Type-Driven Gradual Security with References: Complete Definitions and Proofs

Technical Report TR/DCC-2018-4

MATÍAS TORO, PLEIAD Laboratory, Computer Science Department (DCC), University of Chile
RONALD GARCIA, Software Practices Laboratory, University of British Columbia
ÉRIC TANTER, PLEIAD Laboratory, Computer Science Department (DCC), University of Chile

In security-typed programming languages, types statically enforce noninterference between potentially conspiring values, such as the arguments and results of functions. But to adopt static security types, like other advanced type disciplines, programmers face a steep wholesale transition, often forcing them to refactor working code just to satisfy their type checker. To provide a gentler path to security typing that supports safe and stylish but hard-to-verify programming idioms, researchers have designed languages that blend static and dynamic checking of security types. Unfortunately most of the resulting languages only support static, type-based reasoning about noninterference if a program is entirely statically secured. This limitation substantially weakens the benefits that dynamic enforcement brings to static security typing. Additionally, current proposals are focused on languages with explicit casts, and therefore do not fulfill the vision of gradual typing, according to which the boundaries between static and dynamic checking only arise from the (im)precision of type annotations, and are transparently mediated by implicit checks.

In this technical report we present the complete definitions and proofs of $\mathrm{GSL}_{\mathrm{Ref}}$, a gradual security-typed higher-order language with references. As a gradual language, $\mathrm{GSL}_{\mathrm{Ref}}$ supports the range of static-to-dynamic security checking exclusively driven by type annotations, without resorting to explicit casts. Additionally, $\mathrm{GSL}_{\mathrm{Ref}}$ lets programmers use types to reason statically about termination-insensitive noninterference in *all* programs, even those that enforce security dynamically. We prove that $\mathrm{GSL}_{\mathrm{Ref}}$ satisfies all but one of Siek *et al.*'s criteria for gradually-typed languages, which ensure that programs can seamlessly transition between simple typing and security typing. A notable exception regards the dynamic gradual guarantee, which some specific programs must violate if they are to satisfy noninterference; it remains an open question whether such a language could fully satisfy the dynamic gradual guarantee. To realize this design, we were led to draw a sharp distinction between syntactic type *safety* and semantic type *soundness*, each of which constrains the design of the gradual language.

CCS Concepts: • **Security and privacy → Information flow control**; • **Theory of computation → Type structures**; **Program semantics**;

Additional Key Words and Phrases: Noninterference, language-based security, gradual typing

This work is partially funded by CONICYT FONDECYT Regular Project 1150017.

Contents

$$
\begin{array}{rcll}
S & ::= & \mathsf{Bool}_\ell \mid S \xrightarrow{\ell}_\ell S \mid \mathsf{Ref}_\ell\ S \mid \mathsf{Unit}_\ell & \text{(types)} \\
b & ::= & \mathsf{true} \mid \mathsf{false} & \text{(Booleans)} \\
r & ::= & b \mid \lambda^\ell x : S.t \mid \mathsf{unit} \mid o & \text{(raw values)} \\
v & ::= & r_\ell & \text{(values)} \\
t & ::= & v \mid t\ t \mid t \oplus t \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t & \\
 & & \mathsf{ref}^S\ t \mid !t \mid t{:=}t \mid t :: S \mid \mathsf{prot}_\ell(t) & \text{(terms)} \\
\oplus & ::= & \wedge \mid \vee & \text{(operations)}
\end{array}
$$

Fig. 1. SSL_{Ref}: Syntax

1 OVERVIEW

In this document we present the complete definitions and proofs of the static language SSL_{Ref}, the gradual language GSL_{Ref}, and the evidence augmented language $GSL^{\varepsilon}_{\mathsf{Ref}}$. Section 2 presents the full definitions for the static and gradual languages. Section 3 presents the proof of type safety and noninterference for SSL_{Ref}. Section 4 presents the proofs of soundness and optimality of the Galois connection, and the proof of the static gradual guarantee. Section 5 presents the formalization of evidences for GSL_{Ref}: structure of evidence along with it corresponding Galois connection, initial evidence, evolving evidence (consistent transitivity), algorithmic definitions and their proofs. Section 6 present dynamic properties of $GSL^{\varepsilon}_{\mathsf{Ref}}$. The presentation and proofs follows an intrinsic notation rather than evidence augmented notation, as it is more explicit (although more verbose). We present the proofs of type safety and noninterference, along the proof of the dynamic gradual guarantee for a similar gradual language that does not contain the extra dynamic check added in the runtime semantics.

2 FULL DEFINITIONS FOR THE STATIC AND GRADUAL LANGUAGES

In this section we present the full definition of SSL_{Ref} (sections 2.1 and 2.2) and the full definition of GSL_{Ref} (sections 2.4 and 2.6). Section 2.8 presents the full definitions of noninterference presented in the paper.

2.1 SSL_{Ref}: Static semantics

In this section we present the full definition of the static semantics of SSL_{Ref}. Figure 1 presents the syntax of SSL_{Ref}. Figure 2 presents the complete static semantics of SSL_{Ref}, where the join between types and labels is defined as follows

$$
\begin{array}{c}
\mathsf{Bool}_\ell \curlyvee \ell' = \mathsf{Bool}_{(\ell \curlyvee \ell')} \\
(S_1 \xrightarrow{\ell_c}_\ell S_2) \curlyvee \ell' = S_1 \xrightarrow{\ell_c}_{(\ell \curlyvee \ell')} S_2 \\
\mathsf{Ref}_\ell\ S \curlyvee \ell' = \mathsf{Ref}_{(\ell \curlyvee \ell')}\ S
\end{array}
$$

Figure 3 presents the join and meet type functions.

Definition 2.1 (Valid Type Sets).

$$
\frac{}{valid(\{\,\overline{\mathsf{Bool}_{\ell_i}}\,\})}
\qquad
\frac{valid(\{\,\overline{S_{i1}}\,\}) \quad valid(\{\,\overline{S_{i2}}\,\})}{valid(\{\,\overline{S_{i1} \xrightarrow{\ell_{c\,i}}_{\ell_i} S_{i2}}\,\})}
\qquad
\frac{valid(\{\,\overline{S_i}\,\})}{valid(\{\,\overline{\mathsf{Ref}_{\ell_i}\ S_i}\,\})}
$$

$$
\frac{}{valid(\{\,\overline{\mathsf{Unit}_{\ell_i}}\,\})}
$$

2.2 SSL_{Ref}: Dynamic semantics

In this section we present in Figure 4 the full definition of the dynamic semantics of SSL_{Ref}.

$$\text{(Sx)}\ \frac{x : S \in \Gamma}{\Gamma; \Sigma; \ell_c \vdash x : S} \qquad \text{(Sb)}\ \frac{}{\Gamma; \Sigma; \ell_c \vdash b_\ell : \mathsf{Bool}_\ell} \qquad \text{(Su)}\ \frac{}{\Gamma; \Sigma; \ell_c \vdash \mathsf{unit}_\ell : \mathsf{Unit}_\ell}$$

$$\text{(Sl)}\ \frac{o : S \in \Sigma}{\Gamma; \Sigma; \ell_c \vdash o_\ell : \mathsf{Ref}_\ell\ S} \qquad \text{(S}\lambda\text{)}\ \frac{\Gamma, x : S_1; \Sigma; \ell' \vdash t : S_2}{\Gamma; \Sigma; \ell_c \vdash (\lambda^{\ell'} x : S_1.t)_\ell : S_1 \xrightarrow{\ell'}_\ell S_2}$$

$$\text{(Sprot)}\ \frac{\Gamma; \Sigma; \ell_c \curlyvee \ell \vdash t : S}{\Gamma; \Sigma; \ell_c \vdash \mathsf{prot}_\ell(t) : S \curlyvee \ell} \qquad \text{(S}\oplus\text{)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t_1 : \mathsf{Bool}_{\ell_1} \quad \Gamma; \Sigma; \ell_c \vdash t_2 : \mathsf{Bool}_{\ell_2}}{\Gamma; \Sigma; \ell_c \vdash t_1 \oplus t_2 : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

$$\text{(Sapp)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t_1 : S_{11} \xrightarrow{\ell'}_\ell S_{12} \quad \Gamma; \Sigma; \ell_c \vdash t_2 : S_2 \quad S_2 <: S_{11} \quad \ell_c \curlyvee \ell \preccurlyeq \ell'}{\Gamma; \Sigma; \ell_c \vdash t_1\ t_2 : S_{12} \curlyvee \ell}$$

$$\text{(Sif)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t : \mathsf{Bool}_\ell \quad \Gamma; \Sigma; \ell_c \curlyvee \ell \vdash t_i : S_i}{\Gamma; \Sigma; \ell_c \vdash \mathsf{if}\ t\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1 \dot{\curlyvee} S_2) \curlyvee \ell} \qquad \text{(Sref)}\ \frac{\begin{array}{c}\Gamma; \Sigma; \ell_c \vdash t : S' \quad S' <: S \\ \ell_c \preccurlyeq \mathsf{label}(S)\end{array}}{\Gamma; \Sigma; \ell_c \vdash \mathsf{ref}^S\ t : \mathsf{Ref}_\bot\ S}$$

$$\text{(Sderef)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t : \mathsf{Ref}_\ell\ S}{\Gamma; \Sigma; \ell_c \vdash !t : S \curlyvee \ell}$$

$$\text{(Sasgn)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t_1 : \mathsf{Ref}_\ell\ S_1 \quad \Gamma; \Sigma; \ell_c \vdash t_2 : S_2 \quad S_2 <: S_1 \quad \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S_1)}{\Gamma; \Sigma; \ell_c \vdash t_1 := t_2 : \mathsf{Unit}_\bot}$$

$$\text{(S::)}\ \frac{\Gamma; \Sigma; \ell_c \vdash t : S_1 \quad S_1 <: S_2}{\Gamma; \Sigma; \ell_c \vdash t :: S_2 : S_2}$$

$$\boxed{S <: S} \qquad \frac{\ell \preccurlyeq \ell'}{\mathsf{Bool}_\ell <: \mathsf{Bool}_{\ell'}} \qquad \frac{\ell \preccurlyeq \ell'}{\mathsf{Unit}_\ell <: \mathsf{Unit}_{\ell'}}$$

$$\frac{S_1' <: S_1 \quad S_2 <: S_2' \quad \ell_1 \preccurlyeq \ell_1' \quad \ell_2' \preccurlyeq \ell_2}{S_1 \xrightarrow{\ell_2}_{\ell_1} S_2 <: S_1' \xrightarrow{\ell_2'}_{\ell_1'} S_2'} \qquad \frac{\ell \preccurlyeq \ell'}{\mathsf{Ref}_\ell\ S <: \mathsf{Ref}_{\ell'}\ S}$$

Fig. 2. $\mathsf{SSL}_{\mathsf{Ref}}$: Static Semantics

2.3 $\mathsf{SSL}_{\mathsf{Ref}}$: Noninterference definitions

In this section we present definitions and properties of noninterference for $\mathsf{SSL}_{\mathsf{Ref}}$. Figure 5 presents the full definition of step-indexed logical relations. The proofs can be found in Appendix 3.4.

Definition 2.2. Let ρ be a substitution, Γ and Σ a type substitutions. We say that substitution ρ satisfy environment Γ and Σ, written $\rho \models \Gamma; \Sigma$, if and only if $dom(\rho) = \Gamma$ and $\forall x \in dom(\Gamma), \forall \ell_c, \Gamma; \Sigma; \ell_c \vdash \rho(x) : S'$, where $S' <: \Gamma(x)$.

Definition 2.3 (Related substitutions). Tuples $\langle \ell_1, \rho_1, \mu_1 \rangle$ and $\langle \ell_2, \rho_2, \mu_2 \rangle$ are related on k steps, notation $\Gamma; \Sigma \vdash \langle \ell_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2, \mu_2 \rangle$, if $\rho_i \models \Gamma; \Sigma$, $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$ and

$$\forall x \in \Gamma. \Sigma \vdash \langle \ell_1, \rho_1(x), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2(x), \mu_2 \rangle : \Gamma(x)$$

$$\boxed{S \mathbin{\ddot{\curlyvee}} S, S \mathbin{\ddot{\curlywedge}} S}$$

$$
\begin{aligned}
&\ddot{\curlyvee} : \textsc{Type} \times \textsc{Type} \rightharpoonup \textsc{Type}\\
&\mathsf{Bool}_{\ell} \mathbin{\ddot{\curlyvee}} \mathsf{Bool}_{\ell'} = \mathsf{Bool}_{(\ell \curlyvee \ell')}\\
&(S_{11} \xrightarrow{\ell_c}_{\ell} S_{12}) \mathbin{\ddot{\curlyvee}} (S_{21} \xrightarrow{\ell_c'}_{\ell'} S_{22}) = (S_{11} \mathbin{\ddot{\curlywedge}} S_{21}) \xrightarrow{\ell_c \curlywedge \ell_c'}_{(\ell \curlyvee \ell')} (S_{12} \mathbin{\ddot{\curlyvee}} S_{22})\\
&\mathsf{Ref}_{\ell}\ S \mathbin{\ddot{\curlyvee}} \mathsf{Ref}_{\ell'}\ S = \mathsf{Ref}_{(\ell \curlyvee \ell')}\ S\\
&S \mathbin{\ddot{\curlyvee}} S \text{ undefined otherwise}
\end{aligned}
$$

$$
\begin{aligned}
&\ddot{\curlywedge} : \textsc{Type} \times \textsc{Type} \rightharpoonup \textsc{Type}\\
&\mathsf{Bool}_{\ell} \mathbin{\ddot{\curlywedge}} \mathsf{Bool}_{\ell'} = \mathsf{Bool}_{(\ell \curlywedge \ell')}\\
&(S_{11} \xrightarrow{\ell_c}_{\ell} S_{12}) \mathbin{\ddot{\curlywedge}} (S_{21} \xrightarrow{\ell_c'}_{\ell'} S_{22}) = (S_{11} \mathbin{\ddot{\curlyvee}} S_{21}) \xrightarrow{\ell_c \curlyvee \ell_c'}_{(\ell \curlywedge \ell')} (S_{12} \mathbin{\ddot{\curlywedge}} S_{22})\\
&\mathsf{Ref}_{\ell}\ S \mathbin{\ddot{\curlywedge}} \mathsf{Ref}_{\ell'}\ S = \mathsf{Ref}_{(\ell \curlywedge \ell')}\ S\\
&S \mathbin{\ddot{\curlywedge}} S \text{ undefined otherwise}
\end{aligned}
$$

Fig. 3. $\mathsf{SSL}_{\mathsf{Ref}}$: Join and meet type functions

$\boxed{t \mid \mu \xrightarrow{\ell_c} t \mid \mu}$ **Notion of Reduction**

$$b_{1\ell_1} \oplus b_{2\ell_2} \mid \mu \xrightarrow{\ell_c} (b_1 [\![\oplus]\!] b_2)_{(\ell_1 \curlyvee \ell_2)} \mid \mu \qquad (\lambda^{\ell'} x : S.t)_{\ell}\ v \mid \mu \xrightarrow{\ell_c} \mathsf{prot}_{\ell}([v/x]t) \mid \mu$$

$$\mathsf{if\ true}_{\ell}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid \mu \xrightarrow{\ell_c} \mathsf{prot}_{\ell}(t_1) \mid \mu \qquad \mathsf{if\ false}_{\ell}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid \mu \xrightarrow{\ell_c} \mathsf{prot}_{\ell}(t_2) \mid \mu$$

$$\mathsf{prot}_{\ell}(v) \mid \mu \xrightarrow{\ell_c} v \curlyvee \ell \mid \mu \qquad \mathsf{ref}^S\ v \mid \mu \xrightarrow{\ell_c} o_{\bot} \mid \mu[o \mapsto v \curlyvee \ell_c] \text{ where } o \notin \mathit{dom}(\mu)$$

$$!o_{\ell} \mid \mu \xrightarrow{\ell_c} v \curlyvee \ell \mid \mu \text{ where } \mu(o) = v \qquad o_{\ell}{:=}v \mid \mu \xrightarrow{\ell_c} \mathsf{unit}_{\bot} \mid \mu[o \mapsto v \curlyvee \ell_c \curlyvee \ell]$$

$$v :: S \mid \mu \xrightarrow{\ell_c} v \curlyvee \mathit{label}(S) \mid \mu$$

$\boxed{t \mid \mu \overset{\ell_c}{\longmapsto} t \mid \mu}$ **Reduction**

$$(\mathrm{R}{\rightarrow})\ \frac{t_1 \mid \mu_1 \xrightarrow{\ell_c} t_2 \mid \mu_2}{t_1 \mid \mu_1 \overset{\ell_c}{\longmapsto} t_2 \mid \mu_2} \qquad (\mathrm{R}f)\ \frac{t_1 \mid \mu_1 \overset{\ell_c}{\longmapsto} t_2 \mid \mu_2}{f[t_1] \mid \mu_1 \overset{\ell_c}{\longmapsto} f[t_2] \mid \mu_2}$$

$$(\mathrm{Rprot})\ \frac{t_1 \mid \mu_1 \overset{\ell_c \curlyvee \ell}{\longmapsto} t_2 \mid \mu_2}{\mathsf{prot}_{\ell}(t_1) \mid \mu_1 \overset{\ell_c}{\longmapsto} \mathsf{prot}_{\ell}(t_2) \mid \mu_2}$$

Fig. 4. $\mathsf{SSL}_{\mathsf{Ref}}$: Label Tracking Dynamic Semantics

Definition 2.4 (Semantic Security Typing).

$$
\begin{aligned}
\Gamma; \Sigma; \ell_c \models t : S \iff\ & \forall\, \ell_o \in \textsc{Label}, k \geq 0, \rho_1, \rho_2 \in \textsc{Subst} \text{ and } \mu_1, \mu_2 \in \textsc{Store}\\
& \text{such that } \Sigma \vdash \mu_i \text{ and } \Gamma; \Sigma \vdash \langle \ell_c, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_c, \rho_2, \mu_2 \rangle, \text{ we have}\\
& \Sigma \vdash \langle \ell_c, \rho_1(t), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_c, \rho_2(t), \mu_2 \rangle : \mathcal{C}(S)
\end{aligned}
$$

Proposition 2.5 (Security Type Soundness). *If* $\Gamma; \Sigma; \ell_c \vdash t : S_i' \implies \forall S, S_i' <: S, \Gamma; \Sigma; \ell_c \models t : S$

$$\Sigma \vdash \langle \ell_1, v_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, v_2, \mu_2 \rangle : S \iff \ell_1 \approx_{\ell_o} \ell_2 \wedge \Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \Sigma; \ell_i \vdash v_i : S'_i, S'_i <: S,$$
$$\wedge \left(\mathsf{obs}_{\ell_o}(\ell_i, S) \implies \mathsf{obsRel}^{\Sigma,S}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, , v_2, \mu_2) \right)$$

$$\mathsf{obsRel}^{\Sigma,S}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, v_2, \mu_2) \iff (rval(v_1) = rval(v_2)) \qquad \text{if } S \in \{\mathsf{Bool}_g, \mathsf{Unit}_g, \mathsf{Ref}_g\ S'\}$$

$$\mathsf{obsRel}^{\Sigma,S_1 \xrightarrow{\ell'}_{\ell} S_2}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, v_2, \mu_2) \iff \forall j \leq k. \forall \Sigma \subseteq \Sigma', \Sigma' \vdash \langle \ell_1, v'_1, \mu'_1 \rangle \approx^j_{\ell_o} \langle \ell_2, v'_2, \mu'_2 \rangle : S_1,$$
$$\Sigma' \vdash \langle \ell_1, v_1\ v'_1, \mu'_1 \rangle \approx^j_{\ell_o} \langle \ell_2, v_2\ v'_2, \mu'_2 \rangle : \mathcal{C}(S_2 \mathbin{\widetilde{\curlyvee}} g)$$

$$\Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S) \iff \ell_1 \approx_{\ell_o} \ell_2 \wedge \Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \Sigma; \ell_i \vdash t_i : S'_i, S'_i <: S, \forall j < k$$
$$\left(t_i \mid \mu_i \xmapsto{\ell_i}{}^j t'_i \mid \mu'_i \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu'_1 \approx^{k-j}_{\ell_o} \mu'_2 \wedge \right.$$
$$\left. (irred(t'_i) \implies \Sigma' \vdash \langle \ell_1, t'_1, \mu'_1 \rangle \approx^{k-j}_{\ell_o} \langle \ell_2, t'_2, \mu'_2 \rangle : S) \right)$$

$$\begin{array}{rcl}
\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 & \iff & \Sigma \vdash \mu_i \wedge \forall \ell_i, \ell_1 \approx_{\ell_o} \ell_2, j < k, \forall o \in dom(\mu_1) \cap dom(\mu_2) \\
& & \Sigma \vdash \langle \ell_1, \mu_1(o), \mu_1 \rangle \approx^j_{\ell_o} \langle \ell_2, \mu_2(o), \mu_2 \rangle : \Sigma(o) \\
\ell_1 \approx_{\ell_o} \ell_2 & \iff & \mathsf{obs}_{\ell_o}(\ell_i) \vee \neg \mathsf{obs}_{\ell_o}(\ell_i) \\
\mu_1 \twoheadrightarrow \mu_2 & \iff & dom(\mu_1) \subseteq dom(\mu_2) \\
\mathsf{obs}_{\ell_o}(\ell, S) & \iff & \mathsf{obs}_{\ell_o}(\ell) \wedge \mathsf{obs}_{\ell_o}(label(S)) \\
\mathsf{obs}_{\ell_o}(\ell) & \iff & \ell \preccurlyeq \ell_o
\end{array}$$

Fig. 5. Security logical relations

$$g, g_c, g_r \in \textsc{GLabel}, \quad U \in \textsc{GType}, \quad x \in \textsc{Var}, \quad b \in \textsc{Bool}, \quad \oplus \in \textsc{BoolOp}$$
$$l \in \textsc{Loc}, \quad t \in \textsc{GTerm}, \quad r \in \textsc{RawValue} \quad v \in \textsc{Value}$$
$$\Gamma \in \textsc{Var} \overset{\text{fin}}{\rightharpoonup} \textsc{GType}, \quad \Sigma \in \textsc{Loc} \overset{\text{fin}}{\rightharpoonup} \textsc{GType}$$

$$\begin{array}{lcll}
U & ::= & \mathsf{Bool}_g \mid U \xrightarrow{g_c}_g U \mid \mathsf{Ref}_g\ U \mid \mathsf{Unit}_g & \text{(gradual types)} \\
g & ::= & \ell \ \mid\ ? & \text{(gradual labels)} \\
b & ::= & \mathsf{true} \mid \mathsf{false} & \text{(Booleans)} \\
r & ::= & b \mid \lambda^{g_c} x : U.t \mid \mathsf{unit} \mid o & \text{(base values)} \\
v & ::= & r_g & \text{(values)} \\
t & ::= & v \mid t\ t \mid t \oplus t \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t & \text{(terms)} \\
 & & \mathsf{ref}^U\ t \mid !t \mid t{:=}t \mid \mathsf{prot}_g(t) & \\
\oplus & ::= & \wedge \mid \vee & \text{(operations)}
\end{array}$$

Fig. 6. $\mathsf{GSL}_{\mathsf{Ref}}$: Syntax

2.4 $\mathbf{GSL}_{\mathbf{Ref}}$: Static semantics

In this section we present the syntax and static semantics of $\mathsf{GSL}_{\mathsf{Ref}}$. The syntax of $\mathsf{GSL}_{\mathsf{Ref}}$ is given in Figure 6 and is otherwise identical to that of $\mathsf{SSL}_{\mathsf{Ref}}$. Figure 7 presents the type system of $\mathsf{GSL}_{\mathsf{Ref}}$. Each typing rule is derived from a corresponding $\mathsf{SSL}_{\mathsf{Ref}}$ rule (Figure 2) by lifting labels, types, predicates, and functions to their gradual counterparts. We also present some additional definitions needed in gradualizing $\mathsf{SSL}_{\mathsf{Ref}}$ which are not included in the paper. Finally we present some example typing derivations in Figure 9.

2.4.1 *Additional Definitions.*

Definition 2.6 (Type Concretization). $\gamma_S : \textsc{GType} \to \mathcal{P}(\textsc{Type})$

$$\gamma_S(\mathsf{Bool}_g) = \{\, \mathsf{Bool}_\ell \mid \ell \in \gamma(g) \,\} \qquad \gamma_S(U_1 \xrightarrow{g'}_g U_2) = \gamma_S(U_1) \xrightarrow{\gamma(g')}_{\gamma(g)} \gamma_S(U_2)$$
$$\gamma_S(\mathsf{Unit}_g) = \{\, \mathsf{Unit}_\ell \mid \ell \in \gamma(g) \,\} \qquad \gamma_S(\mathsf{Ref}_g\ U) = \{\, \mathsf{Ref}_\ell\ S \mid \ell \in \gamma(g), S \in \gamma_S(U) \,\}$$

Type concretization induces notions of precision and abstraction.

Definition 2.7 (Type Precision). $U_1 \sqsubseteq U_2$, if and only if $\gamma_S(U_1) \subseteq \gamma_S(U_2)$.

Definition 2.8 (Type Abstraction). $\alpha_S : \mathcal{P}(\textsc{Type}) \to \textsc{GType}$

$$\alpha_S(\{\, \overline{\mathsf{Bool}_{\ell_i}} \,\}) = \mathsf{Bool}_{\alpha(\{\, \overline{\ell_i} \,\})} \qquad \alpha_S(\{\, \overline{\mathsf{Unit}_{\ell_i}} \,\}) = \mathsf{Unit}_{\alpha(\{\, \overline{\ell_i} \,\})}$$

$$\alpha_S(\{\, \overline{S_{i1} \xrightarrow{\ell'_i}_{\ell_i} S_{i2}} \,\}) = \alpha_S(\{\, \overline{S_{i1}} \,\}) \xrightarrow{\alpha(\{\, \overline{\ell'_i} \,\})}_{\alpha(\{\, \overline{\ell_i} \,\})} \alpha_S(\{\, \overline{S_{i2}} \,\}) \qquad \alpha_S(\{\, \overline{\mathsf{Ref}_{\ell_i}\ S_i} \,\}) = \mathsf{Ref}_{\alpha(\{\, \overline{\ell_i} \,\})}\ \alpha_S(\{\, \overline{S_i} \,\})$$

$$\alpha_S(\widehat{S}) \text{ is undefined otherwise}$$

Proposition 2.9 (α_S is Sound and Optimal). *Assuming* $\widehat{S}$ **valid***:*
(i) $\widehat{S} \subseteq \gamma_S(\alpha_S(\widehat{S}))$ *(ii) If* $\widehat{S} \subseteq \gamma_S(U)$ *then* $\alpha_S(\widehat{S}) \sqsubseteq U$.

Definition 2.10 (Gradual label meet).
$g_1 \mathbin{\widetilde{\curlywedge}} g_2 = \alpha(\{\, \ell_1 \curlywedge \ell_2 \mid (\ell_1, \ell_2) \in \gamma(g_1) \times \gamma(g_2) \,\})$.

Algorithmically:

$$\bot \mathbin{\widetilde{\curlywedge}} ? = ? \mathbin{\widetilde{\curlywedge}} \bot = \bot \qquad g \mathbin{\widetilde{\curlywedge}} ? = ? \mathbin{\widetilde{\curlywedge}} g = ? \ \text{ if } g \neq \bot \qquad \ell_1 \mathbin{\widetilde{\curlywedge}} \ell_2 = \ell_1 \curlywedge \ell_2$$

$$\boxed{\Gamma; \Sigma; g \vdash t : U}$$

$$(Ux)\ \frac{x : U \in \Gamma}{\Gamma; \Sigma; g_c \vdash x : U} \qquad (Ub)\ \frac{}{\Gamma; \Sigma; g_c \vdash b_g : \mathsf{Bool}_g} \qquad (Uu)\ \frac{}{\Gamma; \Sigma; g_c \vdash \mathsf{unit}_g : \mathsf{Unit}_g}$$

$$(Uo)\ \frac{o : U \in \Sigma}{\Gamma; \Sigma; g_c \vdash o_g : \mathsf{Ref}_g\ U} \qquad (U\lambda)\ \frac{\Gamma, x : U_1; \Sigma; g_c' \vdash t : U_2}{\Gamma; \Sigma; g_c \vdash (\lambda^{g_c'} x : U_1 . t)_g : U_1 \xrightarrow{g_c'}_g U_2}$$

$$(U\mathrm{prot})\ \frac{\Gamma; \Sigma; g_c \mathbin{\widetilde{\curlyvee}} g \vdash t : U}{\Gamma; \Sigma; g_c \vdash \mathsf{prot}_g(t) : U \mathbin{\widetilde{\curlyvee}} g} \qquad (U\oplus)\ \frac{\Gamma; \Sigma; g_c \vdash t_1 : \mathsf{Bool}_{g_1} \quad \Gamma; \Sigma; g_c \vdash t_2 : \mathsf{Bool}_{g_2}}{\Gamma; \Sigma; g_c \vdash t_1 \oplus t_2 : \mathsf{Bool}_{(g_1 \widetilde{\curlyvee} g_2)}}$$

$$(U\mathrm{app})\ \frac{\begin{array}{c}\Gamma; \Sigma; g_c \vdash t_1 : U_{11} \xrightarrow{g_c'}_g U_{12} \\ \Gamma; \Sigma; g_c \vdash t_2 : U_2 \\ U_2 \lesssim U_{11} \qquad \widetilde{g \curlyvee g_c \preccurlyeq g_c'}\end{array}}{\Gamma; \Sigma; g_c \vdash t_1\ t_2 : U_{12} \mathbin{\widetilde{\curlyvee}} g} \qquad (U\mathrm{if})\ \frac{\begin{array}{c}\Gamma; \Sigma; g_c \vdash t : \mathsf{Bool}_g \\ \Gamma; \Sigma; g_c \mathbin{\widetilde{\curlyvee}} g \vdash t_1 : U_1 \qquad \Gamma; \Sigma; g_c \mathbin{\widetilde{\curlyvee}} g \vdash t_2 : U_2\end{array}}{\Gamma; \Sigma; g_c \vdash \mathsf{if}\ t\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (U_1 \mathbin{\widetilde{\dot{\vee}}} U_2) \mathbin{\widetilde{\curlyvee}} g}$$

$$(U{::})\ \frac{\begin{array}{c}\Gamma; \Sigma; g_c \vdash t : U_1 \\ U_1 \lesssim U_2\end{array}}{\Gamma; \Sigma; g_c \vdash t :: U_2 : U_2} \qquad (U\mathrm{ref})\ \frac{\begin{array}{c}\Gamma; \Sigma; g_c \vdash t : U' \\ U' \lesssim U \qquad g_c \mathrel{\widetilde{\preccurlyeq}} \mathsf{label}(U)\end{array}}{\Gamma; \Sigma; g_c \vdash \mathsf{ref}^U\ t : \mathsf{Ref}_\bot\ U} \qquad (U\mathrm{deref})\ \frac{\Gamma; \Sigma; g_c \vdash t : \mathsf{Ref}_g\ U}{\Gamma; \Sigma; g_c \vdash\ !t : U \mathbin{\widetilde{\curlyvee}} g}$$

$$(U\mathrm{asgn})\ \frac{\Gamma; \Sigma; g_c \vdash t_1 : \mathsf{Ref}_g\ U_1 \qquad \Gamma; \Sigma; g_c \vdash t_2 : U_2 \qquad U_2 \lesssim U_1 \qquad \widetilde{g \curlyvee g_c \preccurlyeq \mathit{label}(U_1)}}{\Gamma; \Sigma; g_c \vdash t_1 := t_2 : \mathsf{Unit}_\bot}$$

Fig. 7. $\mathrm{GSL}_{\mathsf{Ref}}$: Static Semantics

$$\boxed{U \mathbin{\widetilde{\dot{\vee}}} U, U \mathbin{\widetilde{\dot{\wedge}}} U}$$

$\widetilde{\dot{\vee}} : \textsc{Type} \times \textsc{Type} \rightharpoonup \textsc{Type}$

$\mathsf{Bool}_g \mathbin{\widetilde{\dot{\vee}}} \mathsf{Bool}_{g'} = \mathsf{Bool}_{(g \widetilde{\curlyvee} g')}$

$(U_{11} \xrightarrow{g_c}_g U_{12}) \mathbin{\widetilde{\dot{\vee}}} (U_{21} \xrightarrow{g_c'}_{g'} U_{22}) = (U_{11} \mathbin{\widetilde{\dot{\wedge}}} U_{21}) \xrightarrow{g_c \widetilde{\curlywedge} g_c'}_{(g \widetilde{\curlyvee} g')} (U_{12} \mathbin{\widetilde{\dot{\vee}}} U_{22})$

$\mathsf{Ref}_g\ U \mathbin{\dot{\vee}} \mathsf{Ref}_{g'}\ U' = \mathsf{Ref}_{(g \widetilde{\curlyvee} g')}\ U \sqcap U'$

$U \mathbin{\widetilde{\dot{\vee}}} U$ undefined otherwise

$\widetilde{\dot{\wedge}} : \textsc{Type} \times \textsc{Type} \rightharpoonup \textsc{Type}$

$\mathsf{Bool}_g \mathbin{\widetilde{\dot{\wedge}}} \mathsf{Bool}_{g'} = \mathsf{Bool}_{(g \widetilde{\curlywedge} g')}$

$(U_{11} \xrightarrow{g_c}_g U_{12}) \mathbin{\widetilde{\dot{\wedge}}} (U_{21} \xrightarrow{g_c'}_{g'} U_{22}) = (U_{11} \mathbin{\widetilde{\dot{\vee}}} U_{21}) \xrightarrow{g_c \widetilde{\curlyvee} g_c'}_{(g \widetilde{\curlywedge} g')} (U_{12} \mathbin{\widetilde{\dot{\wedge}}} U_{22})$

$\mathsf{Ref}_g\ U \mathbin{\dot{\wedge}} \mathsf{Ref}_{g'}\ U' = \mathsf{Ref}_{(g \widetilde{\curlywedge} g')}\ U \sqcap U'$

$U \mathbin{\widetilde{\dot{\wedge}}} U$ undefined otherwise

Fig. 8. $\mathrm{GSL}_{\mathsf{Ref}}$: consistent join and consistent meet

Definition 2.11 (Gradual label join). $g_1 \mathbin{\widetilde{\curlyvee}} g_2 = \alpha(\{\, \ell_1 \curlyvee \ell_2 \mid (\ell_1, \ell_2) \in \gamma(g_1) \times \gamma(g_2) \,\})$
Algorithmically:

$$\top \mathbin{\widetilde{\curlyvee}} ? = ? \mathbin{\widetilde{\curlyvee}} \top = \top \qquad g \mathbin{\widetilde{\curlyvee}} ? = ? \mathbin{\widetilde{\curlyvee}} g = ? \ \text{ if } g \neq \top \qquad \ell_1 \mathbin{\widetilde{\curlyvee}} \ell_2 = \ell_1 \curlyvee \ell_2$$

Definition 2.12 (Label Meet). $g_1 \sqcap g_2 = \alpha(\gamma(g_1) \cap \gamma(g_2))$.
Algorithmically:

$$g \sqcap g = g \qquad\qquad g \sqcap ? = ? \sqcap g = g$$

Definition 2.13 (Type Meet). $U_1 \sqcap U_2 = \alpha_S(\gamma_S(U_1) \cap \gamma_S(U_2))$.
Algorithmically:

$$\frac{g \sqcap g'}{\mathsf{Bool}_g \sqcap \mathsf{Bool}_{g'}} \qquad \frac{g \sqcap g'}{\mathsf{Unit}_g \sqcap \mathsf{Unit}_{g'}} \qquad \frac{g \sqcap g' \quad U_1 \sqcap U_2}{\mathsf{Ref}_g\ U_1 \sqcap \mathsf{Ref}_{g'}\ U_2}$$

$$\frac{U_1 \sqcap U_1' \quad U_2 \sqcap U_2' \quad g_1 \sqcap g_1' \quad g_2 \sqcap g_2'}{U_1 \xrightarrow{g_2}_{g_1} U_2 \sqcap U_1' \xrightarrow{g_2'}_{g_1'} U_2'}$$

Also, we introduce a function *label*, which yields the security label of a given type:

$$label : \text{GType} \to \text{Label}$$

$$label(\mathsf{Bool}_g) = g \qquad label(\mathsf{Unit}_g) = g \qquad label(U_1 \to_g U_2) = g \qquad label(\mathsf{Ref}_g\ U) = g$$

Definition 2.14 (Type Precision (inductive definition)).

$$\frac{g_1 \sqsubseteq g_2}{\mathsf{Bool}_{g_1} \sqsubseteq \mathsf{Bool}_{g_2}} \qquad \frac{g_1 \sqsubseteq g_2}{\mathsf{Unit}_{g_1} \sqsubseteq \mathsf{Unit}_{g_2}} \qquad \frac{U_{11} \sqsubseteq U_{21} \quad U_{12} \sqsubseteq U_{22} \quad g_1 \sqsubseteq g_2 \quad g_{c1} \sqsubseteq g_{c2}}{U_{11} \xrightarrow{g_{c1}}_{g_1} U_{12} \sqsubseteq U_{21} \xrightarrow{g_{c2}}_{g_2} U_{22}}$$

$$\frac{g_1 \sqsubseteq g_2 \quad U_1 \sqsubseteq U_2}{\mathsf{Ref}_{g_1}\ U_1 \sqsubseteq \mathsf{Ref}_{g_2}\ U_2}$$

Definition 2.15 (Consistent label ordering (inductive definition)).

$$\frac{}{? \mathrel{\widetilde{\preccurlyeq}} g} \qquad \frac{}{g \mathrel{\widetilde{\preccurlyeq}} ?} \qquad \frac{\ell_1 \preccurlyeq \ell_2}{\ell_1 \mathrel{\widetilde{\preccurlyeq}} \ell_2}$$

Definition 2.16 (Consistent subtyping (inductive definition)).

$$\frac{g \mathrel{\widetilde{\preccurlyeq}} g'}{\mathsf{Bool}_g \lesssim \mathsf{Bool}_{g'}} \qquad \frac{g \mathrel{\widetilde{\preccurlyeq}} g'}{\mathsf{Unit}_g \lesssim \mathsf{Unit}_{g'}} \qquad \frac{g \mathrel{\widetilde{\preccurlyeq}} g' \quad U_1 \lesssim U_2 \quad U_2 \lesssim U_1}{\mathsf{Ref}_g\ U_1 \lesssim \mathsf{Ref}_{g'}\ U_2}$$

$$\frac{U_1' \lesssim U_1 \quad U_2 \lesssim U_2' \quad g_1 \mathrel{\widetilde{\preccurlyeq}} g_1' \quad g_2' \mathrel{\widetilde{\preccurlyeq}} g_2}{U_1 \xrightarrow{g_2}_{g_1} U_2 \lesssim U_1' \xrightarrow{g_2'}_{g_1'} U_2'}$$

2.5 $\mathbf{GSL}^{\varepsilon}_{\mathbf{Ref}}$: Static semantics

In this section we present the full definition of the static semantics of $\text{GSL}^{\varepsilon}_{\text{Ref}}$.

Definition 2.17 (Interval). An interval is a bounded unknown label $[\ell_1, \ell_2]$ where ℓ_1 is the upper bound and ℓ_2 is the lower bound.

$$\begin{array}{lll} \imath \in \text{Label}^2 & & \\ \imath & ::= & [\ell, \ell] \quad \text{(interval)} \end{array}$$

Definition 2.18 (Evidence for labels).

$$\varepsilon ::= \langle \imath, \imath \rangle$$

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{}{\ldots \vdash \mathit{pub} : \mathsf{Int}_{L}} \quad \dfrac{}{\ldots \vdash \mathit{priv} : \mathsf{Int}_{?}}}{..\cdot; \mathsf{L} \vdash \mathit{pub} < \mathit{priv} : \mathsf{Int}_{?}} \quad \dfrac{}{..\cdot; ? \vdash 1_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}}} \quad \dfrac{}{\ldots ? \vdash 1_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}}} \\ \hline ., \mathit{pub} : \mathsf{Int}_{\mathsf{L}}, \mathit{priv} : \mathsf{Int}_{\mathsf{H}}; \cdot; \mathsf{L} \vdash \mathsf{if}\ \mathit{pub} < \mathit{priv}\ \mathsf{then}\ 1_{\mathsf{L}}\ \mathsf{else}\ 2_{\mathsf{L}} : \mathsf{Int}_{?} \\ \mathsf{Int}_{?} \lesssim \mathsf{Int}_{\mathsf{L}}\end{array}}{., \mathit{pub} : \mathsf{Int}_{\mathsf{L}}, \mathit{priv} : \mathsf{Int}_{\mathsf{H}}; \cdot; \mathsf{L} \vdash (\mathsf{if}\ \mathit{pub} < \mathit{priv}\ \mathsf{then}\ 1_{\mathsf{L}}\ \mathsf{else}\ 2_{\mathsf{L}}) :: \mathsf{Int}_{\mathsf{L}} : \mathsf{Int}_{\perp}}}{., \mathit{pub} : \mathsf{Int}_{\mathsf{L}}; .; \mathsf{L} \vdash (\lambda^{\top} \mathit{priv} : \mathsf{Int}_{?}.(\mathsf{if}\ \mathit{pub} < \mathit{priv}\ \mathsf{then}\ 1_{\mathsf{L}}\ \mathsf{else}\ 2_{\mathsf{L}}) :: \mathsf{Int}_{\mathsf{L}})_{\mathsf{L}} : \mathsf{Int}_{?} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}}}{\begin{array}{l}\cdot; .; \mathsf{L} \vdash (\lambda^{\top} \mathit{pub} : \mathsf{Int}_{\mathsf{L}}.(\lambda^{\top} \mathit{priv} : \mathsf{Int}_{?}.(\mathsf{if}\ \mathit{pub} < \mathit{priv}\ \mathsf{then}\ 1_{\mathsf{L}}\ \mathsf{else}\ 2_{\mathsf{L}}) :: \mathsf{Int}_{\mathsf{L}})_{\mathsf{L}})_{\mathsf{L}} : \\ \mathsf{Int}_{\mathsf{L}} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{?} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}\end{array}}
$$

$$
\dfrac{\begin{array}{c}\dfrac{\mathcal{D}}{\cdot; .; \mathsf{L} \vdash \mathit{mix}\ 1_{\mathsf{L}} : \mathsf{Int}_{?} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}} \\ \cdot; .; \mathsf{L} \vdash 5_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}} \qquad \mathsf{Int}_{\mathsf{L}} \lesssim \mathsf{Int}_{?}\end{array}}{\cdot; .; \mathsf{L} \vdash (\mathit{mix}\ 1_{\mathsf{L}})\ 5_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}}}
\qquad
\dfrac{\begin{array}{c}\dfrac{\mathcal{D}}{\cdot; .; \mathsf{L} \vdash \mathit{mix}\ 1_{\mathsf{L}} : \mathsf{Int}_{?} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}} \\ \cdot; .; \mathsf{L} \vdash 5_{\mathsf{H}} : \mathsf{Int}_{\mathsf{H}} \qquad \mathsf{Int}_{\mathsf{H}} \lesssim \mathsf{Int}_{?}\end{array}}{\cdot; .; \mathsf{L} \vdash (\mathit{mix}\ 1_{\mathsf{L}})\ 5_{\mathsf{H}} : \mathsf{Int}_{\mathsf{L}}}
$$

$$
\dfrac{\begin{array}{c}\dfrac{\mathcal{D}}{\cdot; .; \mathsf{L} \vdash \mathit{mix}'\ 1_{\mathsf{L}} : \mathsf{Int}_{\mathsf{H}} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}} \\ \cdot; .; \mathsf{L} \vdash 5_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}} \qquad \mathsf{Int}_{\mathsf{L}} \lesssim \mathsf{Int}_{\mathsf{H}}\end{array}}{\cdot; .; \mathsf{L} \vdash (\mathit{mix}'\ 1_{\mathsf{L}})\ 5_{\mathsf{L}} : \mathsf{Int}_{\mathsf{L}}}
\qquad
\dfrac{\begin{array}{c}\dfrac{\mathcal{D}}{\cdot; .; \mathsf{L} \vdash \mathit{mix}'\ 1_{\mathsf{L}} : \mathsf{Int}_{\mathsf{H}} \xrightarrow{\top}_{\mathsf{L}} \mathsf{Int}_{\mathsf{L}}} \\ \cdot; .; \mathsf{L} \vdash 5_{\mathsf{H}} : \mathsf{Int}_{\mathsf{H}} \qquad \mathsf{Int}_{\mathsf{H}} \lesssim \mathsf{Int}_{\mathsf{H}}\end{array}}{\cdot; .; \mathsf{L} \vdash (\mathit{mix}'\ 1_{\mathsf{L}})\ 5_{\mathsf{H}} : \mathsf{Int}_{\mathsf{L}}}
$$

Fig. 9. $\mathsf{GSL}_{\mathsf{Ref}}$: Example typing derivations

$$
\begin{array}{rcl}
t & ::= & v \mid \varepsilon t \mathbin{@_{\varepsilon}} \varepsilon t \mid \varepsilon t \oplus \varepsilon t \mid \mathsf{if}\ \varepsilon t\ \mathsf{then}\ \varepsilon t\ \mathsf{else}\ \varepsilon t \mid \mathsf{ref}^{U}_{\varepsilon}\ \varepsilon t \mid !\varepsilon t \mid \varepsilon t :=_{\varepsilon} \varepsilon t \mid \mathsf{prot}_{\varepsilon_g}\varepsilon g(\varepsilon t) \mid \varepsilon t \\
r & ::= & b \mid (\lambda^{g} x.t) \mid \mathsf{unit} \mid o \\
u & ::= & r_g \mid x \\
v & ::= & u \mid \varepsilon u
\end{array}
$$

Fig. 10. $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Syntax

Definition 2.19 (Type Evidence). An evidence type is a gradual type labeled with an interval:

$$
\begin{array}{lcll}
E \in \mathrm{GEType}, & & \iota \in \mathrm{Label}^2 & \\
E & ::= & \mathsf{Bool}_{\iota} \mid E \xrightarrow{\iota}_{\iota} E \mid \mathsf{Ref}_{\iota}\ E \mid \mathsf{Unit}_{\iota} & \text{(type evidences)}
\end{array}
$$

Definition 2.20 (Evidence for types).

$$\varepsilon ::= \langle E, E \rangle$$

We present the syntax of $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$ in Figure 10 and the static semantics in Figure 11.

2.6 $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Dynamic semantics

In this section we present the full definition of the dynamic semantics of $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$.

We extend the syntax of $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$ with frames defined as follows:

$$
\begin{array}{rcl}
f & ::= & h[\varepsilon] \\
h & ::= & \square \oplus et \mid ev \oplus \square \mid \square \mathbin{@_{\varepsilon}} et \mid ev \mathbin{@_{\varepsilon}} \square \mid \varepsilon\ \square \mid \mathsf{if}\ \square\ \mathsf{then}\ et\ \mathsf{else}\ et \mid !\square \mid \square :=_{\varepsilon} et \mid ev :=_{\varepsilon} \square \mid \mathsf{ref}^{U}_{\varepsilon}\ \square
\end{array}
$$

$$\text{(Ix)}\ \frac{x : U \in \Gamma}{\Gamma; \Sigma; \varepsilon g_c \vdash x : U} \qquad \text{(Ib)}\ \frac{}{\Gamma; \Sigma; \varepsilon g_c \vdash b_g : \mathsf{Bool}_g} \qquad \text{(Iu)}\ \frac{}{\Gamma; \Sigma; \varepsilon g_c \vdash \mathsf{unit}_g : \mathsf{Unit}_g}$$

$$\text{(Il)}\ \frac{o : U \in \Sigma}{\Gamma; \Sigma; \varepsilon g_c \vdash o_g : \mathsf{Ref}_g\ U} \qquad \text{(I}\lambda\text{)}\ \frac{\Gamma, x : U_1; \Sigma; \varepsilon' g' \vdash t : U_2 \qquad \varepsilon' = \mathcal{I}_{\preccurlyeq}^{\cup}(g')}{\Gamma; \Sigma; \varepsilon g_c \vdash (\lambda^{g'} x : U_1.t)_g : U_1 \xrightarrow{g'}_g U_2}$$

$$\text{(Iprot)}\ \frac{\Gamma; \Sigma; \varepsilon' g'_c \vdash t : U' \qquad \varepsilon_1 \vdash U' \lesssim U \qquad \varepsilon_2 \vdash g' \mathrel{\widetilde{\preccurlyeq}} g}{\Gamma; \Sigma; \varepsilon g_c \vdash \mathsf{prot}_{\varepsilon_2 g'} \varepsilon' g'_c (\varepsilon_1 t) : U \mathbin{\widetilde{\curlyvee}} g} \qquad \text{(I}\varepsilon\text{)}\ \frac{\begin{array}{c}\Gamma; \Sigma; \varepsilon g_c \vdash t : U_1 \\ \varepsilon_1 \vdash U_1 \lesssim U_2\end{array}}{\Gamma; \Sigma; \varepsilon g_c \vdash \varepsilon_1 t : U_2}$$

$$\text{(Iapp)}\ \frac{\Gamma; \Sigma; \varepsilon g_c \vdash t_i : U_i \qquad \varepsilon_1 \vdash U_1 \lesssim U_{11} \xrightarrow{g'}_g U_{12} \qquad \varepsilon_2 \vdash U_2 \lesssim U_{11} \qquad \varepsilon_3 \vdash \widetilde{g'_c \vee g \preccurlyeq g'}}{\Gamma; \Sigma; \varepsilon g_c \vdash \varepsilon_1 t_1 \mathbin{@_{\varepsilon_3}} \varepsilon_2 t_2 : U_{12} \mathbin{\widetilde{\curlyvee}} g}$$

$$\text{(Iif)}\ \frac{\begin{array}{cccc}\Gamma; \Sigma; \varepsilon g_c \vdash t_1 : U_1 & \varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_g & & \varepsilon' g'_c = (\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1))(g_c \mathbin{\widetilde{\curlyvee}} g) \\ \Gamma; \Sigma; \varepsilon' g'_c \vdash t_2 : U_2 & \varepsilon_2 \vdash U_2 \lesssim U_2 \mathbin{\widetilde{\dot\curlyvee}} U_3 & \Gamma; \Sigma; \varepsilon' g'_c \vdash t_3 : U_3 & \varepsilon_3 \vdash U_3 \lesssim U_2 \mathbin{\widetilde{\dot\curlyvee}} U_3\end{array}}{\Gamma; \Sigma; \varepsilon g_c \vdash \mathsf{if}\ \varepsilon_1 t_1\ \mathsf{then}\ \varepsilon_2 t_2\ \mathsf{else}\ \varepsilon_3 t_3 : (U_2 \mathbin{\widetilde{\dot\curlyvee}} U_3) \mathbin{\widetilde{\curlyvee}} g}$$

$$\text{(I}\oplus\text{)}\ \frac{\begin{array}{cc}\Gamma; \Sigma; \varepsilon g_c \vdash t_1 : U_1 & \varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_{g_1} \\ \Gamma; \Sigma; \varepsilon g_c \vdash t_2 : U_2 & \varepsilon_2 \vdash U_2 \lesssim \mathsf{Bool}_{g_2}\end{array}}{\Gamma; \Sigma; \varepsilon g_c \vdash \varepsilon_1 t_1 \oplus \varepsilon_2 t_2 : \mathsf{Bool}_{g_1 \widetilde{\curlyvee} g_2}} \qquad \text{(Iref)}\ \frac{\begin{array}{c}\Gamma; \Sigma; \varepsilon g_c \vdash t : U' \\ \varepsilon_1 \vdash U' \lesssim U \qquad \varepsilon_2 \vdash g'_c \mathrel{\widetilde{\preccurlyeq}} label(U)\end{array}}{\Gamma; \Sigma; \varepsilon g_c \vdash \mathsf{ref}^{U}_{\varepsilon_2}\ \varepsilon_1 t : \mathsf{Ref}_{\perp}\ U}$$

$$\text{(Ideref)}\ \frac{\Gamma; \Sigma; \varepsilon g_c \vdash t : U' \qquad \varepsilon' \vdash U' \lesssim \mathsf{Ref}_g\ U}{\Gamma; \Sigma; \varepsilon g_c \vdash !\varepsilon' t : U \mathbin{\widetilde{\curlyvee}} g}$$

$$\text{(Iassgn)}\ \frac{\begin{array}{c}\Gamma; \Sigma; \varepsilon g_c \vdash t_1 : \mathsf{Ref}_{g'}\ U'_1 \qquad \varepsilon_1 \vdash \mathsf{Ref}_{g'}\ U'_1 \lesssim \mathsf{Ref}_g\ U_1 \\ \Gamma; \Sigma; \varepsilon g_c \vdash t_2 : U_2 \qquad \varepsilon_2 \vdash U_2 \lesssim U_1 \qquad \varepsilon_3 \vdash \widetilde{g'_c \vee g \preccurlyeq label(U_1)}\end{array}}{\Gamma; \Sigma; \varepsilon g_c \vdash \varepsilon_1 t_1 \mathrel{:=_{\varepsilon_3}} \varepsilon_2 t_2 : \mathsf{Unit}_{\perp}}$$

Every type rule has the extra judgment $\varepsilon \vdash g_c \mathrel{\widetilde{\preccurlyeq}} g'_c$.

Fig. 11. $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Static Semantics

We present the complete dynamic semantics in Figure 12, and the evaluation frames and reduction in Figure 13. Auxiliary functions for evidence for labels is presented in Figure 14. Auxiliary functions for evidence for types is shown in Figure 15, and the inversion functions for evidence in Figure 16.

