $d', h', s', \rho' : \mathcal{H}'_p; \Gamma'; \Omega'; \mathrm{P}''$ `agree`.

iii) We note that in the original configuration, $\bar{\mathcal{H}}_{out}$ always pins and tracks each r_i, and since $\bar{\mathcal{H}}_{out}$ is also present as an output context in the new configuration $\mathcal{H}'_p; \Gamma'; \Omega'; \mathrm{P}'' \vdash (d', h', s', \rho', \bar{e}' : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}_{out}; \bar{\Gamma}_{out}; \bar{\Omega}_{out} \uplus \Omega_{new}$, we can conclude by lemma 5.13 that each r_i is also pinned in $\mathcal{H}'_p$, and is pinned by lemma 5.14.

We have show that that this new inner configuration preserves all of the properties that our necessary typing rule depends on, so we apply it to conclude that $\mathcal{H}'; \Gamma'; \Omega'; \mathrm{P}'' \vdash (d', h', s', \rho', e' : r\ \tau) \dashv \mathcal{H}_{out}; \Gamma_{out}; \Omega'_{out}$, completing our proof of Preservation. □

5 PROOF OF PROJECTION

We have now seen two versions of the system with different desirable properties. The undecorated system, presented in section 2, built up a powerful region system for tracking reservation safety at compile system, but gauranteed its erasbility at runtime. Unfortunately, too much was erased for statements of undecorated Progress and Preservation, Theorems 2.1, 2.2, to be naturally provable. The decorated system, section 4, remedied this by introducing syntactic annotations that communicated region information to the dynamic semantics. With this information we were able to prove decorated Progress and Preservation, Theormes 4.1 and 4.2. Now, all that remains is to show that these proofs suffice to conclude undecorated progress and preservation as well. We begin with some notation, and a restatement of key Theorems and Lemmas.

5.1 Notation

We note that the expression grammars in the decorated and undecorated systems totally coincide except that the decorated grammar introduces ρ as a regionality map with the same language as P, and the decorated system includes annotations of the form "@r", "@Ω" or "$\oplus_r$" on some expressions. We henceforth let $\mathfrak{e}$ denote a decorated expression, and e denote an undecorated expression. We let STRIP be a function that takes a decorated expression $\mathfrak{e}$ and returns an undecorated expression e identical except for the loss of all annotations. This function also naturally extends to stripping the annotations from decorated evaluation contexts, which we will denote by $\mathfrak{E}[]$, to obtain undecorated evaluation contexts $E[]$.

5.2 Restatement of Soundness Theorems

Our highest level goal is to prove progress and preservation for the undecorated system, given its proof in the decorated system. To this end we restate all 4 theorems here:

THEOREM 4.1 (DECORATED PROGRESS (RESTATED FROM SECTION 4.8.1)). *For any well-typed decorated configuration $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ where $\mathfrak{e} \notin$ LocationNames is a non-blocking decorated expression, there exists* *another decorated dynamic configuration* (d', h', s', ρ')*, static* Ω''*, and decorated expression* $\mathfrak{e}'$ *such that* $(d, h, s, \rho, \Omega', \mathfrak{e}) \xrightarrow{eval} (d', h', s', \rho', \Omega'', \mathfrak{e}')$.

Theorem 4.2 (Decorated Preservation (Restated from Section 4.8.1)). *For any well-typed decorated configuration* $\mathcal{H}, \Gamma, \Omega, P \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega'$ *that steps with the relation* $(d, h, s, \rho, \Omega', \mathfrak{e}) \xrightarrow{eval} (d', h', s', \rho', \Omega'', \mathfrak{e}')$*, there exists some* $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{P}$ *such that the configuration* $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{P} \vdash (d', h', s', \rho', \mathfrak{e}' : r\ \tau) \dashv \mathcal{H}', \Gamma', \Omega''$ *is also well-typed.*

Theorem 2.1 (Undecorated Progress (Restated from Section 2.8)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$ *where* $e \notin$ *LocationNames is a non-blocking undecorated expression, there exists another undecorated dynamic configuration* (d', h', s', e') *such that* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$.

Theorem 2.2 (Undecorated Preservation (Restated from Section 2.8)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$ *that steps with the relation* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$*, the configuration* $\vdash (d', h', s', e' : r\ \tau)$ *is also well-typed.*

4.3 Restatement of Key Lemmas

We claim that that implication of undecorated Progress and Preservation by their decorated counterparts reduces to the following lemmas that were first stated in section 4.8.2:

Lemma 4.3 (Configuration Lifting (Restated from Section 4.8.2)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$*, there exist* $\mathcal{H}, \Gamma, \Omega, P, \mathcal{H}', \Gamma', \Omega', \mathfrak{e}$ *such that* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ *is a well-typed decorated configuration and* $\textsc{strip}(\mathfrak{e}) = e$.

Lemma 4.4 (Configuration Projection (Restated from Section 4.8.2)). *For any well-typed decorated configuration* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega',\vdash (d, h, s, e : r\ \tau)$ *is a well-typed undecorated configuration, where* $e = \textsc{strip}(\mathfrak{e})$.

Lemma 4.5 (Step Lifting (Restated from Section 4.8.2)). *For any undecorated step* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$*, if* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ *is a well-typed decorated configuration and* $\textsc{strip}(\mathfrak{e}) = e$*, then there exist* $P', \Omega'', \mathfrak{e}'$ *such that* $(d, h, s, P, \Omega', \mathfrak{e}) \xrightarrow{eval} (d', h', s', P', \Omega'', \mathfrak{e}')$ *is a decorated step and* $\textsc{strip}(\mathfrak{e}') = e'$.

Lemma 4.6 (Step Projection (Restated from Section 4.8.2)). *For any decorated step* $(d, h, s, \rho, \Omega, \mathfrak{e}) \xrightarrow{eval} (d', h', s', \rho', \Omega', \mathfrak{e}')$*,* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$ *is an undecorated step, where* $e = \textsc{strip}(\mathfrak{e})$ *and* $e' = \textsc{strip}(\mathfrak{e})$.

4.4 Proof that Key Lemmas Suffice

We now give proofs of our claim that these 4 lemmas suffice to project Progress and Preservation from the decorated to undecorated system.

Theorem 2.1 (Undecorated Progress (Restated)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$ *where* $e \notin$ *LocationNames is a non-blocking undecorated expression, there exists another undecorated dynamic configuration* (d', h', s', e') *such that* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$.

Proof of Theorem 2.1 using Theorem 4.1. Let $\vdash (d, h, s, e : r\ \tau)$ be a well-typed undecorated configuration, where $e \notin$ *LocationNames* is a *non-blocking* expression. Lemma 4.3 gives us the well-typed decorated configuration $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \mathrm{P}, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$, where $\textsc{strip}(\mathfrak{e}) = e$. Observing that $\mathfrak{e} \in$ *LocationNames* $\implies$ $e \in$ *LocationNames*, and similarly that self-sufficiency projects downwards, Theorem 4.1 now gives us a decorated step $(d, h, s, \mathrm{P}, \Omega', \mathfrak{e}) \xrightarrow{\mathrm{eval}} (d', h', s', \mathrm{P}', \Omega'', \mathfrak{e}')$. We conclude by applying Lemma 4.6 to obtain the undecorated step $(d, h, s, e) \xrightarrow{\mathrm{eval}} (d', h', s', \textsc{strip}(\mathfrak{e}'))$. □

Theorem 2.2 (Undecorated Preservation (Restated)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$ *that steps with the relation* $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$*, the configuration* $\vdash (d', h', s', e' : r\ \tau)$ *is also well-typed.*