2.7 $\mathsf{GSL}_{\mathsf{Ref}}$: Translation to $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$

In this section we present the translation from terms of $\mathsf{GSL}_{\mathsf{Ref}}$ into terms of $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$ in Figure 17. The initial evidence function for consistent label ordering is presented in Figure 18. The initial evidence function for consistent subtyping is presented in Figure 19 using the following definition of operation pattern:

$$\boxed{\xrightarrow{\varepsilon g_c} : \mathbb{C} \times (\mathbb{C} \cup \{\, \mathbf{error} \,\})}$$

$$(r1) \qquad \varepsilon_1(b_1)_{g_1} \oplus \varepsilon_2(b_2)_{g_2} \mid \mu \xrightarrow{\varepsilon g_c} (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2)(b_1 \llbracket \oplus \rrbracket\, b_2)_{(g_1 \widetilde{\curlyvee} g_2)} \mid \mu$$

$$(r2) \qquad \mathsf{prot}_{\varepsilon_1 g_1} \varepsilon_2 g_2 (\varepsilon_3 u) \mid \mu \xrightarrow{\varepsilon g_c} (\varepsilon_3 \mathbin{\widetilde{\curlyvee}} \varepsilon_1)(u \mathbin{\widetilde{\curlyvee}} g_1) \mid \mu$$

$$(r3) \quad \varepsilon_1(\lambda^{g'} x : U.t)_g \mathbin{@_{\varepsilon_3}} \varepsilon_2 u \mid \mu \xrightarrow{\varepsilon g_c} \begin{cases} \mathsf{prot}_{ilbl(\varepsilon_1)g} \varepsilon_1' g_1' (icod(\varepsilon_1)([\varepsilon_2' u/x]t)) \mid \mu \\ \mathbf{error} \qquad \text{if } \varepsilon_1' \text{ or } \varepsilon_2' \text{ are not defined} \end{cases}$$

where:

$$\begin{aligned} \varepsilon_1' &= (\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_3 \circ^{\preccurlyeq} ilat(\varepsilon_1) \\ \varepsilon_2' &= \varepsilon_2 \circ^{<:} idom(\varepsilon_1) \\ g_1' &= (g_c \mathbin{\widetilde{\curlyvee}} g) \end{aligned}$$

$$(r4) \quad \mathsf{if}\ \varepsilon_1 b_{g_1}\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 \mid \mu \xrightarrow{\varepsilon g_c} \begin{cases} \mathsf{prot}_{ilbl(\varepsilon_1)g_1} \varepsilon' g' (\varepsilon_2 t_2) \mid \mu & \text{if } b = \mathsf{true} \\ \mathsf{prot}_{ilbl(\varepsilon_1)g_1} \varepsilon' g' (\varepsilon_3 t_3) \mid \mu & \text{if } b = \mathsf{false} \end{cases}$$

where:

$$\begin{aligned} \varepsilon' &= \varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1) \\ g' &= g_c \mathbin{\widetilde{\curlyvee}} g_1 \end{aligned}$$

$$(r5) \qquad \mathsf{ref}^{U}_{\varepsilon_2}\ \varepsilon_1 u \mid \mu \xrightarrow{\varepsilon g_c} \begin{cases} o_\perp \mid \mu[o \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} g_c)] \\ \mathbf{error} \qquad \text{if } (\varepsilon \circ^{\preccurlyeq} \varepsilon_2) \text{ is not defined} \end{cases}$$

where:

$$\begin{aligned} & o \notin dom(\mu) \\ \varepsilon' &= \varepsilon_1 \mathbin{\widetilde{\curlyvee}} (\varepsilon \circ^{\preccurlyeq} \varepsilon_2) \end{aligned}$$

$$(r6) \qquad !\varepsilon_1 o_g \mid \mu \xrightarrow{\varepsilon g_c} \mathsf{prot}_{ilbl(\varepsilon_1)g} \varepsilon' g' (iref(\varepsilon_1) v)$$

where:

$$\begin{aligned} \mu(o) &= v \\ \varepsilon' &= \varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1) \\ g' &= g_c \mathbin{\widetilde{\curlyvee}} g \end{aligned}$$

$$(r7) \qquad \varepsilon_1 o_g :=_{\varepsilon_3} \varepsilon_2 u \mid \mu \xrightarrow{\varepsilon g_c} \begin{cases} \mathsf{unit}_\perp \mid \mu[o \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} (g_c \mathbin{\widetilde{\curlyvee}} g))] \\ \mathbf{error} \qquad \text{if } \varepsilon' \text{ is not defined, or } \varepsilon \lfloor \leq \rfloor ilbl(\varepsilon'') \text{does not hold} \end{cases}$$

where:

$$\begin{aligned} \mu(o) &= \varepsilon'' u' \\ \varepsilon' &= (\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \mathbin{\widetilde{\curlyvee}} ((\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_3 \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1))) \end{aligned}$$

$$\boxed{\longrightarrow_{<:} : \textsc{EvTerm} \times (\textsc{EvTerm} \cup \{\, \mathbf{error} \,\})}$$

$$\varepsilon_1(\varepsilon_2 u) \longrightarrow_{<:} \begin{cases} (\varepsilon_2 \circ^{<:} \varepsilon_1) u \\ \mathbf{error} \qquad \text{if not defined} \end{cases}$$

Fig. 12. $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Dynamic semantics

Definition 2.21 (Operation pattern).

$$\begin{array}{llll} \multicolumn{4}{l}{P^T \in \textsc{GPattern}, P^\ell \in \textsc{LPattern}} \\ P^T & ::= & \_ \mid P^T\ op^T\ P^T & \text{(pattern on types)} \\ op^T & ::= & \dot{\curlyvee} \mid \dot{\curlywedge} \mid \sqcap & \text{(operations on types)} \\ P^\ell & ::= & \_ \mid P^\ell\ op^\ell\ P^\ell & \text{(pattern on labels)} \\ op^\ell & ::= & \curlyvee \mid \curlywedge \mid \sqcap & \text{(operations on labels)} \end{array}$$

$$(\mathrm{R}\longrightarrow)\ \frac{t \mid \mu \xrightarrow{\varepsilon g_c} r \qquad r \in \mathbb{C} \cup \{\, \mathbf{error} \,\}}{t \mid \mu \xmapsto{\varepsilon g_c} r} \qquad (\mathrm{R}f)\ \frac{t \mid \mu \xmapsto{\varepsilon g_c} t' \mid \mu'}{f[t] \mid \mu \xmapsto{\varepsilon g_c} f[t'] \mid \mu'}$$

$$(\mathrm{Rprot})\ \frac{t \mid \mu \xmapsto{\varepsilon' g'_c} t' \mid \mu'}{\mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon t) \mid \mu \xmapsto{\varepsilon g_c} \mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon t') \mid \mu'} \qquad (\mathrm{R}h)\ \frac{\varepsilon v \longrightarrow_{<:} \varepsilon' u}{h[\varepsilon v] \mid \mu \xmapsto{\varepsilon g_c} h[\varepsilon' u] \mid \mu}$$

$$(\mathrm{Rprot}h)\ \frac{\varepsilon v \longrightarrow_{<:} \varepsilon' u}{\mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon v) \mid \mu \xmapsto{\varepsilon g_c} \mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon' u) \mid \mu} \qquad (\mathrm{R}f\mathrm{err})\ \frac{t \mid \mu \xmapsto{\varepsilon g_c} \mathbf{error}}{f[t] \mid \mu \xmapsto{\varepsilon g_c} \mathbf{error}}$$

$$(\mathrm{R}h\mathrm{err})\ \frac{\varepsilon v \longrightarrow_{<:} \mathbf{error}}{h[\varepsilon v] \mid \mu \xmapsto{\varepsilon g_c} \mathbf{error}} \qquad (\mathrm{Rproterr})\ \frac{t \mid \mu \xmapsto{\varepsilon' g'_c} \mathbf{error}}{\mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon t) \mid \mu \xmapsto{\varepsilon g_c} \mathbf{error}}$$

$$(\mathrm{Rprot}h\mathrm{err})\ \frac{\varepsilon v \longrightarrow_{<:} \mathbf{error}}{\mathsf{prot}_{\varepsilon_1 g_1} \varepsilon' g'_c(\varepsilon v) \mid \mu \xmapsto{\varepsilon g_c} \mathbf{error}}$$

Fig. 13. $\mathrm{GSL}^{\varepsilon}_{\mathrm{Ref}}$: Evaluation frames and reduction

$$\frac{\ell_1 \curlyvee \ell'_1 \preccurlyeq \ell_2 \curlywedge \ell'_2}{[\ell_1, \ell_2] \sqcap [\ell'_1, \ell'_2] = [\ell_1 \curlyvee \ell'_1, \ell_2 \curlywedge \ell'_2]} \qquad \langle \imath_1, \imath_2 \rangle \sqcap \langle \imath'_1, \imath'_2 \rangle = \langle \imath_1 \sqcap \imath'_1, \imath_2 \sqcap \imath'_2 \rangle$$

$$\langle \imath_1, \imath_2 \rangle \,\widetilde{\curlyvee}\, \langle \imath'_1, \imath'_2 \rangle = \langle \imath_1 \curlyvee \imath'_1, \imath_2 \curlyvee \imath'_2 \rangle \qquad \langle \imath_1, \imath_2 \rangle \,\widetilde{\curlywedge}\, \langle \imath'_1, \imath'_2 \rangle = \langle \imath_1 \curlywedge \imath'_1, \imath_2 \curlywedge \imath'_2 \rangle$$

$$\frac{\ell_1 \preccurlyeq \ell'_2 \qquad \ell'_1 \preccurlyeq \ell''_2 \qquad \ell_1 \preccurlyeq \ell''_2}{\triangle^{\preccurlyeq}([\ell_1, \ell_2], [\ell'_1, \ell'_2], [\ell''_1, \ell''_2]) = \langle [\ell_1, \ell_2 \curlywedge \ell'_2 \curlywedge \ell''_2], [\ell_1 \curlyvee \ell'_1 \curlyvee \ell''_1, \ell''_2] \rangle}$$

$$\langle \imath_1, \imath_{21} \rangle \circ^{\preccurlyeq} \langle \imath_{22}, \imath_3 \rangle = \triangle^{\preccurlyeq}(\imath_1, \imath_{21} \sqcap \imath_{22}, \imath_3) \qquad \frac{\ell_3 \preccurlyeq \ell'_3}{\langle [\ell_1, \ell_2], [\ell_3, \ell_4] \rangle \lfloor \le \rfloor \langle [\ell'_1, \ell'_2], [\ell'_3, \ell'_4] \rangle}$$

Fig. 14. $\mathrm{GSL}^{\varepsilon}_{\mathrm{Ref}}$: Auxiliary functions for the dynamic semantics (Labels)

2.8 Noninterference definitions

The formal definitions of related values and related computations are presented in Figures 20 and 21 respectively.

Definition 2.22 (Related substitutions). Tuples $\langle \hat{g}_1, \rho_1, \mu_1 \rangle$ and $\langle \hat{g}_2, \rho_2, \mu_2 \rangle$ are related on k steps under Γ, Σ and g_c, notation $\Gamma; \Sigma; g_c \vdash \langle \hat{g}_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \hat{g}_2, \rho_2, \mu_2 \rangle$, if $\rho_i \models \Gamma$, $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$ and

$$\forall x \in dom(\Gamma). \Sigma; g_c \vdash \langle \hat{g}_1, \rho_1(x), \mu_1 \rangle \approx^k_{\ell_o} \langle \hat{g}_2, \rho_2(x), \mu_2 \rangle : \Gamma(x)$$

Definition 2.23 (Semantic Security Typing).

$$\Gamma; \Sigma; \hat{g} \models t : U \iff \begin{array}{l} \forall\, \ell_o \in \textsc{Label}, k \geq 0, \rho_1, \rho_2 \in \textsc{Subst} \text{ and } \mu_1, \mu_2 \in \textsc{Store}, \forall g_c, \hat{g} = \varepsilon g, \\ \varepsilon \vdash g \,\widetilde{\preccurlyeq}\, g_c, \text{ such that } \Sigma \vdash \mu_i \text{ and } \Gamma; \Sigma; g_c \vdash \langle \hat{g}, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \hat{g}, \rho_2, \mu_2 \rangle, \\ \text{we have} \Sigma; g_c \vdash \langle \hat{g}, \rho_1(t), \mu_1 \rangle \approx^k_{\ell_o} \langle \hat{g}, \rho_2(t), \mu_2 \rangle : \mathcal{C}(U) \end{array}$$

Proposition 2.24 (Security Type Soundness). $\Gamma; \Sigma; \hat{g} \vdash t : U \implies \Gamma; \Sigma; \hat{g} \models t : U$

$$\mathsf{Bool}_\iota \sqcap \mathsf{Bool}_{\iota'} = \mathsf{Bool}_{\iota \sqcap \iota'} \qquad \mathsf{Ref}_\iota\ E_1 \sqcap \mathsf{Ref}_{\iota'}\ E_2 = \mathsf{Ref}_{\iota \sqcap \iota'}\ E_1 \sqcap E_2$$

$$(E_{11} \xrightarrow{\iota_2}_{\iota_1} E_{12}) \sqcap (E_{21} \xrightarrow{\iota_2'}_{\iota_1'} E_{22}) = (E_{11} \sqcap E_{21}) \xrightarrow{\iota_2 \sqcap \iota_2'}_{\iota_1 \sqcap \iota_1'} (E_{12} \sqcap E_{22}) \qquad E \sqcap E' \text{ undefined otherwise}$$

$$\mathsf{Bool}_{\iota_1}\ \widetilde{\curlyvee}\ \iota_2 = \mathsf{Bool}_{(\iota_1 \widetilde{\curlyvee} \iota_2)} \qquad E_1 \xrightarrow{\iota_2}_{\iota_1} E_2\ \widetilde{\curlyvee}\ \iota_3 = E_1 \xrightarrow{\iota_2}_{(\iota_1 \widetilde{\curlyvee} \iota_3)} E_2 \qquad \mathsf{Ref}_{\iota_1}\ E\ \widetilde{\curlyvee}\ \iota_2 = \mathsf{Ref}_{(\iota_1 \widetilde{\curlyvee} \iota_2)}\ E$$

$$\mathsf{Bool}_{\iota_1}\ \widetilde{\curlywedge}\ \iota_2 = \mathsf{Bool}_{(\iota_1 \widetilde{\curlywedge} \iota_2)} \qquad E_1 \xrightarrow{\iota_2}_{\iota_1} E_2\ \widetilde{\curlywedge}\ \iota_3 = E_1 \xrightarrow{\iota_2}_{(\iota_1 \widetilde{\curlywedge} \iota_3)} E_2 \qquad \mathsf{Ref}_{\iota_1}\ E\ \widetilde{\curlywedge}\ \iota_2 = \mathsf{Ref}_{(\iota_1 \widetilde{\curlywedge} \iota_2)}\ E$$

$$\langle E_1, E_2 \rangle\ \widetilde{\curlyvee}\ \langle \iota_1, \iota_2 \rangle = \langle E_1\ \widetilde{\curlyvee}\ \iota_1, E_2\ \widetilde{\curlyvee}\ \iota_2 \rangle \qquad \langle E_1, E_2 \rangle\ \widetilde{\curlywedge}\ \langle \iota_1, \iota_2 \rangle = \langle E_1\ \widetilde{\curlywedge}\ \iota_1, E_2\ \widetilde{\curlywedge}\ \iota_2 \rangle$$

$$\mathsf{Bool}_{\iota_1}\ \widetilde{\curlyvee}\ \mathsf{Bool}_{\iota_2} = \mathsf{Bool}_{(\iota_1 \widetilde{\curlyvee} \iota_2)} \qquad E_1 \xrightarrow{\iota_2}_{\iota_1} E_2\ \widetilde{\curlyvee}\ E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' = E_1\ \widetilde{\curlywedge}\ E_1' \xrightarrow{\iota_2 \widetilde{\curlywedge} \iota_2'}_{(\iota_1 \widetilde{\curlyvee} \iota_1')} E_2\ \widetilde{\curlyvee}\ E_2'$$

$$\mathsf{Ref}_{\iota_1}\ E_1\ \widetilde{\curlyvee}\ \mathsf{Ref}_{\iota_1'}\ E_1' = \mathsf{Ref}_{(\iota_1 \widetilde{\curlyvee} \iota_1')}\ E_1 \sqcap E_1' \qquad \mathsf{Bool}_{\iota_1}\ \widetilde{\curlywedge}\ \mathsf{Bool}_{\iota_2} = \mathsf{Bool}_{(\iota_1 \widetilde{\curlywedge} \iota_2)}$$

$$E_1 \xrightarrow{\iota_2}_{\iota_1} E_2\ \widetilde{\curlywedge}\ E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' = E_1\ \widetilde{\curlyvee}\ E_1' \xrightarrow{\iota_2 \widetilde{\curlyvee} \iota_2'}_{(\iota_1 \widetilde{\curlywedge} \iota_1')} E_2\ \widetilde{\curlywedge}\ E_2' \qquad \mathsf{Ref}_{\iota_1}\ E_1\ \widetilde{\curlywedge}\ \mathsf{Ref}_{\iota_1'}\ E_1' = \mathsf{Ref}_{(\iota_1 \widetilde{\curlywedge} \iota_1')}\ E_1 \sqcap E_1'$$

$$\langle E_1, E_2 \rangle\ \widetilde{\curlyvee}\ \langle E_1', E_2' \rangle = \langle E_1\ \widetilde{\curlyvee}\ E_1', E_2\ \widetilde{\curlyvee}\ E_2' \rangle \qquad \langle E_1, E_2 \rangle\ \widetilde{\curlywedge}\ \langle E_1', E_2' \rangle = \langle E_1\ \widetilde{\curlywedge}\ E_1', E_2\ \widetilde{\curlywedge}\ E_2' \rangle$$

$$\frac{\triangle^{\preccurlyeq}(\iota_1, \iota_2, \iota_3) = \langle \iota_1', \iota_3' \rangle}{\triangle^{<:}(\mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2}, \mathsf{Bool}_{\iota_3}) = \langle \mathsf{Bool}_{\iota_1'}, \mathsf{Bool}_{\iota_3'} \rangle}$$

$$\frac{\begin{array}{cc} \triangle^{<:}(E_{31}, E_{21}, E_{11}) = \langle E_{31}', E_{11}' \rangle & \triangle^{<:}(E_{12}, E_{22}, E_{32}) = \langle E_{12}', E_{32}' \rangle \\ \triangle^{\preccurlyeq}(\iota_1, \iota_2, \iota_3) = \langle \iota_1', \iota_3' \rangle & \triangle^{\preccurlyeq}(\iota_{13}, \iota_{12}, \iota_{11}) = \langle \iota_{13}', \iota_{11}' \rangle \end{array}}{\triangle^{<:}(E_{11} \xrightarrow{\iota_{11}}_{\iota_1} E_{12}, E_{21} \xrightarrow{\iota_{12}}_{\iota_2} E_{22}, E_{31} \xrightarrow{\iota_{13}}_{\iota_3} E_{32}) = \langle E_{11}' \xrightarrow{\iota_{11}'}_{\iota_1'} E_{12}', E_{31}' \xrightarrow{\iota_{13}'}_{\iota_3'} E_{32}' \rangle}$$

$$\frac{\triangle^{\preccurlyeq}(\iota_1, \iota_2, \iota_3) = \langle \iota_1', \iota_3' \rangle \quad E_1' = E_1 \sqcap E_2 \quad E_3' = E_2 \sqcap E_3}{\triangle^{<:}(\mathsf{Ref}_{\iota_1}\ E_1, \mathsf{Ref}_{\iota_2}\ E_2, \mathsf{Ref}_{\iota_3}\ E_3) = \langle \mathsf{Ref}_{\iota_1'}\ E_1', \mathsf{Ref}_{\iota_3'}\ E_3' \rangle}$$

$$\langle E_1, E_{21} \rangle \circ^{<:} \langle E_{22}, E_3 \rangle = \triangle^{<:}(E_1, E_{21} \sqcap E_{22}, E_3)$$

Fig. 15. $\mathrm{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Auxiliary functions for the dynamic semantics (Types)

Proof. Proof in Appendix 6. □

$$
\begin{aligned}
ilbl(\langle \mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2} \rangle) &= \langle \iota_1, \iota_2 \rangle \\
ilbl(\langle \mathsf{Unit}_{\iota_1}, \mathsf{Unit}_{\iota_2} \rangle) &= \langle \iota_1, \iota_2 \rangle \\
ilbl(\langle \mathsf{Ref}_{\iota_1}\ U_1, \mathsf{Ref}_{\iota_2}\ U_2 \rangle) &= \langle \iota_1, \iota_2 \rangle \\
ilbl(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle \iota_1, \iota_1' \rangle
\end{aligned}
$$

$$
\begin{aligned}
iref(\langle \mathsf{Ref}_{\iota_1}\ E_1, \mathsf{Ref}_{\iota_2}\ E_2 \rangle) &= \langle E_1, E_2 \rangle \\
iref(\langle E_1, E_2 \rangle) &= \text{undefined otherwise}
\end{aligned}
$$

$$
\begin{aligned}
idom(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle E_1', E_1 \rangle \\
idom(\langle E_1, E_2 \rangle) &= \text{undefined otherwise}
\end{aligned}
$$

$$
\begin{aligned}
icod(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle E_2, E_2' \rangle \\
icod(\langle E_1, E_2 \rangle) &= \text{undefined otherwise}
\end{aligned}
$$

Fig. 16. $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Inversion functions for evidence

$$\boxed{\Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : U}$$

$$(Tx)\ \frac{\Gamma(x) = U}{\Gamma; \Sigma; g_c \vdash x \rightsquigarrow x : U} \qquad (Tb)\ \frac{}{\Gamma; \Sigma; g_c \vdash b_g \rightsquigarrow b_g : \mathsf{Bool}_g}$$

$$(Tu)\ \frac{}{\Gamma; \Sigma; g_c \vdash \mathsf{unit}_g \rightsquigarrow \mathsf{unit}_g : \mathsf{Unit}_g} \qquad (T\lambda)\ \frac{\Gamma; \Sigma; g' \vdash t \rightsquigarrow t' : U_2}{\Gamma; \Sigma; g_c \vdash (\lambda^{g'} x : U_1.t)_g \rightsquigarrow (\lambda^{g'} x : U_1.t')_g : U_1 \xrightarrow{g'}_g U_2}$$

$$(T\oplus)\ \frac{\begin{array}{cc} \Gamma; \Sigma; g_c \vdash t_1 \rightsquigarrow t_1' : \mathsf{Bool}_{g_1} & \Gamma; \Sigma; g_c \vdash t_2 \rightsquigarrow t_2' : \mathsf{Bool}_{g_2} \\ \varepsilon_1 = \mathcal{I}^{\cup}[\![\mathsf{Bool}_{g_1}]\!] & \varepsilon_2 = \mathcal{I}^{\cup}[\![\mathsf{Bool}_{g_2}]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash t_1 \oplus t_2 \rightsquigarrow \varepsilon_1 t_1' \oplus \varepsilon_2 t_2' : \mathsf{Bool}_{g_1 \widetilde{\curlyvee} g_2}}$$

$$(Tapp)\ \frac{\begin{array}{ccc} \Gamma; \Sigma; g_c \vdash t_1 \rightsquigarrow t_1' : U_{11} \xrightarrow{g'}_g U_{12} & & \Gamma; \Sigma; g_c \vdash t_2 \rightsquigarrow t_2' : U_2 \\ \varepsilon_1 = \mathcal{I}^{\cup}[\![U_{11} \xrightarrow{g'}_g U_{12}]\!] & \varepsilon_2 = \mathcal{I}[\![U_2 \lesssim U_{11}]\!] & \varepsilon_3 = \mathcal{I}[\![\widetilde{g_c \curlyvee g \preccurlyeq g'}]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash t_1\ t_2 \rightsquigarrow \varepsilon_1 t_1' \mathbin{@_{\varepsilon_3}} \varepsilon_2 t_2' : U_{12} \mathbin{\widetilde{\curlyvee}} g}$$

$$(Tif)\ \frac{\begin{array}{cccc} \Gamma; \Sigma; g_c \vdash t_1 \rightsquigarrow t_1' : \mathsf{Bool}_g & g_c' = g_c \mathbin{\widetilde{\curlyvee}} g & \Gamma; \Sigma; g_c' \vdash t_2 \rightsquigarrow t_2' : U_2 & \Gamma; \Sigma; g_c' \vdash t_3 \rightsquigarrow t_3' : U_3 \\ \varepsilon_1 = \mathcal{I}^{\cup}[\![\mathsf{Bool}_g]\!] & \varepsilon_2 = \mathcal{I}[\![\widetilde{U_2 <: U_2 \mathbin{\dot{\curlyvee}} U_3}]\!] & & \varepsilon_3 = \mathcal{I}[\![\widetilde{U_3 <: U_2 \mathbin{\dot{\curlyvee}} U_3}]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 \rightsquigarrow \mathsf{if}\ \varepsilon_1 t_1\ \mathsf{then}\ \varepsilon_2 t_2\ \mathsf{else}\ \varepsilon_3 t_3 : (U_2 \mathbin{\widetilde{\dot{\curlyvee}}} U_3) \mathbin{\widetilde{\curlyvee}} g}$$

$$(Tassgn)\ \frac{\begin{array}{ccc} \Gamma; \Sigma; g_c \vdash t_1 \rightsquigarrow t_1' : \mathsf{Ref}_g\ U_1 & & \Gamma; \Sigma; g_c \vdash t_2 \rightsquigarrow t_2' : U_2 \\ \varepsilon_1 = \mathcal{I}^{\cup}[\![\mathsf{Ref}_g\ U_1]\!] & \varepsilon_2 = \mathcal{I}[\![U_2 \lesssim U_1]\!] & \varepsilon_3 = \mathcal{I}[\![\widetilde{g_c \curlyvee g \preccurlyeq \mathit{label}(U_1)}]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash t_1 {:=} t_2 \rightsquigarrow \varepsilon_1 t_1' :=_{\varepsilon_3} \varepsilon_2 t_2' : \mathsf{Unit}_\bot}$$

$$(Tref)\ \frac{\begin{array}{cc} \multicolumn{2}{c}{\Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : U'} \\ \varepsilon_1 = \mathcal{I}[\![U' \lesssim U]\!] & \varepsilon_2 = \mathcal{I}[\![g_c \mathrel{\widetilde{\preccurlyeq}} \mathit{label}(U)]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash \mathsf{ref}^U\ t \rightsquigarrow \mathsf{ref}^U_{\varepsilon_2}\ \varepsilon_1 t' : \mathsf{Ref}_\bot\ U} \qquad (Tderef)\ \frac{\begin{array}{c} \Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : \mathsf{Ref}_g\ U \\ \varepsilon = \mathcal{I}^{\cup}[\![\mathsf{Ref}_g\ U]\!] \end{array}}{\Gamma; \Sigma; g_c \vdash !t \rightsquigarrow !\varepsilon t' : U \mathbin{\widetilde{\curlyvee}} g}$$

$$(T{::})\ \frac{\Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : U_1 \qquad \varepsilon = \mathcal{I}[\![U_1 \lesssim U_2]\!]}{\Gamma; \Sigma; g_c \vdash t :: U_2 \rightsquigarrow \varepsilon t' : U_2}$$

where $\mathcal{I}^{\cup}[\![g]\!] = \mathcal{I}[\![g \mathrel{\widetilde{\preccurlyeq}} g]\!]$ and $\mathcal{I}^{\cup}[\![U]\!] = \mathcal{I}[\![U \lesssim U]\!]$

Fig. 17. $\mathsf{GSL}_{\mathsf{Ref}}$: translation to $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$ terms

$$
\begin{aligned}
bounds(?) &= [\bot, \top] \\
bounds(\ell) &= [\ell, \ell] \\
bounds(x_1 \curlyvee x_2) &= bounds(x_1) \curlyvee bounds(x_2) \\
bounds(x_1 \curlywedge x_2) &= bounds(x_1) \curlywedge bounds(x_2) \\
bounds(x_1 \sqcap x_2) &= bounds(x_1) \sqcap bounds(x_2) \\
bounds(F_1(\overline{x_i}) \curlyvee F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \curlyvee bounds(F_2(\overline{x_i})) \\
bounds(F_1(\overline{x_i}) \curlywedge F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \curlywedge bounds(F_2(\overline{x_i})) \\
bounds(F_1(\overline{x_i}) \sqcap F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \sqcap bounds(F_2(\overline{\ell_i}))
\end{aligned}
$$

$$
\frac{bounds(F_1(\overline{g_i})) = [\ell_1, \ell_2] \qquad bounds(F_2(\overline{g_j})) = [\ell_1', \ell_2']}{\mathcal{I}(F_1(g_1, \ldots \widetilde{g_n) \preccurlyeq F_2(g_{n+1}}, \ldots g_{n+m})) = \langle [\ell_1, \ell_2 \curlywedge \ell_2'], [\ell_1 \curlyvee \ell_1', \ell_2'] \rangle}
$$

where $F_1 : \text{GLabel}^n \to \text{GLabel}$ and $F_2 : \text{GLabel}^m \to \text{GLabel}$.

$$
\mathcal{I}^{\cup}(\widetilde{F(g_1, \ldots, g_n)}) = \mathcal{I}(F(g_1, \ldots \widetilde{, g_n) \preccurlyeq F(g_1}, \ldots, g_n))
$$

Fig. 18. $\mathrm{GSL}^{\varepsilon}_{\mathrm{Ref}}$: Initial evidence for gradual labels

$$
\begin{array}{rcl}
liftP(\_) & = & \_ \\
liftP(P_1^T \mathbin{\dot\vee} P_2^T) & = & liftP(P_1^T) \curlyvee liftP(P_2^T) \\
liftP(P_1^T \mathbin{\dot\wedge} P_2^T) & = & liftP(P_1^T) \curlywedge liftP(P_2^T) \\
liftP(P_1^T \sqcap P_2^T) & = & liftP(P_1^T) \sqcap liftP(P_2^T) \\
invert(\_) & = & \_ \\
invert(P_1^T \mathbin{\dot\vee} P_2^T) & = & invert(P_1^T) \mathbin{\dot\wedge} invert(P_2^T) \\
invert(P_1^T \mathbin{\dot\wedge} P_2^T) & = & invert(P_1^T) \mathbin{\dot\vee} invert(P_2^T) \\
invert(P_1^T \sqcap P_2^T) & = & invert(P_1^T) \sqcap invert(P_2^T) \\
tomeet(\_) & = & \_ \\
tomeet(P_1^T \mathbin{\dot\vee} P_2^T) & = & tomeet(P_1^T) \sqcap tomeet(P_2^T) \\
tomeet(P_1^T \mathbin{\dot\wedge} P_2^T) & = & tomeet(P_1^T) \sqcap tomeet(P_2^T) \\
tomeet(P_1^T \sqcap P_2^T) & = & tomeet(P_1^T) \sqcap tomeet(P_2^T)
\end{array}
$$

$$
\frac{\mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_i}) <: liftP(G_2)(\overline{\ell_j})}]\!] = \langle \iota_1, \iota_2 \rangle}{\mathcal{I}[\![\widetilde{G_1(\overline{\mathsf{Bool}_{g_i}}) \preccurlyeq G_2(\overline{\mathsf{Bool}_{g_j}})}]\!] = \langle \mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2} \rangle}
$$

$$
\frac{\begin{array}{c}\mathcal{I}[\![\widetilde{invert(G_2)(\overline{U_{j1}}) <: invert(G_1)(\overline{U_{i1}})}]\!] = \langle E'_{21}, E'_{11} \rangle \qquad \mathcal{I}[\![\widetilde{G_1(\overline{U_{i2}}) <: G_2(\overline{U_{j2}})}]\!] = \langle E_{12}, E_{22} \rangle \\ \mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_{i1}}) <: liftP(G_2)(\overline{\ell_{j1}})}]\!] = \langle \iota_{11}, \iota_{12} \rangle \\ \mathcal{I}[\![\widetilde{liftP(invert(G_2))(\overline{\ell_{j2}}) <: liftP(invert(G_1))(\overline{\ell_{i2}})}]\!] = \langle \iota_{22}, \iota_{21} \rangle\end{array}}{\mathcal{I}[\![\widetilde{G_1(\overline{U_{i1} \xrightarrow{g_{i2}}_{g_{i1}} U_{i2}}) <: G_2(\overline{U_{j1} \xrightarrow{g_{j2}}_{g_{j1}} U_{j2}})}]\!] = \langle E_{11} \xrightarrow{\iota_{21}}_{\iota_{11}} E_{12}, E_{21} \xrightarrow{\iota_{22}}_{\iota_{12}} E_{22} \rangle}
$$

$$
\frac{\begin{array}{c}\mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_i}) <: liftP(G_2)(\overline{\ell_j})}]\!] = \langle \iota_1, \iota_2 \rangle \\ \mathcal{I}[\![\widetilde{tomeet(G_1)(\overline{U_i}) <: tomeet(G_2)(\overline{U_j})}]\!] = \langle E_1, E_2 \rangle \\ \mathcal{I}[\![\widetilde{tomeet(G_2)(\overline{U_j}) <: tomeet(G_1)(\overline{U_i})}]\!] = \langle E'_2, E'_1 \rangle\end{array}}{\mathcal{I}[\![\widetilde{G_1(\overline{\mathsf{Ref}_{g_i}\ U_i}) <: G_2(\overline{\mathsf{Ref}_{g_j}\ U_j})}]\!] = \langle \mathsf{Ref}_{\iota_1}\ E_1 \sqcap E'_1, \mathsf{Ref}_{\iota_2}\ E_2 \sqcap E'_2 \rangle}
$$

where $G_1 : \text{GLabel}^n \to \text{GLabel}$ and $G_2 : \text{GLabel}^m \to \text{GLabel}$, and $G_1(x_1, ..., x_n) = P_1^T(x_1, ..., x_n)$, $G_2(x_1, ..., x_n) = P_2^T(x_1, ..., x_m)$.

$$
\mathcal{I}^{\circlearrowleft}(\widetilde{F(U_1, ..., U_n)}) = \mathcal{I}[\![\widetilde{F(U_1, ..., U_n) <: F(U_1, ..., U_n)}]\!]
$$

Fig. 19. $\mathrm{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Initial evidence for gradual types

$$\Sigma; g_c \vdash \langle \hat{g}_1, v_1, \mu_1\rangle \approx^{k}_{\ell_o} \langle \hat{g}_2, v_2, \mu_2\rangle : U \iff g_c \vdash \hat{g}_1 \approx_{\ell_o} \hat{g}_2 \wedge \Sigma \vdash \mu_1 \approx^{k}_{\ell_o} \mu_2 \wedge \cdot; \Sigma; \hat{g}_i \vdash v_i : U \wedge$$
$$(\mathsf{obsVal}^{U}_{\ell_o}(v_i) \vee \neg\mathsf{obsVal}^{U}_{\ell_o}(v_i)) \wedge \Big((\mathsf{obsVal}^{U}_{\ell_o}(v_i) \wedge \mathsf{obsEv}^{g'_i}_{\ell_o}(\varepsilon_i)) \implies \mathsf{obsRel}^{\Sigma, g_c, U}_{k, \ell_o}(\hat{g}_1, v_1, \mu_1, \hat{g}_2, v_2, \mu_2)\Big)$$

$$\mathsf{obsRel}^{\Sigma, g_c, U}_{k, \ell_o}(\hat{g}_1, v_1, \mu_1, \hat{g}_2, v_2, \mu_2) \iff \mathit{rval}(v_1) = \mathit{rval}(v_2) \qquad \text{if } U \in \{\mathsf{Bool}_g, \mathsf{Unit}_g, \mathsf{Ref}_g\ U'\}$$

$$\mathsf{obsRel}^{\Sigma, g_c, U_1 \xrightarrow{g_{32}}_{g_{31}} U_2}_{k, \ell_o}(\hat{g}_1, v_1, \mu_1, \hat{g}_2, v_2, \mu_2) \iff \forall j \leq k, \forall U' = U'_1 \xrightarrow{g'_{32}}_{g'_{31}} U'_2, \forall U''_1,$$
$$\forall g'_c, \forall \hat{g}'_i = \varepsilon'_i g'_i, \text{ where } \varepsilon'_i \vdash g'_i \mathrel{\widetilde{\preccurlyeq}} g'_c, \text{ s.t. } \hat{g}_i \leq_{\ell_o} \hat{g}'_i,$$
$$\varepsilon_{11} \vdash U_1 \xrightarrow{g_{32}}_{g_{31}} U_2 \lesssim U', \varepsilon_{12} \vdash U''_1 \lesssim U'_1, \text{ and } \varepsilon_{3i} \vdash \widetilde{g'_c \curlyvee g'_{31}} \preccurlyeq g'_{32}, \text{ we have:}$$
$$\forall v'_i, \mu'_i, \Sigma', \Sigma \subseteq \Sigma', \Sigma'; g_c \vdash \langle \hat{g}_1, v'_1, \mu'_1\rangle \approx^{j}_{\ell_o} \langle \hat{g}_2, v'_2, \mu'_2\rangle : U''_1, \mathit{dom}(\mu_i) \subseteq \mathit{dom}(\mu'_i),$$
$$\Sigma'; g_c \vdash \langle \hat{g}_1, (\varepsilon_{11} v_1 \mathbin{@_{\varepsilon_{31}}} \varepsilon_{12} v'_1), \mu'_1\rangle \approx^{j}_{\ell_o} \langle \hat{g}_2, (\varepsilon_{11} v_2 \mathbin{@_{\varepsilon_{32}}} \varepsilon_{12} v'_2), \mu'_2\rangle : \mathcal{C}(U'_2 \mathbin{\widetilde{\curlyvee}} g'_{31})$$

Fig. 20. Related values

$$\Sigma; g_c \vdash \langle \hat{g}_1, t_1, \mu_1\rangle \approx^{k}_{\ell_o} \langle \hat{g}_2, t_2, \mu_2\rangle : \mathcal{C}(U) \iff g_c \vdash \hat{g}_1 \approx_{\ell_o} \hat{g}_2 \wedge \Sigma \vdash \mu_1 \approx^{k}_{\ell_o} \mu_2 \wedge \forall \hat{g}'_i, \text{ s.t. } \hat{g}_i \leq_{\ell_o} \hat{g}'_i \text{ and}$$
$$\cdot; \Sigma; \hat{g}'_i \vdash t_i : U, \forall j < k, \Big(t_i \mid \mu_i \overset{\hat{g}'_i}{\longmapsto}{}^{j} t'_i \mid \mu'_i \implies \exists \Sigma', \Sigma \subseteq \Sigma'$$
$$\Sigma' \vdash \mu'_1 \approx^{k-j}_{\ell_o} \mu'_2 \wedge ((\mathit{irred}(t'_1) \wedge \mathit{irred}(t'_2)) \implies \Sigma'; g_c \vdash \langle \hat{g}_1, t'_1, \mu'_1\rangle \approx^{k-j}_{\ell_o} \langle \hat{g}_2, t'_2, \mu'_2\rangle : U)\Big)$$

Fig. 21. Related computations

3 STATIC SECURITY TYPING WITH REFERENCES

In this section we present the proof of type preservation for SSL_{Ref} in Sec. 3.1, and the definitions and proof of noninterference for SSL_{Ref} in Sec. 3.2.

3.1 SSL_{Ref}: Static type safety

In this section we present the proof of type safety for SSL_{Ref}.

Definition 3.1 (Well typeness of the store). A store μ is said to be *well typed* with respect to a typing context Γ and a store typing Σ, written $\Gamma;\Sigma \vdash \mu$, if $dom(\mu) = dom(\Sigma)$ and $\forall o \in dom(\mu)$, $\Gamma;\Sigma;\bot \vdash \mu(o) : S$ and $S <: \Sigma(o)$.

LEMMA 3.2. *If* $\Gamma;\Sigma;\ell_c \vdash t : S$ *then* $\forall \ell_c' \preccurlyeq \ell_c, \Gamma;\Sigma;\ell_c' \vdash t : S$.

PROOF. By induction on the derivation of $\Gamma;\Sigma;\ell_c \vdash t : S$. Noticing that none of the inferred types of the type rules depend on ℓ_c.

Case (Sx, Sb, Su, Sl). Trivial because neither the premises and the infered type depend on the security effect.

Case (S⊕). Then $t = b_{1\ell_1} \oplus b_{2\ell_2}$ and

$$\text{(S}\oplus\text{)}\dfrac{\text{(Sb)}\dfrac{}{\Gamma;\Sigma;\ell_c \vdash b_{1\ell_1} : \mathsf{Bool}_{\ell_1}} \quad \text{(Sb)}\dfrac{}{\Gamma;\Sigma;\ell_c \vdash b_{2\ell_2} : \mathsf{Bool}_{\ell_2}}}{\Gamma;\Sigma;\ell_c \vdash b_{1\ell_1} \oplus b_{2\ell_2} : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$, then by induction hypotheses on the premises:

$$\text{(S}\oplus\text{)}\dfrac{\text{(Sb)}\dfrac{}{\Gamma;\Sigma;\ell_c' \vdash b_{1\ell_1} : \mathsf{Bool}_{\ell_1'}} \quad \text{(Sb)}\dfrac{}{\Gamma;\Sigma;\ell_c' \vdash b_{2\ell_2} : \mathsf{Bool}_{\ell_2'}}}{\Gamma;\Sigma;\ell_c' \vdash b_{1\ell_1} \oplus b_{2\ell_2} : \mathsf{Bool}_{(\ell_1' \curlyvee \ell_2')}}$$

where $\ell_1' = \ell_1$ and $\ell_2' = \ell_2$ and the result holds.

Case (Sprot). Then $t = \mathsf{prot}_\ell(t)$ and

$$\text{(Sprot)}\dfrac{\overline{\Gamma;\Sigma;\ell_c \curlyvee \ell \vdash t : S}}{\Gamma;\Sigma;\ell_c \vdash \mathsf{prot}_\ell(t) : S \curlyvee \ell}$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$. Considering that $\ell_c' \curlyvee \ell \preccurlyeq \ell_c \curlyvee \ell$, then by induction hypotheses on the premise:

$$\text{(Sprot)}\dfrac{\overline{\Gamma;\Sigma;\ell_c' \curlyvee \ell \vdash t : S}}{\Gamma;\Sigma;\ell_c' \vdash \mathsf{prot}_\ell(t) : S \curlyvee \ell}$$

and therefore the result holds.

Case (Sapp). Then $t = t_1\ t_2$ and

$$\text{(Sapp)}\dfrac{\text{(S}\lambda\text{)}\dfrac{\mathcal{D}_1}{\Gamma;\Sigma;\ell_c \vdash t_1 : S_{11} \xrightarrow{\ell_c''}_{\ell'} S_{12}} \quad \dfrac{\mathcal{D}_2}{\Gamma;\Sigma;\ell_c \vdash t_2 : S_2} \quad \ell_c \curlyvee \ell \preccurlyeq \ell_c'' \quad S_2 <: S_{11}}{\Gamma;\Sigma;\ell_c \vdash t_1\ t_2 : S_{12} \curlyvee \ell}$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$. Then by using induction hypotheses on the premisess, considering $S_{11}' \xrightarrow{\ell_c'''}_{\ell'} S_{12}' <: S_{11} \xrightarrow{\ell_c''}_{\ell} S_{12}$ and $S_2' <: S_2$. As $S_2 <: S_{11}$ and $S_{11} <: S_{11}'$ then $S_2' <: S_{11}'$. Also, by definition of the join operator $\ell_c' \curlyvee \ell' \preccurlyeq \ell_c \curlyvee \ell \preccurlyeq \ell_c'' \preccurlyeq \ell_c'''$, and then:

$$
\text{(Sapp)}\dfrac{\text{(S}\lambda\text{)}\dfrac{\mathcal{D}_1}{\Gamma; \Sigma; \ell_c' \vdash t_1 : S_{11}' \xrightarrow{\ell_c'''}_{\ell'} S_{12}'} \quad \dfrac{\mathcal{D}_2}{\Gamma; \Sigma; \ell_c' \vdash t_2 : S_2'} \quad \ell_c' \curlyvee \ell' \preccurlyeq \ell_c''' \quad S_2' <: S_{11}'}{\Gamma; \Sigma; \ell_c' \vdash t_1\ t_2 : S_{12}' \curlyvee \ell'}
$$

Where $S_{12}' \curlyvee \ell' = S_{12} \curlyvee \ell$ and the result holds.

Case (Sif-true). Then $t = \mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$ and

$$
\text{(Sif)}\dfrac{\dfrac{\mathcal{D}_0}{\Gamma; \Sigma; \ell_c \vdash \mathsf{true}_\ell : \mathsf{Bool}_\ell} \quad \dfrac{\mathcal{D}_1}{\Gamma; \Sigma; \ell_c \curlyvee \ell \vdash t_1 : S_1} \quad \dfrac{\mathcal{D}_2}{\Gamma; \Sigma; \ell_c \curlyvee \ell \vdash t_2 : S_2}}{\Gamma; \Sigma; \ell_c \vdash \mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1 \,\dot{\curlyvee}\, S_2) \curlyvee \ell}
$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$.As $\ell_c' \curlyvee \ell \preccurlyeq \ell_c \curlyvee \ell$, by induction hypotheses in the premises:

$$
\text{(Sif)}\dfrac{\dfrac{\mathcal{D}_0}{\Gamma; \Sigma; \ell_c' \vdash \mathsf{true}_\ell : \mathsf{Bool}_\ell} \quad \dfrac{\mathcal{D}_1}{\Gamma; \Sigma; \ell_c' \curlyvee \ell \vdash t_1 : S_1'} \quad \dfrac{\mathcal{D}_2}{\Gamma; \Sigma; \ell_c' \curlyvee \ell \vdash t_2 : S_2'}}{\Gamma; \Sigma; \ell_c' \vdash \mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1' \,\dot{\curlyvee}\, S_2') \curlyvee \ell}
$$

where $S_1' = S_1$, $S_2' = S_2$. Then $(S_1' \,\dot{\curlyvee}\, S_2') \curlyvee \ell = (S_1 \,\dot{\curlyvee}\, S_2) \curlyvee \ell$ and therefore the result holds.

Case (Sif-false). Analogous to case (if-true).

Case (Sref). Then $t = \mathsf{ref}^S\ v$ and

$$
\text{(Sref)}\dfrac{\Gamma; \Sigma; \ell_c \vdash v : S' \quad S' <: S \quad \ell_c \preccurlyeq \mathsf{label}(S)}{\Gamma; \Sigma; \ell_c \vdash \mathsf{ref}^S\ v : \mathsf{Ref}_\bot\ S}
$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$. By using induction hypotheses in the premise, considering $\ell_c' \preccurlyeq \ell_c \preccurlyeq$ *label*(S):

$$
\text{(Sref)}\dfrac{\Gamma; \Sigma; \ell_c' \vdash v : S' \quad S' <: S \quad \ell_c' \preccurlyeq \mathsf{label}(S)}{\Gamma; \Sigma; \ell_c' \vdash \mathsf{ref}^S\ v : \mathsf{Ref}_\bot\ S}
$$

and the result holds.

Case (Sderef). Then $t = !o_\ell$ and

$$
\text{(Sderef)}\dfrac{\text{(Sl)}\dfrac{o : S \in \Sigma}{\Gamma; \Sigma; \ell_c \vdash o_\ell : \mathsf{Ref}_\ell\ S}}{\Gamma; \Sigma; \ell_c \vdash !o_\ell : S \curlyvee \ell}
$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$, then by using induction hypotheses in the premise:

$$
\text{(Sderef)}\dfrac{\text{(Sl)}\dfrac{o : S \in \Sigma}{\Gamma; \Sigma; \ell_c' \vdash o_\ell : \mathsf{Ref}_{\ell'}\ S}}{\Gamma; \Sigma; \ell_c' \vdash !o_\ell : S \curlyvee \ell'}
$$

where $\ell' = \ell$. and the result holds.

Case (Sassgn). Then $t = o_\ell{:=}v$ and

$$\text{(Sassgn)}\ \frac{\dfrac{o : S \in \Sigma}{\Gamma; \Sigma; \ell_c \vdash o_\ell : \mathsf{Ref}_\ell\ S} \quad \dfrac{\mathcal{D}}{\Gamma; \Sigma; \ell_c \vdash v : S_2} \quad \begin{matrix} S_2 <: S & \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S) \end{matrix}}{\Gamma; \Sigma; \ell_c \vdash o_\ell{:=}v : \mathsf{Unit}_\perp}$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$. Considering that $\ell_c' \curlyvee \ell \preccurlyeq \ell_c \curlyvee \ell \preccurlyeq label(S)$, and $S_2' <: S_2 <: S$, then:

$$\text{(Sassgn)}\ \frac{\dfrac{o : S \in \Sigma}{\Gamma; \Sigma; \ell_c' \vdash o_\ell : \mathsf{Ref}_\ell\ S} \quad \dfrac{\mathcal{D}}{\Gamma; \Sigma; \ell_c' \vdash v : S_2'} \quad \begin{matrix} S_2' <: S & \ell_c' \curlyvee \ell \preccurlyeq \mathsf{label}(S) \end{matrix}}{\Gamma; \Sigma; \ell_c' \vdash o_\ell{:=}v : \mathsf{Unit}_\perp}$$

but

$$\frac{}{\mathsf{Unit}_\perp <: \mathsf{Unit}_\perp}$$

and therefore the result holds.

Case (S::). Then $t = v :: S$ and

$$\text{(S::)}\ \frac{\dfrac{\mathcal{D}}{\Gamma; \Sigma; \ell_c \vdash v : S_1} \quad S_1 <: S}{\Gamma; \Sigma; \ell_c \vdash v :: S : S}$$

Suppose ℓ_c' such that $\ell_c' \preccurlyeq \ell_c$, then by Lemma 3.4

$$\text{(S::)}\ \frac{\dfrac{\mathcal{D}}{\Gamma; \Sigma; \ell_c' \vdash v : S_1} \quad S_1 <: S}{\Gamma; \Sigma; \ell_c' \vdash v :: S : S}$$

and the result holds.

□

Lemma 3.3 (Substitution). *If* $\Gamma, x : S_1; \Sigma; \ell_c \vdash t : S$ *and* $\Gamma; \Sigma; \ell_c \vdash v : S_1'$ *such that* $S_1' <: S_1$, *then* $\Gamma; \Sigma; \ell_c \vdash [v/x]t : S'$ *such that* $S' <: S$.

Proof. By induction on the derivation of $\Gamma, x : S_1; \Sigma; \ell_c \vdash t : S$. □

Lemma 3.4. *If* $\Gamma; \Sigma; \ell_c \vdash v : S$ *then* $\forall \ell_c', \Gamma; \Sigma; \ell_c' \vdash v : S$.

Proof. By induction on the derivation of $\Gamma; \Sigma; \ell_c \vdash v : S$ observing that for values, there is no premise that depends on ℓ_c. □

Proposition 3.5 ($\longrightarrow$ is well defined). *If* $\cdot; \Sigma; \ell_c \vdash t : S$, $\cdot; \Sigma \vdash \mu$ *and* $\forall\ \ell_r$, *such that* $\ell_r \preccurlyeq \ell_c$, $t \mid \mu \xrightarrow{\ell_r} t' \mid \mu'$ *then, for some* $\Sigma' \supseteq \Sigma$, $\cdot; \Sigma'; \ell_c \vdash t' : S'$, *where* $S' <: S$ *and* $\cdot; \Sigma' \vdash \mu'$.

Proof.

Case (S⊕). Then $t = b_{1\ell_1} \oplus b_{2\ell_2}$ and

$$\text{(S}\oplus\text{)}\ \frac{\text{(Sb)}\ \dfrac{}{\cdot; \Sigma; \ell_c \vdash b_{1\ell_1} : \mathsf{Bool}_{\ell_1}} \quad \text{(Sb)}\ \dfrac{}{\cdot; \Sigma; \ell_c \vdash b_{2\ell_2} : \mathsf{Bool}_{\ell_2}}}{\cdot; \Sigma; \ell_c \vdash b_{1\ell_1} \oplus b_{2\ell_2} : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$\begin{array}{ll} & b_{1\ell_1} \oplus b_{2\ell_2} \mid \mu \\ \xrightarrow{\ell_r} & (b_1 \llbracket \oplus \rrbracket b_2)_{(\ell_1 \curlyvee \ell_2)} \mid \mu \end{array}$$

Then

$$(\mathrm{S}\oplus)\dfrac{}{\ell_c \vdash (b_1 \llbracket \oplus \rrbracket b_2)_{(\ell_1 \curlyvee \ell_2)} : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

Case (Sprot). Then $t = \mathsf{prot}_\ell(v)$ and

$$(\text{Sprot})\dfrac{\dfrac{}{\cdot; \Sigma; \ell_c \curlyvee \ell \vdash v : S}}{\cdot; \Sigma; \ell_c \vdash \mathsf{prot}_\ell(v) : S \curlyvee \ell}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$\mathsf{prot}_\ell(v) \mid \mu \xrightarrow{\ell_r} v \curlyvee \ell \mid \mu$$

But by Lemma 3.2, $\cdot; \Sigma; \ell_c \vdash v : S$.

$$\dfrac{}{\cdot; \Sigma; \ell_c \vdash v \curlyvee \ell : S \curlyvee \ell}$$

and the result holds.

Case (Sapp). Then $t = (\lambda^{\ell'_c} x : S_{11}.t)_\ell \; v$ and

$$(\text{Sapp})\dfrac{(\text{S}\lambda)\dfrac{\dfrac{\mathcal{D}_1}{\cdot, x : S_{11}; \Sigma; \ell'_c \vdash t : S_{12}}}{\cdot; \Sigma; \ell_c \vdash (\lambda^{\ell'_c} x : S_{11}.t)_\ell : S_{11} \xrightarrow{\ell'_c}_\ell S_{12}} \quad \dfrac{\mathcal{D}_2}{\cdot; \Sigma; \ell_c \vdash v : S_2} \quad \ell_c \curlyvee \ell \preccurlyeq \ell'_c \quad S_2 <: S_{11}}{\cdot; \Sigma; \ell_c \vdash (\lambda^{\ell'_c} x : S_{11}.t)_\ell \; v : S_{12} \curlyvee \ell}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, and

$$(\lambda^{\ell'_c} x : S_{11}.t)_\ell \; v \mid \mu \xrightarrow{\ell_r} \mathsf{prot}_\ell([v/x]t) \mid \mu$$

But as $\ell_c \curlyvee \ell \preccurlyeq \ell'_c$ then by Lemma 3.2, $\cdot; \Sigma; \ell_c \curlyvee \ell \vdash t : S'_{12}$, where $S'_{12} <: S_{12}$.
By Lemma 3.3 and Lemma 3.4, $\cdot; \Sigma; \ell_c \curlyvee \ell \vdash [v/x]t : S''_{12}$, where $S''_{12} <: S'_{12} <: S_{12}$. Then

$$(\text{Sprot})\dfrac{\dfrac{\mathcal{D}'_1}{\cdot; \Sigma; \ell_c \curlyvee \ell \vdash [v/x]t : S''_{12}}}{\cdot; \Sigma; \ell_c \vdash \mathsf{prot}_\ell([v/x]t) : S''_{12} \curlyvee \ell}$$

Where $S''_{12} \curlyvee \ell <: S_{12} \curlyvee \ell$ and the result holds.

Case (Sif-true). Then $t = \mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$ and

$$(\text{Sif})\dfrac{\dfrac{\mathcal{D}_0}{\cdot; \Sigma; \ell_c \vdash \mathsf{true}_\ell : \mathsf{Bool}_\ell} \quad \dfrac{\mathcal{D}_1}{\cdot; \Sigma; \ell_c \curlyvee \ell \vdash t_1 : S_1} \quad \dfrac{\mathcal{D}_2}{\cdot; \Sigma; \ell_c \curlyvee \ell \vdash t_2 : S_2}}{\cdot; \Sigma; \ell_c \vdash \mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1 \,\dot{\curlyvee}\, S_2) \curlyvee \ell}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then if

$$\mathsf{if}\ \mathsf{true}_\ell\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid \mu \xrightarrow{\ell_r} \mathsf{prot}_\ell(t_1) \mid \mu$$

Then

$$\text{(Sprot)}\dfrac{\dfrac{\mathcal{D}_1}{\cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_1 : S_1}}{\cdot;\Sigma;\ell_c \vdash \mathsf{prot}_\ell(t_1) : S_1 \curlyvee \ell}$$

and by definition of the join operator, $S_1 \curlyvee \ell <: (S_1 \dot{\curlyvee} S_2) \curlyvee \ell$ and the result holds.

Case (Sif-false). Analogous to case (if-true).

Case (Sref). Then $t = \mathsf{ref}^S\ v$ and

$$\text{(Sref)}\dfrac{\cdot;\Sigma;\ell_c \vdash v : S' \qquad S' <: S \qquad \ell_c \preccurlyeq \mathsf{label}(S)}{\cdot;\Sigma;\ell_c \vdash \mathsf{ref}^S\ v : \mathsf{Ref}_\bot\ S}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$\mathsf{ref}^S\ v \mid \mu \xrightarrow{\ell_r} o_\bot \mid \mu[o \mapsto v \curlyvee \ell_r]$$

where $o \notin dom(\mu)$.
Let us take $\Sigma' = \Sigma, o : S$ and let us call $\mu' = \mu[o \mapsto v \curlyvee \ell_r]$. Then as $dom(\mu) = dom(\Sigma)$ then $dom(\mu') = dom(\Sigma')$. Also, as $\ell_r \preccurlyeq \ell_c \preccurlyeq label(S)$ then by Lemma 3.4, $\cdot;\Sigma';\bot \vdash v : S' \curlyvee \ell_r$ and $S' \curlyvee \ell_r <: \Sigma(o) = S$. Therefore $\cdot;\Sigma' \vdash \mu'$.

Then

$$\text{(Sl)}\dfrac{o : S \in \Sigma'}{\cdot;\Sigma';\ell_c \vdash o_\bot : \mathsf{Ref}_\bot\ S}$$

and the result holds.

Case (Sderef). Then $t = !o_\ell$ and

$$\text{(Sderef)}\dfrac{\text{(Sl)}\dfrac{o : S \in \Sigma}{\cdot;\Sigma;\ell_c \vdash o_\ell : \mathsf{Ref}_\ell\ S}}{\cdot;\Sigma;\ell_c \vdash !o_\ell : S \curlyvee \ell}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$!o_\ell \mid \mu \xrightarrow{\ell_r} v \curlyvee \ell \mid \mu \text{ where } \mu(o) = v$$

Also $\cdot;\Sigma \vdash \mu$ then $\cdot;\Sigma;\bot \vdash \mu(o) : S'$ and $S' <: S$. By Lemma 3.4, $\cdot;\Sigma;\ell_c \vdash v : S'$

$$\dfrac{}{\cdot;\Sigma;\ell_c \vdash v \curlyvee \ell : S' \curlyvee \ell}$$

But $S' \curlyvee \ell <: S \curlyvee \ell$ and the result holds.