Proof of Theorem 2.2 using Theorem 4.2. Let $\vdash (d, h, s, e : r\ \tau)$ be a well-typed undecorated configuration, and let $(d, h, s, e) \xrightarrow{\text{eval}} (d', h', s', e')$ be an undecorated step. Lemma 4.3 gives us the decorated configuration $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \mathrm{P}, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$, where $\textsc{strip}(\mathfrak{e}) = e$. Lemma 4.5 then gives us the decorated step $(d, h, s, \mathrm{P}, \Omega', \mathfrak{e}) \xrightarrow{\text{eval}} (d', h', s', \mathrm{P}', \Omega'', \mathfrak{e}')$, where $\textsc{strip}(\mathfrak{e}') = e'$. Theorem 4.2 now allows us to conclude $\bar{\mathcal{H}}, \bar{\Gamma}, \bar{\Omega}, \bar{\mathrm{P}} \vdash (d', h', s', \mathrm{P}', \mathfrak{e}' : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega''$. We conclude by applying Lemma 4.4 to obtain the well-typed undecorated configuration $\vdash (d', h', s', e' : r\ \tau)$. □

5.5 Useful Side Lemmas

The following lemmas are useful in our proof of the key projection lemmas. They are provided here without proof, but were originally stated with proof in the decorated system's Progress and Preservation argument.

Lemma 5.10 (Dynamic Region Arbitrarity (Restated from Section 5.2)). *If* $(d, h, s, \rho, \Omega, e) \xrightarrow{eval} (d', h', s', \rho', \Omega', e')$*, then for any* $\bar{\Omega}$ *containing* Ω *there exists* $\bar{\Omega}'$ *such that* $(d, h, s, \rho, \bar{\Omega}, e) \xrightarrow{eval} (d', h', s', \rho', \bar{\Omega}', e')$.

Lemma 5.15 (Pinned Weakening (Restated from Section 5.2)). $r^{\dagger}\langle X\rangle, \mathcal{H}; \Gamma; \Omega; R \vdash (d, h, s, \rho, e : r\ \tau) \dashv r^{\dagger}\langle X'\rangle, \mathcal{H}'; \Gamma'; \Omega' \implies r^{\cdot}\langle X\rangle, \mathcal{H}; \Gamma; \Omega; R \vdash (d, h, s, \rho, e : r\ \tau) \dashv r^{\cdot}\langle X'\rangle, \mathcal{H}'; \Gamma'; \Omega'$

Lemma 5.21 (Inner Typing Dependence (Restated from Section 5.2)). *For any expression e of the form* $E[\bar{e}]$ *for some expression* $\bar{e} \notin$ *LocationNames, if* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, e : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ *is a well-typed configuration then there exist* $\bar{r}, \bar{\tau}, \bar{\mathcal{H}}', \bar{\Gamma}', \bar{\Omega}'$ *such that* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, \rho, \bar{e} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}'; \bar{\Gamma}'; \bar{\Omega}'$ *is also a well-typed configuration and* $\bar{\Omega}' \subseteq \Omega'$.

4.6 Proofs of Key Lemmas

Lemma 4.3 (Configuration Lifting (Restated)). *For any well-typed undecorated configuration* $\vdash (d, h, s, e : r\ \tau)$*, there exist* $\mathcal{H}, \Gamma, \Omega, P, \mathcal{H}', \Gamma', \Omega', \mathfrak{e}$ *such that* $\mathcal{H}; \Gamma; \Omega; P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega'$ *is a well-typed decorated configuration and* $\textsc{strip}(\mathfrak{e}) = e$.

Proof of Lemma 4.3. We begin with an inversion of F1 on the assumed undecorated well-typedness of the configuration $\vdash (d, h, s, e : r\ \tau)$ to obtain $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash e : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$ and $\vdash d, h, s : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ `agree`. We must translate each of these two into a corresponding judgment in the decorated system.

For the latter, we compare the invariants F1 - F13 in the undecorated system with the invariants F1$^{(\text{D})}$ - F14$^{(\text{D})}$ in the decorated system. We observe that the exact extent of difference is: the latter adds the context ρ to obtain the alternate judgment form $\vdash d, h, s, \rho : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ `agree` from F1$^{(\text{D})}$, uses ρ in place of P in all invariants F2$^{(\text{D})}$ - F13$^{(\text{D})}$, and adds F14$^{(\text{D})}$ requiring P to be a subfunction of ρ. Clearly now, $\vdash d, h, s, \mathrm{P} : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ `agree` in the decorated system is equivalent to $\vdash d, h, s : \mathcal{H}; \Gamma; \Omega; \mathrm{P}$ `agree` in the base system.

Now we need only show that $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash \mathfrak{e} : r\ \tau \dashv \mathcal{H}'; \Gamma'; \Omega'$ with $\textsc{strip}(\mathfrak{e}) = e$ and we will have sufficient information to apply F1$^{(\text{D})}$ from the decorated system and conclude our goal, $\mathcal{H}; \Gamma; \Omega; \mathrm{P} \vdash (d, h, s, \mathrm{P}, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}'; \Gamma'; \Omega')$, holds. This can easily be done by induction on the proof tree of T$^{(\text{D})}$ rules in the undecorated system deriving our undecorated typing judgment for e. All T$^{(\text{D})}$ rules are identical between the decorated and undecorated systems except for their possible application of an annotation to their judged expression and their possible observation of annotations from assumedly-well-typed subexpressions. In most cases (T10, T11, T13R, T13L, T15, T23, T24), it suffices to obtain a well-typed decorated expression by annotating the well-typed undecorated expression with its regionality. The only remaining cases are decorated T9L and T9R, in which we let the annotation $@\Omega_{out}$ be the difference between the two values of Ω in the $\mathrm{H}^{(\mathrm{D})}$ judgment form we get from inversion, and decorated $\mathrm{T17}^{(\mathrm{D})}$, in which the annotation $@r_{new}$ is exactly the region r_{new} as derived even by undecorated T17. This gives a recursive procedure to construct $\mathfrak{e}$ from e while preserving well-typedness under the same static contexts, type, and regionality, so we can conclude $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \mathfrak{e} : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'$, and we are done. □

Lemma 4.4 (Configuration Projection (Restated)). *For any well-typed decorated configuration* $\mathcal{H};\Gamma;\Omega;P \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega'$, $\vdash (d, h, s, e : r\ \tau)$ *is a well-typed undecorated configuration, where* $e = \textsc{strip}(\mathfrak{e})$.

Proof of Lemma 4.4. Similarly to 4.3, we begin with an inversion of $\mathrm{F1}^{(\mathrm{D})}$, this time on the assumed decorated well-typedness of the configuration $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \rho, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega'$. We obtain $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \mathfrak{e} : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'$ and $\vdash d, h, s, \rho : \mathcal{H};\Gamma;\Omega;\mathrm{P}$ `agree`. Obtaining $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash \textsc{strip}(\mathfrak{e}) : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'$ is a trivial induction that just observes, for every $\mathrm{T}^{(\mathrm{D})}$ rule, that if the decorated version judges well-typedness with an annotation then the undecorated version judges well-typedness with the same contexts and no annotation. Obtaining $\vdash d, h, s : \mathcal{H};\Gamma;\Omega;\rho$ follows from the observations made in the prior proof that undecorated F2 - F13 are identical to decorated $\mathrm{F2}^{(\mathrm{D})}$ - $\mathrm{F13}^{(\mathrm{D})}$ except for a global reliance on P instead of ρ, and we additionally note that in the process of inverting $\vdash d, h, s, \rho : \mathcal{H};\Gamma;\Omega;\mathrm{P}$, we obtained that P is subsumed by ρ from $\mathrm{F14}^{(\mathrm{D})}$. This allows us to weaken our derived undecorated typing judgment to $\mathcal{H};\Gamma;\Omega;\rho \vdash e : r\ \tau \dashv \mathcal{H}';\Gamma';\Omega'$, and application of undecorated F1 now allows us to conclude $\vdash (d, h, s, e : r\ \tau)$, so we are done.

□

Lemma 4.5 (Step Lifting (Restated)). *For any undecorated step* $(d, h, s, e)\xrightarrow{eval}(d', h', s', e')$, *if* $\mathcal{H};\Gamma;\Omega;P \vdash (d, h, s, P, \mathfrak{e} : r\ \tau) \dashv \mathcal{H}';\Gamma';\Omega'$ *is a well-typed decorated configuration and* $\textsc{strip}(\mathfrak{e}) = e$, *then there exist* $P', \Omega'', \mathfrak{e}'$ *such that* $(d, h, s, P, \Omega', \mathfrak{e})\xrightarrow{eval}(d', h', s', P', \Omega'', \mathfrak{e}')$ *is a decorated step and* $\textsc{strip}(\mathfrak{e}') = e'$.