Case (Sassgn). Then $t = o_\ell{:=}v$ and

$$\text{(Sassgn)}\dfrac{\begin{array}{cc}\dfrac{o : S \in \Sigma}{\cdot;\Sigma;\ell_c \vdash o_\ell : \mathsf{Ref}_\ell\ S} & \dfrac{\mathcal{D}}{\cdot;\Sigma;\ell_c \vdash v : S_2} \\ S_2 <: S & \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S)\end{array}}{\cdot;\Sigma;\ell_c \vdash o_\ell{:=}v : \mathsf{Unit}_\bot}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$o_\ell{:=}v \mid \mu \xrightarrow{\ell_r} \mathsf{unit}_\bot \mid \mu[o \mapsto v \curlyvee \ell_r \curlyvee \ell]$$

Let us call $\mu' = \mu[o \mapsto v \curlyvee \ell_r \curlyvee \ell]$. Also $\cdot;\Sigma \vdash \mu$ then $dom(\mu') = dom(\Sigma)$, and $\cdot;\Sigma;\ell_c \vdash v : S_2$ where $S_2 <: S$. Therefore $\cdot;\Sigma;\ell_c \vdash v \curlyvee \ell_r \curlyvee \ell : S_2 \curlyvee \ell_r \curlyvee \ell$. But $\ell_r \curlyvee \ell \preccurlyeq \ell_c \curlyvee \ell \preccurlyeq label(S)$, then $S_2 \curlyvee \ell_r \curlyvee \ell <: S$ and therefore $\cdot;\Sigma \vdash \mu'$. Also

$$\text{(Su)}\dfrac{}{\cdot;\Sigma;\ell_c \vdash \mathsf{unit}_\bot : \mathsf{Unit}_\bot}$$

but

$$\frac{}{\mathsf{Unit}_\bot <: \mathsf{Unit}_\bot}$$

and therefore the result holds.

Case (S::). Then $t = v :: S$ and

$$(\text{S::})\frac{\dfrac{\mathcal{D}}{\cdot;\Sigma;\ell_c \vdash v : S_1} \qquad S_1 <: S}{\cdot;\Sigma;\ell_c \vdash v :: S : S}$$

Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$v :: S \mid \mu \xrightarrow{\ell_r} v \curlyvee label(S) \mid \mu$$

But $S_1 <: S$ then $S_1 \curlyvee S = S$ and therefore $S_1 \curlyvee label(S) = S$. Therefore:

$$\frac{}{\Gamma;\Sigma;\ell_c \vdash v \curlyvee label(S) : S}$$

and the result holds.

□

Proposition 3.6 (Canonical forms). *Consider a value v such that $\cdot;\Sigma;\ell_c \vdash v : S$. Then:*

(1) If $S = \mathsf{Bool}_\ell$ then $v = b_\ell$ for some b.
(2) If $S = \mathsf{Unit}_\ell$ then $v = \mathsf{unit}_\ell$.
(3) If $S = S_1 \xrightarrow{\ell_c'}_\ell S_2$ then $v = (\lambda^{\ell_c'} x : S_1.t_2)$ for some t_2 and ℓ_c'.
(4) If $S = \mathsf{Ref}_\ell\ S$ then $v = o_\ell$ for some location o.

Proof. By inspection of the type derivation rules. □

Proposition 3.7 (Type Safety). *If $\cdot;\Sigma;\ell_c \vdash t : S$ then either*

- *t is a value v*
- *for any store μ such that $\Sigma \vdash \mu$ and any $\ell_c' \preccurlyeq \ell_c$, we have $t \mid \mu \xmapsto{\ell_c'} t' \mid \mu'$ and $\cdot;\Sigma';\ell_c \vdash t' : S'$ for some $S' <: S$, and some $\Sigma' \supseteq \Sigma$ such that $\Sigma' \vdash \mu'$.*

Proof. By induction on the structure of t.

Case (Sb, Su, Sλ, Sl). t is a value.

Case (Sprot). Then $t = \mathsf{prot}_\ell(t)$ and

$$(\text{Sprot})\frac{\cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_1 : S_1}{\cdot;\Sigma;\ell_c \vdash \mathsf{prot}_\ell(t_1) : S_1 \curlyvee \ell}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then by (R$\longrightarrow$) and Canonical Forms (Lemma 3.6). $t \mid \mu \xmapsto{\ell_r} t' \mid \mu$ and by Prop 3.5, $\cdot;\Sigma;\ell_c \vdash t' : S'$ where $S' <: S$ and the result holds.
(2) Suppose ℓ_r such that $\ell_r \preccurlyeq \ell_c$, then

$$(\text{Rprot})\frac{t_1 \mid \mu \xmapsto{\ell_r \curlyvee \ell} t_2 \mid \mu'}{\mathsf{prot}_\ell(t_1) \mid \mu \xmapsto{\ell_r} \mathsf{prot}_\ell(t_2) \mid \mu'}$$

As $\ell_r \preccurlyeq \ell_c$ then $\ell_r \curlyvee \ell \preccurlyeq \ell_c \curlyvee \ell$. Using induction hypotheses $\cdot;\Sigma';\ell_c \curlyvee \ell \vdash t_2 : S_1'$ where $S_1' <: S_1$ and $\cdot;\Sigma' \vdash \mu'$. Therefore

$$\text{(Sprot)}\frac{\cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_2 : S_1'}{\cdot;\Sigma;\ell_c \vdash \mathsf{prot}_\ell(t_2) : S_1' \curlyvee \ell}$$

but $S_1' \curlyvee \ell <: S_1 \curlyvee \ell$ and the result holds.

Case (S$\oplus$). Then $t = t_1 \oplus t_2$ and

$$\text{(S}\oplus\text{)}\frac{\cdot;\Sigma;\ell_c \vdash t_1 : \mathsf{Bool}_{\ell_1} \qquad \cdot;\Sigma;\ell_c \vdash t_2 : \mathsf{Bool}_{\ell_2}}{\cdot;\Sigma;\ell_c \vdash t_1 \oplus t_2 : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then by induction on t_2 one of the following holds:
 (a) t_2 is a value. Then by Canonical Forms (Lemma 3.6)

$$\text{(R}\longrightarrow\text{)}\frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu}$$

 and by Prop 3.5, $\cdot;\Sigma;\ell_c \vdash t' : S'$, where $S' <: S$, therefore the result holds.
 (b) $t_2 \mid \mu \xmapsto{\ell_r{}'} t_2' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot;\Sigma';\ell_c \vdash t_2 : \mathsf{Bool}_{\ell_2'}$, where $\mathsf{Bool}_{\ell_2'} <: \mathsf{Bool}_{\ell_2}$ and $\cdot;\Sigma' \vdash \mu'$.
 Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1 \oplus t_2' \mid \mu'$ and:

$$\text{(S}\oplus\text{)}\frac{\cdot;\Sigma;\ell_c \vdash t_1 : \mathsf{Bool}_{\ell_1} \qquad \cdot;\Sigma;\ell_c \vdash t_2' : \mathsf{Bool}_{\ell_2'}}{\cdot;\Sigma;\ell_c \vdash t_1 \oplus t_2' : \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2')}}$$

 but

$$\frac{(\ell_1 \curlyvee \ell_2') \preccurlyeq (\ell_1 \curlyvee \ell_2)}{\mathsf{Bool}_{(\ell_1 \curlyvee \ell_2')} <: \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

 and the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypotheses, $\cdot;\Sigma';\ell_c \vdash t_1' : \mathsf{Bool}_{\ell_1'}$ where $\mathsf{Bool}_{\ell_1'} <: \mathsf{Bool}_{\ell_1}$, and $\cdot;\Sigma \vdash \mu'$. Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1' \oplus t_2 \mid \mu'$ and:

$$\text{(S}\oplus\text{)}\frac{\cdot;\Sigma;\ell_c \vdash t_1' : \mathsf{Bool}_{\ell_1'} \qquad \cdot;\Sigma;\ell_c \vdash t_2 : \mathsf{Bool}_{\ell_2}}{\cdot;\Sigma;\ell_c \vdash t_1' \oplus t_2 : \mathsf{Bool}_{(\ell_1' \curlyvee \ell_2)}}$$

 but

$$\frac{(\ell_1' \curlyvee \ell_2) \preccurlyeq (\ell_1 \curlyvee \ell_2)}{\mathsf{Bool}_{(\ell_1' \curlyvee \ell_2)} <: \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}}$$

 and the result holds.

Case (Sapp). Then $t = t_1\ t_2$, $S = S_{12} \curlyvee \ell$ and

$$\text{(Sapp)}\frac{\begin{array}{cc}\cdot;\Sigma;\ell_c \vdash t_1 : S_{11} \xrightarrow{\ell_c'}_{\ell} S_{12} & \cdot;\Sigma;\ell_c \vdash t_2 : S_2 \\ S_2 <: S_{11} & \ell_c \curlyvee \ell \preccurlyeq \ell_c'\end{array}}{\cdot;\Sigma;\ell_c \vdash t_1\ t_2 : S_{12} \curlyvee \ell}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then by Canonical Forms (Lemma 3.6), and induction on t_2 one of the following holds:

(a) t_2 is a value. Then by Canonical Forms (Lemma 3.6)

$$\text{(R}{\rightarrow}\text{)}\ \frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu}$$

and by Prop 3.5 $\cdot;\Sigma;\ell_c \vdash t' : S'$, where $S' <: S$, therefore the result holds.

(b) $t_2 \mid \mu \xmapsto{\ell_r{}'} t_2' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot;\Sigma';\ell_c \vdash t_2 : S_2'$, where $S_2' <: S_2$ and $\cdot;\Sigma' \vdash \mu'$.

Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1\ t_2' \mid \mu'$. But $S_2' <: S_2 <: S_{11}$ and then:

$$\text{(Sapp)}\ \frac{\begin{array}{cc}\cdot;\Sigma;\ell_c \vdash t_1 : S_{11} \xrightarrow{\ell_c'}_{\ell} S_{12} & \cdot;\Sigma;\ell_c \vdash t_2' : S_2' \\ S_2' <: S_{11} & \ell_c \curlyvee \ell \preccurlyeq \ell_c'\end{array}}{\cdot;\Sigma;\ell_c \vdash t_1\ t_2 : S_{12} \curlyvee \ell}$$

and the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypotheses, $\cdot;\Sigma';\ell_c \vdash t_1' : S_{11}' \xrightarrow{\ell_c''}_{\ell'} S_{12}'$ where $S_{11}' \xrightarrow{\ell_c''}_{\ell'} S_{12}' <: S_{11} \xrightarrow{\ell_c'}_{\ell} S_{12}$, and $\cdot;\Sigma' \vdash \mu'$. Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1'\ t_2 \mid \mu'$. By definition of subtyping, $S_2 <: S_{11} <: S_{11}'$, $\ell_c' \preccurlyeq \ell_c''$ and $\ell' \preccurlyeq \ell$. Therefore $\ell_c \curlyvee \ell' \preccurlyeq \ell_c \curlyvee \ell \preccurlyeq \ell_c' \preccurlyeq \ell_c''$. Then

$$\text{(Sapp)}\ \frac{\begin{array}{cc}\cdot;\Sigma;\ell_c \vdash t_1' : S_{11}' \xrightarrow{\ell_c''}_{\ell'} S_{12}' & \cdot;\Sigma;\ell_c \vdash t_2 : S_2 \\ S_2 <: S_{11}' & \ell_c \curlyvee \ell' \preccurlyeq \ell_c''\end{array}}{\cdot;\Sigma;\ell_c \vdash t_1'\ t_2 : S_{12}' \curlyvee \ell'}$$

but $S_{12}' \curlyvee \ell' <: S_{12} \curlyvee \ell$ and the result holds.

Case (Sif). Then $t = \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$ and

$$\text{(Sif)}\ \frac{\begin{array}{c}\cdot;\Sigma;\ell_c \vdash t_0 : \mathsf{Bool}_{\ell} \\ \cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_1 : S_1 \qquad \cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_2 : S_2\end{array}}{\cdot;\Sigma;\ell_c \vdash \mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1 \mathbin{\dot\curlyvee} S_2) \curlyvee \ell}$$

By induction hypotheses, one of the following holds:

(1) t_0 is a value. Then by Canonical Forms (Lemma 3.6)

$$\text{(R}{\rightarrow}\text{)}\ \frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu}$$

and by Prop 3.5, $\cdot;\Sigma;\ell_c \vdash t' : S'$, where $S' <: S$, therefore the result holds.

(2) $t_0 \mid \mu \xmapsto{\ell_r{}'} t_0' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induciton hypothesis, $\cdot;\Sigma;\ell_c \vdash t_0' : \mathsf{Bool}_{\ell'}$, where $\mathsf{Bool}_{\ell'} <: \mathsf{Bool}_{\ell}$ and $\cdot;\Sigma \vdash \mu'$. Then by (Sf), $t \mid \mu \xmapsto{\ell_r} \mathsf{if}\ t_0'\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \mid \mu'$. As $\ell_c \curlyvee \ell' \preccurlyeq \ell_c \curlyvee \ell$, by Lemma 3.2, $\cdot;\Sigma;\ell_c \curlyvee \ell' \vdash t_1 : S_1'$ and $\cdot;\Sigma;\ell_c \curlyvee \ell' \vdash t_2 : S_2'$, where $S_1' <: S_1$ and $S_2' <: S_2$. Therefore:

$$\text{(Sif)}\ \frac{\begin{array}{c}\cdot;\Sigma;\ell_c \vdash t_0' : \mathsf{Bool}_{\ell'} \\ \cdot;\Sigma;\ell_c \curlyvee \ell' \vdash t_1 : S_1' \qquad \cdot;\Sigma;\ell_c \curlyvee \ell \vdash t_2 : S_2'\end{array}}{\cdot;\Sigma;\ell_c \vdash \mathsf{if}\ t_0'\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : (S_1' \mathbin{\dot\curlyvee} S_2') \curlyvee \ell'}$$

but by definition of join and subtyping $(S_1' \mathbin{\dot\curlyvee} S_2') \curlyvee \ell' <: (S_1 \mathbin{\dot\curlyvee} S_2) \curlyvee \ell$ and the result holds.

Case (S::). Then $t = t_1 :: S_2$ and

$$\text{(S::)}\frac{\cdot; \Sigma; \ell_c \vdash t_1 : S_1 \qquad S_1 <: S_2}{\cdot; \Sigma; \ell_c \vdash t_1 :: S_2 : S_2}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then

$$\text{(R}\rightarrow\text{)}\frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu}$$

and by Prop 3.5, $\cdot; \Sigma; \ell_c \vdash t' : S'$, where $S' <: S$, therefore the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot; \Sigma; \ell_c \vdash t_1' : S_1'$, where $S_1' <: S_1$ and $\cdot; \Sigma' \vdash \mu'$. Then by $(\text{S}f)$, $t \mid \mu \xmapsto{\ell_r} t_1' :: S_2 \mid \mu'$. Also, $S_1' <: S_1 <: S_2$ and therefore:

$$\text{(S::)}\frac{\cdot; \Sigma; \ell_c \vdash t_1' : S_1' \qquad S_1' <: S_2}{\cdot; \Sigma; \ell_c \vdash t_1' :: S_2 : S_2}$$

and the result holds.

Case (Sref). Then $t = \mathsf{ref}^S\ t$ and

$$\text{(Sref)}\frac{\cdot; \Sigma; \ell_c \vdash t_1 : S_1' \qquad S_1' <: S_1 \qquad \ell_c \preccurlyeq \mathsf{label}(S_1)}{\cdot; \Sigma; \ell_c \vdash \mathsf{ref}^{S_1, \ell_c}\ t_1 : \mathsf{Ref}_\perp\ S_1}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then

$$\text{(R}\rightarrow\text{)}\frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu'}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu'}$$

and by Prop 3.5, $\cdot; \Sigma; \ell_c \vdash t' : S'$, where $S' <: S$ and $\cdot; \Sigma' \vdash \mu'$, therefore the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot; \Sigma; \ell_c \vdash t_1' : S_1''$ where $S_1'' <: S_1'$ and $\cdot; \Sigma' \vdash \mu'$. Then by $(\text{S}f)$, $t \mid \mu \xmapsto{\ell_r} \mathsf{ref}^{S_1}\ t_1' \mid \mu'$ and:

$$\text{(Sref)}\frac{\cdot; \Sigma; \ell_c \vdash t_1' : S_1'' \qquad S_1'' <: S_1 \qquad \ell_c \preccurlyeq \mathsf{label}(S_1)}{\cdot; \Sigma; \ell_c \vdash \mathsf{ref}^{S_1}\ t_1' : \mathsf{Ref}_\perp\ S_1}$$

and the result holds.

Case (Sderef). Then $t = !t_1$ and

$$\text{(Sderef)}\frac{\cdot; \Sigma; \ell_c \vdash t_1 : \mathsf{Ref}_\ell\ S_1}{\cdot; \Sigma; \ell_c \vdash !t_1 : S_1 \curlyvee \ell}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then by Canonical Forms (Lemma 3.6)

$$\text{(R}\rightarrow\text{)}\frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu}$$

and by Prop 3.5, $\cdot; \Sigma; \ell_c \vdash t' : S'$, where $S' <: S$, therefore the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot; \Sigma; \ell_c \vdash t_1' : \mathsf{Ref}_{\ell'}\ S_1$ where $\mathsf{Ref}_{\ell'}\ S_1 <: \mathsf{Ref}_{\ell}\ S_1$ and $\cdot; \Sigma' \vdash \mu'$. Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t' \mid \mu'$ and:

$$\text{(Sderef)}\frac{\cdot; \Sigma; \ell_c \vdash t_1' : \mathsf{Ref}_{\ell'}\ S_1}{\cdot; \Sigma; \ell_c \vdash !t_1' : S_1 \curlyvee \ell'}$$

but $S_1 \curlyvee \ell' <: S_1 \curlyvee \ell$ and the result holds.

Case (Sasgn). Then $t = t_1 := t_2$ and

$$\text{(Sasgn)}\frac{\begin{array}{cc}\cdot; \Sigma; \ell_c \vdash t_1 : \mathsf{Ref}_{\ell}\ S_1 & \cdot; \Sigma; \ell_c \vdash t_2 : S_2 \\ S_2 <: S_1 & \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S_1)\end{array}}{\cdot; \Sigma; \ell_c \vdash t_1 := t_2 : \mathsf{Unit}_{\bot}}$$

By induction hypotheses, one of the following holds:

(1) t_1 is a value. Then by Canonical Forms (Lemma 3.6), and induction on t_2 one of the following holds:

(a) t_2 is a value. Then by Canonical Forms (Lemma 3.6)

$$\frac{t \mid \mu \xrightarrow{\ell_r} t' \mid \mu'}{t \mid \mu \xmapsto{\ell_r} t' \mid \mu'}$$

and by Prop 3.5, $\cdot; \Sigma; \ell_c \vdash t' : S'$, where $S' <: S$ and $\cdot; \Sigma' \vdash \mu'$, therefore the result holds.

(b) $t_2 \mid \mu \xmapsto{\ell_r{}'} t_2' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypothesis, $\cdot; \Sigma'; \ell_c \vdash t_2 : S_2'$ where $S_2' <: S_2$ and $\cdot; \Sigma' \vdash \mu'$.
Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1 := t_2' \mid \mu'$. As $S_2' <: S_2 <: S_1$, then:

$$\text{(Sasgn)}\frac{\begin{array}{cc}\cdot; \Sigma; \ell_c \vdash t_1 : \mathsf{Ref}_{\ell}\ S_1 & \cdot; \Sigma; \ell_c \vdash t_2' : S_2' \\ S_2' <: S_1 & \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S_1)\end{array}}{\cdot; \Sigma; \ell_c \vdash t_1 := t_2' : \mathsf{Unit}_{\bot}}$$

and the result holds.

(2) $t_1 \mid \mu \xmapsto{\ell_r} t_1' \mid \mu'$ for all $\ell_r{}'$ such that $\ell_r{}' \preccurlyeq \ell_c$, in particular we pick $\ell_r{}' = \ell_r$. Then by induction hypotheses, $\cdot; \Sigma'; \ell_c \vdash t_1' : \mathsf{Ref}_{\ell'}\ S_1$, where $\mathsf{Ref}_{\ell'}\ S_1 <: \mathsf{Ref}_{\ell}\ S_1$ and $\cdot; \Sigma' \vdash \mu'$. Then by (Sf), $t \mid \mu \xmapsto{\ell_r} t_1' := t_2 \mid \mu'$. As $\ell' \preccurlyeq \ell$ then $\ell_c \curlyvee \ell' \preccurlyeq \ell_c \curlyvee \ell \preccurlyeq \mathsf{label}(S_1)$, and therefore:

$$\text{(Sasgn)}\frac{\begin{array}{cc}\cdot; \Sigma; \ell_c \vdash t_1' : \mathsf{Ref}_{\ell'}\ S_1 & \cdot; \Sigma; \ell_c \vdash t_2 : S_2 \\ S_2 <: S_1 & \ell_c \curlyvee \ell' \preccurlyeq \mathsf{label}(S_1)\end{array}}{\cdot; \Sigma; \ell_c \vdash t_1 := t_2 : \mathsf{Unit}_{\bot}}$$

and the result holds.

□

3.2 SSL$_{\mathsf{Ref}}$: Noninterference

In this section we present the proof of noninterference for SSL$_{\mathsf{Ref}}$. Section 3.3 present some auxiliary definitions and section 3.4 present the proof of noninterference.

$$\Sigma \vdash \langle \ell_1, v_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, v_2, \mu_2 \rangle : S \iff \ell_1 \approx_{\ell_o} \ell_2 \wedge \Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \Sigma; \ell_i \vdash v_i : S'_i, S'_i <: S, \wedge \left(\mathsf{obs}_{\ell_o}(\ell_i, S) \implies \mathsf{obsRel}^{\Sigma,S}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, , v_2, \mu_2) \right)$$

$$\mathsf{obsRel}^{\Sigma,S}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, v_2, \mu_2) \iff (rval(v_1) = rval(v_2)) \qquad \text{if } S \in \{\mathsf{Bool}_g, \mathsf{Unit}_g, \mathsf{Ref}_g\ S'\}$$

$$\mathsf{obsRel}^{\Sigma, S_1 \xrightarrow{\ell'}_{\ell} S_2}_{k,\ell_o}(\ell_1, v_1, \mu_1, \ell_2, v_2, \mu_2) \iff \forall j \leq k. \forall \Sigma \subseteq \Sigma', \Sigma' \vdash \langle \ell_1, v'_1, \mu'_1 \rangle \approx^j_{\ell_o} \langle \ell_2, v'_2, \mu'_2 \rangle : S_1, \Sigma' \vdash \langle \ell_1, v_1\ v'_1, \mu'_1 \rangle \approx^j_{\ell_o} \langle \ell_2, v_2\ v'_2, \mu'_2 \rangle : \mathcal{C}(S_2 \mathbin{\widetilde{\curlyvee}} g)$$

$$\Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S) \iff \ell_1 \approx_{\ell_o} \ell_2 \wedge \Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \Sigma; \ell_i \vdash t_i : S'_i, S'_i <: S, \forall j < k \left(t_i \mid \mu_i \xrightarrow{\ell_i}{}^{j} t'_i \mid \mu'_i \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu'_1 \approx^{k-j}_{\ell_o} \mu'_2 \wedge (irred(t'_i) \implies \Sigma' \vdash \langle \ell_1, t'_1, \mu'_1 \rangle \approx^{k-j}_{\ell_o} \langle \ell_2, t'_2, \mu'_2 \rangle : S) \right)$$

$$\begin{array}{rcl}
\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2 & \iff & \Sigma \vdash \mu_i \wedge \forall \ell_i, \ell_1 \approx_{\ell_o} \ell_2, j < k, \forall o \in dom(\mu_1) \cap dom(\mu_2) \\
& & \Sigma \vdash \langle \ell_1, \mu_1(o), \mu_1 \rangle \approx^j_{\ell_o} \langle \ell_2, \mu_2(o), \mu_2 \rangle : \Sigma(o) \\
\ell_1 \approx_{\ell_o} \ell_2 & \iff & \mathsf{obs}_{\ell_o}(\ell_i) \vee \neg \mathsf{obs}_{\ell_o}(\ell_i) \\
\mu_1 \twoheadrightarrow \mu_2 & \iff & dom(\mu_1) \subseteq dom(\mu_2) \\
\mathsf{obs}_{\ell_o}(\ell, S) & \iff & \mathsf{obs}_{\ell_o}(\ell) \wedge \mathsf{obs}_{\ell_o}(label(S)) \\
\mathsf{obs}_{\ell_o}(\ell) & \iff & \ell \preccurlyeq \ell_o
\end{array}$$

Fig. 22. Security logical relations

3.3 Definitions

To define the fundamental property of the step-indexed logical relations we first define how to relate substitutions:

Definition 3.8. Let ρ be a substitution, Γ and Σ a type substitutions. We say that substitution ρ satisfy environment Γ and Σ, written $\rho \models \Gamma; \Sigma$, if and only if $dom(\rho) = \Gamma$ and $\forall x \in dom(\Gamma), \forall \ell_c, \Gamma; \Sigma; \ell_c \vdash \rho(x) : S'$, where $S' <: \Gamma(x)$.

Definition 3.9 (Related substitutions). Tuples $\langle \ell_1, \rho_1, \mu_1 \rangle$ and $\langle \ell_2, \rho_2, \mu_2 \rangle$ are related on k steps, notation $\Gamma; \Sigma \vdash \langle \ell_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2, \mu_2 \rangle$, if $\rho_i \models \Gamma; \Sigma$, $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$ and

$$\forall x \in \Gamma. \Sigma \vdash \langle \ell_1, \rho_1(x), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2(x), \mu_2 \rangle : \Gamma(x)$$

3.4 Proof of noninterference

Lemma 3.10 (Substitution preserves typing). *If $\Gamma; \Sigma; \ell \vdash t : S$ and $\rho \models \Gamma; \Sigma$ then $\Gamma; \Sigma; \ell \vdash \rho(t) : S'$ and $S' <: S$.*

Proof. By induction on the derivation of $\Gamma; \Sigma; \ell \vdash t \in S$. □

Lemma 3.11. *Consider stores $\mu_1, \mu_2, \mu'_1, \mu'_2$ such that $\mu_i \twoheadrightarrow \mu'_i$, and substitutions ρ_1 and ρ_2, such that $\Gamma; \Sigma \vdash \langle \ell_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2, \mu_2 \rangle$, then if $\forall j \leq k$, if $\Sigma \subseteq \Sigma', \Sigma' \vdash \mu'_1 \approx^j_{\ell_o} \mu'_2$ then $\Gamma; \Sigma' \vdash \langle \ell_1, \rho_1, \mu'_1 \rangle \approx^j_{\ell_o} \langle \ell_2, \rho_2, \mu'_2 \rangle$*

Proof. By definition of related computations and related stores. The key argument is that given that $\mu_i \rightharpoonup \mu_i'$ then μ_i' have at least the same locations of μ_i and the values still are related as well given that they still have the same type. □

Lemma 3.12 (Substitution preserves typing). *If* $\Gamma; \Sigma; \ell \vdash t : S$ *then* $\forall \ell' \preccurlyeq \ell, \Gamma; \Sigma; \ell' \preccurlyeq \ell : S$.

Proof. By induction on the derivation of $\Gamma; \Sigma; \ell \vdash t \in S$. □

Lemma 3.13 (Downward Closed / Monotonicity). *If*

(1) $\Sigma \vdash \langle \ell_1, v_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, v_2, \mu_2 \rangle : S$ *then*
$\forall j \leq k, \Sigma \vdash \langle \ell_1, v_1, \mu_1 \rangle \approx^j_{\ell_o} \langle \ell_2, v_2, \mu_2 \rangle : S$

(2) $\Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S)$ *then*
$\forall j \leq k, \Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^j_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S)$

(3) $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$ *then* $\forall j \leq k, \Sigma \vdash \mu_1 \approx^j_{\ell_o} \mu_2$

Proof. By induction on type S and the definition of related stores. □

Lemma 3.14. *Consider simple values* $v_i : S_i$ *and*
$\Sigma \vdash \langle \ell_1, v_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, v_2, \mu_2 \rangle : S$.
Then

$$\Sigma \vdash \langle \ell_1, (v_1 \curlyvee \ell), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, (v_2 \curlyvee \ell), \mu_2 \rangle : S \curlyvee \ell$$

Proof. By induction on type S. We proceed by definition of related values and observational-monotonicity of the join, considering that the label stamping can only make values non observable. □

Lemma 3.15 (Reduction preserves relations). *Consider* $\Sigma; \ell_i \vdash t_i \in \mathbb{T}[S]$, $\mu_i \in$ *Store*, $\Sigma \vdash \mu_i$, *and* $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$. *Consider* $j < k$, *posing* $t_i \mid \mu_i \overset{\ell_i}{\longmapsto}{}^j t_i' \mid \mu_i'$, $\Sigma \subseteq \Sigma'$, $\Sigma' \vdash \mu_i'$ *we have* $\Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S)$ *if and only if* $\Sigma' \vdash \langle \ell_1, t_1', \mu_1' \rangle \approx^{k-j}_{\ell_o} \langle \ell_2, t_2', \mu_2' \rangle : \mathcal{C}(S)$

Proof. Direct by definition of
$\Sigma \vdash \langle \ell_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, t_2, \mu_2 \rangle : \mathcal{C}(S)$ and transitivity of $\overset{\ell}{\longmapsto}{}^j$. □

Lemma 3.16. *Consider term* $\Sigma; \ell \vdash t : S$, *store* μ *and* $j > 0$,
such that $t \mid \mu \overset{\ell}{\longmapsto}{}^j t' \mid \mu'$. *Then* $\mu \rightharpoonup \mu'$.

Proof. Trivial by induction on the derivation of t. The only rules that change the store are the ones for reference and assignment, neither of which remove locations. □

Lemma 3.17. *Suppose that* $\Sigma \vdash \langle \ell_1 \curlyvee \ell_1', t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2 \curlyvee \ell_2', t_2, \mu_2 \rangle : \mathcal{C}(S)$, *and that* $\ell_i \vdash \mathsf{prot}_{\ell_i'}(t) : S_i' \curlyvee \ell_i', S_i' \curlyvee \ell_i' <: S \curlyvee \ell$ *for* $i \in \{1, 2\}$. *If* $\ell_1 \approx^k_{\ell_o} \ell_2$, *and* $\ell_1' \approx^k_{\ell_o} \ell_2'$,
then $\Sigma \vdash \langle \ell_1, \mathsf{prot}_{\ell_1'}(t_1), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \mathsf{prot}_{\ell_2'}(t_2), \mu_2 \rangle : \mathcal{C}(S \curlyvee \ell)$

Proof. Consider $j < k$, we know by definition of related computations that

$$t_i \mid \mu_i \overset{\ell_i \curlyvee \ell_i'}{\longmapsto}{}^j t_i' \mid \mu_i'$$

then $\mu_1' \approx^j_{\ell_o} \mu_2'$, and by Lemma 3.16 $\mu_i \rightharpoonup \mu_i'$. If t_i' are reducible after $k-1$ steps, then the result holds immediately by (Rprot()). The interest case if t_i' are irreducible after $j < k$ steps:

Suppose that after j steps $t_i' = v_i$, then $\Sigma' \vdash \langle \ell_1 \curlyvee \ell_1', v_1, \mu_1' \rangle \approx_{\ell_o}^{k-j} \langle \ell_2 \curlyvee \ell_2', v_2, \mu_2' \rangle : S$, for some Σ' such that $\Sigma \subseteq \Sigma'$.

Therefore:

$$\begin{array}{ll} & \mathsf{prot}_{\ell_i'}(t_i) \mid \mu_i' \\ \overset{\ell_i}{\longmapsto}{}^{j} & \mathsf{prot}_{\ell_i'}(v_i) \mid \mu_i' \\ \overset{\ell_i}{\longmapsto}{}^{1} & (v_i \curlyvee \ell_i') \mid \mu_i' \end{array}$$

Let us suppose $\Sigma'; \ell_i \vdash v_i : S_i''$, where $S_i'' <: S_i' <: S$. Then $\Sigma'; \ell_i \vdash v_i \curlyvee \ell_i' : S_i'' \curlyvee \ell_i'$, and $S_i'' \curlyvee \ell_i' <: S \curlyvee \ell$. If $\neg\mathsf{obs}_{\ell_o}(\ell_i \curlyvee \ell_i')$ by monotonicity of the join either $\neg\mathsf{obs}_{\ell_o}(\ell_i')$ or $\neg\mathsf{obs}_{\ell_o}(\ell_i)$. If $\neg\mathsf{obs}_{\ell_o}(\ell_i')$ then $\neg\mathsf{obs}_{\ell_o}(S \curlyvee \ell_i')$ and the result holds. If $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ the result holds immediately. If $\mathsf{obs}_{\ell_o}(\ell_i \curlyvee \ell_i', S)$ then $\mathsf{obs}_{\ell_o}(\ell_i, S \curlyvee \ell_i')$, then the result follows by Lemma 3.14, and by backward preservation of the relations (Lemma 3.15).

□

Lemma 3.18. *Consider ℓ, such that $\neg\mathsf{obs}_{\ell_o}(\ell)$, then then $\forall k > 0$, such that, $\Sigma; \ell \vdash t : S$, $\Sigma \vdash \mu$ $t \mid \mu \overset{\ell}{\longmapsto}{}^{k} t' \mid \mu'$, then $\forall \ell'$,*

(1) $\forall o \in dom(\mu') \backslash dom(\mu)$, $\neg\mathsf{obs}_{\ell_o}(\ell', \mu'(o))$.
(2) $\forall o \in dom(\mu') \cap dom(\mu) \land \mu'(o) \neq \mu(o)$, $\neg\mathsf{obs}_{\ell_o}(\mathsf{label}(\Sigma(o)))$.

Proof. We use induction on the derivation of t. The interest cases are the last step of reduction rules for references and assignments.

Case ($t = o_{\ell''}{:=}v$). We are only updating the heap so we only have to prove (1) and (2). Then

$$o_{\ell''}{:=}v \overset{\ell}{\longmapsto} \mathsf{unit}_\perp \mid \mu[o \mapsto (v \curlyvee (\ell \curlyvee \ell''))]$$

Next we have to prove that $\mathsf{obs}_{\ell_o}(\mathsf{label}(\Sigma(o)))$ is not defined. As $\Sigma; \ell \vdash t : S$, then we know that $\ell \curlyvee \ell'' \preccurlyeq \mathsf{label}(\Sigma(o))$, and as $\neg(\mathsf{obs}_{\ell_o}(\ell))$ by monotonicity of the join the result holds.

Case ($t = \mathsf{ref}^{S'}\ v$). We are extending the heap, so we need to only prove (1). Then

$$\mathsf{ref}^{S'}\ v \mid \mu \overset{\ell}{\longmapsto} o_\perp \mid \mu[o \mapsto (v \curlyvee \ell)]$$

where $o \notin dom(\mu)$. We need to prove that $\mathsf{obs}_{\ell_o}(\mathsf{label}(v \curlyvee \ell))$ does not hold, which follows directly by monotonicity of the join.

□

Lemma 3.19. *Consider ℓ, such that $\mathsf{obs}_{\ell_o}(\ell)$ does not hold, then then $\forall k > 0$, such that $\Sigma; \ell \vdash t_i : S_i$, and that $t_i \mid \mu_i \overset{\ell}{\longmapsto}{}^{k} t_i' \mid \mu_i'$, then if $\Sigma \vdash \mu_1 \approx_{\ell_o}^{k} \mu_2$, then $\Sigma' \vdash \mu_1' \approx_{\ell_o}^{k} \mu_2'$ for some Σ' such that $\Sigma \subseteq \Sigma'$ and that $\Sigma'; \ell \vdash t_i' : S_i'$, where $S_i' <: S_i$.*

Proof. By Lemma 3.18 we know three things:

(1) $\forall o \in dom(\mu_i') \backslash dom(\mu_i)$, $\mathsf{obs}_{\ell_o}(\ell, \mu_i'(o))$ does not hold, i.e. new locations are not observable and therefore as $\Sigma'; \ell \vdash \mu_i'(o) : S$ and $S <: \Sigma'(o)$, then $\neg\mathsf{obs}_{\ell_o}(\mathsf{label}(\Sigma(o)))$.
(2) $\forall o \in dom(\mu_i') \cap dom(\mu_i) \land \mu_i'(o) \neq \mu(o)$, $\neg\mathsf{obs}_{\ell_o}(\mathsf{label}(\Sigma(o)))$
i.e. for all updated references they have to be previously not observable, and by definition therefore related, and second they are still non observable after the update, and by definition those locations are still related under ℓ because $\Sigma(o) = \Sigma'(o)$.

Therefore $\Sigma' \vdash \mu_1' \approx_{\ell_o}^{k} \mu_2'$ and the result holds. □

Lemma 3.20. *Suppose that* $\Sigma; \ell_i \vdash \mathsf{prot}_{\ell_i'}(t_i) : S' \curlyvee \ell_i', S' \curlyvee \ell_i' <: S$ *for* $i \in \{1,2\}$, *where* $\neg \mathsf{obs}_{\ell_o}(\ell_i \curlyvee \ell_i')$. *Also consider two stores* μ_i *such that* $\Sigma \vdash \mu_1 \approx^k_{\ell_o} \mu_2$.
Then $\Sigma \vdash \langle \ell_1, \mathsf{prot}_{\ell_1'}(t_1), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \mathsf{prot}_{\ell_2'}(t_2), \mu_2 \rangle : \mathcal{C}(S)$

Proof. Suppose that after at least j more steps, where $j < k$, both subterms reduce to a value :

$$t \mid \mu_i \quad \overset{\ell_i \curlyvee \ell_i'}{\longmapsto}{}^{j} \quad v_i \mid \mu_i'$$

Therefore:

$$\begin{array}{ll} & \mathsf{prot}_{\ell_i'}(t) \mid \mu_i' \\ \overset{\ell_i}{\longmapsto}{}^{j} & \mathsf{prot}_{\ell_i'}(v_i) \mid \mu_i' \\ \overset{\ell_i}{\longmapsto}{}^{1} & (v_i \curlyvee \ell_i') \mid \mu_i' \end{array}$$

As the values can be radically different we have to make sure that both values are not observables. If $\neg \mathsf{obs}_{\ell_o}(\ell_i)$ then the values are not observables because the security context is not observable. Let us assume that $\mathsf{obs}_{\ell_o}(\ell_i)$ holds, but $\mathsf{obs}_{\ell_o}(\ell_i')$ not. Then by monotonicity of the join, $\neg \mathsf{obs}_{\ell_o}(\mathsf{label}(v_i) \curlyvee \ell_i')$ and the result follows.

Now we have to prove that the resulting stores are related, for some Σ' such that $\Sigma \subseteq \Sigma'$. But by Lemma 3.19 the result follows immediately.

□

Next, we present the Noninterference proposition.

Proposition 2.5 (Security Type Soundness). *If* $\Gamma; \Sigma; \ell_c \vdash t : S_i' \implies \forall S, S_i' <: S, \Gamma; \Sigma; \ell_c \models t : S$

Proof. We proceed by proving a more general proposition instead:
If $\Gamma; \Sigma; \ell_i \vdash t : S_i', S_i' <: S$, then $\forall \mu_i \in \textsc{Store}$, $\Sigma \vdash \mu_i$, and $\forall k \geq 0, \forall \rho_i \in \textsc{Subst}, \Gamma; \Sigma \vdash \langle \ell_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2, \mu_2 \rangle$, we have $\Sigma \vdash \langle \ell_1, \rho_1(t), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2(t), \mu_2 \rangle : \mathcal{C}(S)$.
By induction on the derivation of term t. Let us take an arbitrary index $k \geq 0$.

Case (x). $t = x$ and $\Gamma(x) = S$. $\Gamma; \Sigma \vdash \langle \ell_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2, \mu_2 \rangle$ implies by definition that $\Sigma \vdash \langle \ell_1, \rho_1(x), \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2(x), \mu_2 \rangle : S$, and the result holds immediately.

———

Case (b). $t = b_g$. By definition of substitution, $\rho_1(b_g) = \rho_2(b_g) = b_g$. By definition, $\Sigma \vdash \langle \ell_1, b_g, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, b_g, \mu_2 \rangle : \mathsf{Bool}_g$ as required.

———

Case (o). $t = o_{g_1}$ and $\Sigma(o) = S$, where $S = \mathsf{Ref}_{g_1}\ S_1$. By definition of substitution, $\rho_1(o_{g_1}) = \rho_2(o_{g_1}) = o_{g_1}$. We know that $\Sigma; \ell_i \vdash o_{g_1} : \mathsf{Ref}_{g_1}\ S_1$. By definition of related stores, $\Sigma \vdash \langle \ell_1, o_{g_1}, \mu_1 \rangle \approx^k_{\ell_o} \langle \ell_2, o_{g_1}, \mu_2 \rangle : \mathsf{Ref}_{g_1}\ S_1$ as required, and the result holds.

———

Case (λ). $t = (\lambda^{\ell_c''} x : S_1'.t_1)_{\ell'}$. Then $S_i' = S_1' \xrightarrow{\ell_c''}_{\ell_i'} S_{i2}'$, and $S = S_1 \xrightarrow{\ell_c'}_{\ell} S_2$, where $S' <: S$.
By definition of substitution, assuming $x \notin dom(\rho_i)$, and Lemma 3.10:

$$\Gamma; \Sigma; \ell_i \vdash \rho_i(t) = \Gamma; \Sigma; \ell_i \vdash (\lambda^{\ell_c''} x : S_1.\rho_i(t_1))_{\ell'} : S_1' \xrightarrow{\ell_c''}_{\ell'} S_{i2}''$$

where $S_{i2}'' <: S_2'$. Consider $j \leq k$, μ_1', μ_2' such that $\mu_i \twoheadrightarrow \mu_i'$ and $\Sigma \subseteq \Sigma'$ $\Sigma' \vdash \mu_1' \approx^j_{\ell_o} \mu_2'$, and assume two values v_1 and v_2 such that $\Sigma' \vdash \langle \ell_1, v_1, \mu_1' \rangle \approx^j_{\ell_o} \langle \ell_2, v_2, \mu_2' \rangle : S_1$.

We need to show that:

$$\begin{array}{rll} & \Sigma' \vdash \langle \ell_1, (\lambda^{\ell''_c} x : S'_1.\rho_1(t_1))_{\ell'}\ v_1, \mu'_1\rangle & \\ \approx^j_{\ell_o} & \langle \ell_2, (\lambda^{\ell''_c} x : S'_1.\rho_2(t_1))_{\ell'}\ v_2, \mu'_2\rangle & : \mathcal{C}(S_2) \end{array}$$

Then:

$$\begin{array}{rl} & (\lambda^{\ell''_c} x : S'_1.\rho_i(t_1))_{\ell'}\ v_i \mid \mu'_i \\ \overset{\ell_i}{\longmapsto} & \mathsf{prot}_{\ell'}([v_i/x]\rho_i(t_1)) \mid \mu'_i \\ \overset{\ell_i}{\longmapsto}^{*} & \mathsf{prot}_{\ell'}([v_i/x]\rho_i(t_1)) \mid \mu'_i \end{array}$$

We then extend the substitutions to map x to the arguments:

$$\rho'_i = \rho_i\{x \mapsto v_i\}$$

We know that $\Sigma' \vdash \langle \ell_1, v_1, \mu'_1\rangle \approx^j_{\ell_o} \langle \ell_2, v_2, \mu'_2\rangle : S_1$. So as $\mu_i \twoheadrightarrow \mu'_i$ then by Lemma 3.11, $\Gamma, x : S_1; \Sigma' \vdash \langle \ell_1, \rho'_1, \mu'_1\rangle \approx^j_{\ell_o} \langle \ell_2, \rho'_2, \mu'_2\rangle$.

By Lemma 3.10, $\Gamma; \Sigma'; \ell''_c \vdash \rho'_i(t_1) : S''_{i2}$ where $S''_{i2} <: S'_{i2} <: S_2$. We know that $\ell_i \curlyvee \ell' \preccurlyeq \ell''_c$, therefore by Lemma 3.2, $\Gamma; \Sigma'; \ell_i \curlyvee \ell \vdash \rho'_i(t_1) : S''_{i2}$. Then by induction hypothesis and Lemma 3.13:

$$\Sigma' \vdash \langle \ell_1 \curlyvee \ell', \rho'_1(t_1), \mu'_1\rangle \approx^{j-1}_{\ell_o} \langle \ell_2 \curlyvee \ell', \rho'_2(t_1), \mu'_2\rangle : \mathcal{C}(S_2),$$

Finally, by Lemma 3.17:

$$\begin{array}{rl} & \Sigma' \vdash \langle \ell_1, \mathsf{prot}_{\ell'}(\rho'_1(t_1)), \mu'_1\rangle \\ \approx^j_{\ell_o} & \langle \ell_2, \mathsf{prot}_{\ell'}(\rho'_2(t_1)), \mu'_2\rangle : \mathcal{C}(S_2) \end{array}$$

and finally the result holds by backward preservation of the relations (Lemma 3.15).

———

Case (!). $t = !t'$, where $\Sigma; \ell_i \vdash t' : \mathsf{Ref}_{\ell''_i}\ S_1$, where $S_1 \curlyvee \ell''_i <: S = S_1 \curlyvee \ell$.
By definition of substitution:

$$\rho_i(t) = !\rho_i(t')$$

We have to show that

$$\begin{array}{rl} & \Sigma \vdash \langle \ell_1, !\rho_i(t'), \mu_1\rangle \\ \approx^k_{\ell_o} & \langle \ell_2, !\rho_i(t'), \mu_2\rangle : \mathcal{C}(S) \end{array}$$

By Lemma 3.10:

$$\Sigma; \ell_i \vdash !\rho_i(t') : S_1 \curlyvee \ell'''_i$$

where $\ell'''_i \preccurlyeq \ell''_i \preccurlyeq \ell$. By induction hypotheses on the subterm:

$$\Sigma \vdash \langle \ell_1, \rho_1(t'), \mu_1\rangle \approx^k_{\ell_o} \langle \ell_2, \rho_2(t'), \mu_2\rangle : \mathcal{C}(\mathsf{Ref}_\ell\ S_1)$$

Consider $j < k$, then by definition of related computations

$$\rho_i(t') \mid \mu_i \overset{\ell_i}{\longmapsto}{}^j t'_i \mid \mu'_i \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu'_1 \approx^{k-j}_{\ell_o} \mu'_2 \wedge (irred(t'_i) \Rightarrow \Sigma' \vdash \langle \ell_1, t'_1, \mu'_1\rangle \approx^{k-j}_{\ell_o} \langle \ell_2, t'_2, \mu'_2\rangle : \mathsf{Ref}_\ell\ S_1)$$

If terms t'_i are reducible after $j = k - 1$ steps, then
$!\rho_i(t) \mid \mu_i \overset{\ell_i}{\longmapsto}{}^j !t'_i \mid \mu'_i$ and the result holds.
If after at most j steps t'_i is irreducible it means that for some $j' \leq j$, $!\rho_i(t) \mid \mu_i \overset{\ell_i}{\longmapsto}{}^{j'} !v_i \mid \mu'_i$. If $j' = j$ then we use the same same argument for reducible terms and the result holds.

Let us consider now $j' < j$. Then $\Sigma' \vdash \langle \ell_1, v_1, \mu_1' \rangle \approx^{k-j'}_{\ell_o} \langle \ell_2, v_2, \mu_2' \rangle : \mathsf{Ref}_\ell\ S_1$. By Lemma 3.6, each v_i is a location $o_{i\,\ell_i'}$, such that $\Sigma'(o_{i\,\ell_i'}) = \mathsf{Ref}_{\ell_i'}\ S_1$ and $\ell_i' \preccurlyeq \ell'$. Then:

$$\begin{array}{rll} \rho_i(t) \mid \mu & \xmapsto{\ell_i}{}^{j'+1} & !o_{i\,\ell_i'} \mid \mu_i' \\ & \xmapsto{\ell_i}{}^{1} & \mathsf{prot}_{\ell_i'}(v_i') \mid \mu_i' \end{array}$$

with $\ell_i' \preccurlyeq \ell_i''', v_i' = \mu_i'(o_{i\,\ell_i'})$, As $\Sigma' \vdash \langle \ell_1, v_1, \mu_1' \rangle \approx^{k-j'}_{\ell_o} \langle \ell_2, v_2, \mu_2' \rangle : \mathsf{Ref}_\ell\ S_1$, then by By monotonicity of the join either both $\mathsf{obs}_{\ell_o}(\ell_i')$ or $\neg\mathsf{obs}_{\ell_o}(\ell_i')$. Finally as $\Sigma' \vdash \langle \ell_1, v_1', \mu_1' \rangle \approx^{k-j'}_{\ell_o} \langle \ell_2, v_2', \mu_2' \rangle : S_1$, by Lemma 6.59,

$$\begin{array}{rl} & \Sigma' \vdash \langle \ell_1, \mathsf{prot}_{\ell_1'}(v_1'), \mu_1' \rangle \\ \approx^{j}_{\ell_o} & \langle \ell_2, \mathsf{prot}_{\ell_2'}(v_2'), \mu_2' \rangle : \mathcal{C}(S_1 \curlyvee \ell) \end{array}$$

and finally the result holds by backward preservation of the relations (Lemma 3.15).

———

Case (:=). $t = t_1{:=}t_2$. Then $S = \mathsf{Unit}_\perp$.
By definition of substitution:

$$\rho_i(t) = \rho_i(t_1){:=}\rho_i(t_2)$$

and Lemma 3.10:

$$\Sigma; \ell_i \vdash \rho_i(t_1){:=}\rho_i(t_2) : \mathsf{Unit}_\perp$$

We have to show that

$$\begin{array}{rl} & \Sigma \vdash \langle \ell_1, \rho_1(t_1){:=}\rho_1(t_2), \mu_1 \rangle \\ \approx^{k}_{\ell_o} & \langle \ell_2, \rho_2(t_1){:=}\rho_2(t_2), \mu_2 \rangle : \mathcal{C}(S) \end{array}$$

By induction hypotheses

$$\Sigma \vdash \langle \ell_1, \rho_1(t_1), \mu_1 \rangle \approx^{k}_{\ell_o} \langle \ell_2, \rho_2(t_1), \mu_2 \rangle : \mathcal{C}(S_1)$$

Suppose $j_1 < k$, and that $\rho_i(t_1)$ are irreducible after j_1 steps (otherwise, similar to case !, the result holds immediately). Then by definition of related computations:

$$\rho_i(t_1) \mid \mu_i \xmapsto{\ell_i}{}^{j_1} v_i \mid \mu_i' \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu_1' \approx^{k-j_1}_{\ell_o} \mu_2' \wedge \Sigma' \vdash \langle \ell_1, v_1, \mu_1' \rangle \approx^{k-j_1}_{\ell_o} \langle \ell_2, v_2, \mu_2' \rangle : \mathsf{Ref}_\ell\ S_1$$

By Lemma 3.16 $\mu_i \rightarrow \mu_i'$, and $\mu_1' \approx^{k-j_1}_{\ell_o} \mu_2'$ then by Lemma 6.41, $\Sigma' \vdash \langle \ell_1, \rho_1, \mu_1' \rangle \approx^{k-j_1}_{\ell_o} \langle \ell_2, \rho_2, \mu_2' \rangle$. By induction hypotheses:

$$\Sigma' \vdash \langle \ell_1, \rho_1(t_2), \mu_1' \rangle \approx^{k}_{\ell_o} \langle \ell_2, \rho_2(t_2), \mu_2' \rangle : \mathcal{C}(S_2)$$

Again, consider $j_2 = k - j_1$, if after j_2 steps $\rho_1(t_2)$ is reducible or is a value, the result holds immediately. The interest case if after $j_2' < j_2$ steps $\rho_1(t^{S_2})$ reduces to values v_i':

$$\rho_i(t^{S_2}) \mid \mu_i' \xmapsto{\ell_i}{}^{j_2'} v_i' \mid \mu_i'' \implies \Sigma' \subseteq \Sigma'', \Sigma'' \vdash \mu_1'' \approx^{k-j_1-j_2'}_{\ell_o} \mu_2'' \wedge \Sigma'' \vdash \langle \ell_1, v_1', \mu_1'' \rangle \approx^{k-j_1-j_2'}_{\ell_o} \langle \ell_2, v_2', \mu_2'' \rangle : S_2$$

Then

$$\rho_i(t^S) \mid \mu_i \xmapsto{\ell_i}{}^{j_1+j_2'} v_i{:=}v_i' \mid \mu_i'' \wedge \Sigma'' \vdash \mu_1'' \approx^{k-j_1-j_2'}_{\ell_o} \mu_2''$$

As both values v_i are related at some reference type, then by canonical forms (Lemma 3.6) they both must be locations $o_{i\,\ell_i'}$ for some $S_1' <: S_1$. We consider when the values are observable and the locations are identical (otherwise the result is trivial):

$$\begin{array}{rl} & v_i{:=}v_i' \mid \mu_i'' \\ = & o_{\ell_i'}{:=}v_i' \mid \mu_i'' \\ \xmapsto{\ell_i}{}^{1} & \mathsf{unit}_\perp \mid \mu_i''' \end{array}$$

Where $\mu_i''' = \mu_i''[o \mapsto (v_i' \curlyvee (\ell_i \curlyvee \ell_i'))]$. As $\Sigma'' \vdash \langle \ell_1, v_1', \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2'} \langle \ell_2, v_2', \mu_2'' \rangle : S_2$, and as $\ell_i \curlyvee \ell_i' \preccurlyeq \mathsf{label}(S_1)$, where $\ell_i' \preccurlyeq \ell$, and $\mathsf{label}(v_i') \preccurlyeq \mathsf{label}(S_1)$, then $\Sigma''; \ell_i \vdash v_i' \curlyvee (\ell_i \curlyvee \ell_i') : S'$ and $S' <: S_1$. Then by monotonicity of the join Lemma 3.14,

$$\begin{array}{ll} & \Sigma'' \vdash \langle \ell_1, (v_1' \curlyvee (\ell_1 \curlyvee \ell_1')), \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2'} & \langle \ell_2, (v_2' \curlyvee (\ell_2 \curlyvee \ell_2')), \mu_1'' \rangle \end{array}$$

But if $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ then by monotonicity of the join $\neg\mathsf{obs}_{\ell_o}(v_i' \curlyvee (\ell_i \curlyvee \ell_i'))$. Therefore, $\forall \ell_i''$ such that $\ell_1'' \approx_{\ell_o}^{k} \ell_2''$

$$\begin{array}{ll} & \Sigma'' \vdash \langle \ell_1'', (v_1' \curlyvee (\ell_1 \curlyvee \ell_1')), \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2'} & \langle \ell_2'', (v_2' \curlyvee (\ell_2 \curlyvee \ell_2')), \mu_1'' \rangle \end{array}$$

As every values are related at type Unit, we only have to prove that $\Sigma'' \vdash \mu_1''' \approx_{\ell_o}^{k-j_1-j_2'-3} \mu_1'''$, but using monotonicity (Lemma 6.46), it is trivial to prove that because either both both stores update the same location o to values that are related, therefore the result holds.

———

Case (ref). $t = \mathsf{ref}^{S_1}\ t^{S_1'}$. Then $S = \mathsf{Ref}_\perp\ S_1$.
By definition of substitution:

$$\rho_i(t) = \mathsf{ref}^{S_1}\ \rho_i(t')$$

and Lemma 3.10:

$$\ell_i \vdash \mathsf{ref}^{S_1}\ \rho_i(t') : \mathsf{Ref}_\perp\ S_1$$

We have to show that

$$\begin{array}{ll} & \Sigma \vdash \langle \ell_1, \mathsf{ref}^{S_1}\ \rho_1(t'), \mu_1 \rangle \\ \approx_{\ell_o}^{k} & \langle \ell_2, \mathsf{ref}^{S_1}\ \rho_2(t'), \mu_2 \rangle : \mathcal{C}(S_1) \end{array}$$

As $\Sigma; \ell_i \vdash \rho_i(t') : S_i'$ where $S_i' <: S_1$, by induction hypotheses:

$$\Sigma \vdash \langle \ell_1, \rho_1(t'), \mu \rangle \approx_{\ell_o}^{k} \langle \ell_2, \rho_2(t'), \mu \rangle : \mathcal{C}(S_1)$$

Consider $j < k$, by definition of related computations

$$\rho_i(t') \mid \mu_i \xrightarrow{\ell_i}{}^{j} t_i' \mid \mu_i' \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu_1' \approx_{\ell_o}^{k-j} \mu_2' \wedge (irred(t_i') \Rightarrow \Sigma' \vdash \langle \ell_1, t_1', \mu_1' \rangle \approx_{\ell_o}^{k-j} \langle \ell_2, t_2', \mu_2' \rangle : S_1')$$

If terms t_i' are reducible after $j = k-1$ steps, then
$\mathsf{ref}^{S_1}\ \rho_i(t') \mid \mu_i \xrightarrow{\ell_i}{}^{j} \mathsf{ref}^{S_1}\ t_i' \mid \mu_i'$ and the result holds.
If after at most j steps t_i' is irreducible, it means that for some $j' \leq j$ $\mathsf{ref}^{S_1}\ \rho_i(t') \mid \mu_i \xrightarrow{\ell_i}{}^{j'} \mathsf{ref}^{S_1}\ v_i \mid \mu_i'$.
If $j' = j$ then we use the same same argument for reducible terms and the result holds.
Let us consider now $j' < j$. Then:

$$\begin{array}{lll} \rho_i(t) \mid \mu & \xrightarrow{\ell_i}{}^{j'+1} & \mathsf{ref}^{S_1}\ v_i \mid \mu_i' \\ & \xrightarrow{\ell_i}{}^{1} & o_\perp \mid \mu_i'' \end{array}$$

with, $\mu_i'' = \mu_i'[o \mapsto (v_i \curlyvee \ell_i)]$. Also, as $\Sigma' \vdash \langle \ell_1, v_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'} \langle \ell_2, v_2, \mu_2' \rangle : S_1$, then $\Sigma'' \vdash \langle \ell_1, v_1, \mu_1'' \rangle \approx_{\ell_o}^{k-j'} \langle \ell_2, v_2, \mu_2'' \rangle : S_1$, with $\Sigma'' = \Sigma', o : S_1$. And as $\mathsf{label}(v_i) \curlyvee \ell_i \preccurlyeq \mathsf{label}(S_1)$,then by Lemma 3.14, $\Sigma'' \vdash \langle \ell_1, v_1 \curlyvee \ell_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'} \langle \ell_2, v_2 \curlyvee \ell_2, \mu_2' \rangle : S_1$.

If $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ then by monotonicity of the join $\neg\mathsf{obs}_{\ell_o}(\mathsf{label}(v_i' \curlyvee \ell_i))$ and $\neg\mathsf{obs}_{\ell_o}(\mathsf{label}(\Sigma''(o)))$. Therefore, $\forall \ell_i''$ such that $\ell_1'' \approx_{\ell_o}^{k} \ell_2''$ $\Sigma'' \vdash \langle \ell_1'', v_1 \curlyvee \ell_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'} \langle \ell_2'', v_2 \curlyvee \ell_2, \mu_2' \rangle : S_1$. By definition of related stores $\Sigma'' - \mu_1'' \approx_{\ell_o}^{k-j'} \mu_2''$. Then by Monotonicity of the relation (Lemma 6.46) $\Sigma'' - \mu_1'' \approx_{\ell_o}^{k-j'-2} \mu_2''$ and the result holds.

———

Case (⊕). $t = t_1 \oplus; t_2$

By definition of substitution:

$$\rho_i(t) = \rho_i(t_1) \oplus; \rho_i(t_2)$$

and Lemma 3.10:

$$\Sigma; \ell_i \vdash \rho_i(t_1) \oplus; \rho_i(t_2) : S''$$

with $S_i'' <: S_i' <: S$. We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k - 3$ where:

$$\rho_i(t_1) \mid \mu_i \overset{\ell_i}{\longmapsto}{}^{j_1} v_{i1} \mid \mu_i' \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \Sigma' \vdash \langle \ell_1, v_{11}, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \ell_2, v_{21}, \mu_2' \rangle : S_1$$

$$\rho_i(t_2) \mid \mu_i' \overset{\ell_i}{\longmapsto}{}^{j_2} v_{i2} \mid \mu_i'' \implies \Sigma' \subseteq \Sigma'', \Sigma'' \vdash \mu_1'' \approx_{\ell_o}^{k-j_1-j_2} \mu_2'' \wedge \Sigma'' \vdash \langle \ell_1, v_{12}, \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2} \langle \ell_2, v_{22}, \mu_2'' \rangle : S_2$$

By Lemma 3.6, each v_{ij} is a boolean $(b_{ij})_{\ell_{ij}}$ then:

$$\begin{array}{rl} & \rho_i(t) \mid \mu_i'' \\ \longmapsto^{j_1+j_2+2} & (b_{i1})_{\ell_{i1}} \oplus (b_{i2})_{\ell_{i2}} \mid \mu_i'' \\ \longmapsto^{1} & (b_i)_{\ell_i'} \mid \mu_i'' \end{array}$$

with $b_i = b_{i1} [\![\oplus]\!] b_{i2}$, $\ell_i' = \ell_{i1} \curlyvee \ell_{i2}$, and $\ell_i' \preccurlyeq \mathsf{label}(S_i'') \preccurlyeq \mathsf{label}(S)$. It remains to show that:

$$\Sigma'' \vdash \langle \ell_1, (b_1)_{\ell_1'}, \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2-3} \langle \ell_2, (b_2)_{\ell_2'}, \mu_2'' \rangle : S$$

If $\neg\mathsf{obs}_{\ell_o}(\ell_i)$, then the result is trivial because the resulting booleans are also related as they are not observable.

If $\mathsf{obs}_{\ell_o}(\ell_i)$, and $\neg\mathsf{obs}_{\ell_o}(\ell_{i1}')$ or $\neg\mathsf{obs}_{\ell_o}(\ell_{i2}')$, then by monotonicity of the join, $\neg\mathsf{obs}_{\ell_o}(\ell_i')$ and the result holds. If $\mathsf{obs}_{\ell_o}(\ell_{ij})$ then $\mathsf{obs}_{\ell_o}(\ell_i')$ and therefore $b_{11} = b_{21}$ and $b_{12} = b_{22}$, so $b_1 = b_2$, and the result holds.

———

Case (app). $t = t_1\ t_2$, with $\Sigma; \ell_i \vdash t_1 : S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i'} S_{i2}$, and $\Sigma; \ell_i \vdash t_2 : S_{i1}''$. Also $S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i'} S_{i2} <: S_1 \xrightarrow{\ell_c}_{\ell} S_2$, and $S = S_2$.

By definition of substitution:

$$\rho_i(t) = \rho_i(t_1)\ \rho_i(t_2)$$

and Lemma 3.10:

$$\Sigma; \ell_i \vdash \rho_i(t_1)\ \rho_i(t_2) : S_{i2}'$$

with $S_{i2}' <: S_{i2} <: S_2$. We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k$ where by induction hypotheses and the definition of related computations:

$$\rho_i(t_1) \mid \mu_i \overset{\ell_i}{\longmapsto}{}^{j_1} v_{i1} \mid \mu_i' \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \Sigma' \vdash \langle \ell_1, v_{11}, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \ell_2, v_{21}, \mu_2' \rangle : S_1$$

$$\rho_i(t_2) \mid \mu_i' \overset{\ell_i}{\longmapsto}{}^{j_2} v_{i2} \mid \mu_i'' \implies \Sigma' \subseteq \Sigma'', \Sigma' \vdash \mu_1'' \approx_{\ell_o}^{k-j_1-j_2} \mu_2'' \wedge \Sigma'' \vdash \langle \ell_1, v_{12}, \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2} \langle \ell_2, v_{22}, \mu_2'' \rangle : S_2$$

Then

$$\rho_i(t) \mid \mu_i \overset{\ell_i}{\longmapsto}{}^{j_1+j_2} v_{i1}\ v_{i2} \mid \mu_i''$$

If $\mathsf{obs}_{\ell_o}(\ell_i, v_{i1})$ then, by definition of $\approx_{\ell_o}$ at values of function type, we have:

$$\begin{array}{ll} & \Sigma' \vdash \langle \ell_1, (v_{11}\ v_{12}), \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2} & \langle \ell_2, (v_{21}\ v_{22}), \mu_2'' \rangle : \mathcal{C}(S_2 \curlyvee \ell) \end{array}$$

Finally, by backward preservation of the relations (Lemma 3.15) the result holds.

If $\neg\mathsf{obs}_{\ell_o}(\ell_i, v_{i1})$, and we assume by canonical forms that $v_{i1} = (\lambda^{\ell'_{ci}} x.t_i)_{\ell_i''}$ then, either $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ or $\neg\mathsf{obs}_{\ell_o}(\ell_i'')$ and

$$\begin{array}{ll} & (v_{i1}\ v_{i2}) \mid \mu_1'' \\ = & ((\lambda^{\ell'_{ci}} x.t_i)_{\ell_i''}\ v_{i2}) \mid \mu_1'' \\ \xmapsto{\ell_i}{}^{1} & \mathsf{prot}_{\ell_i''}(t_i') \mid \mu_1'' \end{array}$$

If either $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ or $\neg\mathsf{obs}_{\ell_o}(\ell_i'')$ then by Lemma 3.20 ,

$$\begin{array}{ll} & \Sigma'' \vdash \langle \ell_1, \mathsf{prot}_{\ell_1''}(t_1'), \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2} & \langle \ell_2, \mathsf{prot}_{\ell_2''}(t_2'), \mu_2'' \rangle : \mathcal{C}(S_2 \curlyvee \ell) \end{array}$$

Finally, by backward preservation of the relations (Lemma 3.15) the result holds.

———

Case (if). $t = \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$, with $\Sigma; \ell_i \vdash t_1 : S_1$, $\Sigma; \ell_i' \vdash t_2 : S_2$, $\Sigma; \ell_i' \vdash t_3 : S_3$, $\ell_i' = \ell_i \curlyvee \mathsf{label}(S_1)$, and $S' = S_2 \mathbin{\dot\curlyvee} S_3 <: S$
By definition of substitution:

$$\rho_i(t) = \mathsf{if}\ \rho_i(t_1)\ \mathsf{then}\ \rho_i(t_2)\ \mathsf{else}\ \rho_i(t_3)$$

We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k$ where by induction hypotheses and related computations we have that:

$$\rho_i(t_1) \mid \mu_i \xmapsto{\ell_i}{}^{j_1} v_{i1} \mid \mu_i' \implies \Sigma \subseteq \Sigma', \Sigma' \vdash \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \Sigma' \vdash \langle \ell_1, v_{11}, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \ell_2, v_{21}, \mu_2' \rangle : S_1$$

By Lemma 3.6, each v_{i1} is a boolean $(b_{i1})_{\ell_{i1}}$, such that $\Sigma'; \ell_i \vdash (b_{i1})_{\ell_{i1}} : \mathsf{Bool}_{\ell_{i1}}$ and $\mathsf{Bool}_{\ell_{i1}} <: S_1$, implies $S_1 = \mathsf{Bool}_{\ell_1'}$. Then:

$$\rho_i(t) \mid \mu_i \xmapsto{\ell_i}{}^{j_1+1} \mathsf{if}\ (b_{i1})_{\ell_{i1}}\ \mathsf{then}\ \rho_i(t_2)\ \mathsf{else}\ \rho_i(t_3) \mid \mu_i'$$

Let us consider $\neg\mathsf{obs}_{\ell_o}(\ell_i, (b_{i1})_{\ell_{i1}})$. Let us assume the worst case scenario and that both execution reduce via different branches of the conditional.
Then

$$\rho_1(t) \mid \mu_1 \xmapsto{\ell_i}{}^{j_1+2} \mathsf{prot}_{\ell_{11}}(\rho_1(t_2)) \mid \mu_1'$$

$$\rho_2(t) \mid \mu_2 \xmapsto{\ell_i}{}^{j_1+2} \mathsf{prot}_{\ell_{21}}(\rho_2(t_3)) \mid \mu_2'$$

But because $\neg\mathsf{obs}_{\ell_o}(\ell_i, (b_{i1})_{\ell_{i1}})$, then either $\neg\mathsf{obs}_{\ell_o}(\ell_i)$ or $\neg\mathsf{obs}_{\ell_o}(\ell_{i1})$ and therefore, $\neg\mathsf{obs}_{\ell_o}(\ell_i \curlyvee \ell_{i1})$. Then by Lemma 3.20,

$$\Sigma' \vdash \langle \ell_1, \mathsf{prot}_{\ell_{11}}(\rho_1(t_2)), \mu_1' \rangle \approx_{\ell_o}^{k} \langle \ell_2, \mathsf{prot}_{\ell_{21}}(\rho_2(t_3)), \mu_2' \rangle$$

and the result holds by backward preservation of the relations (Lemma 3.15).

Now let us consider if $\mathsf{obs}_{\ell_o}(\ell_i, (b_{i1})_{\ell_{i1}})$ holds. Then by definition of $\approx_{\ell_o}$ on boolean values, $b_{11} = b_{21}$. Because $b_{11} = b_{21}$, both $\rho_1(t)$ and $\rho_2(t)$ step into the same branch of the conditional. Let us assume the condition is true (the other case is similar):

Then by induction hypothesis $\Sigma' \vdash \langle \ell_1 \curlyvee \ell_{11}, \rho_1(t_2), \mu_1' \rangle \approx^k_{\ell_o} \langle \ell_2 \curlyvee \ell_{21}, \rho_2(t_2), \mu_2' \rangle : S_2$, and by Lemma 3.17,

$$\Sigma' \vdash \langle \ell_1, \mathsf{prot}_{\ell_{11}}(\rho_1(t_2)), \mu_1' \rangle \approx^k_{\ell_o} \langle \ell_2, \mathsf{prot}_{\ell_{21}}(\rho_2(t_2)), \mu_2' \rangle : S$$

and the result holds by backward preservation of the relations (Lemma 3.15).

Case (prot()). Direct by using Lemma 3.17.

□

4 GRADUALIZING THE STATIC SEMANTICS

In section 4.1, we show the proof of optimality and soundness of the abstraction. In section 4.2, we present the proof for the Static Gradual Guarantee.

4.1 From Gradual Labels to Gradual Types

Proposition 4.1 (α is Sound). *If $\widehat{\ell} \neq \emptyset$ then $\widehat{\ell} \subseteq \gamma(\alpha(\widehat{\ell}))$.*

Proof. By case analysis on the structure of $\widehat{\ell}$. If $\widehat{\ell} = \{\ell\}$ then $\gamma(\alpha(\{\ell\})) = \gamma(\ell) = \{\ell\} = \widehat{\ell}$, otherwise
$\gamma(\alpha(\widehat{\ell})) = \gamma(?) = \text{Label} \supseteq \widehat{\ell}$. □

Proposition 4.2 (α is Optimal). *If $\widehat{\ell} \subseteq \gamma(g)$ then $\alpha(\widehat{\ell}) \sqsubseteq g$.*

Proof. By case analysis on the structure of g. If $g = \ell$, $\gamma(g) = \{\ell\}$; $\widehat{\ell} \subseteq \{\ell\}, \widehat{\ell} \neq \emptyset$ implies $\alpha(\widehat{\ell}) = \alpha(\{\ell\}) = \ell \sqsubseteq g$ (if $\widehat{\ell} = \emptyset$, $\alpha(\widehat{\ell})$ is undefined). If $g = ?$, $g' \sqsubseteq g$ for all g'. □

Proposition 4.3 (α is Sound and Optimal). *If $\widehat{\ell} \neq \emptyset$ then,*
(i) $\widehat{\ell} \subseteq \gamma(\alpha(\widehat{\ell}))$.

(ii) If $\widehat{\ell} \subseteq \gamma(g)$ then $\alpha(\widehat{\ell}) \sqsubseteq g$.

Proof. Trivial using Prop 4.1 and 4.2. □

Proposition 4.4 (α_S is Sound). *If $\widehat{S}$* **valid**, *then $\widehat{S} \subseteq \gamma_S(\alpha_S(\widehat{S}))$.*

Proof. By well-founded induction on $\widehat{S}$ according to the ordering relation $\widehat{S} \sqsubset \widehat{S}$ defined as follows:

$$\widehat{dom}(\widehat{S}) \sqsubset \widehat{S}$$
$$\widehat{cod}(j\widehat{S}) \sqsubset \widehat{S}$$

Where $\widehat{dom}, \widehat{cod} : \mathcal{P}(\text{GType}) \to \mathcal{P}(\text{GType})$ are the collecting liftings of the domain and codomain functions dom, cod respectively, *e.g.*,

$$\widehat{dom}(\widehat{S}) = \{\, dom(S) \mid S \in \widehat{S} \,\}.$$

We then consider cases on $\widehat{S}$ according to the definition of α_S.

Case ($\{\,\overline{\mathsf{Bool}_{\ell_i}}\,\}$).

$$\begin{aligned}
\gamma_S(\alpha_S(\{\,\overline{\mathsf{Bool}_{\ell_i}}\,\})) &= \gamma_S(\mathsf{Bool}_{\alpha(\{\,\overline{\ell_i}\,\})}) \\
&= \{\, \mathsf{Bool}_\ell \mid \ell \in \gamma(\alpha(\{\,\overline{\ell_i}\,\})) \,\} \\
&\supseteq \{\,\overline{\mathsf{Bool}_{\ell_i}}\,\} \text{ by soundness of } \alpha.
\end{aligned}$$

Case $(\{\overline{S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i} S_{i2}}\})$.

$$\begin{aligned}
& \gamma_S(\alpha_S(\{\overline{S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i} S_{i2}}\})) \\
= \; & \gamma_S(\alpha_S(\{\overline{S_{i1}}\}) \xrightarrow{\alpha(\{\overline{\ell_{ci}}\})}_{\alpha(\{\overline{\ell_i}\})} \alpha_S(\{\overline{S_{i2}}\})) \\
= \; & \gamma_S(\alpha_S(\{\overline{S_{i1}}\})) \xrightarrow{\gamma(\alpha(\{\overline{\ell_{ci}}\}))}_{\gamma(\alpha(\{\overline{\ell_i}\}))} \gamma_S(\alpha_S(\{\overline{S_{i2}}\})) \\
\supseteq \; & \{\overline{S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i} S_{i2}}\}
\end{aligned}$$

by induction hypothesis on $\{\overline{S_{i1}}\}$ and $\{\overline{S_{i2}}\}$, and soundness of α.

Case $(\{\overline{\mathsf{Ref}_{\ell_i}\ S_i}\})$.

$$\begin{aligned}
& \gamma_S(\alpha_S(\{\overline{\mathsf{Ref}_{\ell_i}\ S_i}\})) \\
= \; & \gamma_S(\mathsf{Ref}_{\alpha(\{\overline{\ell_i}\})}\ \alpha_S(\{\overline{S_i}\})) \\
= \; & \{\mathsf{Ref}_\ell\ S \mid \ell \in \gamma(\alpha(\{\overline{\ell_i}\})), S \in \gamma_S(\alpha_S(\{\overline{S_i}\}))\} \\
\supseteq \; & \{\overline{\mathsf{Ref}_{\ell_i}\ S_i}\}
\end{aligned}$$

by induction hypothesis on $\{\overline{S_i}\}$ and soundness of α.

□

Proposition 4.5 (α_S is Optimal). *If $\widehat{S}$* **valid** *and $\widehat{S} \subseteq \gamma_S(U)$ then $\alpha_S(\widehat{S}) \sqsubseteq U$.*

Proof. By induction on the structure of U.

Case (Bool_g). $\gamma_S(\mathsf{Bool}_g) = \{\mathsf{Bool}_\ell \mid \ell \in \gamma(g)\}$
So $\widehat{S} = \{\mathsf{Bool}_\ell \mid \ell \in \widehat{\ell}\}$ for some $\widehat{\ell} \subseteq \gamma(g)$. By optimality of α, $\alpha(\widehat{\ell}) \sqsubseteq g$, so $\alpha_S(\{\mathsf{Bool}_\ell \mid \ell \in \widehat{\ell}\}) = \mathsf{Bool}_{\alpha(\widehat{\ell})} \sqsubseteq \mathsf{Bool}_g$.

Case $(U_1 \to_g U_2)$. $\gamma_S(U_1 \xrightarrow{g_c}_g U_2) = \gamma_S(U_1) \xrightarrow{\gamma(g_c)}_{\gamma(g)} \gamma_S(U_2)$.
So $\widehat{S} = \{\overline{S_{1i} \xrightarrow{\ell_{ci}}_{g_i} S_{2i}}\}$, with $\{\overline{S_{1i}}\} \subseteq \gamma_S(U_1)$,
$\{\overline{S_{1i}}\} \subseteq \gamma_S(U_2)$, $\{\overline{\ell_{ci}}\} \subseteq \gamma(g_c)$ and $\{\overline{\ell_{ci}}\} \subseteq \gamma(g)$. By induction hypothesis, $\alpha_S(\{\overline{S_{1i}}\}) \sqsubseteq U_1$ and $\alpha_S(\{\overline{S_{2i}}\}) \sqsubseteq U_2$, and by optimality of α, $\alpha(\{\overline{\ell_{ci}}\}) \sqsubseteq g_c$ and $\alpha(\{\overline{\ell_i}\}) \sqsubseteq g$. Hence $\alpha_S(\{\overline{S_{1i} \xrightarrow{\ell_{ci}}_{\ell_i} S_{2i}}\}) =$

$\alpha_S(\{\overline{S_{1i}}\}) \xrightarrow{\alpha(\{\overline{\ell_{ci}}\})}_{\alpha(\{\overline{g_i}\})} \alpha_S(\{\overline{S_{2i}}\}) \sqsubseteq U_1 \xrightarrow{g_c}_g U_2$.

Case $(\mathsf{Ref}_g\ U)$. $\gamma_S(\mathsf{Ref}_g\ U) = \{\mathsf{Ref}_\ell\ S \mid \ell \in \gamma(g), S \in \gamma(U)\}$
So $\widehat{S} = \{\mathsf{Ref}_\ell\ S \mid \ell \in \widehat{\ell}, S \in \{\overline{S_i}\}\}$ for some $\{\overline{S_i}\} \subseteq \gamma_S(U)$ and some $\widehat{\ell} \subseteq \gamma(g)$. By induction hypothesis $\alpha_S(\{\overline{S_i}\}) \sqsubseteq U$ and by optimality of α, $\alpha(\widehat{\ell}) \sqsubseteq g$, so $\alpha_S(\{\mathsf{Ref}_\ell\ S \mid \ell \in \widehat{\ell}, S \in \{\overline{S_i}\}\}) = \mathsf{Ref}_{\alpha(\widehat{\ell})}\ \alpha_S(\{\overline{S_i}\}) \sqsubseteq \mathsf{Ref}_g\ U$.

□

Proposition 2.9 (α_S is Sound and Optimal). *Assuming $\widehat{S}$* **valid***:*
(i) $\widehat{S} \subseteq \gamma_S(\alpha_S(\widehat{S}))$ *(ii) If $\widehat{S} \subseteq \gamma_S(U)$ then $\alpha_S(\widehat{S}) \sqsubseteq U$.*

Proof. Trivial using Prop 4.4 and 4.5. □

4.2 Static Criteria for Gradual Typing

In this section we present the proof of Static Gradual Guarantee for $\mathrm{GSL_{Ref}}$.

Proposition 4.6 (Static conservative extension). *Let $\vdash_S$ denote SSL_{Ref}'s type system. Then for any static language term $t \in$ Term, $\cdot; \Sigma; \ell_c \vdash_S t : S$ if and only if $\cdot; \Sigma; \ell_c \vdash t : S$.*

Proof. By induction over the typing derivations. The proof is trivial because static types are given singleton meanings via concretization. □

Definition 4.7 (Term precision).

$$\text{(Px)}\frac{}{x \sqsubseteq x} \qquad \text{(Pb)}\frac{g \sqsubseteq g'}{b_g \sqsubseteq b_{g'}} \qquad \text{(Pu)}\frac{g \sqsubseteq g'}{\mathsf{unit}_g \sqsubseteq \mathsf{unit}_{g'}} \qquad \text{(P}\lambda\text{)}\frac{t \sqsubseteq t' \quad g'_c \sqsubseteq g''_c \quad U_1 \sqsubseteq U'_1 \quad g \sqsubseteq g'}{(\lambda^{g'_c} x : U_1.t)_g \sqsubseteq (\lambda^{g''_c} x : U'_1.t')_{g'}}$$

$$\text{(Pprot)}\frac{t \sqsubseteq t' \quad g \sqsubseteq g'}{\mathsf{prot}_g(t) \sqsubseteq \mathsf{prot}_{g'}(t')} \qquad \text{(P}\oplus\text{)}\frac{t_1 \sqsubseteq t'_1 \quad t_2 \sqsubseteq t'_2}{t_1 \oplus t_2 \sqsubseteq t'_1 \oplus t'_2} \qquad \text{(Papp)}\frac{t_1 \sqsubseteq t'_1 \quad t_2 \sqsubseteq t'_2}{t_1\ t_2 \sqsubseteq t'_1\ t'_2}$$

$$\text{(Pif)}\frac{t \sqsubseteq t \quad t_1 \sqsubseteq t'_1 \quad t_2 \sqsubseteq t'_2}{\mathsf{if}\ t\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \sqsubseteq \mathsf{if}\ t'\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2} \qquad \text{(P::)}\frac{t \sqsubseteq t' \quad U \sqsubseteq U'}{t :: U \sqsubseteq t' :: U'}$$

$$\text{(Pref)}\frac{t \sqsubseteq t' \quad U \sqsubseteq U'}{\mathsf{ref}^U\ t \sqsubseteq \mathsf{ref}^{U'}\ t'} \qquad \text{(Pderef)}\frac{t \sqsubseteq t'}{!t \sqsubseteq !t'} \qquad \text{(Pasgn)}\frac{t_1 \sqsubseteq t'_1 \quad t_2 \sqsubseteq t'_2}{t_1{:=}t_2 \sqsubseteq t'_1{:=}t'_2}$$

Definition 4.8 (Type environment precision).

$$\frac{}{\cdot \sqsubseteq \cdot} \qquad \frac{\Gamma \sqsubseteq \Gamma' \quad U \sqsubseteq U'}{\Gamma, x : U \sqsubseteq \Gamma', x : U'}$$

Lemma 4.9. *If $\Gamma; \cdot; g_c \vdash t : U$ and $\Gamma \sqsubseteq \Gamma'$, then $\Gamma'; \cdot; g_c \vdash t : U'$ for some $U \sqsubseteq U'$.*

Proof. Simple induction on typing derivations. □

Lemma 4.10. *If $U_1 \lesssim U_2$ and $U_1 \sqsubseteq U'_1$ and $U_2 \sqsubseteq U'_2$ then $U'_1 \lesssim U'_2$.*

Proof. By definition of $\lesssim$, there exists $\langle S_1, S_2 \rangle \in \gamma^2(U_1, U_2)$ such that $S_1 <: S_2$. $U_1 \sqsubseteq U'_1$ and $U_2 \sqsubseteq U'_2$ mean that $\gamma(U_1) \subseteq \gamma(U'_1)$ and $\gamma(U_2) \subseteq \gamma(U'_2)$, therefore $\langle S_1, S_2 \rangle \in \gamma^2(U'_1, U'_2)$. □

Lemma 4.11. *If $\widetilde{g_1 \curlyvee g_2 \preccurlyeq g_3}$, $g_1 \sqsubseteq g'_1$, $g_2 \sqsubseteq g'_2$ and $g_3 \sqsubseteq g'_3$, then $\widetilde{g'_1 \curlyvee g'_2 \preccurlyeq g'_3}$.*

Proof. By definition of the consistent judgment, there exists $\langle \ell_1, \ell_2, \ell_3 \rangle \in \gamma^3(g_1, g_2, g_3)$ such that $\ell_1 \curlyvee \ell_2 \preccurlyeq \ell_3$. $g_1 \sqsubseteq g'_1$, $g_2 \sqsubseteq g'_2$ and $g_3 \sqsubseteq g'_3$ mean that $\gamma(g_1) \subseteq \gamma(g'_1)$, $\gamma(g_2) \subseteq \gamma(g'_2)$ and $\gamma(g_3) \subseteq \gamma(g'_3)$ respectively. Therefore $\langle \ell_1, \ell_2, \ell_3 \rangle \in \gamma^3(g_1, g_2, g_3)$. □

Lemma 4.12. *If $g_1 \,\widetilde{\preccurlyeq}\, g_2$, $g_1 \sqsubseteq g'_1$ and $g_2 \sqsubseteq g'_2$, then $g'_1 \,\widetilde{\preccurlyeq}\, g'_2$.*

Proof. Using almost identical argument of Lemma 4.11 □

Proposition 4.13 (Static gradual guarantee). *Suppose $g_{c1} \sqsubseteq g_{c2}$ and $t_1 \sqsubseteq t_2$. If $\cdot; \cdot; g_{c1} \vdash t_1 : U_1$ then $\cdot; \cdot; g_{c2} \vdash t_2 : U_2$ where $U_1 \sqsubseteq U_2$.*

Proof. We prove the property on opens terms instead of closed terms: If $\Gamma; \cdot; g_{c1} \vdash t_1 : U_1$, $g_{c1} \sqsubseteq g_{c2}$ and $t_1 \sqsubseteq t_2$ then $\Gamma; \cdot; g_{c2} \vdash t_2 : U_2$ and $U_1 \sqsubseteq U_2$.

The proof proceed by induction on the typing derivation.

Case (Ux, Ub, Uu). Trivial by definition of $\sqsubseteq$ using (Px), (Pb), (Pu) respectively.

Case $(U\lambda)$. Then $t_1 = (\lambda^{g'_c} x : U'_1.t)_g$ and $U_1 = U'_1 \xrightarrow{g'_c}_g U'_2$. By $(U\lambda)$ we know that:

$$(U\lambda)\frac{\Gamma, x : U'_1; \cdot; g'_c \vdash t : U'_2}{\Gamma; \cdot; g_{c1} \vdash (\lambda^{g'_c} x : U'_1.t)_g : U'_1 \xrightarrow{g'_c}_g U'_2} \tag{1}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = (\lambda^{g_c} x : U''_1.t')'_g$ and therefore

$$(U\lambda)\frac{\begin{array}{c} t \sqsubseteq t' \\ g'_c \sqsubseteq g''_c \qquad U'_1 \sqsubseteq U''_1 \qquad g \sqsubseteq g' \end{array}}{(\lambda^{g'_c} x : U'_1.t)_g \sqsubseteq (\lambda^{g''_c} x : U''_1.t')_{g'}} \tag{2}$$

Using induction hypotheses on the premise of 1, $\Gamma, x : U'_1; \cdot; g_{c2} \vdash t' : U''_2$ with $U'_2 \sqsubseteq U''_2$. By Lemma 4.9, $\Gamma, x : U''_1; \cdot; g_{c2} \vdash t' : U'''_2$ where $U''_2 \sqsubseteq U'''_2$. Then we can use rule $(U\lambda)$ to derive:

$$(U\lambda)\frac{\Gamma, x : U''_1; \cdot; g''_c \vdash t' : U'''_2}{\Gamma; .; g_{c1} \vdash (\lambda^{g''_c} x : U'_1.t')_{g'} : U''_1 \xrightarrow{g''_c}_{g'} U'''_2}$$

Where $U_2 \sqsubseteq U''_2$. Using the premise of 2 and the definition of type precision we can infer that

$$U'_1 \xrightarrow{g'_c}_g U'_2 \sqsubseteq U''_1 \xrightarrow{g''_c}_{g'} U'''_2$$

and the result holds.

Case (Uo). This case can not happen because initial programs do not contain locations.

Case $(U\mathsf{prot})$. Then $t_1 = \mathsf{prot}_g(t)$ and $U_1 = U \mathbin{\widetilde{\curlyvee}} g$. By $(U\mathsf{prot})$ we know that:

$$(U\mathsf{prot})\frac{\Gamma; \cdot; g_{c1} \mathbin{\widetilde{\curlyvee}} g \vdash t : U}{\Gamma; \cdot; g_{c1} \vdash \mathsf{prot}_g(t) : U \mathbin{\widetilde{\curlyvee}} g} \tag{3}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = \mathsf{prot}_{g'}(t')$ and therefore

$$(\mathrm{Pprot})\frac{t \sqsubseteq t' \qquad g \sqsubseteq g'}{\mathsf{prot}_g(t) \sqsubseteq \mathsf{prot}_{g'}(t')} \tag{4}$$

By definition of join on consistent labels, $g_{c1} \mathbin{\widetilde{\curlyvee}} g \sqsubseteq g_{c2} \mathbin{\widetilde{\curlyvee}} g'$. Using induction hypotheses on the premises of 3, we can use rule $(U\mathsf{prot})$ to derive:

$$(U\mathsf{prot})\frac{\Gamma; \cdot; g_{c2} \mathbin{\widetilde{\curlyvee}} g' \vdash t' : U'}{\Gamma; \cdot; g_{c2} \vdash \mathsf{prot}_{g'}(t') : U' \mathbin{\widetilde{\curlyvee}} g'}$$

For some U', where $U \sqsubseteq U'$. Using the premise of 4 and the definition of join we can infer that

$$U \mathbin{\widetilde{\curlyvee}} g \sqsubseteq U' \mathbin{\widetilde{\curlyvee}} g'$$

and the result holds.

Case $(U\oplus)$. Then $t_1 = t'_1 \oplus t'_2$ and $U_1 = \mathsf{Bool}_{(g_1 \widetilde{\curlyvee} g_2)}$. By $(U\oplus)$ we know that:

$$(U\oplus)\frac{\Gamma; \cdot; g_{c1} \vdash t'_1 : \mathsf{Bool}_{g_1} \qquad \Gamma; \cdot; g_{c1} \vdash t'_2 : \mathsf{Bool}_{g_2}}{\Gamma; \cdot; g_{c1} \vdash t'_1 \oplus t'_2 : \mathsf{Bool}_{(g_1 \widetilde{\curlyvee} g_2)}} \tag{5}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = t''_1 \oplus t''_2$ and therefore

$$\text{(P}\oplus\text{)}\frac{t'_1 \sqsubseteq t''_1 \qquad t'_2 \sqsubseteq t''_2}{t'_1 \oplus t'_2 \sqsubseteq t''_1 \oplus t''_2} \tag{6}$$

Using induction hypotheses on the premises of 5, we can use rule $(U\oplus)$ to derive:

$$(U\oplus)\frac{\Gamma; \cdot; g_{c2} \vdash t''_1 : \mathsf{Bool}_{g'_1} \qquad \Gamma; \cdot; g_{c2} \vdash t''_2 : \mathsf{Bool}_{g'_2}}{\Gamma; \cdot; g_{c2} \vdash t''_1 \oplus t''_2 : \mathsf{Bool}_{(g''_1 \widetilde{\curlyvee} g''_2)}}$$

Where $g'_1 \sqsubseteq g''_1$ and $g'_2 \sqsubseteq g''_2$. Using the premise of 6 and the definition of type precision we can infer that

$$\frac{(g'_1 \widetilde{\curlyvee} g'_2) \sqsubseteq (g''_1 \widetilde{\curlyvee} g''_2)}{\mathsf{Bool}_{(g'_1 \widetilde{\curlyvee} g'_2)} \sqsubseteq \mathsf{Bool}_{(g''_1 \widetilde{\curlyvee} g''_2)}}$$

and the result holds.

Case $(U\text{app})$. Then $t_1 = t'_1\ t'_2$ and $U_1 = U_{12} \widetilde{\curlyvee} g$. By $(U\text{app})$ we know that:

$$(U\text{app})\frac{\begin{array}{cc}\Gamma; \cdot; g_{c1} \vdash t'_1 : U_{11} \xrightarrow{g'_c}_{g} U_{12} & \Gamma; \cdot; g_{c1} \vdash t'_2 : U'_2 \\ U'_2 \lesssim U_{11} & g \mathrel{\widetilde{\curlyvee}} g_{c1} \preccurlyeq g'_c\end{array}}{\Gamma; \cdot; g_{c1} \vdash t'_1\ t'_2 : U_{12} \widetilde{\curlyvee} g} \tag{7}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = t''_1\ t''_2$ and therefore

$$\text{(Papp)}\frac{t'_1 \sqsubseteq t''_1 \qquad t'_2 \sqsubseteq t''_2}{t'_1\ t'_2 \sqsubseteq t''_1\ t''_2} \tag{8}$$

Using induction hypotheses on the premises of 7, $\Gamma; \cdot; g_{c2} \vdash t''_1 : U'_{11} \xrightarrow{g''_c}_{g'} U'_{12}$ and $\Gamma; \cdot; g_{c2} \vdash t''_2 : U''_2$, where $U'_2 \sqsubseteq U''_2$, $U_{11} \xrightarrow{g'_c}_{g} U_{12} \sqsubseteq U'_{11} \xrightarrow{g''_c}_{g'} U'_{12}$. By Lemma 4.10, $U''_2 \lesssim U'_{11}$. By definition of precision of types, $g'_c \sqsubseteq g''_c$ and $g \sqsubseteq g'$, therefore by Lemma 4.11, $g' \mathrel{\widetilde{\curlyvee}} g_{c2} \preccurlyeq g''_c$. Then we can use rule $(U\text{app})$ to derive:

$$(U\text{app})\frac{\begin{array}{cc}\Gamma; \cdot; g_{c2} \vdash t''_1 : U'_{11} \xrightarrow{g''_c}_{g'} U'_{12} & \Gamma; \cdot; g_{c2} \vdash t''_2 : U''_2 \\ U''_2 \lesssim U'_{11} & g' \mathrel{\widetilde{\curlyvee}} g_{c2} \preccurlyeq g''_c\end{array}}{\Gamma; \cdot; g_{c2} \vdash t''_1\ t''_2 : U'_{12} \widetilde{\curlyvee} g'}$$

Using the definition of type precision we can infer that

$$U_{12} \widetilde{\curlyvee} g \sqsubseteq U'_{12} \widetilde{\curlyvee} g'$$

and the result holds.

Case $(U\text{if})$. Then $t_1 = \mathsf{if}\ t\ \mathsf{then}\ t'_1\ \mathsf{else}\ t_2$ and $U_1 = (U'_1 \widetilde{\dot{\curlyvee}} U'_2) \widetilde{\curlyvee} g$. By $(U\text{if})$ we know that:

$$(U\text{if})\frac{\begin{array}{c}\Gamma; \cdot; g_{c1} \vdash t : \mathsf{Bool}_g \\ \Gamma; \cdot; g_{c1} \widetilde{\curlyvee} g \vdash t'_1 : U'_1 \qquad \Gamma; \cdot; g_{c1} \widetilde{\curlyvee} g \vdash t_2 : U'_2\end{array}}{\Gamma; \cdot; g_{c1} \vdash \mathsf{if}\ t\ \mathsf{then}\ t'_1\ \mathsf{else}\ t_2 : (U'_1 \widetilde{\dot{\curlyvee}} U'_2) \widetilde{\curlyvee} g} \tag{9}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = \mathsf{if}\ t'\ \mathsf{then}\ t''_1\ \mathsf{else}\ t''_2$ and therefore

$$\text{(Pif)}\frac{t \sqsubseteq t' \qquad t'_1 \sqsubseteq t''_1 \qquad t'_2 \sqsubseteq t''_2}{\mathsf{if}\ t\ \mathsf{then}\ t'_1\ \mathsf{else}\ t'_2 \sqsubseteq \mathsf{if}\ t'\ \mathsf{then}\ t'_1\ \mathsf{else}\ t'_2} \tag{10}$$

Consider any ℓ' such that $\ell \sqsubseteq \ell'$. As $g_{c1} \mathbin{\widetilde{\curlyvee}} g \sqsubseteq g_{c2} \mathbin{\widetilde{\curlyvee}} g'$ then we can use induction hypotheses on the premises of 9 and derive:

$$(U\text{if})\frac{\begin{array}{c}\Gamma; \cdot; g_{c2} \vdash t' : \mathsf{Bool}_{g'} \\ \Gamma; \cdot; g_{c2} \mathbin{\widetilde{\curlyvee}} g' \vdash t_1'' : U_1'' \qquad \Gamma; \cdot; g_{c2} \mathbin{\widetilde{\curlyvee}} g' \vdash t_2'' : U_2''\end{array}}{\Gamma; \cdot; g_{c2} \vdash \mathsf{if}\ t'\ \mathsf{then}\ t_1''\ \mathsf{else}\ t_2'' : (U_1'' \mathbin{\widetilde{\curlyvee}} U_2'') \mathbin{\widetilde{\curlyvee}} g'}$$

Where $U_1' \sqsubseteq U_1''$ and $U_2' \sqsubseteq U_2''$. Using the definition of type precision we can infer that

$$(U_1' \mathbin{\widetilde{\curlyvee}} U_2') \mathbin{\widetilde{\curlyvee}} g \sqsubseteq (U_1'' \mathbin{\widetilde{\curlyvee}} U_2'') \mathbin{\widetilde{\curlyvee}} g'$$

and the result holds.

Case $(U::)$. Then $t_1 = t :: U_1$. By $(U::)$ we know that:

$$(U::)\frac{\Gamma; \cdot; g_{c1} \vdash t : U_1' \qquad U_1' \lesssim U_1}{\Gamma; \cdot; g_{c1} \vdash t :: U_1' : U_1} \tag{11}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = t' :: U_2$ and therefore

$$(\mathrm{P}::)\frac{t \sqsubseteq t' \qquad U_1 \sqsubseteq U_2}{t :: U_1 \sqsubseteq t' :: U_2} \tag{12}$$

Using induction hypotheses on the premises of 11, $\Gamma; \cdot; g_c \vdash t' : U_2'$ where $U_1' \sqsubseteq U_2'$. We can use rule $(U::)$ and Lemma 4.10 to derive:

$$(U::)\frac{\Gamma; \cdot; g_{c2} \vdash t' : U_2' \qquad U_2' \lesssim U_2}{\Gamma; \cdot; g_{c2} \vdash t' :: U_2 : U_2}$$

Where $U_1 \sqsubseteq U_2$ and the result holds.

Case $(U\mathrm{ref})$. Then $t_1 = \mathsf{ref}^U\ t$ and $U_1 = \mathsf{Ref}_{g_c}\ U$. By $(U\mathrm{ref})$ we know that:

$$(U\mathrm{ref})\frac{\Gamma; \cdot; g_{c1} \vdash t : U_1' \qquad U_1' \lesssim U \qquad g_{c1} \mathrel{\widetilde{\preccurlyeq}} \mathsf{label}(U)}{\Gamma; \cdot; g_{c1} \vdash \mathsf{ref}^U\ t : \mathsf{Ref}_\perp\ U} \tag{13}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = \mathsf{ref}^{U'}\ t'$ and therefore

$$(\mathrm{Pref})\frac{t \sqsubseteq t' \qquad U \sqsubseteq U'}{\mathsf{ref}^U\ t \sqsubseteq \mathsf{ref}^{U'}\ t'} \tag{14}$$

Using induction hypotheses on the premises of 13, we can use rule $(U\mathrm{ref})$ and Lemma 4.10 and 4.12 to derive:

$$(U\mathrm{ref})\frac{\Gamma; \cdot; g_{c2} \vdash t' : U_1'' \qquad U_1'' \lesssim U' \qquad g_{c2} \mathrel{\widetilde{\preccurlyeq}} \mathsf{label}(U')}{\Gamma; \cdot; g_{c2} \vdash \mathsf{ref}^{U'}\ t' : \mathsf{Ref}_\perp\ U'}$$

Where $U \sqsubseteq U'$ and $U_1' \sqsubseteq U_1''$. Using the the definition of type precision we can infer that

$$\frac{U \sqsubseteq U'}{\mathsf{Ref}_\perp\ U \sqsubseteq \mathsf{Ref}_\perp\ U'}$$

and the result holds.

Case $(U\mathrm{deref})$. Then $t_1 = !t$ and $U_1 = U \mathbin{\widetilde{\curlyvee}} g$. By $(U\mathrm{deref})$ we know that:

$$(U\mathrm{deref})\frac{\Gamma; \cdot; g_{c1} \vdash t : \mathsf{Ref}_g\ U}{\Gamma; \cdot; g_{c1} \vdash !t : U \mathbin{\widetilde{\curlyvee}} g} \tag{15}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = !t'$ and therefore

$$\text{(Pderef)}\frac{t \sqsubseteq t'}{!t \sqsubseteq !t'} \tag{16}$$

Using induction hypotheses on the premises of 15, we can use rule (Uderef) to derive:

$$(U\text{deref})\frac{\Gamma; \cdot; g_{c2} \vdash t' : \mathsf{Ref}_{g'}\ U'}{\Gamma; \cdot; g_{c2} \vdash !t' : U' \mathbin{\widetilde{\curlyvee}} g'}$$

Where $g \sqsubseteq g'$ and $U \sqsubseteq U'$. Using the premise of 16 and the definition of type precision we can infer that

$$U \mathbin{\widetilde{\curlyvee}} g \sqsubseteq U' \mathbin{\widetilde{\curlyvee}} g'$$

and the result holds.

Case (Uasgn). Then $t_1 = t_1' {:=} t_2'$ and $U_1 = \mathsf{Unit}_\perp$. By ($U$asgn) we know that:

$$(U\text{asgn})\frac{\begin{array}{cc}\Gamma; \cdot; g_{c1} \vdash t_1' : \mathsf{Ref}_g\ U_1' & \Gamma; \cdot; g_{c1} \vdash t_2' : U_2' \\ U_2' \lesssim U_1' & g \widetilde{\vee\, g_{c1} \preccurlyeq} \mathit{label}(U_1')\end{array}}{\Gamma; \cdot; g_{c1} \vdash t_1' {:=} t_2' : \mathsf{Unit}_\perp} \tag{17}$$

Consider g_{c2} such that $g_{c1} \sqsubseteq g_{c2}$ and t_2 such that $t_1 \sqsubseteq t_2$. By definition of term precision t_2 must have the form $t_2 = t_1'' {:=} t_2''$ and therefore

$$\text{(Pasgn)}\frac{t_1' \sqsubseteq t_1'' \qquad t_2' \sqsubseteq t_2''}{t_1' {:=} t_2' \sqsubseteq t_1'' {:=} t_2''} \tag{18}$$

Using induction hypotheses on the premises of 17, $\Gamma; \cdot; g_{c2} \vdash t_1' : \mathsf{Ref}_{g'}\ U_1''$ and $\Gamma; \cdot; g_{c2} \vdash t_2' : U_2'$, where $\mathsf{Ref}_g\ U_1' \sqsubseteq \mathsf{Ref}_{g'}\ U_1''$ and $U_2' \sqsubseteq U_2''$. By definition of precision on types and Lemma 4.10, $U_2'' \lesssim U_1''$. Also, as, $g \sqsubseteq g'$ and $U_1' \sqsubseteq U_1''$, by Lemma 4.11, $g' \widetilde{\vee\, g_{c2} \preccurlyeq} \mathit{label}(U_1')$. Then we can use rule ($U$asgn) to derive:

$$(U\text{asgn})\frac{\begin{array}{cc}\Gamma; \cdot; g_{c2} \vdash t_1' : \mathsf{Ref}_{g'}\ U_1'' & \Gamma; \cdot; g_{c2} \vdash t_2' : U_2'' \\ U_2'' \lesssim U_1'' & g' \widetilde{\vee\, g_{c2} \preccurlyeq} \mathit{label}(U_1'')\end{array}}{\Gamma; \cdot; g_{c2} \vdash t_1'' {:=} t_2'' : \mathsf{Unit}_\perp}$$

Using the definition of type precision we can infer that

$$\mathsf{Unit}_\perp \sqsubseteq \mathsf{Unit}_\perp$$

and the result holds.

□

5 GRADUALIZING THE DYNAMIC SEMANTICS

In this section we present the formalization of the evidences for $\mathrm{GSL}_{\mathsf{Ref}}$. Section 5.1 presents the structure of evidence and the abstraction and concretization functions. In section 5.2, we show how to calculate the initial evidence. In particular we give definition for the initial evidence of consistent judgments for labels and types. In section 5.2, we present how to evolve evidence. We define the consistent transitivity operator, the meet operator and join of evidences. In section 5.4, we present the algorithmic definitions of initial evidence and consistent transitivity. Finally, in section 5.5, we present some of the proofs of the propositions for evidence presented.

5.1 Precise Evidence for Consistent Security Judgments

Definition 5.1 (Interval). An interval is a bounded unknown label $[\ell_1, \ell_2]$ where ℓ_1 is the upper bound and ℓ_2 is the lower bound.

$$\iota \in \textsc{Label}^2$$
$$\iota \quad ::= \quad [\ell, \ell] \quad \text{(interval)}$$

Definition 5.2 (Interval Concretization). Let $\gamma_\iota : \textsc{Label}^2 \to \mathcal{P}(\textsc{Label})$ be defined as follows:

$$\gamma_\iota([\ell_1, \ell_2]) = \{\ell \mid \ell \in \textsc{Label}, \ell_1 \preccurlyeq \ell \preccurlyeq \ell_2\}$$

We can only concretize *valid* intervals:

Definition 5.3 (Valid Gradual Label).

$$\frac{\ell_1 \preccurlyeq \ell_2}{valid([\ell_1, \ell_2])}$$

Definition 5.4 (Label Evidence Concretization). Let $\gamma_{\varepsilon_\ell} : \textsc{Label}^4 \to \mathcal{P}(\textsc{Label}^2)$ be defined as follows:

$$\gamma_{\varepsilon_\ell}(\langle \iota_1, \iota_2 \rangle) = \{\langle \ell_1, \ell_2 \rangle \mid \ell_1 \in \gamma_\iota(\iota_1), \ell_2 \in \gamma_\iota(\iota_2)\}$$

Definition 5.5 (Interval Abstraction). Let $\alpha : \mathcal{P}(\textsc{Label}) \to \textsc{Label}^2$ be defined as follows:

$$\alpha_\iota(\emptyset) \text{ is undefined}$$
$$\alpha_\iota(\{\,\overline{\ell_i}\,\}) = [\curlywedge\overline{\ell_i}, \curlyvee\overline{\ell_i}] \text{ otherwise}$$

Definition 5.6 (Label Evidence Abstraction). Let $\alpha_{\varepsilon_\ell} : \mathcal{P}(\textsc{Label}^2) \to \textsc{Label}^4$ be defined as follows:

$$\alpha_{\varepsilon_\ell}(\emptyset) \text{ is undefined}$$
$$\alpha_{\varepsilon_\ell}(\{\,\overline{\langle \ell_{1i}, \ell_{2i} \rangle}\,\}) = \langle \alpha_\iota(\{\,\overline{\ell_{1i}}\,\}), \alpha_\iota(\{\,\overline{\ell_{2i}}\,\}) \rangle \text{ otherwise}$$

Definition 5.7 (Type Evidence). An evidence type is a gradual type labeled with an interval:

$$E \in \text{GE}\textsc{Type}, \quad \iota \in \textsc{Label}^2$$
$$E \quad ::= \quad \mathsf{Bool}_\iota \mid E \xrightarrow{\iota}_\iota E \mid \mathsf{Ref}_\iota\ E \mid \mathsf{Unit}_\iota \quad \text{(evidence types)}$$

Definition 5.8 (Type Evidence Concretization). Let $\gamma_E : \text{GE}\textsc{Type} \to \mathcal{P}(\textsc{Type})$ be defined as follows:

$$\gamma_E(\mathsf{Bool}_\iota) = \{\, \mathsf{Bool}_\ell \mid \ell \in \gamma_\iota(\iota) \,\}$$
$$\gamma_E(E_1 \xrightarrow{\iota_2}_{\iota_1} E_2) = \gamma_E(E_1) \xrightarrow{\gamma_\iota(\iota_2)}_{\gamma_\iota(\iota_1)} \gamma_E(E_2)$$
$$\gamma_E(\mathsf{Ref}_\iota\ E) = \{\, \mathsf{Ref}_\ell\ S \mid \ell \in \gamma_\iota(\iota), S \in \gamma_E(E) \,\}$$

where $\to$ is the set of all possible combinations of function types, using each member of the sets obtained by the γ_E and γ_ι functions.

Definition 5.9 (Evidence Concretization). Let
$\gamma_{\varepsilon_\ell} : \text{GET}\textsc{ype}^2 \to \mathscr{P}(\textsc{Type}^2)$ be defined as follows:

$$\gamma_{\varepsilon_\ell}(\langle E_1, E_2\rangle) = \{\langle S_1, S_2\rangle \mid S_1 \in \gamma_E(E_1), S_2 \in \gamma_E(E_2)\}$$

Definition 5.10 (Type Evidence Abstraction). Let the abstraction function $\alpha_E : \mathscr{P}(\textsc{Type}) \to \text{GET}\textsc{ype}$ be defined as:

$$\alpha_E(\{\overline{\mathsf{Bool}_{\ell_i}}\}) = \mathsf{Bool}_{\alpha_\iota(\{\overline{\ell_i}\})}$$
$$\alpha_E(\{\overline{S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i} S_{i2}}\}) = \alpha_E(\{\overline{S_{i1}}\}) \xrightarrow{\alpha_\iota(\{\overline{\ell_{ci}}\})}_{\alpha_\iota(\{\overline{\ell_i}\})} \alpha_E(\{\overline{S_{i2}}\})$$
$$\alpha_E(\{\overline{\mathsf{Ref}_{\ell_i}\ S_i}\}) = \mathsf{Ref}_{\alpha_\iota(\{\overline{\ell_i}\})}\ \alpha_E(\{\overline{S_i}\})$$
$$\alpha_E(\widehat{S}) \text{ is undefined otherwise}$$

Definition 5.11 (Evidence Abstraction). Let $\alpha_\varepsilon : \mathscr{P}(\textsc{Type}^2) \to \text{GET}\textsc{ype}^2$ be defined as follows:

$$\alpha_\varepsilon(\emptyset) \text{ is undefined}$$
$$\alpha_\varepsilon(\{\overline{\langle S_{1i}, S_{2i}\rangle}\}) = \langle \alpha_E(\{\overline{S_{1i}}\}), \alpha_E(\{\overline{S_{2i}}\})\rangle \text{ otherwise}$$

We can only abstract *valid* sets of security types, *i.e.* in which elements only defer by security labels.

Definition 5.12 (Valid Type Sets).

$$\frac{}{valid(\{\overline{\mathsf{Bool}_{\ell_i}}\})} \qquad \frac{valid(\{\overline{S_{i1}}\}) \qquad valid(\{\overline{S_{i2}}\})}{valid(\{\overline{S_{i1} \xrightarrow{\ell_{ci}}_{\ell_i} S_{i2}}\})} \qquad \frac{valid(\{\overline{S_i}\})}{valid(\{\overline{\mathsf{Ref}_{\ell_i}\ S_i}\})}$$

$$\frac{}{valid(\{\overline{\mathsf{Unit}_{\ell_i}}\})}$$

Proposition 5.13 (α_ι is Sound). *If $\widehat{\ell}$ is not empty, then $\widehat{\ell} \subseteq \gamma_\iota(\alpha_\iota(\widehat{\ell}))$.*

Proposition 5.14 (α_ι is Optimal). *If $\widehat{\ell}$ is not empty, and $\widehat{\ell} \subseteq \gamma_\iota(\iota)$ then $\alpha_\iota(\widehat{\ell}) \sqsubseteq \iota$.*

Proposition 5.15 (α_E is Sound). *If $valid(\widehat{S})$ then $\widehat{S} \subseteq \gamma_E(\alpha_E(\widehat{S}))$.*

Proposition 5.16 (α_E is Optimal). *If $valid(\widehat{S})$ and $\widehat{S} \subseteq \gamma_E(E)$ then $\alpha_E(\widehat{S}) \sqsubseteq E$.*

With concretization of security type, we can now define security type precision.

Definition 5.17 (Interval and Type Evidence Precision).