Proof of Lemma 4.5. This is the most difficult of the 4 lemmas. We will prove it by induction on the proof tree for the undecorated step, splitting cases by the root $\mathrm{E}^{(\mathrm{D})}$ rule and generalizing $\mathcal{H}', \Gamma', \Omega', r, \tau$ but leaving $\mathcal{H}, \Gamma, \Omega, \mathrm{P}, d, h, s$ fixed. Some cases have the exact same premises in the decorated and undecorated semantics, so such steps trivially lift from the undecorated to decorated semantics, and are skipped in the case analysis.

$\mathrm{E1A}^{(\mathrm{D})}$, $\mathrm{E1B}^{(\mathrm{D})}$**:** This case tells us that $(d, h, s, e)\xrightarrow{\text{eval}}(d', h', s', e')$, where $e = E[\bar{e}]$ and $e' = E[\bar{e}']$, from which we can also conclude $\mathfrak{e} = \mathfrak{E}[\bar{\mathfrak{e}}]$, where $\textsc{strip}(\bar{\mathfrak{e}}) = \bar{e}$ and $\textsc{strip}(\mathfrak{E}[]) = E[]$. From the lemma 5.21, we can conclude that $\mathcal{H};\Gamma;\Omega;\mathrm{P} \vdash (d, h, s, \mathrm{P}, \bar{\mathfrak{e}} : \bar{r}\ \bar{\tau}) \dashv \bar{\mathcal{H}}';\bar{\Gamma}';\bar{\Omega}'$ for some $\bar{r}, \bar{\tau}, \bar{\mathcal{H}}', \bar{\Gamma}', \bar{\Omega}'$ with $\bar{\Omega}' \subseteq \Omega'$. We can now apply the inductive hypothesis to conclude that there exist $\mathrm{P}', \bar{\Omega}'', \bar{\mathfrak{e}}'$ such that $(d, h, s, \mathrm{P}, \bar{\Omega}', \bar{\mathfrak{e}})\xrightarrow{\text{eval}}(d', h', s', \mathrm{P}', \bar{\Omega}'', \bar{\mathfrak{e}}')$ and $\textsc{strip}(\bar{\mathfrak{e}}') = \bar{e}'$. Application of $\mathrm{E1A}^{(\mathrm{D})}$ or $\mathrm{E1B}^{(\mathrm{D})}$ now yields $(d, h, s, \mathrm{P}, \bar{\Omega}', \mathfrak{E}[\bar{\mathfrak{e}}])\xrightarrow{\text{eval}}(d, h, s, \mathrm{P}', \bar{\Omega}'', \mathfrak{E}[\bar{\mathfrak{e}}'])$. All that is left is to note $\mathfrak{E}[\bar{\mathfrak{e}}] = \mathfrak{e}$ from above, $\textsc{strip}(\mathfrak{E}[\bar{\mathfrak{e}}']) = E[\bar{e}'] = e'$ by assumption, and $\bar{\Omega}' \subseteq \Omega'$, then to apply lemma 5.10, to conclude $(d, h, s, \mathrm{P}, \Omega', e)\xrightarrow{\text{eval}}(d', h', s', \mathrm{P}', \Omega'', e)$ for some Ω''.