(1) ι_1 is less imprecise than ι_2, notation $\iota_1 \sqsubseteq \iota_2$, if and only if $\gamma_{\varepsilon_\ell}(\iota_1) \subseteq \gamma_{\varepsilon_\ell}(\iota_2)$; inductively:

$$\frac{\ell_3 \preccurlyeq \ell_1 \qquad \ell_2 \preccurlyeq \ell_4}{[\ell_1, \ell_2] \sqsubseteq [\ell_3, \ell_4]}$$

(2) E_1 is less imprecise than E_2, notation $E_1 \sqsubseteq E_2$, if and only if $\gamma_E(E_1) \subseteq \gamma_E(E_2)$; inductively:

$$\frac{\iota_1 \sqsubseteq \iota_2}{\mathsf{Bool}_{\iota_1} \sqsubseteq \mathsf{Bool}_{\iota_2}} \qquad \frac{\begin{array}{cc} E_{11} \sqsubseteq E_{21} & E_{12} \sqsubseteq E_{22} \\ \iota_1 \sqsubseteq \iota_2 & \iota_1' \sqsubseteq \iota_2' \end{array}}{E_{11} \xrightarrow{\iota_1'}_{\iota_1} E_{12} \sqsubseteq E_{21} \xrightarrow{\iota_2'}_{\iota_2} E_{22}} \qquad \frac{\iota_1 \sqsubseteq \iota_2 \qquad E_1 \sqsubseteq E_2}{\mathsf{Ref}_{\iota_1}\ E_1 \sqsubseteq \mathsf{Ref}_{\iota_2}\ E_2}$$

5.2 Initial evidence

With the definition of concretization and abstraction we can now define the initial evidence of label ordering and subtyping:

Definition 5.18 (Initial Evidence of label ordering). Let $F_1 : \textsc{Label}^n \longrightarrow \textsc{Label}$ and $F_2 : \textsc{Label}^m \longrightarrow \textsc{Label}$ be functions over labels. The initial evidence of the judgment $F_1(\overline{g_i}) \mathrel{\widetilde{\preccurlyeq}} F_2(\overline{g_j})$, notation $\mathcal{I}[\![F_1(\overline{g_i}) \mathrel{\widetilde{\preccurlyeq}} F_2(\overline{g_j})]\!]$, is defined as follows:

$$\mathcal{I}[\![F_1(g_1, ...g_n) \mathrel{\widetilde{\preccurlyeq}} F_2(g_{n+1}, ...g_{n+m})]\!] = \alpha_{\varepsilon_\ell}(\{\langle F_1(\overline{\ell_i}), F_2(\overline{\ell_j})\rangle \mid \langle\overline{\ell_i}\rangle \in \gamma^n(\overline{g_i}_{[1/n]}), \langle\overline{\ell_j}\rangle \in \gamma^m(\overline{g_i}_{[n+1/m]}) \mid F_1(\overline{\ell_i}) \preccurlyeq F_2(\overline{\ell_j})\})$$

Suppose $F_1 = F_{11}$

Definition 5.19 (Initial Evidence of subtyping). Let $F_1 : \textsc{Type}^n \longrightarrow \textsc{Type}$ and $F_2 : \textsc{Type}^m \longrightarrow \textsc{Type}$ be functions over types. The initial evidence of the judgment $F_1(\overline{U_i}) \mathrel{\widetilde{\preccurlyeq}} F_2(\overline{U_j})$, notation $\mathcal{I}[\![F_1(\overline{U_i}) \mathrel{\widetilde{<:}} F_2(\overline{U_j})]\!]$, is defined as follows:

$$\mathcal{I}[\![F_1(U_1, ...U_n) \mathrel{\widetilde{<:}} F_2(U_{n+1}, ...U_{n+m})]\!] = \alpha_{\varepsilon_\ell}(\{\langle F_1(\overline{S_i}), F_2(\overline{S_j})\rangle \mid \langle\overline{S_i}\rangle \in \gamma_S^n(\overline{U_i}_{[1/n]}), \langle\overline{S_j}\rangle \in \gamma_S^m(\overline{U_i}_{[n+1/m]}) \mid F_1(\overline{S_i}) <: F_2(\overline{S_j})\})$$

Proposition 5.20. *[Elaboration preserves typing] Consider* $\Gamma; \Sigma; g_c \vdash t : U$ *then if* $\Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : U$*, and* $\varepsilon = \mathcal{I}^{\cup}_{\preccurlyeq}(\ell_c)$*, then* $\Gamma; \Sigma; \varepsilon g_c \vdash t' : U$

Proof. Straightforward induction on type U. □

5.3 Evolving evidence: Consistent Transitivity

Now that we know how to extract initial evidence from consistent judgments, we need a way to combine evidences to use during program evaluation, i.e. we need to find a way to *evolve* evidence. We define *consistent transitivity* for label ordering and subtyping, $\circ^{\preccurlyeq}$ and $\circ^{<:}$ respectively, to combine evidences as follows:

Definition 5.21 (Consistent transitivity for label ordering). Let function $\circ^{\preccurlyeq} : \textsc{Interval}^2 \times \textsc{Interval}^2 \rightarrow \textsc{Label}^2$ be defined as:

$$\langle\imath_{11}, \imath_{12}\rangle \circ^{\preccurlyeq} \langle\imath_{21}, \imath_{22}\rangle = \alpha_{\varepsilon_\ell}(\{\langle\ell_{11}, \ell_{22}\rangle \in \gamma_{\varepsilon_\ell}(\langle\imath_{11}, \imath_{22}\rangle) \mid \exists\ell \in \gamma_\imath(\imath_{12}) \cap \gamma_\imath(\imath_{21}).\ell_{11} \preccurlyeq \ell \wedge \ell \preccurlyeq \ell_{22}\})$$

Proposition 5.22. *Suppose* $\varepsilon_1 \vdash F_1(\overline{g_i}) \mathrel{\widetilde{\preccurlyeq}} F_2(\overline{g_j})$ *and* $\varepsilon_2 \vdash F_2(\overline{g_j}) \mathrel{\widetilde{\preccurlyeq}} F_3(\overline{g_k})$*. If* $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ *is defined, then* $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 \vdash F_1(\overline{g_i}) \mathrel{\widetilde{\preccurlyeq}} F_3(\overline{g_k})$

Proposition 5.23. $\gamma_\imath(\imath_1 \sqcap \imath_2) = \gamma_\imath(\imath_1) \cap \gamma_\imath(\imath_2)$.

where $\imath \sqcap \imath' = \alpha(\gamma(\imath) \cap \gamma(\imath'))$.

Proposition 5.24. $\langle\imath_1, \imath_{21}\rangle \circ^{\preccurlyeq} \langle\imath_{22}, \imath_3\rangle = \triangle^{\preccurlyeq}(\imath_1, \imath_{21} \sqcap \imath_{22}, \imath_3)$

where

$$\triangle^{\preccurlyeq}(\imath_1, \imath_2, \imath_3) = \alpha_\varepsilon(\{\langle\ell_1, \ell_3\rangle \in \gamma_\varepsilon(\langle\imath_1, \imath_3\rangle) \mid \exists\ell_2 \in \gamma_\imath(\imath_2).\ell_1 \preccurlyeq \ell_2 \wedge \ell_2 \preccurlyeq \ell_3\})$$

Definition 5.25 (Consistent transitivity for subtyping). Suppose

$$\langle E_{11}, E_{12}\rangle \vdash F_1(\overline{U_i}) \widetilde{<:} F_2(\overline{U_j}) \qquad \langle E_{21}, E_{22}\rangle \vdash F_2(\overline{U_j}) \widetilde{<:} F_3(\overline{U_k})$$

We deduce evidence for consistent transitivity for subtyping:
$\langle E_{11}, E_{12}\rangle \circ^{<:} \langle E_{21}, E_{22}\rangle \vdash F_1(\overline{U_i}) \widetilde{<:} F_3(\overline{U_k})$
where $\circ^{<:}$: $\text{ETYPE}^2 \times \text{ETYPE}^2 \to \text{ETYPE}^2$ is defined as:

$$\langle E_{11}, E_{12}\rangle \circ^{<:} \langle E_{21}, E_{22}\rangle = \alpha_\varepsilon(\{\langle S_{11}, S_{22}\rangle \in \gamma_\varepsilon(\langle E_{11}, E_{22}\rangle) \mid \exists S \in \gamma_E(E_{12}) \cap \gamma_E(E_{21}).S_{11} <: S \wedge S <: S_{22}\})$$

Proposition 5.26. $\gamma_E(E_1 \sqcap E_2) = \gamma_E(E_1) \cap \gamma_E(E_2)$.

Then following AGT,

Proposition 5.27.

$$\langle E_1, E_{21}\rangle \circ^{<:} \langle E_{22}, E_3\rangle = \triangle^{<:}(E_1, E_{21} \sqcap E_{22}, E_3)$$

where

$$\triangle^{<:}(E_1, E_2, E_3) = \alpha_\varepsilon(\{\langle S_1, S_3\rangle \in \gamma_\varepsilon(\langle E_1, E_3\rangle) \mid \exists S_2 \in \gamma_\imath(E_2).S_1 <: S_2 \wedge S_2 <: S_3\})$$

Definition 5.28 (Intervals join).

$$[\ell_1, \ell_2] \mathbin{\widetilde{\curlyvee}} [\ell_3, \ell_4] = [\ell_1 \curlyvee \ell_3, \ell_2 \curlyvee \ell_4]$$

Definition 5.29 (Evidence label join).

$$\langle \imath_1, \imath_2\rangle \mathbin{\widetilde{\curlyvee}} \langle \imath_3, \imath_4\rangle = \langle \imath_1 \mathbin{\widetilde{\curlyvee}} \imath_3, \imath_2 \mathbin{\widetilde{\curlyvee}} \imath_4\rangle$$

Definition 5.30.

$$\begin{aligned}
\mathsf{Bool}_{\imath_1} \mathbin{\widetilde{\curlyvee}} \imath_2 &= \mathsf{Bool}_{(\imath_1 \widetilde{\curlyvee} \imath_2)} \\
E_1 \xrightarrow{\imath_2}_{\imath_1} E_2 \mathbin{\widetilde{\curlyvee}} \imath_3 &= E_1 \xrightarrow{\imath_2}_{(\imath_1 \widetilde{\curlyvee} \imath_3)} E_2 \\
\mathsf{Ref}_{\imath_1}\ E \mathbin{\widetilde{\curlyvee}} \imath_2 &= \mathsf{Ref}_{(\imath_1 \widetilde{\curlyvee} \imath_2)}\ E
\end{aligned}$$

Definition 5.31.

$$\langle E_1, E_2\rangle \mathbin{\widetilde{\curlyvee}} \langle \imath_1, \imath_2\rangle = \langle E_1 \mathbin{\widetilde{\curlyvee}} \imath_1, E_2 \mathbin{\widetilde{\curlyvee}} \imath_2\rangle$$

Proposition 5.32. *If* $\varepsilon_S \vdash U_1 \lesssim U_2$ *and* $\varepsilon_l \vdash g_1 \mathrel{\widetilde{\preccurlyeq}} g_2$ *then* $\varepsilon_S \mathbin{\widetilde{\curlyvee}} \varepsilon_l \vdash U_1 \mathbin{\widetilde{\curlyvee}} g_1 <: U_2 \mathbin{\widetilde{\curlyvee}} g_2$

5.4 Algorithmic definitions

This section gives algorithmic definitions of consistent transitivity and initial evidence for label ordering and subtyping.

5.4.1 Label Evidences.

Definition 5.33 (Intervals join).

$$[\ell_1, \ell_2] \mathbin{\widetilde{\curlyvee}} [\ell_3, \ell_4] = [\ell_1 \curlyvee \ell_3, \ell_2 \curlyvee \ell_4]$$

Definition 5.34 (Intervals meet).

$$[\ell_1, \ell_2] \mathbin{\widetilde{\curlywedge}} [\ell_3, \ell_4] = [\ell_1 \curlywedge \ell_3, \ell_2 \curlywedge \ell_4]$$

Definition 5.35. Let $F_1 : \text{GLabel}^n \to \text{GLabel}$ and $F_2 : \text{GLabel}^m \to \text{GLabel}$. The initial evidence for consistent judgment $F_1(\overline{g_i}) \mathrel{\widetilde{\preccurlyeq}} F_2(\overline{g_j})$ is defined as follows:

$$\begin{aligned}
bounds(?) &= [\bot, \top]\\
bounds(\ell) &= [\ell, \ell]\\
bounds(x_1 \curlyvee x_2) &= bounds(x_1) \curlyvee bounds(x_2)\\
bounds(x_1 \curlywedge x_2) &= bounds(x_1) \curlywedge bounds(x_2)\\
bounds(x_1 \sqcap x_2) &= bounds(x_1) \sqcap bounds(x_2)\\
bounds(F_1(\overline{x_i}) \curlyvee F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \curlyvee bounds(F_2(\overline{x_i}))\\
bounds(F_1(\overline{x_i}) \curlywedge F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \curlywedge bounds(F_2(\overline{x_i}))\\
bounds(F_1(\overline{x_i}) \sqcap F_2(\overline{x_i})) &= bounds(F_1(\overline{x_i})) \sqcap bounds(F_2(\overline{\ell_i}))
\end{aligned}$$

$$\frac{bounds(F_1(\overline{g_i})) = [\ell_1, \ell_2] \qquad bounds(F_2(\overline{g_j})) = [\ell_1', \ell_2']}{\mathcal{I}(F_1(g_1, \ldots g_n) \mathrel{\widetilde{\preccurlyeq}} F_2(g_{n+1}, \ldots g_{n+m})) = \langle [\ell_1, \ell_2 \curlywedge \ell_2'], [\ell_1 \curlyvee \ell_1', \ell_2'] \rangle}$$

where $F_1 : \text{GLabel}^n \to \text{GLabel}$ and $F_2 : \text{GLabel}^m \to \text{GLabel}$.

$$\mathcal{I}^{\cup}(\widetilde{F(g_1, \ldots, g_n)}) = \mathcal{I}(F(g_1, \ldots, g_n) \mathrel{\widetilde{\preccurlyeq}} F(g_1, \ldots, g_n))$$

The algorithmic definition of meet:

$$\begin{aligned}
[\ell_1, \ell_2] \sqcap [\ell_3, \ell_4] &= [\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4] \quad \text{if } valid([\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4])\\
\iota \sqcap \iota' & \text{ undefined otherwise}
\end{aligned}$$

We calculate the algorithmic definition of $\triangle^{\preccurlyeq}$:

$$\frac{\ell_1 \preccurlyeq \ell_4 \qquad \ell_3 \preccurlyeq \ell_6 \qquad \ell_1 \preccurlyeq \ell_6}{\triangle^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3, \ell_4], [\ell_5, \ell_6]) = \langle [\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5, \ell_6] \rangle}$$

5.4.2 Type Evidences. We define a function *liftP*() to transform functions over types into functions over labels. Also we define function *invert*() to invert the operator on types, used in the domain and latent effect of function types. Finally we define function *tomeet*() to transform type operators into meets, given the invariant property of references.

We start defining a pattern of operations:

Definition 5.36 (Operation pattern).

$$\begin{array}{llll}
\multicolumn{4}{l}{P^T \in \text{GPattern}, P^\ell \in \text{LPattern}}\\
P^T & ::= & _ \mid P^T\ op^T\ P^T & \text{(pattern on types)}\\
op^T & ::= & \dot{\curlyvee} \mid \dot{\curlywedge} \mid \sqcap & \text{(operations on types)}\\
P^\ell & ::= & _ \mid P^\ell\ op^\ell\ P^\ell & \text{(pattern on labels)}\\
op^\ell & ::= & \curlyvee \mid \curlywedge \mid \sqcap & \text{(operations on labels)}
\end{array}$$

$$
\begin{aligned}
liftP(\_) &= \_ \\
liftP(P_1^T \mathbin{\dot\vee} P_2^T) &= liftP(P_1^T) \curlyvee liftP(P_2^T) \\
liftP(P_1^T \mathbin{\dot\wedge} P_2^T) &= liftP(P_1^T) \curlywedge liftP(P_2^T) \\
liftP(P_1^T \sqcap P_2^T) &= liftP(P_1^T) \sqcap liftP(P_2^T) \\
invert(\_) &= \_ \\
invert(P_1^T \mathbin{\dot\vee} P_2^T) &= invert(P_1^T) \mathbin{\dot\wedge} invert(P_2^T) \\
invert(P_1^T \mathbin{\dot\wedge} P_2^T) &= invert(P_1^T) \mathbin{\dot\vee} invert(P_2^T) \\
invert(P_1^T \sqcap P_2^T) &= invert(P_1^T) \sqcap invert(P_2^T) \\
tomeet(\_) &= \_ \\
tomeet(P_1^T \mathbin{\dot\vee} P_2^T) &= tomeet(P_1^T) \sqcap tomeet(P_2^T) \\
tomeet(P_1^T \mathbin{\dot\wedge} P_2^T) &= tomeet(P_1^T) \sqcap tomeet(P_2^T) \\
tomeet(P_1^T \sqcap P_2^T) &= tomeet(P_1^T) \sqcap tomeet(P_2^T)
\end{aligned}
$$

We use case-based analysis to calculate the algorithmic rules for the initial evidence of consistent subtyping on gradual security types:

$$
\frac{\mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_i}) <: liftP(G_2)(\overline{\ell_j})}]\!] = \langle \iota_1, \iota_2 \rangle}{\mathcal{I}[\![\widetilde{G_1(\overline{\mathsf{Bool}_{g_i}}) \preccurlyeq G_2(\overline{\mathsf{Bool}_{g_j}})}]\!] = \langle \mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2} \rangle}
$$

$$
\frac{\begin{array}{c}\mathcal{I}[\![\widetilde{invert(G_2)(\overline{U_{j1}}) <: invert(G_1)(\overline{U_{i1}})}]\!] = \langle E'_{21}, E'_{11} \rangle \qquad \mathcal{I}[\![\widetilde{G_1(\overline{U_{i2}}) <: G_2(\overline{U_{j2}})}]\!] = \langle E_{12}, E_{22} \rangle \\ \mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_{i1}}) <: liftP(G_2)(\overline{\ell_{j1}})}]\!] = \langle \iota_{11}, \iota_{12} \rangle \\ \mathcal{I}[\![\widetilde{liftP(invert(G_2))(\overline{\ell_{j2}}) <: liftP(invert(G_1))(\overline{\ell_{i2}})}]\!] = \langle \iota_{22}, \iota_{21} \rangle\end{array}}{\mathcal{I}[\![\widetilde{G_1(\overline{U_{i1} \xrightarrow{g_{i2}}_{g_{i1}} U_{i2}}) <: G_2(\overline{U_{j1} \xrightarrow{g_{j2}}_{g_{j1}} U_{j2}})}]\!] = \langle E_{11} \xrightarrow{\iota_{21}}_{\iota_{11}} E_{12}, E_{21} \xrightarrow{\iota_{22}}_{\iota_{12}} E_{22} \rangle}
$$

$$
\frac{\begin{array}{c}\mathcal{I}[\![\widetilde{liftP(G_1)(\overline{\ell_i}) <: liftP(G_2)(\overline{\ell_j})}]\!] = \langle \iota_1, \iota_2 \rangle \\ \mathcal{I}[\![\widetilde{tomeet(G_1)(\overline{U_i}) <: tomeet(G_2)(\overline{U_j})}]\!] = \langle E_1, E_2 \rangle \\ \mathcal{I}[\![\widetilde{tomeet(G_2)(\overline{U_j}) <: tomeet(G_1)(\overline{U_i})}]\!] = \langle E'_2, E'_1 \rangle\end{array}}{\mathcal{I}[\![\widetilde{G_1(\overline{\mathsf{Ref}_{g_i}\ U_i}) <: G_2(\overline{\mathsf{Ref}_{g_j}\ U_j})}]\!] = \langle \mathsf{Ref}_{\iota_1}\ E_1 \sqcap E'_1, \mathsf{Ref}_{\iota_2}\ E_2 \sqcap E'_2 \rangle}
$$

where $G_1 : \textsc{GLabel}^n \to \textsc{GLabel}$ and $G_2 : \textsc{GLabel}^m \to \textsc{GLabel}$, and $G_1(x_1, \ldots, x_n) = P_1^T(x_1, \ldots, x_n)$, $G_2(x_1, \ldots, x_n) = P_2^T(x_1, \ldots, x_m)$.

$$
\mathcal{I}^{\circlearrowleft}(\widetilde{F(U_1, \ldots, U_n)}) = \mathcal{I}[\![\widetilde{F(U_1, \ldots, U_n) <: F(U_1, \ldots, U_n)}]\!]
$$

We calculate a recursive meet operator for gradual types:

$$
\begin{gathered}
\mathsf{Bool}_\iota \sqcap \mathsf{Bool}_{\iota'} = \mathsf{Bool}_{\iota \sqcap \iota'} \\
(E_{11} \xrightarrow{\iota_2}_{\iota_1} E_{12}) \sqcap (E_{21} \xrightarrow{\iota'_2}_{\iota'_1} E_{22}) = (E_{11} \sqcap E_{21}) \xrightarrow{\iota_2 \sqcap \iota'_2}_{\iota_1 \sqcap \iota'_1} (E_{12} \sqcap E_{22}) \\
\mathsf{Ref}_\iota\ E_1 \sqcap \mathsf{Ref}_{\iota'}\ E_2 = \mathsf{Ref}_{\iota \sqcap \iota'}\ E_1 \sqcap E_2 \\
U \sqcap U' \text{ undefined otherwise}
\end{gathered}
$$

We calculate a recursive definition for $\triangle^{<:}$ by case analysis on the structure of the second argument,

$$\dfrac{\triangle^{\preccurlyeq}(\iota_1,\iota_2,\iota_3) = \langle \iota_1', \iota_3' \rangle}{\triangle^{<:}(\mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2}, \mathsf{Bool}_{\iota_3}) = \langle \mathsf{Bool}_{\iota_1'}, \mathsf{Bool}_{\iota_3'} \rangle}$$

$$\dfrac{\begin{array}{c}\triangle^{<:}(E_{31}, E_{21}, E_{11}) = \langle E_{31}', E_{11}' \rangle \\ \triangle^{<:}(E_{12}, E_{22}, E_{32}) = \langle E_{12}', E_{32}' \rangle \\ \triangle^{\preccurlyeq}(\iota_1,\iota_2,\iota_3) = \langle \iota_1', \iota_3' \rangle \\ \triangle^{\preccurlyeq}(\iota_{13},\iota_{12},\iota_{11}) = \langle \iota_{13}', \iota_{11}' \rangle\end{array}}{\begin{array}{l}\triangle^{<:}(E_{11} \xrightarrow{\iota_{11}}_{\iota_1} E_{12}, E_{21} \xrightarrow{\iota_{12}}_{\iota_2} E_{22}, E_{31} \xrightarrow{\iota_{13}}_{\iota_3} E_{32}) \\ = \langle E_{11}' \xrightarrow{\iota_{11}'}_{\iota_1'} E_{12}', E_{31}' \xrightarrow{\iota_{13}'}_{\iota_3'} E_{32}' \rangle\end{array}}$$

$$\dfrac{\begin{array}{c}\triangle^{\preccurlyeq}(\iota_1,\iota_2,\iota_3) = \langle \iota_1', \iota_3' \rangle \\ E_1' = E_1 \sqcap E_2 \qquad E_3' = E_2 \sqcap E_3\end{array}}{\triangle^{<:}(\mathsf{Ref}_{\iota_1}\ E_1, \mathsf{Ref}_{\iota_2}\ E_2, \mathsf{Ref}_{\iota_3}\ E_3) = \langle \mathsf{Ref}_{\iota_1'}\ E_1', \mathsf{Ref}_{\iota_3'}\ E_3' \rangle}$$

5.4.3 Evidence inversion functions. The evidence inversion functions are defined as follows

$$\begin{aligned} ilbl(\langle \mathsf{Bool}_{\iota_1}, \mathsf{Bool}_{\iota_2} \rangle) &= \langle \iota_1, \iota_2 \rangle \\ ilbl(\langle \mathsf{Unit}_{\iota_1}, \mathsf{Unit}_{\iota_2} \rangle) &= \langle \iota_1, \iota_2 \rangle \\ ilbl(\langle \mathsf{Ref}_{\iota_1}\ U_1, \mathsf{Ref}_{\iota_2}\ U_2 \rangle) &= \langle \iota_1, \iota_2 \rangle \\ ilbl(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle \iota_1, \iota_1' \rangle \end{aligned}$$

$$\begin{aligned} iref(\langle \mathsf{Ref}_{\iota_1}\ E_1, \mathsf{Ref}_{\iota_2}\ E_2 \rangle) &= \langle E_1, E_2 \rangle \\ iref(\langle E_1, E_2 \rangle) &= \text{undefined otherwise} \end{aligned}$$

$$\begin{aligned} idom(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle E_1', E_1 \rangle \\ idom(\langle E_1, E_2 \rangle) &= \text{undefined otherwise} \end{aligned}$$

$$\begin{aligned} icod(\langle E_1 \xrightarrow{\iota_2}_{\iota_1} E_2, E_1' \xrightarrow{\iota_2'}_{\iota_1'} E_2' \rangle) &= \langle E_2, E_2' \rangle \\ icod(\langle E_1, E_2 \rangle) &= \text{undefined otherwise} \end{aligned}$$

5.5 Proofs

Proposition 5.13 (α_ι is Sound). *If $\widehat{\ell}$ is not empty, then $\widehat{\ell} \subseteq \gamma_\iota(\alpha_\iota(\widehat{\ell}))$.*

Proof. Suppose $\widehat{\ell} = \{\, \overline{\ell_i} \,\}$. By definition of $\alpha_{\varepsilon_\ell}$, $\alpha_\iota(\{\, \overline{\ell_i} \,\}) = [\curlywedge\overline{\ell_i}, \curlyvee\overline{\ell_i}]$. Therefore

$$\gamma_\iota(\alpha_\iota(\{\, \overline{\ell_i} \,\})) = \{\ell \mid \ell \in \textsc{Label}, \curlywedge\overline{\ell_i} \preccurlyeq \ell \preccurlyeq \curlyvee\overline{\ell_i}\}$$

And it is easy to see that if $\ell \in \{\, \overline{\ell_i} \,\}$, then $\ell \in \gamma_\iota(\alpha_\iota(\{\, \overline{\ell_i} \,\}))$, and therefore the result holds. □

Proposition 5.14 (α_ι is Optimal). *If $\widehat{\ell}$ is not empty, and $\widehat{\ell} \subseteq \gamma_\iota(\iota)$ then $\alpha_\iota(\widehat{\ell}) \sqsubseteq \iota$.*

Proof. By case analysis on the structure of ι. If $\iota = [\ell_1, \ell_2]$, $\gamma_{\varepsilon_\ell}(\iota) = \{\ell \mid \ell \in \textsc{Label}, \ell_1 \preccurlyeq \ell \preccurlyeq \ell_2\}$; $\widehat{\ell} \subseteq \{\ell \mid \ell \in \textsc{Label}, \ell_1 \preccurlyeq \ell \preccurlyeq \ell_2\}$, $\widehat{\ell} \neq \emptyset$ implies $\alpha_{\varepsilon_\ell}(\widehat{\ell}) = [\ell_3, \ell_4]$, where $\ell_1 \preccurlyeq \ell_3$ and $\ell_4 \preccurlyeq \ell_2$, therefore $[\ell_3, \ell_4] \sqsubseteq \iota$ (if $\widehat{\ell} = \emptyset$, $\alpha_{\varepsilon_\ell}(\widehat{\ell})$ is undefined). □

Proposition 5.15 (α_E is Sound). *If $valid(\widehat{S})$ then $\widehat{S} \subseteq \gamma_E(\alpha_E(\widehat{S}))$.*

Proof. By well-founded induction on $\widehat{S}$. Similar to Prop 4.4. □

Proposition 5.16 (α_E is Optimal). *If $valid(\widehat{S})$ and $\widehat{S} \subseteq \gamma_E(E)$ then $\alpha_E(\widehat{S}) \sqsubseteq E$.*

Proof. By induction on the structure of U. Similar to Prop 4.5. □

Proposition 5.23. $\gamma_\iota(\iota_1 \sqcap \iota_2) = \gamma_\iota(\iota_1) \cap \gamma_\iota(\iota_2)$.

Proof.

$$\begin{aligned} \gamma_\iota(\iota_1 \sqcap \iota_2) &= \gamma_\iota(\alpha_\iota(\gamma_\iota(\iota_1) \cap \gamma_\iota(\iota_1))) \\ &\subseteq \gamma_\iota(\iota_1) \cap \gamma_\iota(\iota_1) \quad \text{(soundness of } \alpha_\iota) \end{aligned}$$

Let $\ell \in \gamma_\iota(\iota_1) \cap \gamma_\iota(\iota_1)$. We now that $\gamma_\iota(\iota_1 \sqcap \iota_2)$ is defined. Suppose $\iota_1 = [\ell_1, \ell_2]$ and $\iota_2 = [\ell_3, \ell_4]$. Therefore $\iota_1 \sqcap \iota_2 = [\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4]$.

But $\gamma_\iota(\iota_1) \cap \gamma_\iota(\iota_1) = \{\ell \mid \ell \in \text{Label}, \ell_1 \preccurlyeq \ell \preccurlyeq \ell_2\} \cap \{\ell \mid \ell \in \text{Label}, \ell_3 \preccurlyeq \ell \preccurlyeq \ell_4\}$. Which is equivalent to $\{\ell \mid \ell \in \text{Label}, \ell_1 \preccurlyeq \ell \preccurlyeq \ell_2 \wedge \ell_3 \preccurlyeq \ell \preccurlyeq \ell_4\}$, equivalent to $\{\ell \mid \ell \in \text{Label}, \ell_1 \curlyvee \ell_3 \preccurlyeq \ell \preccurlyeq \ell_2 \curlywedge \ell_4\}$. Which is by definition $\gamma_\iota([\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4])$, and the result holds.

□

Proposition 5.24. $\langle \iota_1, \iota_{21} \rangle \circ^{\preccurlyeq} \langle \iota_{22}, \iota_3 \rangle = \triangle^{\preccurlyeq}(\iota_1, \iota_{21} \sqcap \iota_{22}, \iota_3)$

Proof. Follows directly from the definition of consistent transitivity and Prop 5.23. □

Proposition 5.26. $\gamma_E(E_1 \sqcap E_2) = \gamma_E(E_1) \cap \gamma_E(E_2)$.

Proof. By induction on evidence types ε_1 and ε_2 and Prop 5.23. □

Proposition 5.27.

$$\langle E_1, E_{21} \rangle \circ^{<:} \langle E_{22}, E_3 \rangle = \triangle^{<:}(E_1, E_{21} \sqcap E_{22}, E_3)$$

where

$$\triangle^{<:}(E_1, E_2, E_3) = \alpha_\varepsilon(\{\langle S_1, S_3 \rangle \in \gamma_\varepsilon(\langle E_1, E_3 \rangle) \mid \exists S_2 \in \gamma_\iota(E_2). S_1 <: S_2 \wedge S_2 <: S_3\})$$

Proof. Follows directly from the definition of consistent transitivity and Prop 5.26. □

Proposition 5.32. *If $\varepsilon_S \vdash U_1 \lesssim U_2$ and $\varepsilon_l \vdash g_1 \widetilde{\preccurlyeq} g_2$ then $\varepsilon_S \mathbin{\widetilde{\curlyvee}} \varepsilon_l \vdash U_1 \mathbin{\widetilde{\curlyvee}} g_1 <: U_2 \mathbin{\widetilde{\curlyvee}} g_2$*

Proof. By induction on types U_1 and U_2, using the definition of $\mathcal{I}_{<:}$ and Proposition 5.43. □

Proposition 5.37. $[\ell_1, \ell_2] \curlyvee [\ell_3, \ell_4] = [\ell_1 \curlyvee \ell_3, \ell_2 \curlyvee \ell_4]$

Proof. Follows directly by definition of γ and $\curlyvee$. □

Proposition 5.38.

$$\langle \iota_1, \iota_2 \rangle \mathbin{\widetilde{\curlyvee}} \langle \iota_1', \iota_2' \rangle = \langle \iota_1 \curlyvee \iota_1', \iota_2 \curlyvee \iota_2' \rangle$$

Proof. Follows directly from the definition of consistent join monotonicity and Prop 5.37. □

Proposition 5.39.

$$\begin{aligned} [\ell_1, \ell_2] \sqcap [\ell_3, \ell_4] &= [\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4] \quad \textit{if } \ell_1 \curlyvee \ell_3 \preccurlyeq \ell_2 \curlywedge \ell_4 \\ \iota \sqcap \iota' &\ \textit{undefined otherwise} \end{aligned}$$

Proof. By definition of meet:

$$[\ell_1, \ell_2] \sqcap [\ell_3, \ell_4] = \alpha_\iota(\{\ell' \mid \ell' \in \gamma([\ell_1, \ell_2]) \cap \gamma([\ell_3, \ell_4])\})$$

But by definition of intersection on intervals, $\gamma([\ell_1, \ell_2]) \cap \gamma([\ell_3, \ell_4]) = \gamma([\ell_1 \curlyvee \ell_3, \ell_2 \curlywedge \ell_4])$ if $\ell_1 \curlyvee \ell_3 \preccurlyeq \ell_2 \curlywedge \ell_4$ (otherwise the intersection is empty), and the result follows by definition of α_ι. □

Proposition 5.40.

$$\frac{\ell_1 \preccurlyeq \ell_4 \qquad \ell_3 \preccurlyeq \ell_6 \qquad \ell_1 \preccurlyeq \ell_6}{\triangle^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3, \ell_4], [\ell_5, \ell_6]) = \langle [\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5, \ell_6] \rangle}$$

Proof. By definition:

$$\triangle^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3, \ell_4], [\ell_5, \ell_6]) = \alpha_\varepsilon(\{\langle \ell'_1, \ell'_3 \rangle \in \gamma_\varepsilon(\langle [\ell_1, \ell_2], [\ell_5, \ell_6] \rangle) \mid \exists \ell'_2 \in \gamma_\imath([\ell_3, \ell_4]).\ell'_1 \preccurlyeq \ell'_2 \preccurlyeq \ell'_3\})$$

It is easy to see that $\alpha_\imath(\{\ell'_{1i}\}) = [\ell_1, \ell'_{12}]$, for some ℓ'_{12}. We know that $\ell'_{12} \preccurlyeq \ell_2$, $\ell'_{12} \preccurlyeq \ell_4$ and $\ell'_{12} \preccurlyeq \ell_6$, *i.e.* $\ell'_{12} \preccurlyeq \ell_2 \curlywedge \ell_4 \curlywedge \ell_6$. But $\ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \preccurlyeq \ell_4 \preccurlyeq \ell_6$ therefore

$$\langle \ell_2 \curlywedge \ell_4 \curlywedge \ell_6, \ell_6 \rangle \in \{\langle \ell'_1, \ell'_3 \rangle \in \gamma_\varepsilon(\langle [\ell_1, \ell_2], [\ell_5, \ell_6] \rangle) \mid \\ \exists \ell'_2 \in \gamma_\imath([\ell_3, \ell_4]).\ell'_1 \preccurlyeq \ell'_2 \preccurlyeq \ell'_3\}$$

and by definition of $\alpha_\imath$, $\ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \preccurlyeq \ell'_{12}$, then $\alpha_\imath(\{\ell'_{1i}\}) = [\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6]$. Similar argument is used to prove that $\alpha_\imath(\{\ell'_{3i}\}) = [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5, \ell_6]$. □

Lemma 5.41. *Let $\ell_i \in$ Label, then $(\ell_1 \curlywedge \ell_2) \curlyvee (\ell_3 \curlywedge \ell_4) \preccurlyeq (\ell_1 \curlyvee \ell_3) \curlywedge (\ell_2 \curlyvee \ell_4)$.*

Proof.

$$\begin{array}{ll} & (\ell_1 \curlywedge \ell_2) \curlyvee (\ell_3 \curlywedge \ell_4) \\ \preccurlyeq & (\ell_1 \curlyvee (\ell_3 \curlywedge \ell_4)) \curlywedge (\ell_2 \curlyvee (\ell_3 \curlywedge \ell_4)) \\ \preccurlyeq & ((\ell_1 \curlyvee \ell_3) \curlywedge (\ell_1 \curlyvee \ell_4)) \curlywedge ((\ell_2 \curlyvee \ell_3) \curlywedge (\ell_2 \curlyvee \ell_4)) \\ \preccurlyeq & (\ell_1 \curlyvee \ell_3) \curlywedge (\ell_2 \curlyvee \ell_4) \end{array}$$

□

Proposition 5.42. *Suppose $\varepsilon_1 \vdash \widetilde{F_1(\overline{g_i}) \preccurlyeq F_2(\overline{g_j})}$ and $\varepsilon_2 \vdash \widetilde{F_2(\overline{g_j}) \preccurlyeq F_3(\overline{g_k})}$. If $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ is defined, then $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 \vdash \widetilde{F_1(\overline{g_i}) \preccurlyeq F_3(\overline{g_k})}$*

Proof. Suppose $\varepsilon_1 = \langle \imath_{11}, \imath_{12} \rangle$ and $\varepsilon_2 = \langle \imath_{21}, \imath_{22} \rangle$. Then by definition of initial evidence:

$$\langle \imath_{11}, \imath_{12} \rangle = \langle [\ell_1, \ell_2], [\ell_3, \ell_4] \rangle \sqsubseteq \mathcal{I}[\![\widetilde{F_1(\overline{g_i}) \preccurlyeq F_2(\overline{g_j})}]\!] = \langle \imath'_{11}, \imath'_{12} \rangle$$

and

$$\langle \imath_{21}, \imath_{22} \rangle = \langle [\ell_5, \ell_6], [\ell_7, \ell_8] \rangle \sqsubseteq \mathcal{I}[\![\widetilde{F_2(\overline{g_j}) \preccurlyeq F_3(\overline{g_k})}]\!] = \langle \imath'_{21}, \imath'_{22} \rangle$$

Suppose that $\mathcal{I}[\![\widetilde{F_1(\overline{g_i}) \preccurlyeq F_3(\overline{g_k})}]\!] = \langle \imath'_1, \imath'_3 \rangle$. We have to prove that $\langle \imath_{11}, \imath_{12} \rangle \circ^{\preccurlyeq} \langle \imath_{21}, \imath_{22} \rangle \sqsubseteq \langle \imath'_1, \imath'_3 \rangle$.

If $bounds(F_1(\overline{g_i})) = [\ell'_1, \ell'_2]$, $bounds(F_2(\overline{g_j})) = [\ell'_3, \ell'_4]$, and $bounds(F_3(\overline{g_i})) = [\ell'_5, \ell'_6]$ We know that $\mathcal{I}[\![\widetilde{F_1(\overline{g_i}) \preccurlyeq F_2(\overline{g_j})}]\!] = \langle [\ell'_1, \ell'_2 \curlywedge \ell'_4], [\ell'_1 \curlyvee \ell'_3, \ell'_4] \rangle$. Therefore $\ell'_1 \preccurlyeq \ell_1$, $\ell_2 \preccurlyeq \ell'_2 \curlywedge \ell'_4$, $\ell'_1 \curlyvee \ell'_2 \preccurlyeq \ell_3$ and $\ell_4 \preccurlyeq \ell'_4$.

Using the same argument,

$\mathcal{I}[\![\widetilde{F_2(\overline{g_j}) \preccurlyeq F_3(\overline{g_k})}]\!] = \langle [\ell'_3, \ell'_4 \curlywedge \ell'_6], [\ell'_3 \curlyvee \ell'_5, \ell'_6] \rangle$. Therefore $\ell'_3 \preccurlyeq \ell_5$, $\ell_6 \preccurlyeq \ell'_4 \curlywedge \ell'_6$, $\ell'_3 \curlyvee \ell'_5 \preccurlyeq \ell_7$ and $\ell_8 \preccurlyeq \ell'_6$.

But $\mathcal{I}[\![\widetilde{F_1(\overline{g_i}) \preccurlyeq F_3(\overline{g_k})}]\!] = \langle [\ell'_1, \ell'_2 \curlywedge \ell'_6], [\ell'_1 \curlyvee \ell'_5, \ell'_6] \rangle$ and

$$\begin{array}{l} \langle \imath_{11}, \imath_{12} \rangle \circ^{\preccurlyeq} \langle \imath_{21}, \imath_{22} \rangle = \triangle^{\preccurlyeq}(\imath_{11}, \imath_{12} \sqcap \imath_{21}, \imath_{22}) = \\ \qquad \triangle^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3 \curlyvee \ell_5, \ell_4 \curlywedge \ell_6], [\ell_7, \ell_8]) \\ \qquad\qquad = \langle [\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7, \ell_8] \rangle \end{array}$$

we need to prove that

$$\langle[\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7, \ell_8]\rangle \sqsubseteq \langle[\ell'_1, \ell'_2 \curlywedge \ell'_6], [\ell'_1 \curlyvee \ell'_5, \ell'_6]\rangle$$

. But we know that $\ell'_1 \preccurlyeq \ell_1$. Also that $\ell_2 \preccurlyeq \ell'_2 \curlywedge \ell'_4$ and therefore $\ell_2 \preccurlyeq \ell'_2$. The same for $\ell_6 \preccurlyeq \ell'_6$ and therefore $\ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8 \preccurlyeq \ell'_2 \curlywedge \ell'_6$, i.e. $[\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8] \sqsubseteq [\ell'_1, \ell'_2 \curlywedge \ell'_6]$. The argument is applied for the second components and the result holds.

□

Proposition 5.43. *Suppose* $\varepsilon_1 \vdash \widetilde{F_{11}(\overline{g_i}) \preccurlyeq F_{12}(\overline{g_j})}$ *and* $\varepsilon_2 \vdash \widetilde{F_{21}(\overline{g_i}) \preccurlyeq F_{22}(\overline{g_j})}$ *Then* $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2 \vdash \widetilde{F_{11}(\overline{g_i}) \curlyvee F_{21}(\overline{g_i}) \preccurlyeq F_{12}(\overline{g_j}) \curlyvee F_{22}(\overline{g_j})}$

Proof. By definition of initial evidence noticing that $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2$ can be more precise than the initial evidence of judgment

Suppose $\varepsilon_1 = \langle[\ell_1, \ell_2], [\ell_3, \ell_4]\rangle$, and $\varepsilon_2 = \langle[\ell_5, \ell_6], [\ell_7, \ell_8]\rangle$, then $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2 = \langle[\ell_1 \curlyvee \ell_5, \ell_2 \curlyvee \ell_6], [\ell_3 \curlyvee \ell_6, \ell_4 \curlyvee \ell_8]\rangle$.

If $bounds(F_{11}(\overline{g_i})) = [\ell'_{111}, \ell'_{112}]$, $bounds(F_{12}(\overline{g_i})) = [\ell'_{121}, \ell'_{122}]$, $bounds(F_{21}(\overline{g_i})) = [\ell'_{211}, \ell'_{212}]$ and $bounds(F_{22}(\overline{g_i})) = [\ell'_{221}, \ell'_{222}]$.

We know that $\mathcal{I}[\![\widetilde{F_{11}(\overline{g_i}) \preccurlyeq F_{12}(\overline{g_j})}]\!] = \langle[\ell'_{111}, \ell'_{112} \curlywedge \ell'_{122}], [\ell'_{111} \curlyvee \ell'_{121}, \ell'_{122}]\rangle$. Therefore $\ell'_{111} \preccurlyeq \ell_1$, $\ell_2 \preccurlyeq \ell'_{112} \curlywedge \ell'_{122}$, $\ell'_{111} \curlyvee \ell'_{121} \preccurlyeq \ell_3$ and $\ell_4 \preccurlyeq \ell'_{122}$. Using the same argument, $\mathcal{I}[\![\widetilde{F_{21}(\overline{g_i}) \preccurlyeq F_{22}(\overline{g_j})}]\!] = \langle[\ell'_{211}, \ell'_{212} \curlywedge \ell'_{222}], [\ell'_{211} \curlyvee \ell'_{221}, \ell'_{222}]\rangle$. Therefore $\ell'_{211} \preccurlyeq \ell_5$, $\ell_6 \preccurlyeq \ell'_{212} \curlywedge \ell'_{222}$, $\ell'_{211} \curlyvee \ell'_{221} \preccurlyeq \ell_7$ and $\ell_8 \preccurlyeq \ell'_{222}$.

But the $\mathcal{I}[\![\widetilde{F'_1(\overline{g_i}) \preccurlyeq F'_2(\overline{g_j})}]\!] = \langle[\ell'_1, \ell'_2 \curlywedge \ell'_4], [\ell'_1 \curlyvee \ell'_3, \ell'_4]\rangle$ where
$bounds(F'_1(\overline{g_i})) = bounds(F_{11}(\overline{g_i})) \curlyvee bounds(F_{21}(\overline{g_i})) = [\ell'_{111}, \ell'_{112}] \curlyvee [\ell'_{211}, \ell'_{212}] = [\ell'_{111} \curlyvee \ell'_{211}, \ell'_{112} \curlyvee \ell'_{212}]$,
and
$bounds(F'_2(\overline{g_i})) = bounds(F_{12}(\overline{g_i})) \curlyvee bounds(F_{22}(\overline{g_i})) = [\ell'_{121}, \ell'_{122}] \curlyvee [\ell'_{221}, \ell'_{222}] = [\ell'_{121} \curlyvee \ell'_{221}, \ell'_{122} \curlyvee \ell'_{222}]$.

We need to prove that $[\ell_1 \curlyvee \ell_5, \ell_2 \curlyvee \ell_6] \sqsubseteq [\ell'_{111} \curlyvee \ell'_{211}, \ell'_{112} \curlyvee \ell'_{212}]$, *i.e.* $\ell'_{111} \curlyvee \ell'_{211} \preccurlyeq \ell_1 \curlyvee \ell_5$ and $\ell_2 \curlyvee \ell_6 \preccurlyeq \ell'_{112} \curlyvee \ell'_{212}$. But $\ell'_{111} \preccurlyeq \ell_1$ and $\ell'_{211} \preccurlyeq \ell_5$, therefore $\ell'_{111} \curlyvee \ell'_{211} \preccurlyeq \ell_1 \curlyvee \ell_5$. Similarly, as $\ell_2 \preccurlyeq \ell'_{112} \curlywedge \ell'_{122}$ and $\ell_6 \preccurlyeq \ell'_{212} \curlywedge \ell'_{222}$, then $\ell_2 \curlyvee \ell_6 \preccurlyeq \ell'_{112} \curlyvee \ell'_{212}$. Therefore $[\ell_1 \curlyvee \ell_5, \ell_2 \curlyvee \ell_6] \sqsubseteq [\ell'_{111} \curlyvee \ell'_{211}, \ell'_{112} \curlyvee \ell'_{212}]$.

Using analogous argument, we also know that $[\ell_3 \curlyvee \ell_6, \ell_4 \curlyvee \ell_8] \sqsubseteq [\ell'_{121} \curlyvee \ell'_{221}, \ell'_{122} \curlyvee \ell'_{222}]$. Therefore $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2 \sqsubseteq \mathcal{I}[\![\widetilde{F'_1(\overline{g_i}) \preccurlyeq F'_2(\overline{g_j})}]\!]$, and the result holds.

□

Lemma 5.44. *Let* $S_1, S_2 \in$ Type. *Then*

(1) If $(S_1 \mathbin{\dot{\curlyvee}} S_2)$ *is defined then* $S_1 <: (S_1 \mathbin{\dot{\curlyvee}} S_2)$.
(2) If $(S_1 \mathbin{\dot{\curlywedge}} S_2)$ *is defined then* $(S_1 \mathbin{\dot{\curlywedge}} S_2) <: S_1$.

Proof. We start by proving (1) assuming that $(S_1 \curlyvee S_2)$ is defined. We proceed by case analysis on S_1.

Case (Bool_ℓ). If $S_1 = \mathsf{Bool}_{\ell_1}$ then as $(S_1 \mathbin{\dot{\curlyvee}} S_2)$ is defined then S_2 must have the form Bool_{ℓ_2} for some ℓ_2. Therefore $(S_1 \mathbin{\dot{\curlyvee}} S_2) = \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}$. But by definition of $\preccurlyeq$, $\ell_1 \preccurlyeq (\ell_1 \curlyvee \ell_2)$ and therefore we use $(<:_{\mathsf{Bool}})$ to conclude that $\mathsf{Bool}_{\ell_1} <: \mathsf{Bool}_{(\ell_1 \curlyvee \ell_2)}$, i.e. $S_1 <: (S_1 \mathbin{\dot{\curlyvee}} S_2)$.

Case $(S \rightarrow_\ell S)$. If $S_1 = S_{11} \rightarrow_{\ell_1} S_{12}$ then as $(S_1 \mathbin{\dot{\curlyvee}} S_2)$ is defined then S_2 must have the form $S_{21} \rightarrow_{\ell_2} S_{22}$ for some S_{21}, S_{22} and ℓ_2.

We also know that $(S_1 \curlyvee S_2) = (S_{11} \curlywedge S_{21}) \rightarrow_{(\ell_1 \curlyvee \ell_2)} (S_{12} \curlywedge S_{22})$. By definition of $\preccurlyeq$, $\ell_1 \preccurlyeq (\ell_1 \curlyvee \ell_2)$. Also, as $(S_1 \curlyvee S_2)$ is defined then $(S_{11} \curlywedge S_{21})$ is defined. Using the induction hypothesis of (2) on S_{11}, $(S_{11} \curlywedge S_{21}) <: S_{11}$. Also, using the induction hypothesis of (1) on S_{12} we also know that $S_{12} <: (S_{12} \curlywedge S_{22})$. Then by $(<:_{\rightarrow})$ we can conclude that $S_{11} \rightarrow_{\ell_1} S_{12} <: (S_{11} \curlywedge S_{21}) \rightarrow_{(\ell_1 \curlyvee \ell_2)} (S_{12} \curlywedge S_{22})$, i.e. $S_1 <: (S_1 \curlyvee S_2)$.

The proof of (2) is similar to (1) but using the argument that $(\ell_1 \curlywedge \ell_2) \preccurlyeq \ell_1$. □

Lemma 5.45. *Let $S \in$ Type and $\ell \in$ Label. Then $S <: S \curlyvee \ell$.*

Proof. Straigthforward case analysis on type S using the fact that $\ell \preccurlyeq (\ell' \curlyvee \ell)$ for any ℓ'. □

Lemma 5.46. *Let $S_1, S_2 \in$ Type such that $S_1 <: S_2$, and let $\ell_1, \ell_2 \in$ Label such that $\ell_1 \preccurlyeq \ell_2$. Then $S_1 \curlyvee \ell_1 <: S_2 \curlyvee \ell_2$.*

Proof. Straightforward case analysis on type S using the definition of *label stamping* on types. □

6 $\text{GSL}^{\varepsilon}_{\text{Ref}}$: DYNAMIC PROPERTIES

Notice that for convenience, the proofs and properties are defined over intrinsic terms [Garcia et al. 2016] instead of terms of the internal language. They are actually the same as terms of the internal language, but keeping all static annotations explicitly. First we introduce the static semantics of intrinsic terms in Sec. 6.1. Their dynamic semantics in Sec. 6.2. The relation between intrinsic and evidence-augmented terms in Sec. 6.3. Then the proof of type safety is presented Sec. 6.4, the proof of dynamic gradual guarantee for $\text{GSL}^{\varepsilon}_{\text{Ref}}$ without the specific check in rule $(r7)$ in section 6.5, and the proof of noninterference in Sec. 6.6.

6.1 Intrinsic Terms: Static Semantics

Following Garcia et al. [2016], we develop *intrinsically typed* terms [Church 1940]: a term notation for gradual type derivations. These terms serve as our internal language for dynamic semantics: they play the same role that cast calculi play in typical presentations of gradual typing [Siek and Taha 2006]. Intrinsically-typed terms t^U comprise a family $\mathbb{T}[U]$ of type-indexed sets, such that ill-typed terms do not exist. They are built up from disjoint families $x^U \in \mathbb{V}[U]$ and $o^U \in \mathbb{L}[U]$ of intrinsically typed variables and locations respectively. Unless required, we omit the type exponent on intrinsic terms, writing $\check{t} \in \mathbb{T}[U]$.

To each typing rule corresponds an intrinsic term formation rule that captures all the information needed to ensure that an intrinsic term is isomorphic to a typing derivation. Because intrinsic variables and locations reflect their typings, intrinsic terms do not need explicit type environments Γ or store environments Σ; however, the typing judgment depends on a security effect g_c, which intrinsic terms must account for.

Additionally, because intrinsic terms represent typing derivations of programs *as they reduce*, they must account for the possibility that runtime values have more precise types than those used in the original typing derivation. For instance, the term in function position of an application can be a subtype of the function type used to type-check the program originally. The formation rule of the application intrinsic term must permit this extra subtyping leeway, justified by evidence. The same holds for the security information. Therefore, an intrinsic term has the general form $\phi \triangleright \check{t}$, where the context information $\phi \triangleq \langle \varepsilon g_c, g_c \rangle$ contains the static program counter label g_c used

$$\varepsilon \in \text{Evidence}, \quad et \in \text{EvTerm}, \quad ev \in \text{EvValue}, \quad v \in \text{Value},$$
$$u \in \text{SimpleValue}, g \in \text{EvFrame}, \quad f \in \text{TmFrame}$$

$$\begin{array}{lcl}
u & ::= & x^U \mid b_g \mid (\lambda^g x^U.\check{t})_g \mid o^U_g \mid \mathsf{unit}_g \\
v & ::= & u \mid \varepsilon u :: U \\
f & ::= & h[\varepsilon] \\
\mu & := & \bullet \mid \mu, o^U \mapsto v \\
p & ::= & x^U \mid o^U \\
q & ::= & p \mid \varepsilon p :: U
\end{array}
\qquad
\begin{array}{lcl}
\varepsilon & ::= & \langle E_1, E_2\rangle \mid \langle \iota_1, \iota_2\rangle \\
et & ::= & \varepsilon\check{t} \\
ev & ::= & \varepsilon u \\
e\ell & ::= & \varepsilon g \\
\phi & ::= & \langle \varepsilon g, g\rangle
\end{array}$$

$$\begin{array}{lcl}
h & ::= & \square \oplus^g et \mid ev \oplus^g \square \mid \square @^U_\varepsilon\ et \mid ev @^U_\varepsilon \square \mid \square :: U \mid \mathsf{if}^g\ \square\ \mathsf{then}\ et\ \mathsf{else}\ et \\
& \mid & !^U \square \mid \square \overset{g,U}{:=}_\varepsilon et \mid ev \overset{g,U}{:=}_\varepsilon \square \mid \mathsf{ref}^U_\varepsilon\ \square \mid \mathsf{prot}^{g,U}_{\varepsilon g}\ \phi'(\varepsilon t)
\end{array}$$

Fig. 23. GSL_{Ref}: Syntax of the Intrinsic Term Language

to type-check the source term, as well as the runtime program counter label g_c, along with the evidence $\varepsilon \vdash g_c \mathrel{\widetilde{\preccurlyeq}} g_c$.[1] For simplicity we define accessors $\phi.g_c \triangleq g_c$, $\phi.g_c \triangleq g_c$, and $\phi.\varepsilon \triangleq \varepsilon$.

Figure 23, presents the syntax of intrinsic terms. Fig. 24 presents the intrinsic terms formation rules for GSL_{Ref}. In rule (Iprot), labels g and g' represent the static and dynamic information of the label used to increase the program counter label in the subterm, respectively. Evidence ε_1 justifies that the type of the subterm is a consistent subtype of U, the static type of the subterm. ϕ' represents the context information associated to the subterm $\check{t}$: $\phi'.g_c$ (resp. $\phi'.g_c$) is the program counter label used to typecheck (resp. evaluate) $\check{t}$.

In the intrinsic term formation rule for applications (Iapp), U_1 is the runtime type of the function term. We annotate the initial static type information with @. The evidence ε_2 for the label ordering premise is also annotated, since it is needed to reconstruct the derivation. The intrinsic term of a conditional, described in Rule (Iif)[2], carries the static information of the label of the conditional term g. The context information ϕ' used for both branches is obtained by joining the term context ϕ point-wise with the evidence and labels associated with the consistent subtyping judgment of the conditional. Evidences ε_2 and ε_3 justify that the type of each branch is a consistent subtype of the join of both types. Finally, rule (Iassgn) is built similarly to the application rule (Iapp).

6.2 Intrinsic Terms: Dynamic Semantics

Next we present the full definition of the intrinsic reduction rules in Figure 25, and the full definition of notions of intrinsic reduction in Figure 26.

Because the security context information of a term is maintained at each step, we also adopt the lightweight notation $\check{t}_1 \mid \mu_1 \overset{\phi}{\longmapsto} \check{t}_2 \mid \mu_2$, to denote the reduction of the intrinsic term $\phi \triangleright \check{t}_1 \in \mathbb{T}[U]$ in store μ_1 to the intrinsic term $\phi \triangleright \check{t}_2 \in \mathbb{T}[U]$ in store μ_2. We note $\mathbb{C}[U]$ the combination of a term $\check{t} \in \mathbb{T}[U]$ (without context) and a store μ. Function applications reduce to to an error if consistent transitivity fails to justify $U_2 <: U_{11}$. Conditionals similarly reduce to a new prot term, which is constructed using the static and dynamic information of the conditional term. Assignments may reduce to an ascribed unit value. Similarly to references, the stored value is ascribed the statically determined type U. Therefore consistent transitivity may fail to justify that the actual type of the

[1] We use color to make distinctions when is needed: green is for effects and static information; orange is for the runtime information of the security effect.

[2] Evidence inversion functions (*idom*, *icod*, *iref*, *ilbl* and *ilat*) manifest the evidence for the inversion principles on consistent subtyping judgments; *e.g.* starting from the evidence that $U_1 \lesssim U_2$, *ilbl* produces the evidence of the judgment $label(U_1) \mathrel{\widetilde{\preccurlyeq}} label(U_2)$.

$$\text{(Ix)}\ \frac{}{\phi \triangleright x^U \in \mathbb{T}[U]} \qquad \text{(Ib)}\ \frac{}{\phi \triangleright b_g \in \mathbb{T}[\mathsf{Bool}_g]} \qquad \text{(Iu)}\ \frac{}{\phi \triangleright \mathsf{unit}_g \in \mathbb{T}[\mathsf{Unit}_g]}$$

$$\text{(Il)}\ \frac{}{\phi \triangleright o^U_g \in \mathbb{T}[\mathsf{Ref}_g\ U]} \qquad \text{(I}\lambda\text{)}\ \frac{\phi' = \langle \varepsilon, g', g' \rangle \quad \phi \triangleright \check{t} \in \mathbb{T}[U_2] \quad \varepsilon \vdash g' \mathrel{\widetilde{\preccurlyeq}} g'}{\phi \triangleright (\lambda^{g'} x^{U_1}.\check{t})_g \in \mathbb{T}[U_1 \xrightarrow{g'}_g U_2]}$$

$$\text{(Iprot)}\ \frac{\begin{array}{c}\phi' \triangleright \check{t} \in \mathbb{T}[U'] \\ \varepsilon_1 \vdash U' \lesssim U \quad \varepsilon_2 \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \mathsf{prot}^{g,U}_{\varepsilon_2 g'} \phi'(\varepsilon_1 \check{t}) \in \mathbb{T}[U \mathbin{\widetilde{\curlyvee}} g]} \qquad \text{(I}\oplus\text{)}\ \frac{\begin{array}{cc}\phi \triangleright \check{t}_1 \in \mathbb{T}[U_1] & \varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_{g_1} \\ \phi \triangleright \check{t}_2 \in \mathbb{T}[U_2] & \varepsilon_2 \vdash U_2 \lesssim \mathsf{Bool}_{g_2}\end{array}}{\phi \triangleright \varepsilon_1 \check{t}_1 \oplus^{g_1 \widetilde{\curlyvee} g_2} \varepsilon_2 \check{t}_2 \in \mathbb{T}[\mathsf{Bool}_{g_1 \widetilde{\curlyvee} g_2}]}$$

$$\text{(Iapp)}\ \frac{\phi \triangleright \check{t}_i \in \mathbb{T}[U_i] \quad \varepsilon_1 \vdash U_1 \lesssim U_{11} \xrightarrow{g'}_g U_{12} \quad \varepsilon_2 \vdash U_2 \lesssim U_{11} \quad \varepsilon_3 \vdash \widetilde{\phi.g_c \curlyvee g \preccurlyeq g'}}{\phi \triangleright \varepsilon_1 \check{t}_1 \mathbin{@^{U_{11} \xrightarrow{g'}_g U_{12}}_{\varepsilon_3}} \varepsilon_2 \check{t}_2 \in \mathbb{T}[U_{12} \mathbin{\widetilde{\curlyvee}} g]}$$

$$\text{(Iif)}\ \frac{\begin{array}{c}\phi \triangleright \check{t}_1 \in \mathbb{T}[U_1] \quad \varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_g \quad \phi' = \phi \mathbin{\widetilde{\curlyvee}} \langle \mathit{ilbl}(\varepsilon_1), \mathit{label}(U_1), g \rangle \\ \phi' \triangleright \check{t}_2 \in \mathbb{T}[U_2] \quad \varepsilon_2 \vdash U_2 \lesssim U_2 \mathbin{\widetilde{\vee}} U_3 \quad \phi' \triangleright \check{t}_3 \in \mathbb{T}[U_3] \quad \varepsilon_3 \vdash U_3 \lesssim U_2 \mathbin{\widetilde{\vee}} U_3\end{array}}{\phi \triangleright \mathsf{if}^g\ \varepsilon_1 \check{t}_1\ \mathsf{then}\ \varepsilon_2 \check{t}_2\ \mathsf{else}\ \varepsilon_3 \check{t}_3 \in \mathbb{T}[(U_2 \mathbin{\widetilde{\vee}} U_3) \mathbin{\widetilde{\curlyvee}} g]}$$

$$\text{(Iref)}\ \frac{\begin{array}{c}\phi \triangleright \check{t} \in \mathbb{T}[U'] \\ \varepsilon_1 \vdash U' \lesssim U \quad \varepsilon_2 \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \mathit{label}(U)\end{array}}{\phi \triangleright \mathsf{ref}^U_{\varepsilon_2}\ \varepsilon_1 \check{t} \in \mathbb{T}[\mathsf{Ref}_\perp\ U]} \qquad \text{(Ideref)}\ \frac{\begin{array}{c}\phi \triangleright \check{t} \in \mathbb{T}[U'] \\ \varepsilon \vdash U' \lesssim \mathsf{Ref}_g\ U\end{array}}{\phi \triangleright !^{\mathsf{Ref}_g\ U} \varepsilon \check{t} \in \mathbb{T}[U \mathbin{\widetilde{\curlyvee}} g]}$$

$$\text{(Iassgn)}\ \frac{\begin{array}{c}\phi \triangleright \check{t}_1 \in \mathbb{T}[\mathsf{Ref}_{g'}\ U_1'] \quad \varepsilon_1 \vdash \mathsf{Ref}_{g'}\ U_1' \lesssim \mathsf{Ref}_g\ U_1 \\ \phi \triangleright \check{t}_2 \in \mathbb{T}[U_2] \quad \varepsilon_2 \vdash U_2 \lesssim U_1 \quad \varepsilon_3 \vdash \widetilde{\phi.g_c \curlyvee g \preccurlyeq \mathit{label}(U_1)}\end{array}}{\phi \triangleright \varepsilon_1 \check{t}_1 \mathrel{\overset{g,U_1}{:=}_{\varepsilon_3}} \varepsilon_2 \check{t}_2 \in \mathbb{T}[\mathsf{Unit}_\perp]} \qquad \text{(I::)}\ \frac{\begin{array}{c}\phi \triangleright \check{t} \in \mathbb{T}[U_1] \\ \varepsilon_1 \vdash U_1 \lesssim U_2\end{array}}{\phi \triangleright \varepsilon_1 \check{t} :: U_2 \in \mathbb{T}[U_2]}$$

Fig. 24. $\mathrm{GSL}_{\mathsf{Ref}}$: Gradual Intrinsic Terms

stored value is a subtype of U. As the value is stamped with actual labels, the term may also reduce to an error if consistent transitivity cannot support the judgment $\widetilde{\phi.g_c \curlyvee \ell \preccurlyeq U}$.

6.3 Relating Intrinsic and Evidence-augmented Terms

In this section we present the translation rules from $\mathrm{GSL}_{\mathsf{Ref}}$ terms to intrinsic terms in Figure 27. Also this section presents the erasure function in in Figure 28—highlighting the syntactics differences between terms in gray—along properties that relates evidence-augmented terms and intrinsic terms.

In particular we identify four important properties. First, that given a source language the erasure of the translation to intrinsic term is equal to the translation of the source term to an evidence-augmented term:

Proposition 6.1. *If* $\Gamma; \Sigma; g_c \vdash t \rightsquigarrow \check{t} : U$ *and* $\Gamma; \Sigma; g_c \vdash t \rightsquigarrow t' : U$, *then* $|\check{t}| = t'$.

Proof. By induction on the type derivation of t. □

Second, given a reducible intrinsic term $\check{t}$, if it reduces to an error, then it erasure also reduces to an error; or, if reduces to an intrinsic term $\check{t}'$, then the erasure of $\check{t}'$ also reduces to the erasure of $\check{t}'$:

Proposition 6.2. *Consider* $\phi = \varepsilon g_c$, $\phi \triangleright \check{t} \in \mathbb{T}[U]$, *and* $\cdot; \Sigma; \varepsilon g_c \vdash t : U$, *such that* $\Sigma \models \mu_2$. *Then if* $\check{t} = t$ *and* $\mu_1 = \mu_1'$ *then either*

$$\boxed{\overset{\phi}{\longmapsto}: \mathbb{C}[U] \times (\mathbb{C}[U] \cup \{\, \mathbf{error} \,\})}$$

$$(\mathrm{R}\longrightarrow)\frac{t^U \mid \mu \;\overset{\phi}{\longrightarrow}\; r \qquad r \in \mathbb{C}[U] \cup \{\, \mathbf{error} \,\}}{t^U \mid \mu \;\overset{\phi}{\longmapsto}\; r} \qquad (\mathrm{R}f)\frac{\check{t}_1 \mid \mu \;\overset{\phi}{\longmapsto}\; \check{t}_2 \mid \mu'}{f[\check{t}_1] \mid \mu \;\overset{\phi}{\longmapsto}\; f[\check{t}_2] \mid \mu'}$$

$$(\mathrm{Rprot})\frac{\check{t}_1 \mid \mu \;\overset{\phi'}{\longmapsto}\; \check{t}_2 \mid \mu'}{\mathsf{prot}_{\varepsilon\ell}\phi'\ (\varepsilon\check{t}_1) \mid \mu \;\overset{\phi}{\longmapsto}\; \mathsf{prot}_{\varepsilon\ell}\phi'\ (\varepsilon\check{t}_2) \mid \mu'} \qquad (\mathrm{R}h)\frac{et \longrightarrow_c et'}{h[et] \mid \mu \;\overset{\phi}{\longmapsto}\; h[et'] \mid \mu'}$$

$$(\mathrm{Rproth})\frac{et \longrightarrow_c et'}{\mathsf{prot}_{\varepsilon\ell}\phi'\ (et) \mid \mu \;\overset{\phi}{\longmapsto}\; \mathsf{prot}_{\varepsilon\ell}\phi'\ (et') \mid \mu'} \qquad (\mathrm{R}f\mathrm{err})\frac{\check{t} \mid \mu \;\overset{\phi}{\longmapsto}\; \mathbf{error}}{f[\check{t}] \mid \mu \;\overset{\phi}{\longmapsto}\; \mathbf{error}}$$

$$(\mathrm{R}h\mathrm{err})\frac{et \longrightarrow_c \mathbf{error}}{h[et] \mid \mu \;\overset{\phi}{\longmapsto}\; \mathbf{error}} \qquad (\mathrm{Rproterr})\frac{\check{t} \mid \mu \;\overset{\phi'}{\longmapsto}\; \mathbf{error}}{\mathsf{prot}_{\varepsilon\ell}\phi'\ (\varepsilon\check{t}) \mid \mu \;\overset{\phi}{\longmapsto}\; \mathbf{error}}$$

$$(\mathrm{Rprotherr})\frac{et \longrightarrow_c \mathbf{error}}{\mathsf{prot}_{\varepsilon\ell}\phi'\ (et) \mid \mu \;\overset{\phi}{\longmapsto}\; \mathbf{error}}$$

Fig. 25. $\mathrm{GSL}_{\mathrm{Ref}}$: Intrinsic Reduction

- $\check{t} \mid \mu_1 \;\overset{\phi}{\longmapsto}\; \check{t}' \mid \mu_2 \Rightarrow |\check{t}| \mid |\mu_2| \;\overset{\varepsilon g_c}{\longmapsto}\; |\check{t}'| \mid |\mu_2'|$, *or*
- $\check{t} \mid \mu_1 \;\overset{\phi}{\longmapsto}\; \mathbf{error} \Rightarrow |\check{t}| \mid |\mu_2| \mathbf{error}$

Proof. By induction on the type derivation of $\check{t}$.

Case (I::). Then $\check{t} = \varepsilon_1\check{t}' :: U$ and by (E::), $t = \varepsilon_1 t'$ for some t' such that $\check{t}' = t'$. Suppose that $\varepsilon_1 \vdash U' \lesssim U$. By inspection on the type derivations, $\phi \triangleright \check{t}' \in \mathbb{T}[U']$ and $\cdot; \Sigma; \varepsilon g_c \vdash t' : U'$.

Let us suppose that $\check{t}' \mid \mu_1 \;\overset{\phi}{\longmapsto}\; \check{t}'' \mid \mu_2$, then by induction hypothesis $t' \mid \mu_2 \;\overset{\varepsilon g_c}{\longmapsto}\; t'' \mid \mu_2'$ and $\check{t}'' = t''$ and $\mu_1' = \mu_2'$. Then $\varepsilon_1\check{t}' :: U \mid \mu_1 \;\overset{\phi}{\longmapsto}\; \varepsilon_1\check{t}'' :: U \mid \mu_1'$ and $\varepsilon_1 t' \mid \mu_2 \;\overset{\varepsilon g_c}{\longmapsto}\; \varepsilon_1 t'' \mid \mu_2'$. But as $\mu_1' = \mu_2'$, and by (E::) $\varepsilon_1\check{t}'' :: U = \varepsilon_1 t''$, the result holds.

Let us suppose now that $\check{t}' = \varepsilon_2 u :: U'$. Then as $\check{t}' = t'$, $t' = \varepsilon_2 u'$, for some u' such that $u = u'$. If $\varepsilon_2 \circ^{<:} \varepsilon_1$ is not defined the result holds immediately. Suppose $\varepsilon_2 \circ^{<:} \varepsilon_1 = \varepsilon'$, then $\varepsilon_1(\varepsilon_2 u :: U') :: U \mid \mu_1 \;\overset{\phi}{\longmapsto}\; \varepsilon' u :: U \mid \mu_1$ and $\varepsilon_1(\varepsilon_2 u') \mid \mu_2 \;\overset{\varepsilon g_c}{\longmapsto}\; \varepsilon' u' \mid \mu_2$. But as $\mu_1 = \mu_2$, and by (E::) $\varepsilon' u :: U = \varepsilon' u'$, the result holds.

If $\check{t}' = u$, then as $\check{t}' = t'$, $t' = \varepsilon_2 u'$, for some u' such that $u = u'$, and the result holds immediately.

The other cases proceed analogous. □

Fourth, if an intrinsic term type checks, then its erasure also type checks to the same type.

Proposition 6.3. *Consider $\phi \triangleright \check{t} \in \mathbb{T}[U]$ then, for $\Gamma \models \check{t}$ and $\Sigma \models \check{t}$, $\Gamma; \Sigma; |\phi| \vdash |\check{t}| : U$.*

Proof. By induction on the type derivation of $\check{t}$. □

Notions of Reduction

$$\boxed{\xrightarrow{\phi} : \mathbb{C}[U] \times (\mathbb{C}[U] \cup \{\, \mathbf{error} \,\})}$$

$$\varepsilon_1(b_1)_{g_1} \oplus^g \varepsilon_2(b_2)_{g_2} \mid \mu \xrightarrow{\phi} (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2)(b_1 \llbracket\oplus\rrbracket\, b_2)_{(g_1 \widetilde{\curlyvee} g_2)} :: \mathsf{Bool}_g \mid \mu$$

$$\mathsf{prot}^{g,U}_{\varepsilon_2 g'} \phi'(\varepsilon_1 u) \mid \mu \xrightarrow{\phi} (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2)(u \mathbin{\widetilde{\curlyvee}} g') :: U \mathbin{\widetilde{\curlyvee}} g \mid \mu$$

$$\varepsilon_1(\lambda^{g_2'} x^{U_{11}}.t^*)_{g_2} \mathbin{@}_{\varepsilon_3}^{U_1 \xrightarrow{g_1'}_{g_1} U_2} \varepsilon_2 u \mid \mu \xrightarrow{\phi} \begin{cases} \mathsf{prot}^{g_1,U_2}_{ilbl(\varepsilon_1) g_2} \phi'(icod(\varepsilon_1)([(\varepsilon u :: U_{11})/x^{U_{11}}]t^*)) \mid \mu \\ \mathbf{error} \quad \text{if } \varepsilon \text{ or } \varepsilon' \text{ are not defined} \end{cases}$$
$$\text{where} \quad \varepsilon = \varepsilon_2 \circ^{<:} idom(\varepsilon_1), \quad \varepsilon' = (\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_3 \circ^{\preccurlyeq} ilat(\varepsilon_1)$$
$$\text{and } \phi' = \langle \varepsilon',\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g_2,\ g_2' \rangle$$

$$\mathsf{if}^g\ \varepsilon_1\mathsf{true}_{g_1} \text{ then } \varepsilon_2 t^{U_2} \text{ else } \varepsilon_3 t^{U_3} \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g,U}_{ilbl(\varepsilon_1) g_1} \phi'(\varepsilon_2 t^{U_2}) \mid \mu$$
$$\text{where } \phi' = \langle \phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1),\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g_1,\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g \rangle \text{ and } U = (U_2 \mathbin{\widetilde{\curlyvee}} U_3)$$

$$\mathsf{if}^g\ \varepsilon_1\mathsf{false}_{g_1} \text{ then } \varepsilon_2 t^{U_2} \text{ else } \varepsilon_3 t^{U_3} \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g,U}_{ilbl(\varepsilon_1) g_1} \phi'(\varepsilon_3 t^{U_3}) \mid \mu$$
$$\text{where } \phi' = \langle \phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1),\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g_1,\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g \rangle \text{ and } U = (U_2 \mathbin{\widetilde{\curlyvee}} U_3)$$

$$\mathsf{ref}^U_{\varepsilon_\ell}\ \varepsilon u \mid \mu \xrightarrow{\phi} \begin{cases} o^U_\perp \mid \mu[o^U \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} \phi.g_c) :: U] \text{ where } o^U \notin dom(\mu) \\ \mathbf{error} \quad \text{if } (\phi.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell) \text{ is not defined} \end{cases}$$
$$\text{where } \varepsilon' = \varepsilon \mathbin{\widetilde{\curlyvee}} (\phi.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell)$$

$$!^{\mathsf{Ref}_g\ U}\ \varepsilon o^{U'}_{g'} \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g,U}_{ilbl(\varepsilon) g'} \phi'(iref(\varepsilon) v)$$
$$\text{where } \mu(o^{U'}) = v \text{ and } \phi' = \langle \phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon),\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g',\ \phi.g_c \mathbin{\widetilde{\curlyvee}} g \rangle$$

$$\varepsilon_1 o^U_g \overset{g',U_1}{:=}_{\varepsilon_3} \varepsilon_2 u \mid \mu \xrightarrow{\phi} \begin{cases} \mathsf{unit}_\perp \mid \mu[o^U \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} (\phi.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U] \\ \mathbf{error} \quad \text{if } \varepsilon' \text{ is not defined, or} \\ \qquad\qquad \phi.\varepsilon \lfloor\leq\rfloor ilbl(\varepsilon) \text{ does not hold} \end{cases}$$
$$\text{where } \varepsilon' = (\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \mathbin{\widetilde{\curlyvee}} ((\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_3 \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1)))$$
$$\text{and } \mu(o^U) = \varepsilon u' :: U$$

$$\boxed{\longrightarrow_c : \textsc{EvTerm} \times (\textsc{EvTerm} \cup \{\, \mathbf{error} \,\})}$$

$$\varepsilon_1(\varepsilon_2 v :: U) \longrightarrow_c \begin{cases} (\varepsilon_2 \circ^{<:} \varepsilon_1) v \\ \mathbf{error} \quad \text{if not defined} \end{cases}$$

$$\langle [\ell_1, \ell_2], [\ell_3, \ell_4] \rangle \lfloor\leq\rfloor \langle [\ell_1', \ell_2'], [\ell_3', \ell_3'] \rangle \iff \ell_3 \preccurlyeq \ell_4'$$

Fig. 26. $\mathsf{GSL}_{\mathsf{Ref}}$: Intrinsic Notions of Reduction

Finally, if an evidence-augmented term type checks, then there must exists some intrinsic term that have the same type and that it erasure is the original evidence-augmented term.

Proposition 6.4. *Consider $\Gamma; \Sigma; \varepsilon g_c \vdash t : U$. Then $\exists \check{t}, \exists \phi$ such that $|\check{t}| = t$ and $|\phi| = \varepsilon g_c$ and $\phi \triangleright \check{t} \in \mathbb{T}[U]$*

Proof. By induction on the type derivation of t.

$$\boxed{\Gamma;\Sigma;g_c \vdash t \leadsto \check{t} : U}$$

$$(T\mathrm{x})\;\frac{\Gamma(x) = U}{\Gamma;\Sigma;g_c \vdash x \leadsto x^U : U} \qquad (T\mathrm{b})\;\frac{}{\Gamma;\Sigma;g_c \vdash b_g \leadsto b_g : \mathsf{Bool}_g}$$

$$(T\mathrm{u})\;\frac{}{\Gamma;\Sigma;g_c \vdash \mathsf{unit}_g \leadsto \mathsf{unit}_g : \mathsf{Unit}_g} \qquad (T\lambda)\;\frac{\Gamma;\Sigma;g' \vdash t \leadsto \check{t} : U_2}{\Gamma;\Sigma;g_c \vdash (\lambda^{g'} x : U_1.t)_g \leadsto (\lambda^{g'} x^{U_1}.\check{t})_g : U_1 \xrightarrow{g'}_g U_2}$$

$$(T\oplus)\;\frac{\begin{array}{cc}\Gamma;\Sigma;g_c \vdash t_1 \leadsto \check{t}_1 : \mathsf{Bool}_{g_1} & \Gamma;\Sigma;g_c \vdash t_2 \leadsto \check{t}_2 : \mathsf{Bool}_{g_2} \\ \varepsilon_1 = \mathcal{I}_{<:}(\mathsf{Bool}_{g_1}, \mathsf{Bool}_{g_1}) & \varepsilon_2 = \mathcal{I}_{<:}(\mathsf{Bool}_{g_2}, \mathsf{Bool}_{g_2})\end{array}}{\Gamma;\Sigma;g_c \vdash t_1 \oplus t_2 \leadsto \varepsilon_1\check{t}_1 \oplus^{g_1 \widetilde{\curlyvee} g_2} \varepsilon_2\check{t}_2 : \mathsf{Bool}_{g_1 \widetilde{\curlyvee} g_2}}$$

$$(T\mathrm{app})\;\frac{\begin{array}{ccc}\Gamma;\Sigma;g_c \vdash t_1 \leadsto \check{t}_1 : U_{11} \xrightarrow{g'}_g U_{12} & & \Gamma;\Sigma;g_c \vdash t_2 \leadsto \check{t}_2 : U_2 \\ \varepsilon_1 = \mathcal{I}^{\cup}_{<:}(U_{11} \xrightarrow{g'}_g U_{12}) & \varepsilon_2 = \mathcal{I}_{<:}(U_2, U_{11}) & \varepsilon_3 = \mathcal{I}_{\preccurlyeq}(g_c, g, g')\end{array}}{\Gamma;\Sigma;g_c \vdash t_1\; t_2 \leadsto \varepsilon_1\check{t}_1 \mathbin{@}^{U_{11} \xrightarrow{g'}_g U_{12}}_{\varepsilon_3} \varepsilon_2\check{t}_2 : U_{12} \,\widetilde{\curlyvee}\, g}$$

$$(T\mathrm{if})\;\frac{\begin{array}{cccc}\Gamma;\Sigma;g_c \vdash t_1 \leadsto \check{t}_1 : U_1 & g_c' = g_c \,\widetilde{\curlyvee}\, g & \Gamma;\Sigma;g_c' \vdash t_2 \leadsto \check{t}_2 : U_2 & \Gamma;\Sigma;g_c' \vdash t_3 \leadsto \check{t}_3 : U_3 \\ \varepsilon_1 = \mathcal{I}_{<:}(U_1, \mathsf{Bool}_g) & & \varepsilon_2 = \mathcal{I}_{<:}(U_2, U_2, U_3) & \varepsilon_3 = \mathcal{I}_{<:}(U_3, U_2, U_3)\end{array}}{\Gamma;\Sigma;g_c \vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 \leadsto \mathsf{if}^g\ \varepsilon_1\check{t}_1\ \mathsf{then}\ \varepsilon_2\check{t}_2\ \mathsf{else}\ \varepsilon_3\check{t}_3 : (U_2 \,\widetilde{\curlyvee}\, U_3) \,\widetilde{\curlyvee}\, g}$$

$$(T\mathrm{assgn})\;\frac{\begin{array}{ccc}\Gamma;\Sigma;g_c \vdash t_1 \leadsto \check{t}_1 : \mathsf{Ref}_g\ U_1 & & \Gamma;\Sigma;g_c \vdash t_2 \leadsto \check{t}_2 : U_2 \\ \varepsilon_1 = \mathcal{I}^{\cup}_{<:}(\mathsf{Ref}_g\ U_1) & \varepsilon_2 = \mathcal{I}_{<:}(U_2, U_1) & \varepsilon_3 = \mathcal{I}_{\preccurlyeq}(g_c, g, \mathit{label}(U_1))\end{array}}{\Gamma;\Sigma;g_c \vdash t_1{:=}t_2 \leadsto \varepsilon_1\check{t}_1 \overset{g,U_1}{:=}_{\varepsilon_3} \varepsilon_2\check{t}_2 : \mathsf{Unit}_\bot}$$

$$(T\mathrm{ref})\;\frac{\begin{array}{cc}\multicolumn{2}{c}{\Gamma;\Sigma;g_c \vdash t \leadsto \check{t} : U'} \\ \varepsilon_1 = \mathcal{I}_{<:}(U', U) & \varepsilon_2 \vdash \mathcal{I}_{\preccurlyeq}(g_c, \mathit{label}(U))\end{array}}{\Gamma;\Sigma;g_c \vdash \mathsf{ref}^U\ t \leadsto \mathsf{ref}^U_{\varepsilon_2}\ \varepsilon_1\check{t} : \mathsf{Ref}_\bot\ U} \qquad (T\mathrm{deref})\;\frac{\begin{array}{c}\Gamma;\Sigma;g_c \vdash t \leadsto \check{t} : \mathsf{Ref}_g\ U \\ \varepsilon = \mathcal{I}^{\cup}_{<:}(\mathsf{Ref}_g\ U)\end{array}}{\Gamma;\Sigma;g_c \vdash\ !t \leadsto\ !^{\mathsf{Ref}_g\ U}\varepsilon\check{t} : U \,\widetilde{\curlyvee}\, g}$$

$$(T{::})\;\frac{\Gamma;\Sigma;g_c \vdash t \leadsto \check{t} : U_1 \qquad \varepsilon = \mathcal{I}_{<:}(U_1, U_2)}{\Gamma;\Sigma;g_c \vdash t :: U_2 \leadsto \varepsilon\check{t} :: U_2 : U_2}$$

$$\text{where } \mathcal{I}^{\cup}_{\preccurlyeq}(g) = \mathcal{I}_{\preccurlyeq}(g, g) \text{ and } \mathcal{I}^{\cup}_{<:}(U) = \mathcal{I}_{<:}(U, U)$$

Fig. 27. $\mathrm{GSL}_{\mathsf{Ref}}$: translation to $\mathrm{GSL}_{\mathsf{Ref}}$ intrinsic terms

Case $(\varepsilon' t')$. Then $t = \varepsilon' t'$, for some ε', t'. But we know that $\Gamma;\Sigma;\varepsilon g_c \vdash \varepsilon' t' : U$ and suppose $\varepsilon' \vdash U' \lesssim U$ and $\varepsilon \vdash g_c \lesssim g_c'$. Then by choosing $\phi = \langle \varepsilon, g_c \rangle g_c'$ and induction hypothesis on t', $\exists \check{t}'$ such that $\phi \triangleright \check{t}' \in \mathbb{T}[U']$.

The other cases proceed analogous. ◻

Lemma 6.5. *Consider* $\phi \triangleright \check{t}_1 \in \mathbb{T}[U]$. *If* $\check{t}_1 \sqsubseteq \check{t}_2$ *then* $|\check{t}_1| \sqsubseteq |\check{t}_2|$.

Proof. By induction on the type derivation of $\check{t}_1$ and the definition of $||$. ◻

Lemma 6.6. *Consider* $\phi \triangleright \check{t}_1 \in \mathbb{T}[U]$. *If* $|\check{t}_1| \sqsubseteq t_2$, *then* $\exists \check{t}_2$, *such that* $\check{t}_1 \sqsubseteq \check{t}_2$ *and that* $|\check{t}_2| = t_2$.

$$\text{(Ex)}\;\frac{}{|x^U| = x} \qquad \text{(Eb)}\;\frac{}{|b_g| = b_g} \qquad \text{(Eu)}\;\frac{}{|\mathsf{unit}_g| = \mathsf{unit}_g} \qquad \text{(E}o\text{)}\;\frac{}{|o^U_g| = o_g}$$

$$\text{(E}\lambda\text{)}\;\frac{|\check{t}| = t}{|(\lambda^{g'} x^{U_1}.\check{t})_g| = (\lambda^{g'} x : U_1.\check{t})_g} \qquad \text{(Eprot)}\;\frac{|\phi'| = \varepsilon_2 g_2 \qquad |\check{t}| = t}{|\mathsf{prot}^{g_1', U}_{\varepsilon_1 g_1}\ \phi'(\varepsilon_3 \check{t})| = \mathsf{prot}_{\varepsilon_1 g_1} \varepsilon_2 g_2(\varepsilon_3 t)}$$

$$\text{(E}\oplus\text{)}\;\frac{|\check{t}_1| = t_1 \qquad |\check{t}_2| = t_2}{|\varepsilon_1 \check{t}_1 \oplus^{g_1 \mathbin{\widetilde{\curlyvee}} g_2}\ \varepsilon_2 \check{t}_2| = \varepsilon_1 t_1 \oplus \varepsilon_2 t_2} \qquad \text{(Eapp)}\;\frac{|\check{t}_i| = t_i}{|\varepsilon_1 \check{t}_1\ @^{U_{11} \xrightarrow{g'}_g U_{12}}_{\varepsilon_3}\ \varepsilon_2 \check{t}_2| = \varepsilon_1 t_1\ @_{\varepsilon_3}\ \varepsilon_2 t_2}$$

$$\text{(Eif)}\;\frac{|\check{t}_i| = t_i}{|\mathsf{if}^{g}\ \varepsilon_1 \check{t}_1 \mathsf{\ then\ } \varepsilon_2 \check{t}_2 \mathsf{\ else\ } \varepsilon_3 \check{t}_3| = \mathsf{if}\ \varepsilon_1 t_1 \mathsf{\ then\ } \varepsilon_2 t_2 \mathsf{\ else\ } \varepsilon_3 t_3} \qquad \text{(Eref)}\;\frac{|\check{t}| = t}{|\mathsf{ref}^U_{\varepsilon_2}\ \varepsilon_1 \check{t}| = \mathsf{ref}^U_{\varepsilon_2}\ \varepsilon_1 t}$$

$$\text{(Ederef)}\;\frac{|\check{t}| = t}{|!^{\mathsf{Ref}_g\ U}\ \varepsilon \check{t}| = !\varepsilon t} \qquad \text{(Eassgn)}\;\frac{|\check{t}_i| = t}{|\varepsilon_1 \check{t}_1 \overset{g,\ U_1}{:=}_{\varepsilon_3}\ \varepsilon_2 \check{t}_2| = \varepsilon_1 t_1 :=_{\varepsilon_3} \varepsilon_2 t_2} \qquad \text{(E::)}\;\frac{|\check{t}| = t}{|\varepsilon \check{t} :: U_2| = \varepsilon t|}$$

$$\frac{}{|\bullet| = \bullet} \qquad \frac{|\mu_1| = \mu_2 \qquad |x^U| = x \qquad |v| = v'}{|\mu_1, x^U \mapsto v| = \mu_2, x \mapsto v'} \qquad |\langle \varepsilon, g, g' \rangle| = \varepsilon g$$

Fig. 28. $\mathsf{GSL}^{\varepsilon}_{\mathsf{Ref}}$: Equivalence between intrinsic terms and evidence-augmented terms

Proof. By induction on $\check{t}_1$ and the definition of $||$.

Case (I::). Then $\check{t}_1 = \varepsilon_1 \check{t}_1' :: U$, and $|\check{t}_1| = \varepsilon_1 |\check{t}_1'|$. By definition of $\sqsubseteq$, t_2 has the form $\varepsilon_2 t_2'$, where $\varepsilon_2 \sqsubseteq \varepsilon_2$ and $|\check{t}_1'| \sqsubseteq t_2'$. By induction hypothesis, $\exists \check{t}_2'$ such that $\check{t}_1' \sqsubseteq \check{t}_2'$ and that $|\check{t}_2'| = t_2'$. By definition of evidence, we can build the term $\varepsilon_2 \check{t}_2' :: ?$, but we know that $\varepsilon_1 \check{t}_1' :: U \sqsubseteq \varepsilon_2 \check{t}_2' :: ?$ and that $|\varepsilon_2 \check{t}_2' :: ?| = \varepsilon_2 |\check{t}_2'| = \varepsilon_2 t_2$ and the result holds.

The other cases proceed analogous. □

6.4 Type Safety

In this section we present the proof of type safety for $\mathsf{GSL}_{\mathsf{Ref}}$.

We define what it means for a store to be well typed with respect to a term. Informally, all free locations of a term and of the contents of the store must be defined in the domain of that store. Also, the store must preserve types between intrinsic locations and underlying values.

Definition 6.7 (μ is well typed). A store μ is said to be *well typed* with respect to an intrinsic term t^U, written $t^U \vdash \mu$, if

(1) $\mathit{freeLocs}(t^U) \subseteq \mathit{dom}(\mu)$, and
(2) $\forall\, v \in \mathit{cod}(\mu), v \vdash \mu$ and
(3) $\forall\, o^U \in \mathit{dom}(\mu), \forall \phi$, then $\phi \triangleright \mu(o^U) \in \mathbb{T}[U]$.

Lemma 6.8. *Suppose $\phi \triangleright t^U \in \mathbb{T}[U]$, then $\forall g_r', \forall \varepsilon_r'$, such that $g_r' \preccurlyeq \phi.g_c$ and $\varepsilon_r' \vdash g_r' \preccurlyeq \phi.g_c$, $\phi' = \langle \varepsilon_r' g_r', \phi.g_c \rangle$ then $\phi' \triangleright t^U \in \mathbb{T}[U]$.*

Proof. By induction on the derivation of $\phi \triangleright t^U \in \mathbb{T}[U]$. Noticing that no typing derivation depends on $\varepsilon_r' g_r'$, save for the judgement $\varepsilon_r' \vdash g_r' \preccurlyeq g_c$ which is premise of this lemma. □

Lemma 6.9. *Suppose $\phi \triangleright v \in \mathbb{T}[U]$, then $\forall \phi'$, then $\phi' \triangleright v \in \mathbb{T}[U]$.*

Proof. By induction on the derivation of $\phi' \triangleright v$ observing that for values, there is no premise that depends on the security effect. $\square$

Lemma 6.10 (Canonical forms). *Consider a value $v \in \mathbb{T}[U]$. Then either $v = u$, or $v = \varepsilon u :: U$ with $u \in \mathbb{T}[U']$ and $\varepsilon \vdash U' \lesssim U$. Furthermore:*

(1) If $U = \mathsf{Bool}_g$ then either $v = b_g$ or $v = \varepsilon b_{g'} :: \mathsf{Bool}_g$ with $b_{g'} \in \mathbb{T}[\mathsf{Bool}_{g'}]$ and $\varepsilon \vdash \mathsf{Bool}_{g'} \lesssim \mathsf{Bool}_g$.

(2) If $U = U_1 \xrightarrow{g_c}_g U_2$ then either $v = (\lambda^{g_c} x^{U_1}.t^{U_2})_g$ with $t^{U_2} \in \mathbb{T}[U_2]$ or $v = \varepsilon(\lambda^{g'_c} x^{U'_1}.t^{U'_2})_{g'} :: U_1 \xrightarrow{g_c}_g U_2$ with $t^{U'_2} \in \mathbb{T}[U'_2]$ and $\varepsilon \vdash U'_1 \xrightarrow{g'_c}_{g'} U'_2 \lesssim U_1 \xrightarrow{g_c}_g U_2$.

(3) If $U = \mathsf{Ref}_g\ U_1$ then either $v = o_g^{U_1}$ or $v = \varepsilon o_{g'}^{U'_1} :: \mathsf{Ref}_g\ U_1$ with $o_{g'}^{U'_1} \in \mathsf{Ref}_{g'}\ U'_1$ and $\varepsilon \vdash \mathsf{Ref}_{g'}\ U'_1 \lesssim \mathsf{Ref}_g\ U_1$.

Proof. By direct inspection of the formation rules of gradual intrinsic terms (Figure 24). $\square$

Lemma 6.11 (Substitution). *If $\phi \triangleright t^U \in \mathbb{T}[U]$ and $\phi \triangleright v \in \mathbb{T}[U_1]$, then $\phi \triangleright [v/x^{U_1}]t^U \in \mathbb{T}[U]$.*

Proof. By induction on the derivation of $\phi \triangleright t^U$. $\square$

Proposition 6.12 ($\longrightarrow$ is well defined). *If $t^U \mid \mu \longrightarrow r$ and $t^U \vdash \mu$, then $r \in$ Config$_U \cup \{$ **error** $\}$ and if $r = t'^U \mid \mu' \in$ Config$_U$ then also $t'^U \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$.*

Proof. By induction on the structure of a derivation of $t^{\widetilde{T}} \mid \mu \longrightarrow r$, considering the last rule used in the derivation.

Case (I$\oplus$). Then $t^U = b_{1\ell_1} \oplus^g b_{2\ell_2}$. By construction we can suppose that $g = g'_1 \mathbin{\widetilde{\curlyvee}} g'_2$, then

$$\text{(I}\oplus\text{)}\ \frac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi \triangleright b_{1\ell_1} \in \mathsf{Bool}_{g_1} \qquad \varepsilon_1 \vdash \mathsf{Bool}_{g_1} \lesssim \mathsf{Bool}_{g'_1} \\ \phi \triangleright b_{2\ell_2} \in \mathsf{Bool}_{g_2} \qquad \varepsilon_2 \vdash \mathsf{Bool}_{g_2} \lesssim \mathsf{Bool}_{g'_2}\end{array}}{\phi \triangleright \varepsilon_1 b_{1\ell_1} \oplus^g \varepsilon_2 b_{2\ell_2} \in \mathbb{T}[\mathsf{Bool}_g]}$$

Therefore

$$\begin{array}{ll} & \varepsilon_1(b_1)_{g_1} \oplus^g \varepsilon_2(b_2)_{g_2} \mid \mu \\ \xrightarrow{\phi} & (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2)(b_1 \llbracket\oplus\rrbracket b_2)_{(g_1 \widetilde{\curlyvee} g_2)} :: \mathsf{Bool}_g \mid \mu \end{array}$$

Then

$$\text{(I}\oplus\text{)}\ \frac{\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c}{\phi \triangleright (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2)(b_1 \llbracket\oplus\rrbracket b_2)_{(g_1 \widetilde{\curlyvee} g_2)} :: \mathsf{Bool}_g \in \mathbb{T}[\mathsf{Bool}_g]}$$

and the result holds.

Case (Iprot). Then $t^U = \phi \triangleright \mathsf{prot}^{g,U}_{\varepsilon g'}\ \phi'(\varepsilon u)$ and

$$\text{(Iprot)}\ \frac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \qquad \varepsilon'_r \vdash \widetilde{g_r \curlyvee g'} \preccurlyeq g'_c \\ \phi' \triangleright u \in \mathbb{T}[U'] \\ \varepsilon \vdash U' \lesssim U \qquad \varepsilon_\ell \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \mathsf{prot}^{g,U}_{\varepsilon g'}\ \phi'(\varepsilon u) \in \mathbb{T}[U \mathbin{\widetilde{\curlyvee}} g]}$$

Therefore

$$\mathsf{prot}^{g,U}_{\varepsilon g'}\ \phi'(\varepsilon u) \mid \mu \xrightarrow{\phi} (\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_\ell)(u \mathbin{\widetilde{\curlyvee}} g') :: U \mathbin{\widetilde{\curlyvee}} g \mid \mu$$

But by Lemma 6.9, $\phi \triangleright u \in \mathbb{T}[U']$. Therefore by definition of join $\phi \triangleright (u \mathbin{\widetilde{\curlyvee}} g') \in \mathbb{T}[U' \mathbin{\widetilde{\curlyvee}} g']$. Then using Lemma 5.43

$$\text{I::}\dfrac{\begin{array}{c}\phi \triangleright (u \mathbin{\widetilde{\curlyvee}} g') \in \mathbb{T}[U' \mathbin{\widetilde{\curlyvee}} g'] \\ (\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_\ell) \vdash \widetilde{U' \curlyvee g'} \lesssim \widetilde{U \curlyvee g}\end{array}}{\phi \triangleright (\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_\ell)(u \mathbin{\widetilde{\curlyvee}} g') :: U \mathbin{\widetilde{\curlyvee}} g \in \mathbb{T}[U \curlyvee \ell]}$$

and the result holds.

Case (Iapp). Then $t^U = \varepsilon_1(\lambda^{g''_c} x^{U_{11}}.t^{U_{12}})_{g_1} \mathbin{@}_{\varepsilon_\ell}^{U_1 \xrightarrow{g'_c}_g U_2} \varepsilon_2 u$ and $U = U_2 \mathbin{\widetilde{\curlyvee}} g$. Then

$$\text{(Iapp)}\dfrac{\begin{array}{c}\dfrac{\dfrac{\mathcal{D}_1}{\phi \triangleright t^{U12} \in \mathbb{T}[U_{12}]} \quad \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c}{\phi \triangleright (\lambda^{g''_c} x^{U_{11}}.t^{U_{12}})_{g_1} \in \mathbb{T}[U_{11} \xrightarrow{g''_c}_{g_1} U_{12}]} \\ \dfrac{\mathcal{D}_2}{\phi \triangleright u \in \mathbb{T}[U'_2]} \qquad \varepsilon_2 \vdash U'_2 \lesssim U_1 \\ \varepsilon_1 \vdash U_{11} \xrightarrow{g''_c}_{g_1} U_{12} \lesssim U_1 \xrightarrow{g'_c}_g U_2 \\ \varepsilon_\ell \vdash \widetilde{g_c \curlyvee g} \preccurlyeq g'_c \qquad \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c\end{array}}{\phi \triangleright \varepsilon_1(\lambda^{g''_c} x^{U_{11}}.t^{U_{12}})_{g_1} \mathbin{@}_{\varepsilon_\ell}^{U_1 \xrightarrow{g'_c}_g U_2} \varepsilon_2 u \in \mathbb{T}[U_2 \mathbin{\widetilde{\curlyvee}} g]}$$

If $\varepsilon' = (\varepsilon_2 \circ^{<:} idom(\varepsilon_1))$ or $\varepsilon'_r = (\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell \circ^{\preccurlyeq} ilat(\varepsilon_1)$ are not defined, then $t^U \mid \mu \xrightarrow{\phi}$ **error**, and then the result hold immediately. Suppose that consistent transitivity does hold, then if $\phi' = \langle \phi'.\varepsilon(\phi.g_c \mathbin{\widetilde{\curlyvee}} g_1), g''_c \rangle$

$$\varepsilon_1(\lambda^{g''_c} x^{U_{11}}.t^{U_{12}})_{g_1} \mathbin{@}_{\varepsilon_\ell}^{U_1 \xrightarrow{g'_c}_g U_2} \varepsilon_2 u \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g, U_2}_{ilbl(\varepsilon_1)g_1} \phi'(icod(\varepsilon_1)([(\varepsilon' u :: U_{11})/x^{U_{11}}]t^{U_{12}})) \mid \mu$$

As $\varepsilon_2 \vdash U'_2 \lesssim U_1$ and by inversion lemma $idom(\varepsilon_1) \vdash U_1 \lesssim U_{11}$, then $\varepsilon' \vdash U'_2 \lesssim U_{11}$. Therefore $\phi \triangleright \varepsilon' u :: U_{11} \in \mathbb{T}[U_{11}]$, and by Lemma 6.11,
$\phi \triangleright [(\varepsilon' u :: U_{11})/x^{U_{11}}]t^{U_{12}} \in \mathbb{T}[U_{12}]$.

We know that $\varepsilon_\ell \vdash \widetilde{g_c \curlyvee g} \preccurlyeq g'_c$. By inversion on the label of types, $ilbl(\varepsilon_1) \vdash g_1 \mathrel{\widetilde{\preccurlyeq}} g$. Also by monotonicity of the join, $\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1) \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g_1 \mathrel{\widetilde{\preccurlyeq}} g_c \mathbin{\widetilde{\curlyvee}} g$. Then, by inversion on the latent effect of function types, $ilat(\varepsilon_1) \vdash g'_c \mathrel{\widetilde{\preccurlyeq}} g''_c$. Therefore combining evidences, as $\phi'.\varepsilon = (\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell \circ^{\preccurlyeq} ilat(\varepsilon_1)$, we may justify the runtime judgment $\phi'.\varepsilon \vdash \widetilde{\phi.g_c \curlyvee g_1} \preccurlyeq g''_c$.

Let us call $t'^{U_{12}} = [(\varepsilon' u :: U_{11})/x^{U_{11}}]t^{U_{12}}$. By Lemma 6.8, $\phi' \triangleright t'^{U_{12}} \in \mathbb{T}[U_{12}]$. Then

$$\text{(Iprot)}\dfrac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi' \triangleright t'^{U_{12}} \in \mathbb{T}[U_{12}] \\ icod(\varepsilon_1) \vdash U_{12} \lesssim U_2 \qquad ilbl(\varepsilon_1) \vdash g_1 \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \mathsf{prot}^{g, U_2}_{ilbl(\varepsilon_1)g_1} \phi'(icod(\varepsilon_1)(t'^{U_{12}})) \in \mathbb{T}[U_2 \mathbin{\widetilde{\curlyvee}} g]}$$

and the result holds.

Case (Iif-true). Then $t^U = \text{if}^g\ \varepsilon_1 b_{g_1}$ then $\varepsilon_2 t^{U_2}$ else $\varepsilon_3 t^{U_3}$, $U = (U_2 \mathbin{\widetilde{\curlyvee}} U_3) \mathbin{\widetilde{\curlyvee}} g$ and

$$\text{(If)}\ \frac{\begin{array}{cc} \phi \triangleright b_{g_1} \in \mathbb{T}[\mathsf{Bool}_{g_1}] & \varepsilon_1 \vdash \mathsf{Bool}_{g_1} \lesssim \mathsf{Bool}_g \\ \phi' = \langle \phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)(\phi.g_c \mathbin{\widetilde{\curlyvee}} g_1), \phi.g_c \mathbin{\widetilde{\curlyvee}} g\rangle & \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi' \triangleright t^{U_2} \in \mathbb{T}[U_2] & \varepsilon_2 \vdash U_2 \lesssim (U_2 \mathbin{\widetilde{\curlyvee}} U_3) \\ \phi' \triangleright t^{U_3} \in \mathbb{T}[U_3] & \varepsilon_3 \vdash U_3 \lesssim (U_2 \mathbin{\widetilde{\curlyvee}} U_3) \end{array}}{\phi \triangleright \text{if}^g\ \varepsilon_1 b_{g_1} \text{ then } \varepsilon_2 t^{U_2} \text{ else } \varepsilon_3 t^{U_3} \in \mathbb{T}[(U_2 \mathbin{\widetilde{\curlyvee}} U_3) \mathbin{\widetilde{\curlyvee}} g]}$$

Therefore

$$\text{if}\ ^g\varepsilon_1 b_{g_1} \text{ then } \varepsilon_2 t^{U_2} \text{ else } \varepsilon_3 t^{U_3} \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g,(U_2 \widetilde{\curlyvee} U_3)}_{ilbl(\varepsilon_1)g_1} \phi'(\varepsilon_2 t^{U_2}) \mid \mu$$

But

$$\text{(Iprot)}\ \frac{\begin{array}{c} \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi' \triangleright t^{U_2} \in \mathbb{T}[U_2] \\ \varepsilon_2 \vdash U_2 \lesssim U_2 \mathbin{\widetilde{\curlyvee}} U_3 \qquad ilbl(\varepsilon_1) \vdash g_1 \mathrel{\widetilde{\preccurlyeq}} g \end{array}}{\phi \triangleright \mathsf{prot}^{g,(U_2 \widetilde{\curlyvee} U_3)}_{ilbl(\varepsilon_1)g_1} \phi'(\varepsilon_2 t^{U_2}) \in \mathbb{T}[(U_2 \mathbin{\widetilde{\curlyvee}} U_3) \mathbin{\widetilde{\curlyvee}} g]}$$

and the result holds.

Case (Iif-false). Analogous to case (if-true).

Case (Iref). Then $t^U = \mathsf{ref}^{U'}_{\varepsilon_\ell}\ \varepsilon u$ and

$$\text{(Iref)}\ \frac{\begin{array}{cc} \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c & \phi \triangleright u \in \mathbb{T}[U''] \\ \varepsilon \vdash U'' \lesssim U' & \varepsilon_\ell \vdash g_c \mathrel{\widetilde{\preccurlyeq}} label(U') \end{array}}{\phi \triangleright \mathsf{ref}^{U'}_{\varepsilon_\ell}\ \varepsilon u \in \mathbb{T}[\mathsf{Ref}_\perp\ U']}$$

If $\varepsilon' = \varepsilon \mathbin{\widetilde{\curlyvee}} (\phi.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell)$ is not defined, then $t^{U'} \mid \mu \xrightarrow{\phi}$ **error**, and then the result hold immediately. Suppose that consistent transitivity does hold, then

$$\mathsf{ref}^{U'}_{\varepsilon_\ell}\ \varepsilon u \mid \mu \xrightarrow{\phi} o^{U'}_\perp \mid \mu[o^{U'} \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} \phi.g_c) :: U']$$

where $o^{U'} \notin dom(\mu)$.

We know that $\varepsilon_\ell \vdash g_c \mathrel{\widetilde{\preccurlyeq}} label(U')$, therefore $\phi.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} label(U')$. We also know that $\varepsilon \vdash U'' \lesssim U'$. Therefore combining both evidences we can justify that $\varepsilon \mathbin{\widetilde{\curlyvee}} (\phi.\varepsilon \circ^{<:} \varepsilon_\ell) \vdash \widetilde{U_2' \curlyvee \phi.g_c} <: U'$. But

$$\text{(Il)}\ \frac{\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c}{o^{U'}_\perp \in \mathbb{T}[\mathsf{Ref}_\perp\ U']}$$

Let us call $\mu' = \mu[o^{U'} \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} \phi.g_c) :: U']$. It is easy to see that $freeLocs(o^{U'}) = o^{U'}$ and $dom(\mu') = dom(\mu) \cup o^{U'}$, then $freeLocs(o^{U'}) \subseteq dom(\mu')$. Given that $t^{U'} \vdash \mu$ then $freeLocs(u) \subseteq dom(\mu)$, and therefore $\forall\, v \in cod(\mu') = cod(\mu) \cup (\varepsilon'(u \mathbin{\widetilde{\curlyvee}} \phi.g_c) :: U'), freeLocs(v') \subseteq dom(\mu')$. Finally as $t^{U'} \vdash \mu$ and $\mu'(o^{U'}) = \varepsilon'(u \mathbin{\widetilde{\curlyvee}} \phi.g_c) :: U' \in \mathbb{T}[U']$ then we can conclude that $l^{U'} \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$, and the result holds.

Case (Ideref). Then $t^U = !^{\mathsf{Ref}_g\ U'} \varepsilon o^{U''}_{g'}$, $U = U' \mathbin{\widetilde{\curlyvee}} g$ and

$$\text{(Ideref)}\ \frac{\begin{array}{c} \phi \triangleright o^{U''}_{g'} \in \mathbb{T}[\mathsf{Ref}_{g'}\ U''] \\ \varepsilon \vdash \mathsf{Ref}_{g'}\ U'' \lesssim \mathsf{Ref}_g\ U' \\ \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \end{array}}{\phi \triangleright !^{\mathsf{Ref}_g\ U'} \varepsilon o^{U''}_{g'} \in \mathbb{T}[U' \mathbin{\widetilde{\curlyvee}} g]}$$

Then for $\phi' = \langle(\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon))(\phi.g_c \mathbin{\widetilde{\curlyvee}} g'), \phi.g_c \mathbin{\widetilde{\curlyvee}} g\rangle$

$$!^{\mathsf{Ref}_g\ U'}\varepsilon o^{U''}_{g'} \mid \mu \xrightarrow{\phi} \mathsf{prot}^{g,U'}_{ilbl(\varepsilon)g'}\phi'(iref(\varepsilon)v) \mid \mu$$

where $\mu(o^{U''}) = v$. As the store is well typed, therefore $\phi \triangleright v \in \mathbb{T}[U'']$. By Lemma 6.9, $\phi' \triangleright v \in \mathbb{T}[U'']$. By inversion lemma on references, $ilbl(\varepsilon) \vdash g' \mathrel{\widetilde{\preccurlyeq}} g$ and $iref(\varepsilon) \vdash U'' \lesssim U'$

$$\text{(Iprot)}\frac{\begin{array}{cc}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c & \phi' \triangleright v \in \mathbb{T}[U''] \\ iref(\varepsilon) \vdash U'' \lesssim U' & ilbl(\varepsilon) \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \mathsf{prot}^{g,U'}_{ilbl(\varepsilon)g'}\phi'(iref(\varepsilon)v) \in \mathbb{T}[U' \mathbin{\widetilde{\curlyvee}} g]}$$

and the result holds.

Case (Iassgn). Then $t^U = \varepsilon_1 o^{U'_1}_{g'} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u$ and

$$\text{(Iassgn)}\frac{\begin{array}{cc}\varepsilon_1 \vdash \mathsf{Ref}_{g'}\ U'_1 \lesssim \mathsf{Ref}_g\ U_1 & \phi \triangleright o^{U'_1}_{g'} \in \mathbb{T}[\mathsf{Ref}_{g'}\ U'_1] \\ \varepsilon_2 \vdash U_2 \lesssim U_1 & \phi \triangleright u \in \mathbb{T}[U_2] \\ \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c & \varepsilon_\ell \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g \preccurlyeq label(U_1)\end{array}}{\phi \triangleright \varepsilon_1 o^{U'_1}_{g'} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u \in \mathbb{T}[\mathsf{Unit}_\perp]}$$

If $\varepsilon' = (\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \mathbin{\widetilde{\curlyvee}} ((\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1))$ is not defined, then $t^{U'} \mid \mu \xrightarrow{\phi}$ **error**, and then the result hold immediately. Suppose that consistent transitivity does hold, then

$$\varepsilon_1 o^{U'_1}_{g} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u \mid \mu \xrightarrow{\phi} \mathsf{unit}_\perp \mid \mu[o^U \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} (\phi.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U'_1]$$

We know that $\varepsilon_\ell \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g \preccurlyeq label(U_1)$. Then by inversion on reference evidence types and inversion in the label of types, $ilbl(iref(\varepsilon_1)) \vdash label(U_1) \mathrel{\widetilde{\preccurlyeq}} label(U'_1)$. But $ilbl(\varepsilon_1) \vdash g' \mathrel{\widetilde{\preccurlyeq}} g$, using monotonicity of the join, $\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1) \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g' \preccurlyeq \phi.g_c \mathbin{\widetilde{\curlyvee}} g$. Therefore $((\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell) \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1)) \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g' \preccurlyeq label(U'_1)$. We also know that if $u \in \mathbb{T}[U_2]$, then $(\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \vdash U_2 \lesssim U'_1$. Combining both evidences, $\varepsilon' = (\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \mathbin{\widetilde{\curlyvee}} (((\phi.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell) \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1)))$, and by Proposition 5.43 we can then justify that $\varepsilon' \vdash U_2 \mathbin{\widetilde{\curlyvee}} (\phi.g_c \mathbin{\widetilde{\curlyvee}} g) <: U'_1$ and therefore justify the ascription in the heap.

Let us call $\mu' = \mu[o^{U'_1} \mapsto \varepsilon'(u \mathbin{\widetilde{\curlyvee}} (\phi.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U'_1]$. As $freeLocs(\mathsf{unit}_\perp) = \emptyset$ then $freeLocs(\mathsf{unit}_\perp) \subseteq \mu'$.
As $t^U \vdash \mu$ then $freeLocs(u) \in dom(\mu)$, and as $dom(\mu) = dom(\mu')$ then it is trivial to see that $\forall\, v' \in cod(\mu'), freeLocs(v') \subseteq dom(\mu')$, and the result holds.

□

Proposition 6.13 ($\longmapsto$ is well defined). *If $t^U \mid \mu \overset{\phi}{\longmapsto} r$ and $t^U \vdash \mu$, then $r \in$ Config$_U \cup \{$* **error** *$\}$ and if $r = t'^U \mid \mu' \in$ Config$_U$ then also $t'^U \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$.*

Proof. By induction on the structure of a derivation of $t^U \mid \mu \overset{\phi}{\longmapsto} r$.

Case (R$\longrightarrow$). $t^U \mid \mu \xrightarrow{\phi} r$. By well-definedness of $\longrightarrow$ (Prop 6.12), $r \in$ Config$_{\widetilde{T}} \cup \{$ **error** $\}$ and if $r = t'^U \mid \mu' \in$ Config$_U$ then also $t'^U \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$.

Case (Rprot). $t^U = \text{prot}^{g,U'}_{\varepsilon g'}\ \phi'(\varepsilon t^{U''}_1)$ and

$$\text{(Iprot)}\ \frac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \quad \varepsilon'_r \vdash \widetilde{g_r \vee g'} \preccurlyeq g'_c \\ \phi' \triangleright t^{U''}_1 \in \mathbb{T}[U''] \\ \varepsilon \vdash U'' \lesssim U' \quad \varepsilon_\ell \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \text{prot}^{g,U'}_{\varepsilon g'}\ \phi'(\varepsilon t^{U''}_1) \in \mathbb{T}[U' \mathrel{\widetilde{\curlyvee}} g]}$$

Using induction hypothesis on the premise of (Rprot()), then

$$\text{(Rprot())}\ \frac{t^{U''}_1 \mid \mu \xrightarrow{\phi'} t^{U''}_2 \mid \mu'}{\text{prot}^{g,U'}_{\varepsilon g'}\ \phi'(\varepsilon t^{U''}_1) \mid \mu \xrightarrow{\phi} \text{prot}^{g,U'}_{\varepsilon g'}\ \phi'(\varepsilon t^{U''}_2) \mid \mu'}$$

where $\phi' \triangleright t^{U''}_2 \in \mathbb{T}[U'']$, $t^{U''}_2 \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$. Therefore

$$\text{(Iprot)}\ \frac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \quad \varepsilon'_r \vdash \widetilde{g_r \vee g'} \preccurlyeq g'_c \\ \phi' \triangleright t^{U''}_2 \in \mathbb{T}[U''] \\ \varepsilon \vdash U'' \lesssim U' \quad \varepsilon_\ell \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \text{prot}^{g,U'}_{\varepsilon g'}\ \phi'(\varepsilon t^{U''}_2) \in \mathbb{T}[U' \mathrel{\widetilde{\curlyvee}} g]}$$

and the result holds.