$\mathrm{E3}^{(\mathrm{D})}$**:** We are given the predicate `extracts-fresh-heap-regfree`$(h, \tau; h_{new}, l)$, and must show that there exists ρ_{new} such that `extracts-fresh-heap`$(\Omega, \mathrm{P}, r, \tau; \rho_{new}, h_{new}, l)$, where r is the region syntactically annotated onto `new-`τ in $\mathfrak{e}$. From the definition of `extracts-fresh-heap-regfree`, there exist $r_{root}, \rho_{old}, \rho'_{new}$ such that `extracts-fresh-heap`$(\cdot; \rho_{old}, r_{root}, \tau; \rho'_{new}, h$ and $\rho_{old} \restriction_{\tau} \equiv h \restriction_{\tau}$. We also know that $h \restriction_{\tau} \equiv \mathrm{P} \restriction_{\tau}$ from inversion of $\mathrm{F7}^{(\mathrm{D})}$ on the assumed decorated well-typed configuration, so $\rho_{old} \restriction_{t} \equiv \mathrm{P} \restriction_{\tau}$, which notably implies they have the same domain. Let Ω_{new} be a set of region names disjoint from Ω with a bijection ϕ from $range(\rho'_{new} \restriction r)$ to $\Omega_{new} \uplus \{r\}$ that sends r_{root} to r. We claim now that $\texttt{extracts-fresh-heap}(\Omega; \mathrm{P}, r, \tau; \phi(\rho'_{new}), h_{new}, l)$ holds, which would complete our proof of the lemma in this case.

First, we see that:

$$\texttt{extracts-fresh-heap}(\cdot; \rho_{old}, r_{root}, \tau; \rho'_{new}, h_{new}, l) \implies \texttt{extracts-fresh-heap}(\Omega; \rho_{old}, r, \tau; \phi(\rho'_{new}), h_{new}, l)$$

by noting **Rootedness**, **Heap-Sanity**, and **Simplicity** are all preserved by bijective region renaming (including of r_{root}), and that **Disjointness** and **Region-Freshness** are both guaranteed by our choice of Ω_{new}. Next we note that the only conditions on ρ_{old} are through **Disjointness**, and replacing it with P would not invalidate the LHS of the conjunction because $dom(\rho_{old}) = dom(\mathrm{P})$, as seen above, and would not invalidate the RHS of the conjunction again by our choice of Ω_{new} being disjoint from Ω, which contains $dom(\mathrm{P})$ by the assumed well-typedness of the LHS decorated configuration. We can now conclude $\texttt{extracts-fresh-heap}(\Omega; \mathrm{P}, r, \tau; \phi(\rho'_{new}), h_{new}, l)$ and thus apply decorated E3$^{(\text{D})}$, concluding this case.

E8$^{(\text{D})}$**:** We note very simply that it suffices to choose Ω_{new} disjoint from Ω and of equal cardinality to that of $NR(e_{body}) \uplus NR(e_{bool})$, as applying ϕ, a region renaming, to a decorated expression will not effect its projection as an undecorated expression, nor will adding the $@r_u$ annotation, so we can conclude that the step in this case indeed lifts.

E11$^{(\text{D})}$**:** Here as in E8$^{(\text{D})}$, we simply choose Ω_{new} disjoint from Ω, except for an intersection of Ω'_{out} if it is nonempty, and of equal cardinality to $NR(e)$. We do note that by the details of function definitions, our lambda expression in the decorated system will provide us with a Ω_{out} of equal cardinality to Ω'_{out}, and so these two cardinality properties suffice to show the existence of a bijection ϕ as needed. As above, application of ϕ to part of the LHS decorated expression will not effect its projection into the undecorated language, so we can conclude that in this case, too, the step lifts.

E18$^{(\text{D})}$**:** These have additional premises in the decorated system, but the premises all reduce to showing that certain locations are in the domain of P, and all follow from basic properties of the decorated well-typedness of the decorated configuration containing $\mathfrak{e}$, so they lift.

□

Lemma 4.6 (Step Projection (Restated)). *For any decorated step* $(d, h, s, \rho, \Omega, \mathfrak{e}) \xrightarrow{eval} (d', h', s', \rho', \Omega', \mathfrak{e}')$, $(d, h, s, e) \xrightarrow{eval} (d', h', s', e')$ *is an undecorated step, where* $e = \textsc{strip}(\mathfrak{e})$ *and* $e' = \textsc{strip}(\mathfrak{e})$.

Proof of Lemma 4.6. Trivial inspection of E$^{(\text{D})}$ rules in respective systems - the undecorated stepping relation has strictly weaker premises in all cases so one can conclude the undecorated step follows from the decorated step by just inverting the decorated E$^{(\text{D})}$ rule to obtain all necessary information to apply the corresponding undecorated E rule and conclude the step projects. □