Case (Rf). $t^U = f[t^{U'}_1]$, $\phi \triangleright f[t^{U'}] \in \mathbb{T}[U]$, $t^{U'}_1 \mid \mu \xrightarrow{\phi} t^{U'}_2 \mid \mu'$, and consider $F : \mathbb{T}[U'] \to \mathbb{T}[U]$, where $F(\phi \triangleright t^{U'}) = \phi \triangleright f[t^{U'}]$. By induction hypothesis, $\phi \triangleright t^{U'}_2 \in \mathbb{T}[U']$, so $F(\phi \triangleright t^{U'}_2) = \phi \triangleright f[t^{U'}_2] \in \mathbb{T}[U]$.
By induction hypothesis we also know that $t^{U'}_2 \vdash \mu'$.
If $freeLocs(t^{U'}_2) \subseteq \mu'$, $freeLocs(f[t^U_1]) \subseteq \mu$, and $dom(\mu) \subseteq dom(\mu')$, then it is easy to see that $freeLocs(f[t^{U'}_2]) \subseteq \mu'$, and therefore conclude that $f[t^{U_2}] \vdash \mu'$.

Case (R$f$err, R$h$err, Rprot()$f$err, Rprot()$h$err). $r = \textbf{error}$.

Case (Rh). $t^U = h[et]$, $\phi \triangleright h[t^{U'}] \in \mathbb{T}[U]$, and consider $G : \text{EvLabel} \times \text{GLabel} \times \text{GLabel} \times \text{EvTerm} \to \mathbb{T}[U]$, $G(\phi, et) = \phi \triangleright h[et]$ and $et \longrightarrow_c et'$. Then there exists U_e, U_x such that $et = \varepsilon_e t^{U_e}_e$ and $\varepsilon_e \vdash U_e \lesssim U_x$. Also, $t_e = \varepsilon_v v :: U_e$, with $v \in \mathbb{T}[U_v]$ and $\varepsilon_v \vdash U_v \lesssim U_e$.
We know that $\varepsilon_c = \varepsilon_v \circ^{<:} \varepsilon_e$ is defined, and $et = \varepsilon_e t_e \longrightarrow_c \varepsilon_c v = et'$. By definition of $\circ^{<:}$ we have $\varepsilon_c \vdash U_v \lesssim U_x$, so $G(\phi, et') = \phi \triangleright h[et'] \in \mathbb{T}[U]$.
As $freeLocs(et) = freeLocs(et')$ and $\mu' = \mu$ then it is easy to conclude that $h[et'] \vdash \mu$.

Case (Rprot()h). Similar case to (Rh) case, using $P : \text{EvTerm} \to \mathbb{T}[U]$, $P(et) = \phi \triangleright \text{prot}^{g,U}_{\varepsilon g'}\ \phi'(et)$.

□

Now we can establish type safety: programs do not get stuck, though they may terminate with cast errors. Also the store of a program is well typed.

Proposition 6.14 (Type Safety). *If $\phi \triangleright t^U \in \mathbb{T}[U]$ then either t^U is a value v; $t^U \mid \mu \xrightarrow{\phi}$ **error**; or if $t^U \vdash \mu$ then $t^U \mid \mu \xrightarrow{\phi} t'^U \mid \mu'$ for some term $\phi \triangleright t'^U \in \mathbb{T}[U]$ and some μ' such that $t'^U \vdash \mu'$ and $dom(\mu) \subseteq dom(\mu')$.*

Proof. By induction on the structure of $\phi \triangleright t^U$.

Case (Iu,Il, Ib, Ix, Iλ). t^U is a value.

Case (Iprot). $t^U = \mathsf{prot}^{g,U}_{\varepsilon g'}\ \phi'(\varepsilon t^{U'})$, and

$$\text{(Iprot)}\ \frac{\begin{array}{c}\phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \qquad \varepsilon'_r \vdash \widetilde{g_r \curlyvee g' \preccurlyeq g'_c} \\ \phi' \triangleright t^{U'} \in \mathbb{T}[U'] \\ \varepsilon \vdash U' \lesssim U \qquad \varepsilon_\ell \vdash g' \mathrel{\widetilde{\preccurlyeq}} g\end{array}}{\phi \triangleright \mathsf{prot}^{g,U}_{\varepsilon g'}\ \phi'(\varepsilon t^{U'}) \in \mathbb{T}[U \mathrel{\widetilde{\curlyvee}} g]}$$

By induction hypothesis on $t^{U'}$, one of the following holds:

(1) $t^{U'}$ is a simple value , then by (R$\longrightarrow$), $t^U \mid \mu \overset{\phi}{\longmapsto} v \mid \mu$, and by Prop 6.13, $\phi \triangleright v \in \mathbb{T}[U]$ and the result holds.
(2) $t^{U'}$ is an ascribed value v, then, $\varepsilon t^{U'} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rg), or (Rgerr).
(3) $t^{U'} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \mathbb{T}[U_1] \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rprot()), or (Rprot()ferr).

Case (I::). $t^U = \varepsilon_1 t^{U_1} :: U_2$, and

$$\text{(I::)}\ \frac{\begin{array}{c}\phi \triangleright t^{U_1} \in \mathbb{T}[U_1] \\ \varepsilon_1 \vdash U_1 \lesssim U_2 \qquad \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c\end{array}}{\phi \triangleright \varepsilon_1 t^{U_1} :: U_2 \in \mathbb{T}[U_2]}$$

By induction hypothesis on t^{U_1}, one of the following holds:

(1) t^{U_1} is a value, in which case t^U is also a value.
(2) $t^{U_1} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \mathbb{T}[U_1] \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rf), or (R$f$err).

Case (IUif). $t^U = \mathsf{if}^g\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3}$ and

$$\text{(If)}\ \frac{\begin{array}{c}\phi \triangleright t^{U_1} \in \mathbb{T}[U_1] \qquad \varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_g \qquad \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi' = \langle (\phi.\varepsilon \mathrel{\widetilde{\curlyvee}} \mathit{ilbl}(\varepsilon_1))(\phi.g_c \mathrel{\widetilde{\curlyvee}} \mathit{label}(U_1)), g_c \mathrel{\widetilde{\curlyvee}} g\rangle \\ \phi' \triangleright t^{U_2} \in \mathbb{T}[U_2] \qquad \varepsilon_2 \vdash U_2 \lesssim (U_2 \mathrel{\widetilde{\veebar}} U_3) \\ \phi' \triangleright t^{U_3} \in \mathbb{T}[U_3] \qquad \varepsilon_3 \vdash U_3 \lesssim (U_2 \mathrel{\widetilde{\veebar}} U_3)\end{array}}{\phi \triangleright \mathsf{if}^g\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \in \mathbb{T}[(U_2 \mathrel{\widetilde{\veebar}} U_3) \mathrel{\widetilde{\curlyvee}} g]}$$

By induction hypothesis on t^{U_1}, one of the following holds:

(1) t^{U_1} is a value u, then by (R$\longrightarrow$), $t^U \mid \mu \overset{\phi}{\longmapsto} r$ and $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13.
(2) t^{U_1} is an ascribed value v, then, $\varepsilon_1 t^{U_1} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rg), or (Rgerr).
(3) $t^{U_1} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \mathbb{T}[U_1] \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rf), or (R$f$err).

Case (Iapp). $t^U = \varepsilon_1 t^{U_1} \mathbin{@}_{\varepsilon_\ell}^{U_{11} \xrightarrow{g'_c}_g U_{12}} \varepsilon_2 t^{U_2}$

$$\text{(Iapp)}\ \frac{\begin{array}{cc} \phi \triangleright t^{U_1} \in \mathbb{T}[U_1] & \varepsilon_1 \vdash U_1 \lesssim U_{11} \xrightarrow{g'_c}_g U_{12} \\ \phi \triangleright t^{U_2} \in \mathbb{T}[U_2] & \varepsilon_2 \vdash U_2 \lesssim U_{11} \\ \varepsilon_\ell \vdash \widetilde{g_c \curlyvee g} \preccurlyeq g'_c & \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \end{array}}{\phi \triangleright \varepsilon_1 t^{U_1} \mathbin{@}_{\varepsilon_\ell}^{U_{11} \xrightarrow{g'_c}_g U_{12}} \varepsilon_2 t^{U_2} \in \mathbb{T}[U_{12} \mathbin{\widetilde{\curlyvee}} g]}$$

By induction hypothesis on t^{U_1}, one of the following holds:

(1) t^{U_1} is a value $(\lambda x^{U'_{11}}.t^{U'_{12}})_{g'}$ (by canonical forms Lemma 6.10), posing $U_1 = U'_{11} \xrightarrow{g''_c}_{g'} U'_{12}$. Then by induction hypothesis on t^{U_2}, one of the following holds:

  (a) t^{U_2} is a value u, then by (R$\longrightarrow$), $t^U \mid \mu \overset{\phi}{\longmapsto} r$ and $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 6.13.
  (b) t^{U_2} is an ascribed value v, then, $\varepsilon_2 t^{U_2} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rg), or (Rgerr).
  (c) $t^{U_2} \mid \mu \overset{\phi}{\longmapsto} r_2$ for some $r_2 \in \textsc{Config}_{U_2} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

(2) t^{U_1} is an ascribed value v, then, $\varepsilon_1 t^{U_1} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rg), or (Rgerr).

(3) $t^{U_1} \mid \mu \longmapsto r_1$ for some $r_1 \in \textsc{Config}_{U_1} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

Case (I$\oplus$). Similar case to (Iapp)

Case (Iref). $t^U = \mathsf{ref}_{\varepsilon_\ell}^{U'} \varepsilon t^{U''}$ and

$$\text{(Iref)}\ \frac{\begin{array}{cc} \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c & \phi \triangleright t^{U''} \in \mathbb{T}[U''] \\ \varepsilon \vdash U'' \lesssim U' & \varepsilon_\ell \vdash g_c \mathrel{\widetilde{\preccurlyeq}} label(U') \end{array}}{\phi \triangleright \mathsf{ref}_{\varepsilon_\ell}^{U'} \varepsilon t^{U''} \in \mathbb{T}[\mathsf{Ref}_\perp\ U']}$$

By induction hypothesis on $t^{U''}$, one of the following holds:

(1) $t^{U''}$ is a value v, then by (R$\longrightarrow$), $t^{U'} \mid \mu \overset{\phi}{\longmapsto} r$ and $r \in \textsc{Config}_{U'}$ by Prop 6.13. Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

(2) $t^{U''}$ is an ascribed value v, then, $\varepsilon t^{U'_1} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^{U'} \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_{U'} \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rg), or (Rgerr).

(3) $t^{U''} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \textsc{Config}_{U''} \cup \{\,\mathbf{error}\,\}$. Hence $t^{U'} \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_{U'} \cup \{\,\mathbf{error}\,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

Case (Ideref). $t^U = !^{\mathsf{Ref}_g\ U'} \varepsilon t^{U''}$

$$\text{(Ideref)}\ \frac{\begin{array}{c} \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c \\ \phi \triangleright t^{U''} \in \mathbb{T}[U''] \quad \varepsilon \vdash U'' \lesssim \mathsf{Ref}_g\ U' \end{array}}{\phi \triangleright !^{\mathsf{Ref}_g\ U'} \varepsilon t^{U''} \in \mathbb{T}[U' \mathbin{\widetilde{\curlyvee}} g]}$$

By induction hypothesis on $t^{U''}$, one of the following holds:

(1) $t^{U''}$ is a value $l^{U'''}$ (by canonical forms Lemma 6.10), where $U'' = \mathsf{Ref}_{g'}\ U'''$, then by (R⟶), $t^U \mid \mu \overset{\phi}{\longmapsto} r$ and $r \in \textsc{Config}_U$ by Prop 6.13.

(2) $t^{U''}$ is an ascribed value v, then, $\varepsilon t^{U''} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rg), or (Rgerr).

(3) $t^{U''} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \textsc{Config}_{U''} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

Case (IUassign). $t^U = \varepsilon_1 t^{U_1''} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 t^{U_2}$ and

$$\text{(Iassgn)}\ \frac{\begin{array}{cc} \varepsilon_1 \vdash \mathsf{Ref}_{g'}\ U_1' \lesssim \mathsf{Ref}_g\ U_1 & \phi \triangleright t^{U_1''} \in \mathbb{T}[\mathsf{Ref}_{g'}\ U_1'] \\ \varepsilon_2 \vdash U_2 \lesssim U_1 & \phi \triangleright t^{U_2} \in \mathbb{T}[U_2] \\ \phi.\varepsilon \vdash \phi.g_c \mathrel{\widetilde{\preccurlyeq}} \phi.g_c & \varepsilon_\ell \vdash \phi.g_c \mathbin{\widetilde{\curlyvee}} g \preccurlyeq label(U_1) \end{array}}{\phi \triangleright \varepsilon_1 t^{U_1''} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 t^{U_2} \in \mathbb{T}[\mathsf{Unit}_\perp]}$$

By induction hypothesis on $t^{U_1''}$, one of the following holds:

(1) $t^{U_1''}$ is a value $l^{U_1'''}$ (by canonical forms Lemma 6.10), where $U_1'' = \mathsf{Ref}_{g'}\ U_1'''$. Then by induction hypothesis on t^{U_2}, one of the following holds:

  (a) t^{U_2} is a value u, then by (R⟶), $t^U \mid \mu \overset{\phi}{\longmapsto} r$ and $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13. Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

  (b) t^{U_2} is an ascribed value v, then, $\varepsilon_2 t^{U_2} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rg), or (Rgerr).

  (c) $t^{U_2} \mid \mu \overset{\phi}{\longmapsto} r_2$ for some $r_2 \in \textsc{Config}_{U_2} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

(2) $t^{U_1''}$ is an ascribed value v, then, $\varepsilon_1 t^{U_1} \longrightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rg), or (Rgerr).

(3) $t^{U_1''} \mid \mu \overset{\phi}{\longmapsto} r_1$ for some $r_1 \in \textsc{Config}_{U_1''} \cup \{\, \mathbf{error} \,\}$. Hence $t^U \mid \mu \overset{\phi}{\longmapsto} r$ for some $r \in \textsc{Config}_U \cup \{\, \mathbf{error} \,\}$ by Prop 6.13 and either (Rf), or (R$f$err). Also by Prop 6.13, if $r = t'^U \mid \mu' \in \mathbb{T}[U]$ then $dom(\mu) \subseteq dom(\mu')$.

□

Proposition 6.15 (Static terms do not fail). *Let us define* StaticTerm *the set of evidence augmented terms with full static annotations. Then consider* $t_s \in \textsc{StaticTerm}$, $\phi = \langle \varepsilon \ell_c', \ell_c \rangle$, *and* μ_s, *such that* $\varepsilon = \mathcal{I}[\![\ell_c' \mathrel{\widetilde{\preccurlyeq}} \ell_c]\!]$, $\phi \triangleright t_s \in \mathbb{T}[S]$, *and that* $\forall v_s \in cod(\mu_s)$, $v_s \in \textsc{StaticTerm}$. *Then either* t_s *is a value, or*

$$t_s \mid \mu_s \overset{\phi}{\longmapsto} t_s' \mid \mu_s'$$

Proof. We know that if you follow AGT to derive the dynamic semantics of a gradual language, then by construction the resulting language satisfy the dynamic conservative extension property. As we follow AGT to derive the dynamic semantics, we get this property by construction, save for the assignment elimination reduction rule. In this rule we add an extra check of the form $\phi.\varepsilon \lfloor \leq \rfloor\ ilbl(\varepsilon)$. So if we prove that the extra check is always satisfied, then the result holds.

Let us consider a t_1' fully static like so:

$$\text{(Iassgn)}\ \frac{\begin{array}{cc} \varepsilon_1 \vdash \mathsf{Ref}_{\ell'}\ S_1' \lesssim \mathsf{Ref}_{\ell}\ S_1 & \phi \triangleright o_{\ell'}^{S_1'} \in \mathbb{T}[\mathsf{Ref}_{\ell'}\ S_1'] \\ \varepsilon_2 \vdash S_2 \lesssim S_1 & \phi \triangleright u \in \mathbb{T}[S_2] \\ \phi.\varepsilon \vdash \ell_c' \mathrel{\widetilde{\preccurlyeq}} \ell_c & \varepsilon_\ell \vdash \widetilde{\ell_c \curlyvee \ell \preccurlyeq label(S_1)} \end{array}}{\phi \triangleright \varepsilon_1 o_{\ell'}^{S_1'} \overset{\ell, S_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u \in \mathbb{T}[\mathsf{Unit}_\perp]}$$

By inspection of the reduction rules we have to prove that $\phi.\varepsilon \lfloor\leq\rfloor\ ilbl(\varepsilon)$. $\phi.\varepsilon \lfloor\leq\rfloor\ ilbl(\varepsilon)$. We know by definition of interior between two static labels that $\varepsilon = \mathcal{I}[\![\ell_c' \mathrel{\widetilde{\preccurlyeq}} \ell_c]\!] = \langle[\ell_c', \ell_c'], [\ell_c, \ell_c]\rangle$. Also, ff $\mu_s(o^{S_1'}) = \varepsilon u' :: S_1'$, as everything is static, $ilbl(\varepsilon)$ have to have the form $\langle[\ell_u, \ell_u], [label(S_1'), label(S_1')]\rangle$, for some ℓ_u. Then we have to prove that $\ell_c \preccurlyeq label(S_1')$, but notice that as everything is static, $\varepsilon_\ell \vdash \widetilde{\ell_c \curlyvee \ell \preccurlyeq label(S_1)}$ is equivalent to $\varepsilon_\ell \vdash \ell_c \curlyvee \ell \preccurlyeq label(S_1)$, therefore we know that $\ell_c \preccurlyeq label(S_1)$ and the result holds.

□

6.5 Dynamic Gradual Guarantee

In this section we present the proof the Dynamic Gradual Guarantee for $\mathrm{GSL}_{\mathsf{Ref}}$ without the specific check in rule $(r7)$.

Definition 6.16 (Intrinsic term precision). Let $\Omega \in \mathcal{P}(\mathbb{V}[*] \times \mathbb{V}[*]) \cup \mathcal{P}(\mathrm{Loc}_* \times \mathrm{Loc}_*)$ be defined as $\Omega ::= \{\, \overline{x^{U_{i1}} \sqsubseteq x^{U_{i2}}}, \overline{o^{U_{i1}} \sqsubseteq o^{U_{i2}}} \,\}$ We define an ordering relation $(\cdot \vdash \cdot \sqsubseteq \cdot) \in (\mathcal{P}(\mathbb{V}[*] \times \mathbb{V}[*]) \cup \mathcal{P}(\mathrm{Loc}_* \times \mathrm{Loc}_*)) \times \mathbb{T}[*] \times \mathbb{T}[*]$ shown in Figure 29.

Definition 6.17 (Well Formedness of Ω). We say that Ω is well formed iff $\forall \{\, l^{U_{i1}} \sqsubseteq l^{U_{i2}} \,\} \in \Omega. U_{i1} \sqsubseteq U_{i2}$

Before proving the gradual guarantee, we first establish some auxiliary properties of precision. For the following propositions, we assume Well Formedness of Ω (Definition 6.17).

Proposition 6.18. *If $\Omega \vdash t^{U_1} \sqsubseteq t^{U_2}$ for some $\Omega \in \mathcal{P}(\mathbb{V}[*] \times \mathbb{V}[*]) \cup \mathcal{P}(\mathrm{Loc}_* \times \mathrm{Loc}_*)$, then $U_1 \sqsubseteq U_2$.*

Proof. Straightforward induction on $\Omega \vdash t^{U_1} \sqsubseteq t^{U_2}$, since the corresponding precision on types is systematically a premise (either directly or transitively). □

Proposition 6.19. *Let $g_1, g_2 \in \mathrm{EvFrame}$ such that $\phi_1 \triangleright g_1[\varepsilon_{11} t_1^{U_1}] \in \mathbb{T}[U_1'], \phi_2 \triangleright g_2[\varepsilon_{21} t_1^{U_2}] \in \mathbb{T}[U_2']$, with $U_1' \sqsubseteq U_2'$. Then if $g_1[\varepsilon_{11} t_1^{U_1}] \sqsubseteq g_2[\varepsilon_{21} t_1^{U_2}]$, $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$ and $t_2^{U_1} \sqsubseteq t_2^{U_2}$, then $g_1[\varepsilon_{12} t_2^{U_1}] \sqsubseteq g_2[\varepsilon_{22} t_2^{U_2}]$*

Proof. We proceed by case analysis on g_i.

Case $(\square \mathbin{@^{U}_{\varepsilon}} et)$. Then for $i \in \{\,1, 2\,\}$ g_i must have the form $\square \mathbin{@^{U_i''}_{\varepsilon_i}} \varepsilon_i' t^{U_i'}$ for some U_i'', ε_i' and $t^{U_i'}$. As $g_1[\varepsilon_{11} t_1^{U_1}] \sqsubseteq g_2[\varepsilon_{21} t_1^{U_2}]$ then by $\sqsubseteq_{APP}$ $\varepsilon_1 \sqsubseteq \varepsilon_2, \varepsilon_1' \sqsubseteq \varepsilon_2'$, $U_1'' \sqsubseteq U_2''$ and $t^{U_1'} \sqsubseteq t^{U_2'}$.

As $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$ and $t_2^{U_1} \sqsubseteq t_2^{U_2}$, then by $\sqsubseteq_{APP}$ $\varepsilon_{12} t_2^{U_1} \mathbin{@^{U_1''}_{\varepsilon_1}} \varepsilon_1' t^{U_1'} \sqsubseteq \varepsilon_{22} t_2^{U_2} \mathbin{@^{U_2''}_{\varepsilon_2}} \varepsilon_2' t^{U_2'}$, and the result holds.

Case $(\square \oplus^{g} et, ev \oplus^{g} \square, ev \mathbin{@^{U}_{\varepsilon_\ell}} \square, \square :: U, !^{U} \square, \square \overset{g, U_1}{:=}_{\varepsilon_\ell} et,$ $ev \overset{g, U_1}{:=}_{\varepsilon_\ell} \square, \mathsf{if}^{g}\ \square\ \mathsf{then}\ et\ \mathsf{else}\ et)$. Straightforward using similar argument to the previous case.

□

$$\frac{}{\Omega \cup \{\, x^{U_1} \sqsubseteq x^{U_2} \,\} \vdash x^{U_1} \sqsubseteq x^{U_2}} \qquad \frac{g_1 \sqsubseteq g_2}{\Omega \vdash b_{g_1} \sqsubseteq b_{g_2}} \qquad \frac{g_1 \sqsubseteq g_2}{\Omega \vdash \mathsf{unit}_{g_1} \sqsubseteq \mathsf{unit}_{g_2}}$$

$$\frac{g_1 \sqsubseteq g_2}{\Omega \cup \{\, o^{U_1} \sqsubseteq o^{U_2} \,\} \vdash o^{U_1}_{g_1} \sqsubseteq o^{U_2}_{g_2}} \qquad \frac{\begin{array}{c} U_{11} \sqsubseteq U_{12} \quad {g_{c1}}' \sqsubseteq {g_{c2}}' \quad g_1 \sqsubseteq g_2 \\ \Omega \cup \{\, x^{U_{11}} \sqsubseteq x^{U_{12}} \,\} \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \end{array}}{\Omega \vdash (\lambda^{{g_{c1}}'} x^{U_{11}}.t^{U_{12}})_{g_1} \sqsubseteq (\lambda^{{g_{c2}}'} x^{U_{21}}.t^{U_{22}})_{g_2}}$$

$$\frac{\begin{array}{ccc} & g_1 \sqsubseteq g_2 & \\ g'_1 \sqsubseteq g'_1 & \phi'_1 \sqsubseteq \phi'_2 & \varepsilon_1 \sqsubseteq \varepsilon_2 \\ U_1 \sqsubseteq U_2 & \Omega \vdash t^{U'_1} \sqsubseteq t^{U'_2} & \varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2} \end{array}}{\Omega \vdash \mathsf{prot}^{g_1, U_1}_{\varepsilon_{\ell 1} g'_1} \phi'_1(\varepsilon_1 t^{U'_1}) \sqsubseteq \mathsf{prot}^{g_2, U_2}_{\varepsilon_{\ell 2} g'_2} \phi'_2(\varepsilon_2 t^{U'_2})} \qquad \frac{\begin{array}{c} \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \\ U_{12} \sqsubseteq U_{22} \quad \varepsilon_1 \sqsubseteq \varepsilon_2 \end{array}}{(\varepsilon_1 t^{U_{11}} :: U_{12}) \sqsubseteq (\varepsilon_2 t^{U_{21}} :: U_{22})}$$

$$\frac{\begin{array}{ccc} g_{c1} \sqsubseteq g_{c2} & \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} & \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \\ \varepsilon_{11} \sqsubseteq \varepsilon_{21} & \varepsilon_{12} \sqsubseteq \varepsilon_{22} & \varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2} \\ U_1 \sqsubseteq U_3 & U_2 \sqsubseteq U_4 & g_1 \sqsubseteq g_2 \end{array}}{\Omega \vdash \varepsilon_{11} t^{U_{11}} \mathbin{@}^{U_1 \xrightarrow{g_{c1}}_{g_1} U_2}_{\varepsilon_{\ell 1}} [12] t^{U_{12}} \sqsubseteq \varepsilon_{21} t^{U_{21}} \mathbin{@}^{U_3 \xrightarrow{g_{c2}}_{g_2} U_4}_{\varepsilon_{\ell 2}} [22] t^{U_{22}}}$$

$$\frac{\begin{array}{cc} g_1 \sqsubseteq g_2 & \\ \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} & \varepsilon_{11} \sqsubseteq \varepsilon_{21} \\ \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{23}} & \varepsilon_{12} \sqsubseteq \varepsilon_{22} \\ \Omega \vdash t^{U_{13}} \sqsubseteq t^{U_{23}} & \varepsilon_{13} \sqsubseteq \varepsilon_{23} \end{array}}{\begin{array}{l} \Omega \vdash \mathsf{if}^{g_1}\ \varepsilon_{11} t^{U_{11}}\ \mathsf{then}\ \varepsilon_{12} t^{U_{12}}\ \mathsf{else}\ \varepsilon_{13} t^{U_{13}} \sqsubseteq \\ \mathsf{if}^{g_2}\ \varepsilon_{21} t^{U_{21}}\ \mathsf{then}\ \varepsilon_{22} t^{U_{22}}\ \mathsf{else}\ \varepsilon_{23} t^{U_{23}} \end{array}}$$

$$\frac{\begin{array}{c} \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \quad \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \\ \varepsilon_{11} \sqsubseteq \varepsilon_{21} \quad \varepsilon_{12} \sqsubseteq \varepsilon_{22} \quad g_1 \sqsubseteq g_2 \end{array}}{\Omega \vdash (\varepsilon_{11} t^{U_{11}} \oplus^{g_1} \varepsilon_{12} t^{U_{12}}) \sqsubseteq (\varepsilon_{21} t^{U_{21}} \oplus^{g_2} \varepsilon_{22} t^{U_{22}})} \qquad \frac{\begin{array}{c} U_1 \sqsubseteq U_2 \quad \varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2} \quad g_{c1} \sqsubseteq g_{c2} \\ \varepsilon_1 \sqsubseteq \varepsilon_2 \quad \Omega \vdash t^{U'_1} \sqsubseteq t^{U'_2} \end{array}}{\Omega \vdash \mathsf{ref}^{U_1}_{\varepsilon_{\ell 1}}\ \varepsilon_1 t^{U'_1} \sqsubseteq \mathsf{ref}^{U_2}_{\varepsilon_{\ell 2}}\ \varepsilon_2 t^{U'_2}}$$

$$\frac{\begin{array}{c} \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \\ U_1 \sqsubseteq U_2 \quad \varepsilon_1 \sqsubseteq \varepsilon_2 \end{array}}{\Omega \vdash !^{U_1} \varepsilon_1 t^{U_{11}} \sqsubseteq\ !^{U_2} \varepsilon_2 t^{U_{21}}} \qquad \frac{\begin{array}{c} \Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \qquad \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \\ \varepsilon_{11} \sqsubseteq \varepsilon_{21} \quad \varepsilon_{12} \sqsubseteq \varepsilon_{22} \quad \varepsilon_1 \sqsubseteq \varepsilon_2 \quad g_1 \sqsubseteq g_2 \quad U_1 \sqsubseteq U_2 \end{array}}{\Omega \vdash \varepsilon_{11} t^{U_{11}} \overset{g_1, U_1}{:=}_{\varepsilon_1} \varepsilon_{12} t^{U_{12}} \sqsubseteq \varepsilon_{21} t^{U_{21}} \overset{g_2, U_2}{:=}_{\varepsilon_2} \varepsilon_{22} t^{U_{22}}}$$

$$\frac{\begin{array}{c} \forall o^{U_1} \in dom(\mu_1). \exists o^{U_2} \in dom(\mu_2)\ s.t. \\ \Omega \vdash o^{U_1} \sqsubseteq o^{U_2} \qquad \Omega \vdash \mu_1(l^{U_1}) \sqsubseteq \mu_2(l^{U_2}) \end{array}}{\Omega \vdash \mu_1 \sqsubseteq \mu_2}$$

$$\text{where } \phi_1 \sqsubseteq \phi_2 \iff \phi_1.\varepsilon \sqsubseteq \phi_2.\varepsilon \wedge \phi_1.g_c \sqsubseteq \phi_2.g_c \wedge \phi_1.g_c \sqsubseteq \phi_2.g_c$$

Fig. 29. Intrinsic term precision

Proposition 6.20. *Let $g_1, g_2 \in$ EvFrame such that $\phi_1 \triangleright g_1[\varepsilon_1 t^{U_1}] \in \mathbb{T}[U'_1]$, $\phi_2 \triangleright g_2[\varepsilon_2 t^{U_2}] \in \mathbb{T}[U'_2]$, with $U'_1 \sqsubseteq U'_2$. Then if $g_1[\varepsilon_1 t^{U_1}] \sqsubseteq g_2[\varepsilon_2 t^{U_2}]$ then $t^{U_1} \sqsubseteq t^{U_2}$ and $\varepsilon_1 \sqsubseteq \varepsilon_2$.*

Proof. We proceed by case analysis on g_i.

Case ($\square \ @^{U}_{\varepsilon}\ et$). Then there must exist some $\varepsilon_{\ell i}, U_i, \varepsilon'_i$ and $t^{U'_i}$ such that $g[\varepsilon_1 t^{U_1}] = \varepsilon_1 t^{U_1} \ @^{U''_1}_{\varepsilon_1}\ \varepsilon'_1 t^{U'_1}$ and $g[\varepsilon_2 t^{U_2}] = \varepsilon_2 t^{U_2} \ @^{U''_2}_{\varepsilon_{\ell 2}}\ \varepsilon'_2 t^{U'_2}$. Then by the hypothesis and the premises of ($\sqsubseteq_{APP}$), $t^{U_1} \sqsubseteq t^{U_2}$ and $\varepsilon_1 \sqsubseteq \varepsilon_2$, and the result holds immediately.

Case ($\square \oplus^{g} et, ev \oplus^{g} \square, ev \ @^{U}_{\varepsilon_\ell} \square, \square :: U, !^{U} \square, \square \overset{g,U_1}{:=}_{\varepsilon_\ell} et,$
$ev \overset{g,U_1}{:=}_{\varepsilon_\ell} \square, \mathsf{if}^{g}\ \square$ then et else et). Straightforward using similar argument to the previous case.

$\square$

Proposition 6.21. *Let $f_1, f_2 \in$ EvFrame such that $\phi_1 \triangleright f_1[t_1^{U_1}] \in \mathbb{T}[U'_1]$, $\phi_2 \triangleright f_2[t_1^{U_2}] \in \mathbb{T}[U'_2]$, with $U'_1 \sqsubseteq U'_2$. Then if $f_1[t_1^{U_1}] \sqsubseteq f_2[t_1^{U_2}]$ and $t_2^{U_1} \sqsubseteq t_2^{U_2}$, then $f_1[t_1^{U_1}] \sqsubseteq f_2[t_1^{U_2}]$*

Proof. Suppose $f_i[t_1^{U_1}] = g_i[\varepsilon_i t_1^{U_i}]$. We know that $\phi_1 \triangleright g_1[\varepsilon_1 t_1^{U_1}] \in \mathbb{T}[U'_1]$, $\phi_1 \triangleright g_2[\varepsilon_2 t_1^{U_2}] \in \mathbb{T}[U'_2]$ and $U'_1 \sqsubseteq U'_2$. Therefore if $g_1[\varepsilon_1 t_1^{U_1}] \sqsubseteq g_1[\varepsilon_1 t_1^{U_2}]$, by Prop 6.20, $\varepsilon_1 \sqsubseteq \varepsilon_2$. Finally by Prop 6.19 we conclude that $g_1[\varepsilon_1 t_2^{U_1}] \sqsubseteq g_1[\varepsilon_1 t_2^{U_2}]$. $\square$

Proposition 6.22. *Let $f_1, f_2 \in$ EvFrame such that $\phi_1 \triangleright f_1[t^{U_1}] \in \mathbb{T}[U'_1]$, $\phi_2 \triangleright f_2[t^{U_2}] \in \mathbb{T}[U'_2]$, with $U'_1 \sqsubseteq U'_2$. Then if $f_1[t^{U_1}] \sqsubseteq f_2[t^{U_2}]$ then $t^{U_1} \sqsubseteq t^{U_2}$.*

Proof. Suppose $f_i[t^{U_1}] = g_i[\varepsilon_i t_1^{U_i}]$. We know that $\phi_1 \triangleright g_1[\varepsilon_1 t_1^{U_1}] \in \mathbb{T}[U'_1]$, $\phi_1 \triangleright g_2[\varepsilon_2 t_1^{U_2}] \in \mathbb{T}[U'_2]$ and $U'_1 \sqsubseteq U'_2$. Therefore if $g_1[\varepsilon_1 t_1^{U_1}] \sqsubseteq g[\varepsilon_2 t_1^{U_2}]$, then using Prop 6.20 we conclude that $t^{U_1} \sqsubseteq t^{U_2}$.
$\square$

Proposition 6.23 (Substitution preserves precision). *If $\Omega \cup \{x^{U_3} \sqsubseteq x^{U_4}\} \vdash t^{U_1} \sqsubseteq t^{U_2}$ and $\Omega \vdash t^{U_3} \sqsubseteq t^{U_4}$, then $\Omega \vdash [t^{U_3}/x^{U_3}]t^{U_1} \sqsubseteq [t^{U_4}/x^{U_4}]t^{U_2}$.*

Proof. By induction on the derivation of $t^{U_1} \sqsubseteq t^{U_2}$, and case analysis of the last rule used in the derivation. All cases follow either trivially (no premises) or by the induction hypotheses. $\square$

Proposition 6.24 (Monotone precision for $\circ^{<:}$). *If $\varepsilon_1 \sqsubseteq \varepsilon_2$ and $\varepsilon_3 \sqsubseteq \varepsilon_4$ then $\varepsilon_1 \circ^{<:} \varepsilon_3 \sqsubseteq \varepsilon_2 \circ^{<:} \varepsilon_4$.*

Proof. By definition of consistent transitivity for $<:$ and the definition of precision. $\square$

Proposition 6.25 (Monotone precision for $\circ^{\preccurlyeq}$). *If $\varepsilon_1 \sqsubseteq \varepsilon_2$ and $\varepsilon_3 \sqsubseteq \varepsilon_4$ then $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3 \sqsubseteq \varepsilon_2 \circ^{\preccurlyeq} \varepsilon_4$.*

Proof. By definition of consistent transitivity for $\preccurlyeq$ and the definition of precision. $\square$

Proposition 6.26 (Monotone precision for join). *If $\varepsilon_1 \sqsubseteq \varepsilon_2$ and $\varepsilon_3 \sqsubseteq \varepsilon_4$ then $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_3 \sqsubseteq \varepsilon_2 \mathbin{\widetilde{\curlyvee}} \varepsilon_4$.*

Proof. By definition of join and the definition of precision. $\square$

Proposition 6.27. *If $\mathsf{Ref}\ U_1 \sqsubseteq \mathsf{Ref}\ U_2$ then $U_1 \sqsubseteq U_2$.*

Proof. By definition of precision we know that
$\{ \mathsf{Ref}\ T \mid T \in \gamma(U_1) \} \subseteq \{ \mathsf{Ref}\ T \mid T \in \gamma(U_2) \}$. This relation is true only if $\gamma(U_1) \subseteq \gamma(U_2)$ which is equivalent to $U_1 \sqsubseteq U_2$. $\square$

Proposition 6.28. *If $U_{11} \sqsubseteq U_{12}$ and $U_{21} \sqsubseteq U_{22}$ then $U_{11} \mathbin{\widetilde{\dot\vee}} U_{21} \sqsubseteq U_{12} \mathbin{\widetilde{\dot\vee}} U_{22}$.*

Proof. By induction on the type derivation of the types and consistent join. $\square$

Lemma 6.29. *If* $\varepsilon_1 \vdash \mathsf{Ref}_{g_{11}}\ U_{11} \lesssim \mathsf{Ref}_{g_{12}}\ U_{12}$ *and* $\varepsilon_2 \vdash \mathsf{Ref}_{g_{21}}\ U_{21} \lesssim \mathsf{Ref}_{g_{22}}\ U_{22}$*, and* $\varepsilon_1 \sqsubseteq \varepsilon_2$*, then* $iref(\varepsilon_1) \sqsubseteq iref(\varepsilon_2)$.

Proof. By definition of precision and *iref*. □

Proposition 6.30 (Dynamic guarantee for $\longrightarrow$). *Suppose* $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$*,* $\phi_1 \sqsubseteq \phi_2$*, and* $\Omega \vdash \mu_1 \sqsubseteq \mu_2$*. If* $t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} t_2^{U_1} \mid \mu_1'$ *then* $t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} t_2^{U_2} \mid \mu_2'$ *where* $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$ *and* $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$*, for some* $\Omega' \supseteq \Omega$.

Proof. By induction on the structure of $t_1^{U_1} \sqsubseteq t_1^{U_2}$. For simplicity we omit the $\Omega \vdash$ notation on precision relations when it is not relevant for the argument.

Case ($\longrightarrow \oplus$). We know that $t_1^{U_1} = (\varepsilon_{11}(b_1)_{g_{11}} \oplus^{g_1} \varepsilon_{12}(b_2)_{g_{12}})$ then by ($\sqsubseteq_\oplus$) $t_1^{U_2} = (\varepsilon_{21}(b_1)_{g_{21}} \oplus^{g_1} \varepsilon_{22}(b_2)_{g_{22}})$ for some $\varepsilon_{21}, \varepsilon_{22}, g_{21}, g_{22}$ such that $\varepsilon_{11} \sqsubseteq \varepsilon_{21}$, $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$, $g_{11} \sqsubseteq g_{21}$ and $g_{12} \sqsubseteq g_{22}$.
If $t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} b_3 \mid \mu_1$ where $b_3 = (\varepsilon_{11} \mathbin{\tilde\curlyvee} \varepsilon_{12})(b_1 [\![\oplus]\!]\ b_2)_{(g_{11} \tilde\curlyvee g_{21})} :: \mathsf{Bool}_{g_1}$, then
$t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} b_3' \mid \mu_2$ where $b_3' = (\varepsilon_{21} \mathbin{\tilde\curlyvee} \varepsilon_{22})(b_1 [\![\oplus]\!]\ b_2)_{(g_{21} \tilde\curlyvee g_{22})} :: \mathsf{Bool}_{g_2}$. By Lemma 6.26, $(\varepsilon_{11} \mathbin{\tilde\curlyvee} \varepsilon_{12}) \sqsubseteq (\varepsilon_{21} \mathbin{\tilde\curlyvee} \varepsilon_{22})$. Also $(g_{11} \mathbin{\tilde\curlyvee} g_{21}) \sqsubseteq (g_{21} \mathbin{\tilde\curlyvee} g_{22})$.

$$\dfrac{\dfrac{(g_{11} \mathbin{\tilde\curlyvee} g_{21}) \sqsubseteq (g_{12} \mathbin{\tilde\curlyvee} g_{22})}{\Omega \vdash (b_1 [\![\oplus]\!]\ b_2)_{(g_{11} \tilde\curlyvee g_{21})} \sqsubseteq (b_1 [\![\oplus]\!]\ b_2)_{(g_{21} \tilde\curlyvee g_{22})}} \qquad \mathsf{Bool}_{g_1} \sqsubseteq \mathsf{Bool}_{g_2} \qquad (\varepsilon_{11} \mathbin{\tilde\curlyvee} \varepsilon_{12}) \sqsubseteq (\varepsilon_{21} \mathbin{\tilde\curlyvee} \varepsilon_{22})}{\begin{array}{c}(\varepsilon_{11} \mathbin{\tilde\curlyvee} \varepsilon_{12})(b_1 [\![\oplus]\!]\ b_2)_{(g_{11} \tilde\curlyvee g_{21})} :: \mathsf{Bool}_{g_1} \sqsubseteq \\ (\varepsilon_{21} \mathbin{\tilde\curlyvee} \varepsilon_{22})(b_1 [\![\oplus]\!]\ b_2)_{(g_{21} \tilde\curlyvee g_{22})} :: \mathsf{Bool}_{g_2}\end{array}}$$

Therefore $t_2^{U_1} \sqsubseteq t_2^{U_2}$. As $\Omega' = \Omega$, $\mu_1' = \mu_1$ and $\mu_2 = \mu_2'$ then $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$.

Case ($\longrightarrow$prot). We know that $t_1^{U_1} = \mathsf{prot}^{g_1, U_1}_{\varepsilon_{\ell 1} g_1'} \phi_1'(\varepsilon_1 u_1)$, then by ($\sqsubseteq_{\mathsf{prot}()}$) $t_1^{U_2} = \mathsf{prot}^{g_2, U_2}_{\varepsilon_{\ell 2} g_2'} \phi_2'(\varepsilon_2 u_2)$, and therefore

$$\dfrac{\begin{array}{c} g_1' \sqsubseteq g_1' \\ U_1 \sqsubseteq U_2 \end{array} \qquad \begin{array}{c} g_1 \sqsubseteq g_2 \\ \phi_1' \sqsubseteq \phi_2' \\ \Omega \vdash u_1 \sqsubseteq u_2 \end{array} \qquad \begin{array}{c} \varepsilon_1 \sqsubseteq \varepsilon_2 \\ \varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2} \end{array}}{\Omega \vdash \mathsf{prot}^{g_1, U_1}_{\varepsilon_{\ell 1} g_1'} \phi_1'(\varepsilon_1 u_1) \sqsubseteq \mathsf{prot}^{g_2, U_2}_{\varepsilon_{\ell 2} g_2'} \phi_2'(\varepsilon_2 u_2)}$$

for some ε_2, u_2, U_2 and $\varepsilon_{\ell 2}$, where $u_1 \in \mathbb{T}[U_1']$ and $u_2 \in \mathbb{T}[U_2']$. If
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} (\varepsilon_1 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 1})(u_1 \mathbin{\tilde\curlyvee} g_1') :: U_1 \mathbin{\tilde\curlyvee} g_1 \mid \mu_1$. Therefore, $t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} (\varepsilon_2 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 2})(u_2 \mathbin{\tilde\curlyvee} g_2') :: U_2 \mathbin{\tilde\curlyvee} g_2 \mid \mu_2$.
By Lemma 6.26, $(\varepsilon_1 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 1}) \sqsubseteq (\varepsilon_2 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 2})$, and as join is monotone $U_1 \mathbin{\tilde\curlyvee} g_1 \sqsubseteq U_2 \mathbin{\tilde\curlyvee} g_2$ and $(u_1 \mathbin{\tilde\curlyvee} g_1') \sqsubseteq (u_2 \mathbin{\tilde\curlyvee} g_2')$. Therefore by $\sqsubseteq_{::}$, $(\varepsilon_1 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 1})(u_1 \mathbin{\tilde\curlyvee} g_1') :: U_1 \mathbin{\tilde\curlyvee} g_1 \sqsubseteq (\varepsilon_2 \mathbin{\tilde\curlyvee} \varepsilon_{\ell 2})(u_2 \mathbin{\tilde\curlyvee} g_2') :: U_2 \mathbin{\tilde\curlyvee} g_2$. As $\Omega' = \Omega$, $\mu_1' = \mu_1$ and $\mu_2 = \mu_2'$ then $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$.

Case ($\longrightarrow$app). We know that

$t_1^{U_1} = \varepsilon_{11}(\lambda x^{U_{11}}.t^{U_{12}})_{g_1'} \mathbin{@}^{U_1 \xrightarrow{g_{c1}'}_{g_1} U_2}_{\varepsilon_{\ell 1}} \varepsilon_{12} u$ then by ($\sqsubseteq_{app}$) $t_1^{U_2}$ must have the form

$t_1^{U_2} = \varepsilon_{21}(\lambda x^{U_{21}}.t^{U_{22}})_{g_2'} \mathbin{@}^{U_3 \xrightarrow{g_{c2}'}_{g_2} U_4}_{\varepsilon_{\ell 2}} \varepsilon_{22} u_2$ for some $\varepsilon_{21}, x^{U_{21}}, t^{U_{22}}, U_3, U_4, \varepsilon_{22}, g_{c2}', g_2$ and u_2.
Let us pose $\varepsilon_1 = \varepsilon_{12} \circ^{<:} idom(\varepsilon_{11})$ and $\varepsilon_{r1}' = (\phi_1.\varepsilon \mathbin{\tilde\curlyvee} ilbl(\varepsilon_{11})) \circ^{\preccurlyeq} \varepsilon_{\ell 1} \circ^{\preccurlyeq} ilat(\varepsilon_{11})$,
$\phi_1' = \langle \varepsilon_{r1}'(g_1' \mathbin{\tilde\curlyvee} \phi_1.g_c), g_1 \mathbin{\tilde\curlyvee} \phi_1.g_c \rangle$. Then
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} \mathsf{prot}^{g_1, U_2}_{ilbl(\varepsilon_{11}) g_1'} \phi_1'(icod(\varepsilon_{11}) t_1') \mid \mu_1$ with $t_1' = [(\varepsilon_1 u_1 :: U_{11})/x^{U_{11}}]t^{U_{12}}$.
Also, let us pose $\varepsilon_2 = \varepsilon_{22} \circ^{<:} idom(\varepsilon_{21})$ and $\varepsilon_{r2}' = (\phi_2.\varepsilon \mathbin{\tilde\curlyvee} ilbl(\varepsilon_{21})) \circ^{\preccurlyeq} \varepsilon_{\ell 2} \circ^{\preccurlyeq} ilat(\varepsilon_{21})$, $\phi_2' =$

$\langle \varepsilon'_{r2}(g'_2 \mathbin{\widetilde{\curlyvee}} \phi_2.g_c), g_2 \mathbin{\widetilde{\curlyvee}} \phi_2.g_c\rangle$. Then
$t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} \mathrm{prot}^{g_2, U_4}_{ilbl(\varepsilon_{21})g'_2} \phi'_2(icod(\varepsilon_{21})t'_2) \mid \mu_2$ with $t'_2 = [(\varepsilon_2 u_2 :: U_{21})/x^{U_{21}}]t^{U_{22}}$.
As $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$, then $u_1 \sqsubseteq u_2$, $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$ and $idom(\varepsilon_{11}) \sqsubseteq idom(\varepsilon_{21})$ as well, then by Prop 6.24 $\varepsilon_1 \sqsubseteq \varepsilon_2$. Then $\varepsilon_1 u_1 :: U_{11} \sqsubseteq \varepsilon_2 u_2 :: U_{21}$ by ($\sqsubseteq_{::}$).
We also know by ($\sqsubseteq_{APP}$) and ($\sqsubseteq_\lambda$) that $\Omega \cup \{x^{U_{21}} \sqsubseteq x^{U_{21}}\} \vdash t^{U_{12}} \sqsubseteq t^{U_{22}}$. By Substitution preserves precision (Prop 6.23) $t'_1 \sqsubseteq t'_2$, therefore $icod(\varepsilon_{11})t'_1 :: U_2 \sqsubseteq icod(\varepsilon_{21})t'_2 :: U_4$ by ($\sqsubseteq_{::}$). Also $g_1 \sqsubseteq g_2$, $ilbl(\varepsilon_{11}) \sqsubseteq ilbl_{21}$, $g'_1 \sqsubseteq g'_2$ and by Lemma 6.24 and 6.26, $\varepsilon'_{r1} \sqsubseteq \varepsilon'_{r2}$. Also, as $\phi_1.g_c \sqsubseteq \phi_2.g_c$ by monotonicity of the join $g_1 \mathbin{\widetilde{\curlyvee}} \phi_1.g_c \sqsubseteq g_2 \mathbin{\widetilde{\curlyvee}} \phi_2.g_c$, and as $\phi_1.g_c \sqsubseteq \phi_2.g_c$ also by monotonicity of the join $g'_1 \mathbin{\widetilde{\curlyvee}} \phi_1.g_c \sqsubseteq g'_2 \mathbin{\widetilde{\curlyvee}} \phi_2.g_c$. Then by ($\sqsubseteq_{\mathrm{prot}()}$) $t_2^{U_1} \sqsubseteq t_2^{U_2}$. As $\Omega' = \Omega$, $\mu'_1 = \mu_1$ and $\mu_2 = \mu'_2$ then $\Omega' \vdash \mu'_1 \sqsubseteq \mu'_2$.

Case ($\longrightarrow$if-true). $t_1^{U_1} = \mathrm{if}^{g_1}\ \varepsilon_{11}\mathrm{true}_{g'_1}$ then else $\varepsilon_{12}t^{U_{12}}\varepsilon_{13}t^{U_{13}}$ then by ($\sqsubseteq_{if}$) $t_1^{U_2}$ has the form
$t_1^{U_2} = \mathrm{if}^{g_2}\ \varepsilon_{21}\mathrm{true}_{g'_2}$ then else $\varepsilon_{22}t^{U_{22}}\varepsilon_{23}t^{U_{23}}$ for some
$\varepsilon_{21}, \varepsilon_{22}, t^{U_22}, \varepsilon_{23}$, and $t^{U_{32}}$. Then
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} \mathrm{prot}^{g_1,(U_{12} \mathbin{\widetilde{\curlyvee}} U_{13})}_{ilbl(\varepsilon_{11})g'_1} \phi'_1(\varepsilon_{12}t^{U_{12}}) \mid \mu_1$, and
$t_1^{U_2} \mid \mu 2 \xrightarrow{\phi_2} \mathrm{prot}^{g_2,(U_{22} \mathbin{\widetilde{\curlyvee}} U_{23})}_{ilbl(\varepsilon_{21})g'_2} \phi'_2(\varepsilon_{22}t^{U_{22}}) \mid \mu_2$.

Where $\phi'_i = \langle(\phi_i.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{i2})(g'_i \mathbin{\widetilde{\curlyvee}} \phi_i.g_c), \phi_i.g_c \mathbin{\widetilde{\curlyvee}} g_i]\rangle$. Using the fact that $t_1^{U_1} \sqsubseteq t_2^{U_2}$ we know that $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$, $t^{U_{12}} \sqsubseteq t^{U_{22}}$, $g'_1 \sqsubseteq g'_2$, as $\phi_1.g_c \sqsubseteq \phi_2.g_c$ and $g_1 \sqsubseteq g_2$, and as join is monotone, $\phi_1.g_c \mathbin{\widetilde{\curlyvee}} g_1 \sqsubseteq \phi_2.g_c \mathbin{\widetilde{\curlyvee}} g_2$. Also as $\phi_1.g_c \sqsubseteq \phi_2.g_c$ and $g'_1 \sqsubseteq g'_2$, and as join is monotone, $\phi_1.g_c \mathbin{\widetilde{\curlyvee}} g'_1 \sqsubseteq \phi_2.g_c \mathbin{\widetilde{\curlyvee}} g'_2$. By Prop 6.18, we know that $U_{12} \sqsubseteq U_{22}$ and $U_{13} \sqsubseteq U_{23}$. Therefore by Prop 6.28 $(U_{12} \mathbin{\widetilde{\curlyvee}} U_{13}) \sqsubseteq (U_{22} \mathbin{\widetilde{\curlyvee}} U_{23})$. Also as $\phi_1.\varepsilon \sqsubseteq \phi_2.\varepsilon$ and $ilbl(\varepsilon_{12}) \sqsubseteq ilbl(\varepsilon_{22})$ then by Lemma 6.26 $(\phi_1.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{12})) \sqsubseteq (\phi_2.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{22}))$. Then using ($\sqsubseteq_{\mathrm{prot}()}$), $t_2^{U_1} \sqsubseteq t_2^{U_2}$. As $\Omega' = \Omega$, $\mu'_1 = \mu_1$ and $\mu_2 = \mu'_2$ then $\Omega' \vdash \mu'_1 \sqsubseteq \mu'_2$.

Case ($\longrightarrow$if-false). Same as case $\longrightarrow$if-true, using the fact that $\varepsilon_{13} \sqsubseteq \varepsilon_{23}$ and $t^{U_{13}} \sqsubseteq t^{U_{23}}$.

Case ($\longrightarrow$ref). We know that $t_1^{U_1} = \mathrm{ref}^{U''_1}_{\varepsilon_{\ell 1}}\ \varepsilon_1 u_1$, then by ($\sqsubseteq_{ref}$) $t_1^{U_2} = \mathrm{ref}^{U''_2}_{\varepsilon_{\ell 2}}\ \varepsilon_2 u_2$, and therefore

$$\frac{\begin{array}{ccc} U''_1 \sqsubseteq U''_2 & \varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2} & g_{c1} \sqsubseteq g_{c2} \\ \varepsilon_1 \sqsubseteq \varepsilon_2 & \Omega \vdash u_1 \sqsubseteq u_2 & \end{array}}{\Omega \vdash \mathrm{ref}^{U''_1}_{\varepsilon_{\ell 1}}\ \varepsilon_1 u_1 \sqsubseteq \mathrm{ref}^{U''_2}_{\varepsilon_{\ell 2}}\ \varepsilon_2 u_2}$$

for some $\varepsilon_2, u_2, U''_2$ and $\varepsilon_{\ell 2}$, where $u_1 \in \mathbb{T}[U'_1]$ and $u_2 \in \mathbb{T}[U'_2]$. If
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} o_\perp^{U''_1} \mid \mu_1[l^{U''_1} \mapsto v'_1]$, for some $l^{U''_1} \notin \mu_1$ and where $v'_1 = \varepsilon'_1(u_1 \mathbin{\widetilde{\curlyvee}} g_{r1}) :: U''_1$, $\varepsilon'_1 = \varepsilon_1 \mathbin{\widetilde{\curlyvee}} (\phi_1.\varepsilon \circ^{\preccurlyeq} \varepsilon_{\ell 1})$. Therefore, $t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} o_\perp^{U''_2} \mid \mu_2[l^{U''_2} \mapsto v'_2]$, for some $l^{U''_2} \notin \mu_2$ and where $v'_2 = \varepsilon'_2(u_2 \mathbin{\widetilde{\curlyvee}} g_{r2}) :: U''_2$, $\varepsilon'_2 = \varepsilon_2 \mathbin{\widetilde{\curlyvee}} (\phi_2.\varepsilon \circ^{\preccurlyeq} \varepsilon_{\ell 2})$. By Lemma 6.26 and 6.24, $\varepsilon'_1 \sqsubseteq \varepsilon'_2$. Also as $\phi_1.\varepsilon \sqsubseteq \phi_2.\varepsilon$ and $U_1 \sqsubseteq U_2$, then by definition of rf, $\varepsilon''_1 \sqsubseteq \varepsilon''_2$. Then using $\Omega' = \Omega \cup \{\, l^{U''_1} \sqsubseteq l^{U''_2} \,\}$ and that $\perp \sqsubseteq \perp$, by ($\sqsubseteq_l$) we can see that $\Omega' \vdash l_\perp^{U''_1} \sqsubseteq l_\perp^{U''_2}$. As $g_{r1} \sqsubseteq g_{r2}$, by monotonicity of the join, $u_1 \mathbin{\widetilde{\curlyvee}} g_{r1} \sqsubseteq u_2 \mathbin{\widetilde{\curlyvee}} g_{r2}$. Therefore using $\sqsubseteq_{::}$, $\Omega' \vdash v'_1 \sqsubseteq v'_2$. Also because $\Omega \subseteq \Omega'$, then by the fact that $\Omega \vdash \mu_1 \sqsubseteq \mu_2$, it is easy to see that $\Omega \cup \{\, l^{U''_1} \sqsubseteq l^{U''_2} \,\} \vdash \mu_1[l^{U''_1} \mapsto v'_1] \sqsubseteq \mu_2[l^{U''_2} \mapsto v'_2]$, i.e. $\Omega' \vdash \mu'_1 \sqsubseteq \mu'_2$.

Case ($\longrightarrow$deref). We know that $t_1^{U_1} = !^{\mathrm{Ref}_{g_1}\ U'_1}\varepsilon_1 l^{U''_1}_{g'_1}$, $t_1^{U_2} = !^{\mathrm{Ref}_{g_2}\ U'_2}\varepsilon_2 l^{U''_2}_{g'_2}$ and so
$\Omega \vdash !^{\mathrm{Ref}_{g_1}\ U'_1}\varepsilon_1 l^{U''_1}_{g'_1} \sqsubseteq !^{\mathrm{Ref}_{g_2}\ U'_2}\varepsilon_2 l^{U''_2}_{g'_2}$. As $\Omega \vdash \mu_1 \sqsubseteq \mu_2$, using ($\sqsubseteq_\mu$) then $\Omega \vdash \mu_1(l^{U''_1}) \sqsubseteq \mu_2(l^{U''_2})$. Then
$!^{\mathrm{Ref}_{g_1}\ U'_1}\varepsilon_1 l^{U''_1}_{g'_1} \mid \mu \xrightarrow{\phi_1} \mathrm{prot}^{g_1, U'_1}_{\varepsilon'_1 g'_1} \phi'_1(iref(\varepsilon_1)\mu_1(o^{U''_1}))$ where $\varepsilon'_1 = ilbl(\varepsilon_1)$. Therefore
$!^{\mathrm{Ref}_{g_2}\ U'_2}\varepsilon_2 l^{U''_2}_{g'_2} \mid \mu \xrightarrow{\phi_2} \mathrm{prot}^{g_2, U'_2}_{\varepsilon'_2 g'_2} \phi'_2(iref(\varepsilon_2)\mu_2(o^{U''_2}))$ where $\varepsilon'_2 = ilbl(\varepsilon_2)$.

Where $\phi_i' = \langle(\phi_i.\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_i')(\phi_i.g_c \mathbin{\widetilde{\curlyvee}} g_i'), \phi_i.g_c \mathbin{\widetilde{\curlyvee}} g_i\rangle$. By monotonicity of the join $\phi_1.g_c \mathbin{\widetilde{\curlyvee}} g_1 \sqsubseteq \phi_2.g_c \mathbin{\widetilde{\curlyvee}} g_2$, $\phi_1.g_c \mathbin{\widetilde{\curlyvee}} g_1' \sqsubseteq \phi_2.g_c \mathbin{\widetilde{\curlyvee}} g_2'$ and $(\phi_1.\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_1') \sqsubseteq (\phi_2.\varepsilon \mathbin{\widetilde{\curlyvee}} \varepsilon_2')$. As $\varepsilon_1 \sqsubseteq \varepsilon_2$, then by Lemma 6.29, $iref(\varepsilon_1) \sqsubseteq iref(\varepsilon_2)$. Then Using ($\sqsubseteq_{\mathsf{prot}()}$) we can conclude that $\Omega \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$. As $\Omega' = \Omega$, $\mu_1 = \mu_1'$ and $\mu_2 = \mu_2'$ then also $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$.

Case ($\longrightarrow$assign). We know that $t_1^{U_1} = \varepsilon_{11} l_{g_1}^{U_{11}} \overset{g_1, U_1'}{:=}_{\varepsilon_{\ell 1}} \varepsilon_{12} u_1$, $t_1^{U_2} = \varepsilon_{21} l_{g_2}^{U_{21}} \overset{g_2, U_2'}{:=}_{\varepsilon_{\ell 2}} \varepsilon_{22} u_2$ and so $\Omega \vdash \varepsilon_{11} l_{g_1}^{U_{11}} \overset{g_1, U_1'}{:=}_{\varepsilon_{\ell 1}} \varepsilon_{12} u_1 \sqsubseteq \varepsilon_{21} l_{g_2}^{U_{21}} \overset{g_2, U_2'}{:=}_{\varepsilon_{\ell 2}} \varepsilon_{22} u_2$. Then
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} \mathsf{unit}_\perp \mid \mu_1[l^{U_{11}} \mapsto v_1]$, where $v_1 = \varepsilon_1'(u_1 \mathbin{\widetilde{\curlyvee}} (g_{r1} \mathbin{\widetilde{\curlyvee}} g_1)) :: U_{11}$, and $\varepsilon_1' = (\varepsilon_{12} \circ^{<:} iref(\varepsilon_{11})) \mathbin{\widetilde{\curlyvee}} ((\phi_1.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{11})) \circ^{\preccurlyeq} \varepsilon_{\ell 1} \circ^{\preccurlyeq} ilbl(iref(\varepsilon_{11}))$. Similarly, then
$t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} \mathsf{unit}_\perp \mid \mu_2[l^{U_{21}} \mapsto v_2]$, where $v_2 = \varepsilon_2'(u_2 \mathbin{\widetilde{\curlyvee}} (g_{r2} \mathbin{\widetilde{\curlyvee}} g_2)) :: U_{21}$, and $\varepsilon_2' = (\varepsilon_{22} \circ^{<:} iref(\varepsilon_{21})) \mathbin{\widetilde{\curlyvee}} ((\phi_2.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{21})) \circ^{\preccurlyeq} \varepsilon_{\ell 2} \circ^{\preccurlyeq} ilbl(iref(\varepsilon_{21}))$. We need to prove that $\mu_1' = \mu_1[l^{U_{11}} \mapsto v_1] \sqsubseteq \mu_2' = \mu_2[l^{U_{21}} \mapsto v_2]$. Because $\Omega \vdash \mu_1 \sqsubseteq \mu_2$ then $\Omega \vdash l^{U_{11}} \sqsubseteq l^{U_{21}}$ by ($\sqsubseteq_\mu$). By well formedness of Ω we also know that $U_{11} \sqsubseteq U_{21}$. Therefore, by Lemmas 6.24, 6.25 and 6.26 $\varepsilon_1' \sqsubseteq \varepsilon_2'$. Then using $\sqsubseteq_{::}$, $v_1 \sqsubseteq v_2$, following that $\Omega' = \Omega \vdash \mu_1' \sqsubseteq \mu_2'$.

□

Proposition 6.31 (Dynamic guarantee). *Suppose $t_1^{U_1} \sqsubseteq t_1^{U_2}$, $\phi_1 \sqsubseteq \phi_2$, and $\mu_1 \sqsubseteq \mu_2$. If $t_1^{U_1} \mid \mu_1 \overset{\phi_1}{\longmapsto} t_2^{U_1} \mid \mu_1'$ then $t_1^{U_2} \mid \mu_2 \overset{\phi_2}{\longmapsto} t_2^{U_2} \mid \mu_2'$ where $t_2^{U_1} \sqsubseteq t_2^{U_2}$ and $\mu_1' \sqsubseteq \mu_2'$.*

Proof. We prove the following property instead: Suppose $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$, $\phi_1 \sqsubseteq \phi_2$, and $\Omega \vdash \mu_1 \sqsubseteq \mu_2$. If $t_1^{U_1} \mid \mu_1 \overset{\phi_1}{\longmapsto} t_2^{U_1} \mid \mu_1'$ then $t_1^{U_2} \mid \mu_2 \overset{\phi_2}{\longmapsto} t_2^{U_2} \mid \mu_2'$ where $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$ and $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$, for some $\Omega' \supseteq \Omega$.

By induction on the structure of a derivation of $t_1^{U_1} \sqsubseteq t_1^{U_2}$. For simplicity we omit the $\Omega \vdash$ notation on precision relations when it is not relevant for the argument.

Case (R$\longrightarrow$). $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$, $\Omega \vdash \mu_1 \sqsubseteq \mu_2$ and
$t_1^{U_1} \mid \mu_1 \xrightarrow{\phi_1} t_2^{U_1} \mid \mu_1'$. By dynamic guarantee of $\longrightarrow$ (Prop 6.30), $t_1^{U_2} \mid \mu_2 \xrightarrow{\phi_2} t_1^{U_2} \mid \mu_2'$ where $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$, $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$ for some $\Omega' \supseteq \Omega$. And the result holds immediately.

Case (Rf). $t_1^{U_1} = f_1[t_1^{U_1'}]$, $t_1^{U_2} = f_2[t_1^{U_2'}]$. We know that $\Omega \vdash f_1[t_1^{U_1'}] \sqsubseteq f_2[t_1^{U_2'}]$. By using Prop 6.18, $U_1' \sqsubseteq U_2'$. By Prop 6.22, we also know that $\Omega \vdash t_1^{U_1'} \sqsubseteq t_1^{U_2'}$. By induction hypothesis, $t_1^{U_1'} \mid \mu_1 \overset{\phi_1}{\longmapsto} t_2^{U_1'} \mid \mu_1'$, $t_1^{U_2'} \mid \mu_2 \overset{\phi_2}{\longmapsto} t_2^{U_2'} \mid \mu_2'$, $\Omega' \vdash t_2^{U_1'} \sqsubseteq t_2^{U_2'}$ and $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$ for some $\Omega' \supseteq \Omega$.
Then by Prop 6.21 then $\Omega' \vdash f_1[t_2^{U_1'}] \sqsubseteq f_2[t_2^{U_2'}]$ and the result holds.

Case (Rprot). Then $t_1^{U_1} = \mathsf{prot}^{g_1, U_1}_{\varepsilon_{\ell 1} g_1'} \phi_1'(\varepsilon_1 t_1^{U_1'})$ and $t_1^{U_2} = \mathsf{prot}^{g_2, U_2}_{\varepsilon_{\ell 2} g_2'} \phi_2'(\varepsilon_2 t_1^{U_2'})$

As $t_1^{U_1} \sqsubseteq t_1^{U_2}$ then by ($\sqsubseteq_{\mathsf{prot}()}$), $t_1^{U_1'} \sqsubseteq t_1^{U_2'}$, $\phi_1 \sqsubseteq \phi_2$, $\varepsilon_{\ell 1} \sqsubseteq \varepsilon_{\ell 2}$, $g_1 \sqsubseteq g_2$, $g_1' \sqsubseteq g_2'$, and $\varepsilon_1 \sqsubseteq \varepsilon_2$. By (Rprot), $t_1^{U_i'} \mid \mu \overset{\phi_i'}{\longmapsto} t_2^{U_i'} \mid \mu'$ and by induction hypothesis, $t_2^{U_1'} \sqsubseteq t_2^{U_2'}$ and $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$ for some $\Omega' \supseteq \Omega$.

But then by ($\sqsubseteq_{\mathsf{prot}()}$),
$\Omega' \vdash \mathsf{prot}^{g_1, U_1}_{\varepsilon_{\ell 1} g_1'} \phi_1'(\varepsilon_1 t_2^{U_1'}) \sqsubseteq \mathsf{prot}^{g_2, U_2}_{\varepsilon_{\ell 2} g_2'} \phi_2'(\varepsilon_2 t_2^{U_2'})$ and the result holds.

Case (Rg). $t_1^{U_1} = g_1[et_1]$, $t_1^{U_2} = g_2[et_2]$, where $\Omega \vdash g_1[et_1] \sqsubseteq g_2[et_2]$. Also $et_1 \longrightarrow_c et_1'$ and $et_2 \longrightarrow_c et_2'$.

Then there exists U_1, ε_{11}, ε_{12} and v_1 such that $et_1 = \varepsilon_{11}(\varepsilon_{12}v_1 :: U_1)$. Also there exists U_2, ε_{21}, ε_{22} and v_2 such that $et_2 = \varepsilon_{21}(\varepsilon_{22}v_2 :: U_2)$. By Prop 6.20, $\varepsilon_{11} \sqsubseteq \varepsilon_{21}$, and by ($\sqsubseteq_{::}$) $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$, $v_1 \sqsubseteq v_2$ and $U_1 \sqsubseteq U_2$. Then as $et_1 \longrightarrow_c (\varepsilon_{12} \circ^{<:} \varepsilon_{11})v_1$ and $et_2 \longrightarrow_c (\varepsilon_{22} \circ^{<:} \varepsilon_{21})v_2$ then, by Prop 6.24 we know that $\varepsilon_{12} \circ^{<:} \varepsilon_{11} \sqsubseteq \varepsilon_{22} \circ^{<:} \varepsilon_{21}$. Then using this information, and the fact that $v_1 \sqsubseteq v_2$, by Prop 6.19, it follows that $\Omega \vdash g_1[et_1'] \sqsubseteq g_1[et_2']$. As $\Omega' = \Omega$, $\mu_1' = \mu_1$ and $\mu_2 = \mu_2'$ then $\Omega' \vdash \mu_1' \sqsubseteq \mu_2'$.

Case (Rprotg). Analogous to (Rprot) case but using $\longrightarrow_c$ instead.

□

6.6 Noninterference

In this section we present the proof of noninterference for $\mathrm{GSL}_{\mathsf{Ref}}$. We use a logical relation that is more general than the one presented in the paper. The main difference (beside using intrinsic terms), is that the logical relation is no longer indexed by a static security effect. As ϕ embeds the static security effect information, we generalize the logical relation to also relate two different static security effects as well. Section 6.6.1 present some auxiliary definitions. Section 6.6.2 presents the proof of Noninterference (Prop 6.64), which implies Security Type Soundness (Prop 2.24) presented in the paper.

6.6.1 Definitions. We introduce a function *uval*, which strips away ascriptions from a simple value:

$$uval : \text{GType} \rightarrow \textsc{SimpleValue}$$
$$uval(u) = u$$
$$uval(\varepsilon u :: U) = u.$$

In order to compare the observable results of program, we introduce the $rval(v)$ operator, which strips away any checking-related information like labels or evidence-carrying ascriptions:

$$rval : \textsc{Value} \rightarrow \textsc{RawValue}$$
$$rval(b_g) = b$$
$$rval(\varepsilon b_g :: U) = b$$
$$rval(\mathsf{unit}_g) = \mathsf{unit}$$
$$rval(\varepsilon \mathsf{unit}_g :: U) = \mathsf{unit}$$
$$rval(o_g^U) = o$$
$$rval(\varepsilon o_g^{U'} :: U) = o$$
$$rval((\lambda^{g'} x^{U_1}.t^{U_2})_g) = (\lambda^{g'} x^{U_1}.t^{U_2})$$
$$rval(\varepsilon (\lambda^{g'} x^{U_1}.t^{U_2})_g :: U) = (\lambda^{g'} x^{U_1}.t^{U_2})$$

Definition 6.32 (Gradual security logical relations). For an arbitrary element ℓ_o of the security lattice, the ℓ_o-level gradual security relations are step-indexed and type-indexed binary relations on tuples of security effect, closed terms and stores defined inductively as presented in Figure 30. The notation $\langle\phi_1, v_1, \mu_1\rangle \approx^k_{\ell_o} \langle\phi_2, v_2, \mu_2\rangle : U$ indicates that the tuple of security effect ϕ_1, value v_1 and store μ_1 is related to the tuple of security effect ϕ_2, value v_2 and store μ_2 at type U for k steps when observed at the security level ℓ_o. Similarly, the notation $\langle\phi_{\approx_{\ell_o}}, t_{\approx_{\ell_o}}, \mu_{\approx_{\ell_o}}\rangle^k \langle\phi_{:}, t_{:}, \mu_{:}\rangle\, \mathcal{C}(U)$ indicates that the tuple of security effect ϕ_1, term t_1 and store μ_1, and the tuple of security effect ϕ_2, term t_2 and store μ_2 are related computations for k steps, that produce related values and related stores at type U when observed at the security level ℓ_o. Notation $\mu_1 \approx^k_{\ell_o} \mu_1$ relates stores μ_1 and μ_2 for k steps when observed at security level ℓ_o. Finally, notation $\phi_1 \approx_{\ell_o} \phi_2$, relates security effect ϕ_1 and ϕ_2 for any number of steps at security level ℓ_o.

We say that a value is *observable* at level ℓ_o if, given a security effect ϕ, the value is typeable, the security effect is observable, and the label of the value is sublabel of ℓ_o. Also, as value v can be a casted value, we need to analyze if its underlying evidence justifies that the security level of the bare value is also subsumed by the observer security level. We do this by demanding that the underlying evidence and label is also observable. We say that a security effect is observable if its underlying evidence and static label is also observable. We say that an evidence and label

$$\langle \phi_1, v_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, v_2, \mu_2\rangle : U \iff \phi_1 \approx_{\ell_o} \phi_2 \wedge \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \phi_i \triangleright v_i \in \mathbb{T}[U] \wedge \big(\mathsf{obs}_{\ell_o}(\phi_i \triangleright v_i) \vee \neg \mathsf{obs}_{\ell_o}(\phi_i \triangleright v_i)\big) \wedge \big(\mathsf{obs}_{\ell_o}(\phi_i \triangleright v_i) \implies \mathsf{obsRel}^U_{k,\ell_o}(\phi_1, v_1, \mu_1, \phi_2, v_2, \mu_2)\big)$$

$$\mathsf{obsRel}^U_{k,\ell_o}(\phi_1, v_1, \mu_1, \phi_2, v_2, \mu_2) \iff (rval(v_1) = rval(v_2)) \qquad \text{if } U \in \{\mathsf{Bool}_g, \mathsf{Unit}_g, \mathsf{Ref}_g\ U'\}$$

$$\mathsf{obsRel}^{U_1 \xrightarrow{g_1'}_{g_1} U_2}_{k,\ell_o}(\phi_1, v_1, \mu_1, \phi_2, v_2, \mu_2) \iff \forall j \le k.\, \forall U' = U_1'' \xrightarrow{g_2''}_{g_2'} U_2'', U_1', \forall \phi_i', \text{ s.t. } \phi_i \le_{\ell_o} \phi_i', \varepsilon_1' \vdash U_1 \xrightarrow{g_1'}_{g_1} U_2 \lesssim U', \text{ and } \varepsilon_2' \vdash U_1' \lesssim U_1'', \varepsilon_i'' \vdash \widetilde{\phi_i' g_c \curlyvee g_2' \preccurlyeq g_2''}, \text{ we have: } \forall v_i', \mu_i', \langle \phi_1, v_1', \mu_1'\rangle \approx^j_{\ell_o} \langle \phi_2, v_2', \mu_2'\rangle : U_1', dom(\mu_i) \subseteq dom(\mu_i'), \langle \phi_1, (\varepsilon_1' v_1 @^{U'}_{\varepsilon_1''} \varepsilon_2' v_1')_1, \mu_1'\rangle \approx^j_{\ell_o} \langle \phi_2, (\varepsilon_1' v_2 @^{U'}_{\varepsilon_2''} \varepsilon_2' v_2')_2, \mu_2'\rangle : \mathcal{C}(U_2'' \,\widetilde{\curlyvee}\, g_2')$$

$$\langle \phi_1, t_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, t_2, \mu_2\rangle : \mathcal{C}(U) \iff \phi_1 \approx_{\ell_o} \phi_2 \wedge \mu_1 \approx^k_{\ell_o} \mu_2 \wedge \forall \phi_i', \text{ s.t. } \phi_i \le \phi_i' \text{ and } \phi_i' \triangleright t_i \in \mathbb{T}[U] \text{ we have } \forall j < k \; \big(t_i \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^j t_i' \mid \mu_i' \implies \mu_1' \approx^{k-j}_{\ell_o} \mu_2' \wedge (irred(t_i') \implies \langle \phi_1, t_1', \mu_1'\rangle \approx^{k-j}_{\ell_o} \langle \phi_2, t_2', \mu_2'\rangle : U)\big)$$

$$\begin{aligned}
\mu_1 \approx^k_{\ell_o} \mu_2 &\iff \forall \phi_i, \phi_1 \approx_{\ell_o} \phi_2, j < k, \forall o^U \in dom(\mu_1) \cap dom(\mu_2) \\
&\qquad \langle \phi_1 \triangleright \mu_1(o^U), \mu_1\rangle \approx^j_{\ell_o} \langle \phi_2 \triangleright \mu_2(o^U), \mu_2\rangle : U \\
\phi_1 \approx_{\ell_o} \phi_2 &\iff \mathsf{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c) \vee \neg \mathsf{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c) \\
\phi_1 \le_{\ell_o} \phi_2 &\iff \mathsf{obs}_{\ell_o}(\phi_2.\varepsilon\phi_2.g_c) \Rightarrow \mathsf{obs}_{\ell_o}(\phi_1.\varepsilon\phi_1.g_c) \\
\mu_1 \twoheadrightarrow \mu_2 &\iff dom(\mu_1) \subseteq dom(\mu_2) \\
\mathsf{obs}_{\ell_o}(\phi \triangleright v) &\iff \phi \triangleright v \in \mathbb{T}[U] \wedge \mathsf{obs}_{\ell_o}(\phi) \wedge \widetilde{label}(U) \,\widetilde{\preccurlyeq}\, \ell_o \wedge \Big((v = \varepsilon u :: U) \implies \mathsf{obs}_{\ell_o}(ilbl(\varepsilon)\widetilde{label}(U))\Big) \\
\mathsf{obs}_{\ell_o}(\phi) &\iff \mathsf{obs}_{\ell_o}(\phi.\varepsilon\phi.g_c) \\
\mathsf{obs}_{\ell_o}(\varepsilon g) = &\iff \varepsilon \circ^{\preccurlyeq} \varepsilon' \text{ is defined, where } \varepsilon' = \mathcal{I}_{\preccurlyeq}(g, \ell_o)
\end{aligned}$$

Fig. 30. Gradual security logical relations

are observable, if any value with that underlying evidence and static label, can be used used as argument of a function that expects a value with security level ℓ_o. If the consistent transitivity check of the reduction of the application does not hold, then it is not plausible that the security level of the value is subsumed by ℓ_o, and therefore is *not* observable. For instance, consider $\ell_o = \mathsf{L}$, evidence $\varepsilon = \langle [\mathsf{H}, \top], [\bot, \top]\rangle$ and static label $g = ?$. We can construct any value such as $v = \varepsilon\mathsf{true}_? :: \mathsf{Bool}_g$. The level of the value and the bare value are sublabel of ℓ_o. But the evidence describes that at some point during reduction, the security level of the bare value was required to be at least as high as H. Therefore, v is not observable at level L (considering $\mathsf{L} \preccurlyeq \mathsf{H}$), because as $\mathcal{I}_{\preccurlyeq}(?, \ell_o) = \langle [\bot, \mathsf{L}], [\mathsf{L}, \mathsf{L}]\rangle$, the consistent transitivity operation $\langle [\mathsf{H}, \top], [\bot, \top]\rangle \circ^{<:} \langle [\bot, \mathsf{L}], [\mathsf{L}, \mathsf{L}]\rangle$ does not hold.

Two stores are related at k steps if each value in the heap of the locations they have in common, are related at $j < k$ steps for any related security effects. We say that store μ_2 is the evolution of store μ_1, annotated $\mu_1 \twoheadrightarrow \mu_2$ if the domain of μ_1 is a subset of μ_2.

Two tuples of security effects, values and stores are related for k steps at type Bool_g if the security effects are related, the stores are related for k steps, the values can be typed as Bool_g using the security effects as context (any security effect will do, given that the typing of values do not depend on the security effect). Additionally, both security effect and values must both be either observable or not observable. If the security effect and values are observable then the raw values are the same. Two tuples are observables at type Unit_g and $\mathsf{Ref}_g\ U$ analogous to booleans.

Pairs are related at function types similarly to booleans. The difference is that functions can not be compared as booleans. Two functions are related if, given two related values and stores for $j \leq k$ steps at the argument type, the application of those function to the related values are also related for j steps at at the return type.

Two tuples of terms and stores are related computations for k steps at type U, first, if the security effects are related, and the stores are related for k steps. Second the terms must be typed as U using a observationally higher security effect. Third, if for any $j < k$ both terms can be reduced for at least j steps, then the resulting stores are related for the remaining $k - j$ steps Finally, if after at least j steps the resulting terms are irreducible, then the resulting terms are also related values for the remaining $k - j$ steps at type U. Notice that the logical relation also relates programs that do not terminate as long as after k steps the new stores are also related.

To define the fundamental property of the step-indexed logical relations we first define how to relate substitutions:

Definition 6.33. Let ρ be a substitution and Γ a type substitution. We say that substitution ρ satisfy environment Γ, written $\rho \models \Gamma$, if and only if $dom(\rho) = \Gamma$.

Definition 6.34 (Related substitutions). Tuples $\langle \phi_1, \rho_1, \mu_1 \rangle$ and $\langle \phi_2, \rho_2, \mu_2 \rangle$ are related on k steps under Γ, notation $\Gamma \vdash \langle \phi_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2, \mu_2 \rangle$, if $\rho_i \models \Gamma$, $\mu_1 \approx^k_{\ell_o} \mu_2$ and

$$\forall x^U \in \Gamma.\, \langle \phi_1, \rho_1(x^U), \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2(x^U), \mu_2 \rangle : U$$

6.6.2 *Proof of noninterference.*

Lemma 6.35 (Noninterference for booleans). *Suppose $k > 0$, and*

- *an open term $\phi \triangleright t^U \in \mathbb{T}[\mathsf{Bool}_{\ell_o}]$ where $FV(t) = \{ x^{U_1} \}$ with $label(U_1) \,\widetilde{\not\preccurlyeq}\, \ell_o$*
- *two compatible valid stores $t^U \vdash \mu_i$, $\mu_1 \approx^k_{\ell_o} \mu_2$*

Then for any $j < k$, $v_1, v_2 \in \mathbb{T}[U_1]$, if both

- $t^U[v_1/x^{U_1}] \mid \mu_1 \overset{\phi}{\longmapsto}{}^j v_1' \mid \mu_1'$
- $t^U[v_2/x^{U_1}] \mid \mu_2 \overset{\phi}{\longmapsto}{}^j v_2' \mid \mu_2'$

we have that $rval(v_1') = rval(v_2')$, and $\mu_1' \approx^k_{\ell_o} \mu_2'$.

Proof. The result follows as a special case of Proposition 6.64 below. □

In this theorem, we treat t^U as a program that takes x^{U_1} as its input. Furthermore, the security level $g' = \widetilde{label}(U_1)$ of the input is not subsumed by the security level ℓ_o of the observer. As such, noninterference dictates that changing non-observable input must not change the observable *value* of the output (*i.e.*, change true to false or vice-versa). However, this theorem is technically *termination-insensitive* in that it is vacuously true if a change of inputs changes a program that terminates with a value into one that either terminates with an **error**, or does not terminate at all. If a program does not terminate after any number of steps, then at least the stores are related at observation level ℓ_o.

Note that we compare equality of *raw* values at first-order type. Restricting attention to first-order types (*i.e.*, Bool) is common when investigating observational equivalence of typed languages. We strip away security information because a person or client who uses the program ultimately observes only the raw value that the program produces.

Also, gradual security *dynamically* traps some information leaks, so a change in equivalent inputs may cause a program that previously yielded a value or diverged to now produce an **error**. This change in behavior falls under the notion of *termination-insensitive*, since yielding an error is simply a third form of termination behavior (in addition to producing a value and diverging).

Finally, we use notation $t^S \mid \mu \overset{\phi}{\longmapsto}{}^{k} t'^S \mid \mu'$ to describe that configuration $t^S \mid \mu$ reduces, in at most k steps, to configuration $t'^S \mid \mu'$.

Lemma 6.36. *Consider* $\varepsilon_1 \vdash g \mathrel{\widetilde{\preccurlyeq}} g'$. *If* $\forall \varepsilon_2$ *such that* $\varepsilon_2 \vdash g' \mathrel{\widetilde{\preccurlyeq}} \ell_o$, $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 \vdash g \mathrel{\widetilde{\preccurlyeq}} \ell_o$ *is not defined. Then if* $\varepsilon_3 \vdash g' \mathrel{\widetilde{\preccurlyeq}} g''$, *then* $\forall \varepsilon_4$ *such that* $\varepsilon_4 \vdash g'' \mathrel{\widetilde{\preccurlyeq}} \ell_o$,
then $(\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3) \circ^{\preccurlyeq} \varepsilon_4 \vdash g \mathrel{\widetilde{\preccurlyeq}} \ell_o$ *is not defined*

Proof. Applying associativity: $(\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3) \circ^{\preccurlyeq} \varepsilon_4 = \varepsilon_1 \circ^{\preccurlyeq} (\varepsilon_3 \circ^{\preccurlyeq} \varepsilon_4)$, but $(\varepsilon_3 \circ^{\preccurlyeq} \varepsilon_4) \vdash g' \mathrel{\widetilde{\preccurlyeq}} g_o$, and we know that $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ is not defined $\forall \varepsilon_2$ such that $\varepsilon_2 \vdash g' \mathrel{\widetilde{\preccurlyeq}} \ell_o$. Therefore $(\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3) \circ^{\preccurlyeq} \varepsilon_4 \vdash g \mathrel{\widetilde{\preccurlyeq}} \ell_o$ is not defined and the result holds. □

Lemma 6.37. *Consider* $\varepsilon_1 \vdash g \mathrel{\widetilde{\preccurlyeq}} g'$. *If* $\forall \varepsilon_2$ *such that* $\varepsilon_2 \vdash g' \mathrel{\widetilde{\preccurlyeq}} \ell_o$, $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 \vdash g \mathrel{\widetilde{\preccurlyeq}} \ell_o$ *is not defined. Also* $\varepsilon_0 \vdash g_1 \mathrel{\widetilde{\preccurlyeq}} g_2$, *if* $\varepsilon_3 \vdash \widetilde{g_2 \curlyvee g' \preccurlyeq \ell_o}$, *then* $(\varepsilon_0 \mathbin{\widetilde{\curlyvee}} \varepsilon_1) \circ^{\preccurlyeq} \varepsilon_3 \vdash \widetilde{g_1 \curlyvee g \preccurlyeq \ell_o}$ *is not defined*

Proof. Let us prove that if $(\varepsilon_0 \mathbin{\widetilde{\curlyvee}} \varepsilon_1) \circ^{\preccurlyeq} \varepsilon_3 \vdash \widetilde{g_1 \curlyvee g \preccurlyeq \ell_o}$ is defined, then $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ is defined.

As join is monotone $\exists \varepsilon_0'$ such that $\varepsilon_0' \vdash \widetilde{g' \preccurlyeq g_2 \curlyvee g'}$.

Suppose $\varepsilon_1 = \langle [\ell_{11}, \ell_{12}], [\ell_{21}, \ell_{22}] \rangle$, $\varepsilon_0 = \langle [\ell_{31}, \ell_{32}], [\ell_{41}, \ell_{42}] \rangle$, $\varepsilon_0' = \langle [\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}] \rangle$, and $\varepsilon_3 = \langle [\ell_{71}, \ell_{72}], [\ell_{81}, \ell_{82}] \rangle$.
As $\varepsilon_0 \mathbin{\widetilde{\curlyvee}} \varepsilon_1 = \langle [\ell_{11} \curlyvee \ell_{31}, \ell_{12} \curlyvee \ell_{32}], [\ell_{21} \curlyvee \ell_{41}, \ell_{22} \curlyvee \ell_{42}] \rangle$ is defined, then $\ell_{11} \curlyvee \ell_{31} \preccurlyeq \ell_{12} \curlyvee \ell_{32}$ and $\ell_{21} \curlyvee \ell_{41} \preccurlyeq \ell_{22} \curlyvee \ell_{42}$. Also as

$$
\begin{aligned}
(\varepsilon_0 \mathbin{\widetilde{\curlyvee}} \varepsilon_1) \circ^{\preccurlyeq} \varepsilon_3 = \langle &[\ell_{11} \curlyvee \ell_{31}, (\ell_{12} \curlyvee \ell_{32}) \curlywedge ((\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72}) \curlywedge \ell_{82}], \\
&[\ell_{11} \curlyvee \ell_{31} \curlyvee \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{72} \curlyvee \ell_{81}, \ell_{82}] \rangle
\end{aligned}
$$

is defined then $\ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq (\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72}$, $\ell_{11} \curlyvee \ell_{31} \preccurlyeq (\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72}$, $\ell_{11} \curlyvee \ell_{31} \preccurlyeq \ell_{82}$, and $\ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq \ell_{82}$.

If we choose ε_0' as the interior of the judgment, then we do not get new information, therefore $[\ell_{21}, \ell_{22}] \sqsubseteq [\ell_{51}, \ell_{52}]$, *i.e.* $\ell_{51} \preccurlyeq \ell_{21}$ and $\ell_{22} \preccurlyeq \ell_{52}$. Using the same argument $\ell_{61} \preccurlyeq \ell_{71}$ and $\ell_{72} \preccurlyeq \ell_{62}$.

Then

$$
\begin{aligned}
& \varepsilon_0' \circ^{\preccurlyeq} \varepsilon_3 \\
= \; & \triangle^{\preccurlyeq}([\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}] \sqcap [\ell_{71}, \ell_{72}], [\ell_{81}, \ell_{82}]) \\
= \; & \triangle^{\preccurlyeq}([\ell_{51}, \ell_{52}], [\ell_{61} \curlyvee \ell_{71}, \ell_{62} \curlywedge \ell_{72}], [\ell_{81}, \ell_{82}]) \\
= \; & \triangle^{\preccurlyeq}([\ell_{51}, \ell_{52}], [\ell_{71}, \ell_{72}], [\ell_{81}, \ell_{82}])
\end{aligned}
$$

which is defined if $\ell_{51} \preccurlyeq \ell_{72}$, $\ell_{71} \preccurlyeq \ell_{82}$ and $\ell_{51} \preccurlyeq \ell_{82}$. But $\ell_{51} \preccurlyeq \ell_{21} \preccurlyeq \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq (\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72} \preccurlyeq \ell_{72}$, $\ell_{51} \preccurlyeq \ell_{21} \preccurlyeq \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq \ell_{82}$ and $\ell_{71} \preccurlyeq \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq \ell_{82}$.

Therefore

$$\varepsilon_0' \circ^{\preccurlyeq} \varepsilon_3 = \langle [\ell_{51}, \ell_{52} \curlywedge \ell_{72} \curlywedge \ell_{82}], [\ell_{51} \curlyvee \ell_{71} \curlyvee \ell_{81}, \ell_{82}] \rangle$$

Using the same method, $\varepsilon_1 \circ^{\preccurlyeq} (\varepsilon_0' \circ^{\preccurlyeq} \varepsilon_3)$ is defined if $\ell_{21} \curlyvee \ell_{51} \preccurlyeq \ell_{22} \curlywedge (\ell_{52} \curlywedge \ell_{72} \curlywedge \ell_{82})$, $\ell_{11} \preccurlyeq \ell_{22} \curlywedge (\ell_{52} \curlywedge \ell_{72} \curlywedge \ell_{82})$, and $\ell_{11} \preccurlyeq \ell_{82}$.

But by definition of ι $\ell_{21} \preccurlyeq \ell_{22}$, also $\ell_{21} \preccurlyeq \ell_{22} \preccurlyeq \ell_{52}$, $\ell_{21} \preccurlyeq \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq (\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72} \preccurlyeq \ell_{72}$, $\ell_{21} \preccurlyeq \ell_{21} \curlyvee \ell_{41} \curlyvee \ell_{71} \preccurlyeq \ell_{82}$, and $\ell_{51} \preccurlyeq \ell_{71} \preccurlyeq \ell_{72}$, therefore $\ell_{21} \curlyvee \ell_{51} \preccurlyeq \ell_{22} \curlywedge (\ell_{52} \curlywedge \ell_{72} \curlywedge \ell_{82})$.

Also $\ell_{11} \preccurlyeq \ell_{22} \preccurlyeq \ell_{52}$, $\ell_{11} \preccurlyeq \ell_{11} \curlyvee \ell_{31} \preccurlyeq (\ell_{22} \curlyvee \ell_{42}) \curlywedge \ell_{72} \preccurlyeq \ell_{72}$, and $\ell_{11} \preccurlyeq \ell_{11} \curlyvee \ell_{31} \preccurlyeq \ell_{82}$, therefore $\ell_{11} \preccurlyeq \ell_{22} \curlywedge (\ell_{52} \curlywedge \ell_{72} \curlywedge \ell_{82})$, and $\ell_{11} \preccurlyeq \ell_{82}$.

Then as $\varepsilon_1 \circ^{\preccurlyeq} (\varepsilon_0' \circ^{\preccurlyeq} \varepsilon_3)$ is defined then if we choose $\varepsilon_2 = (\varepsilon_0' \circ^{\preccurlyeq} \varepsilon_3) \vdash g' \preccurlyeq \ell_o$, the result holds. □

Lemma 6.38 (Associativity). *Consider $\varepsilon_1, \varepsilon_2$ and ε_3, such that $\varepsilon_1 \vdash g_1 \preccurlyeq g_2$, $\varepsilon_2 \vdash g_2 \preccurlyeq g_3$ and $\varepsilon_3 \vdash g_3 \preccurlyeq g_4$. $(\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2) \circ^{\preccurlyeq} \varepsilon_3 = \varepsilon_1 \circ^{\preccurlyeq} (\varepsilon_2 \circ^{\preccurlyeq} \varepsilon_3)$*

Proof. Suppose $\varepsilon_1 = \langle[\ell_{11}, \ell_{12}], [\ell_{21}, \ell_{22}]\rangle$, $\varepsilon_2 = \langle[\ell_{31}, \ell_{32}], [\ell_{41}, \ell_{42}]\rangle$, and $\varepsilon_3 = \langle[\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}]\rangle$ Then

$$\begin{array}{rl}
& (\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2) \circ^{\preccurlyeq} \varepsilon_3 \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12}], [\ell_{21}, \ell_{22}] \sqcap [\ell_{31}, \ell_{32}], [\ell_{41}, \ell_{42}]) \circ^{\preccurlyeq} \varepsilon_3 \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12}], [\ell_{21} \curlyvee \ell_{31}, \ell_{22} \curlywedge \ell_{32}], [\ell_{41}, \ell_{42}]) \circ^{\preccurlyeq} \varepsilon_3 \\
= & \langle[\ell_{11}, \ell_{12} \curlywedge (\ell_{22} \curlywedge \ell_{32}) \curlywedge \ell_{42}], [\ell_{11} \curlyvee (\ell_{21} \curlyvee \ell_{31}) \curlyvee \ell_{41}, \ell_{42}]\rangle \\
& \circ^{\preccurlyeq}\langle[\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}]\rangle \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12} \curlywedge (\ell_{22} \curlywedge \ell_{32}) \curlywedge \ell_{42}], [\ell_{11} \curlyvee (\ell_{21} \curlyvee \ell_{31}) \curlyvee \ell_{41}, \ell_{42}] \sqcap \\
& [\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}]) \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12} \curlywedge (\ell_{22} \curlywedge \ell_{32}) \curlywedge \ell_{42}], \\
& [\ell_{11} \curlyvee (\ell_{21} \curlyvee \ell_{31}) \curlyvee \ell_{41} \curlyvee \ell_{51}, \ell_{42} \curlywedge \ell_{52}], \\
& [\ell_{61}, \ell_{62}]) \\
= & \langle[\ell_{11}, \ell_{21}'], [\ell_{61}', \ell_{62}]\rangle
\end{array}$$

where $\ell_{21}' = \ell_{12} \curlywedge (\ell_{22} \curlywedge \ell_{32}) \curlywedge \ell_{42} \curlywedge \ell_{52} \curlywedge \ell_{62}$ and $\ell_{61}' = \ell_{11} \curlyvee (\ell_{21} \curlyvee \ell_{31}) \curlyvee \ell_{41} \curlyvee \ell_{51} \curlyvee \ell_{61}$. But

$$\begin{array}{rl}
& \varepsilon_1 \circ^{\preccurlyeq} (\varepsilon_2 \circ^{\preccurlyeq} \varepsilon_3) \\
= & \varepsilon_1 \circ^{\preccurlyeq} \triangle^{\preccurlyeq}([\ell_{31}, \ell_{32}], [\ell_{41}, \ell_{42}] \sqcap [\ell_{51}, \ell_{52}], [\ell_{61}, \ell_{62}]) \\
= & \varepsilon_1 \circ^{\preccurlyeq} \triangle^{\preccurlyeq}([\ell_{31}, \ell_{32}], [\ell_{41} \curlyvee \ell_{51}, \ell_{42} \curlywedge \ell_{52}], [\ell_{61}, \ell_{62}]) \\
= & \langle[\ell_{11}, \ell_{12}], [\ell_{21}, \ell_{22}]\rangle \circ^{\preccurlyeq} \\
& \langle[\ell_{31}, \ell_{32} \curlywedge (\ell_{42} \curlywedge \ell_{52}) \curlywedge \ell_{62}], [\ell_{31} \curlyvee (\ell_{41} \curlyvee \ell_{51}) \curlyvee \ell_{61}, \ell_{62}]\rangle \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12}], [\ell_{21}, \ell_{22}] \sqcap \\
& [\ell_{31}, \ell_{32} \curlywedge (\ell_{42} \curlywedge \ell_{52}) \curlywedge \ell_{62}], [\ell_{31} \curlyvee (\ell_{41} \curlyvee \ell_{51}) \curlyvee \ell_{61}, \ell_{62}]) \\
= & \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12}], \\
& [\ell_{21}, \ell_{22} \curlywedge (\ell_{32} \curlywedge \ell_{42}) \curlywedge \ell_{52} \curlywedge \ell_{62}], \\
& [\ell_{31} \curlyvee (\ell_{41} \curlyvee \ell_{51}) \curlyvee \ell_{61}, \ell_{62}] \\
= & \langle[\ell_{11}, \ell_{21}'], [\ell_{61}', \ell_{62}]\rangle
\end{array}$$

where $\ell_{21}' = \ell_{12} \curlywedge (\ell_{22} \curlywedge \ell_{32}) \curlywedge \ell_{42} \curlywedge \ell_{52} \curlywedge \ell_{62}$ and $\ell_{61}' = \ell_{11} \curlyvee (\ell_{21} \curlyvee \ell_{31}) \curlyvee \ell_{41} \curlyvee \ell_{51} \curlyvee \ell_{61}$, and the result holds. □

Lemma 6.39. *Consider $\varepsilon_1, \varepsilon_2$ and ε_3 such that $\varepsilon_1 \vdash g_1 \preccurlyeq g_2$, $\varepsilon_2 \vdash g_2 \preccurlyeq g_3$ and $\varepsilon_3 \vdash g_3 \preccurlyeq g_4$. If $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} (\varepsilon_2 \circ^{\preccurlyeq} \varepsilon_3)$ is defined, then $(\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2) \circ^{\preccurlyeq} (\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_3)$ is defined*

Proof. By definition of join and consistent transitivity, using the property that the join operator is monotone. □

Lemma 6.40. *If $\nexists\varepsilon_1$, such that $\varepsilon_1 \vdash g_1 \lesssim g_2$, then $\nexists\varepsilon_2$, such that $\varepsilon_2 \vdash \widetilde{g_1 \curlyvee g_3} \preccurlyeq g_2$.*

Proof. By definition of join and consistent transitivity, using the property that the join operator is monotone. □

Lemma 6.41. *Consider stores $\mu_1, \mu_2, \mu_1', \mu_2'$ such that $\mu_i \twoheadrightarrow \mu_i'$, and substitutions ρ_1 and ρ_2, such that $\Gamma \vdash \langle\phi_1, \rho_1, \mu_1\rangle \approx^k_{\ell_o} \langle\phi_2, \rho_2, \mu_2\rangle$, then if $\forall j \leq k$, if $\mu_1' \approx^j_{\ell_o} \mu_2'$ then $\Gamma \vdash \langle\phi_1, \rho_1, \mu_1'\rangle \approx^j_{\ell_o} \langle\phi_2, \rho_2, \mu_2'\rangle$*

Proof. By definition of related computations and related stores. The key argument is that given that $\mu_i \rightarrow \mu_i'$ then μ_i' have at least the same locations of μ_i and the values still are related as well given that they still have the same type. □

Lemma 6.42 (Substitution preserves typing). *If* $\phi \triangleright t^U \in \mathbb{T}[U]$ *and* $\rho \models FV(t^U)$ *then* $\phi \triangleright \rho(t^U) \in \mathbb{T}[U]$.

Proof. By induction on the derivation of $\phi \triangleright t^U \in \mathbb{T}[U]$ □

Lemma 6.43 (Reduction preserves relations). *Consider* $\phi_i \leq_{\ell_o} \phi_i'$, $\phi_i' \triangleright t_i \in \mathbb{T}[U]$, $\mu_i \in$ *Store*, $t_i \vdash \mu_i$, *and* $\mu_1 \approx^k_{\ell_o} \mu_2$. *Consider* $j < k$, *posing* $t_i \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^j t_i' \mid \mu_i'$, *we have*
$\langle \phi_1, t_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, t_2, \mu_2\rangle : \mathcal{C}(U)$ *if and only if* $\langle \phi_1, t_1', \mu_1'\rangle \approx^{k-j}_{\ell_o} \langle \phi_2, t_2', \mu_2'\rangle : \mathcal{C}(U)$

Proof. Direct by definition of
$\langle \phi_1, t_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, t_2, \mu_2\rangle : \mathcal{C}(U)$ and transitivity of $\overset{\phi'}{\longmapsto}{}^j$. □

Lemma 6.44 (Ascription preserves relation). *Suppose* $\varepsilon \vdash U' \lesssim U$.

(1) If $\langle \phi_1, v, \mu\rangle 1 \approx^k_{\ell_o} \langle \phi_2, v, \mu\rangle 2 : U'$ *then*
$\langle \phi_1, \varepsilon v_1 :: U, \mu_1\rangle \approx^{k+1}_{\ell_o} \langle \phi_2, \varepsilon v_2 :: U, \mu_2\rangle : \mathcal{C}(U)$.

(2) If $\langle \phi_1, t, \mu\rangle 1 \approx^k_{\ell_o} \langle \phi_2, t, \mu\rangle 2 : \mathcal{C}(U')$ *then*
$\langle \phi_1, \varepsilon t_1 :: U, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, \varepsilon t_2 :: U, \mu_2\rangle : \mathcal{C}(U)$.

Proof. Following Zdancewic [2002], the proof proceeds by induction on the judgment $\varepsilon \vdash U' \lesssim U$. The difference here is that consistent subtyping is justified by evidence, and that the terms have to be ascribed to exploit subtyping. In particular, case 1 above establishes a computation-level relation because each ascribed term ($\varepsilon v_i :: U$) may not be a value: each value v_i is either a bare value u_i or a casted value $\varepsilon_i u_i :: U_i$, with $\varepsilon_i \vdash S_i \lesssim U$. In the latter case, ($\varepsilon(\varepsilon_i u_i :: U_i) :: U$) either steps to **error** (in which case the relation is vacuously established), or steps to $\varepsilon' u_i :: U$, which is a value. Next if both values were originally observables, then whatever the label of U both values are going to be related. If both values were originally not observables, then by Lemma 6.44 both values are going to be still non observables.

□

Lemma 6.45. *If* $\langle \phi_1, v_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, v_2, \mu_2\rangle : U$ *and,* $\phi_i \triangleright uval(v_i) \in \mathbb{T}[U_i]$ *where* $U_i \lesssim U$, *then* $\forall U', U \lesssim U', \varepsilon_i \vdash U_i \lesssim U', \langle \phi_1, \varepsilon_1 uval(v_1) :: U', \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, \varepsilon_2 uval(v_2) :: U', \mu_1\rangle : U'$.

Proof. Consider U' and ε, such that $\varepsilon \vdash U \lesssim U'$. By Lemma 6.44.1, $\langle \phi_1, \varepsilon v_1 :: U', \mu_1\rangle \approx^{k+1}_{\ell_o} \langle \phi_2, \varepsilon v_2 :: U', \mu_2\rangle : \mathcal{C}(U)$. Next we consider the case were evidence combination do not fails. In case of a failure the lemma vacuously holds. Then as $\phi_i' \triangleright \varepsilon v_i :: U' \in \mathbb{T}[U']$, $\varepsilon v_i :: U' \mid \mu_i \overset{\phi_i'}{\longmapsto} \varepsilon_i uval(v_i) :: U' \mid \mu_i$ and the result follows using Lemma 6.43 and observational monotonicity of the transitivity (Lemma 6.51).

□

Lemma 6.46 (Downward Closed / Monotonicity). *If*

(1) $\langle \phi_1, v_1, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, v_2, \mu_2\rangle : U$ *then*
$\forall j \leq k, \langle \phi_1, v_1, \mu_1\rangle \approx^j_{\ell_o} \langle \phi_2, v_2, \mu_2\rangle : U$

(2) $\langle \phi_1, t_1^U, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, t_2^U, \mu_2\rangle : \mathcal{C}(U)$ *then* $\forall j \leq k, \langle \phi_1, t_1^U, \mu_1\rangle \approx^j_{\ell_o} \langle \phi_2, t_2^U, \mu_2\rangle : \mathcal{C}(U)$

(3) $\mu_1 \approx^k_{\ell_o} \mu_2$ *then* $\forall j \leq k, \mu_1 \approx^j_{\ell_o} \mu_2$

Proof. By induction on type U and the definition of related stores. □

Lemma 6.47. *Consider* $\varepsilon_1 \vdash g_1' \mathrel{\widetilde{\preccurlyeq}} g_1$ *and* $\varepsilon_2 \vdash g_2' \mathrel{\widetilde{\preccurlyeq}} g_2$*. Then* $(\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_1) \wedge \varepsilon_1 \lfloor\leq\rfloor \varepsilon_2) \Rightarrow \neg\mathrm{obs}_{\ell_o}(\varepsilon_2 g_2)$.

Proof. Suppose $\varepsilon_1 = \langle[\ell_{11}, \ell_{12}], [\ell_{13}, \ell_{14}]\rangle$ and $\varepsilon_2 = \langle[\ell_{21}, \ell_{22}], [\ell_{23}, \ell_{24}]\rangle$.
Also consider $\varepsilon_1' = \mathcal{I}_{\preccurlyeq}(g_1, \ell_o) = \langle[\ell_{11}', \ell_{12}'], [\ell_o, \ell_o]\rangle$ and $\varepsilon_2' = \mathcal{I}_{\preccurlyeq}(g_2, \ell_o) = \langle[\ell_{21}', \ell_{22}'], [\ell_o, \ell_o]\rangle$.
If $\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_1' = \triangle^{\preccurlyeq}([\ell_{11}, \ell_{12}], [\ell_{13} \curlyvee \ell_{11}', \ell_{14} \curlywedge \ell_{12}'], [\ell_o, \ell_o])$ is not defined then

(1) $\ell_{13} \curlyvee \ell_{11}' \neg\preccurlyeq \ell_{14} \curlywedge \ell_{12}'$,
(2) $\ell_{11} \neg\preccurlyeq \ell_{14} \curlywedge \ell_{12}'$, or
(3) $\ell_{13} \curlyvee \ell_{11}' \neg\preccurlyeq \ell_o$ or
(4) $\ell_{11} \neg\preccurlyeq \ell_o$.

By construction we know that $\ell_{11} \preccurlyeq \ell_{14}$. By $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_2$ we know that $\ell_{13} \preccurlyeq \ell_{23}$.

If $g_1 = \ell$, then $[\ell_{11}', \ell_{12}'] = [\ell_{13}, \ell_{14}] = [\ell, \ell]$, therefore $\ell \preccurlyeq \ell_{23}$. If $\ell \neg\preccurlyeq \ell_o$, then $\ell_{23} \curlyvee \ell_{21}' \neg\preccurlyeq \ell_o$ and the result holds immediately. If $\ell \preccurlyeq \ell_o$, by construction of evidence we know that it must be the case that $\ell_{11} \preccurlyeq \ell_{13}$, then either

(1) $\ell \curlyvee \ell \neg\preccurlyeq \ell \curlywedge \ell$ (which is impossible),
(2) $\ell_{11} \neg\preccurlyeq \ell \curlywedge \ell$ (which is a contradiction by construction of evidence), or
(3) $\ell \curlyvee \ell \neg\preccurlyeq \ell_o$ (which contradicts $\ell \preccurlyeq \ell_o$) or
(4) $\ell_{11} \neg\preccurlyeq \ell_o$.

so the only possibility is that $\ell_{11} \neg\preccurlyeq \ell_o$, but we know that $\ell_{11} \preccurlyeq \ell_{13}$, i.e. $\ell_{11} \preccurlyeq \ell$ and that $\ell \preccurlyeq \ell_o$, then by transitivity $\ell_{11} \preccurlyeq \ell_o$ which is a contradiction so $\ell \preccurlyeq \ell_o$ case cannot happen.

If $g_1 = ?$, then $[\ell_{11}', \ell_{12}'] = [\bot, \ell_o]$.
If (1) holds, i.e. $\ell_{13} \neg\preccurlyeq \ell_{14} \curlywedge \ell_o$, by construction we know that $\ell_{13} \preccurlyeq \ell_{14}$, therefore it must be the case that $\ell_{13} \neg\preccurlyeq \ell_o$, but $\ell_{13} \preccurlyeq \ell_{23}$ and the result holds because (3) does not hold for ε_2.
If (2) holds, i.e. $\ell_{11} \neg\preccurlyeq \ell_{14} \curlywedge \ell_o$, by construction we know that $\ell_{11} \preccurlyeq \ell_{14}$, therefore it must be the case that $\ell_{11} \neg\preccurlyeq \ell_o$. We also know by construction that $\ell_{11} \preccurlyeq \ell_{13}$, then $\ell_{13} \neg\preccurlyeq \ell_o$. As $\ell_{13} \preccurlyeq \ell_{23}$, then $\ell_{23} \preccurlyeq \ell_o$, and therefore (3) does not hold for ε_2, i.e. $\ell_{23} \curlyvee \ell_{21}' \neg\preccurlyeq \ell_o$. If (3) holds, i.e. $\ell_{13} \curlyvee \bot \neg\preccurlyeq \ell_o$, then $\ell_{13} \neg\preccurlyeq \ell_o$, but $\ell_{13} \preccurlyeq \ell_{23}$ and the result holds because (3) does not hold for ε_2.
If (4) holds, i.e. $\ell_{11} \neg\preccurlyeq \ell_o$, as $\ell_{11} \preccurlyeq \ell_{13} \preccurlyeq \ell_{23}$ then $\ell_{23} \neg\preccurlyeq \ell_o$, and therefore (3) does not hold for ε_2, i.e. $\ell_{23} \curlyvee \ell_{21}' \neg\preccurlyeq \ell_o$. □

Lemma 6.48. *Consider* $\varepsilon_1 \vdash g_1' \mathrel{\widetilde{\preccurlyeq}} g_1$, $\varepsilon_2 \vdash g_2' \mathrel{\widetilde{\preccurlyeq}} g_2$*, and* $\varepsilon_3 = \varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2$ *such that* $\varepsilon_3 \vdash \widetilde{g_1' \curlyvee g_2' \preccurlyeq g_1 \curlyvee g_2}$*. Then* $(\mathrm{obs}_{\ell_o}(\varepsilon_1 g_1) \wedge \mathrm{obs}_{\ell_o}(\varepsilon_2 g_2)) \Rightarrow \mathrm{obs}_{\ell_o}(\varepsilon_3(g_1 \mathbin{\widetilde{\curlyvee}} g_2))$.

Proof. Suppose $\varepsilon_1 = \langle[\ell_{11}, \ell_{12}], [\ell_{13}, \ell_{14}]\rangle$ and $\varepsilon_2 = \langle[\ell_{21}, \ell_{22}], [\ell_{23}, \ell_{24}]\rangle$.
Then $\varepsilon_1 \mathbin{\widetilde{\curlyvee}} \varepsilon_2 = \varepsilon_3 = \langle[\ell_{11} \curlyvee \ell_{21}, \ell_{12} \curlyvee \ell_{22}], [\ell_{13} \curlyvee \ell_{23}, \ell_{14} \curlyvee \ell_{24}]\rangle$. Also consider $\varepsilon_1' = \mathcal{I}_{\preccurlyeq}(g_1, \ell_o) = \langle[\ell_{11}', \ell_{12}'], [\ell_o, \ell_o]\rangle$, $\varepsilon_2' = \mathcal{I}_{\preccurlyeq}(g_2, \ell_o) = \langle[\ell_{21}', \ell_{22}'], [\ell_o, \ell_o]\rangle$, and $\varepsilon_3' = \mathcal{I}_{\preccurlyeq}(g_2 \mathbin{\widetilde{\curlyvee}} g_3, \ell_o) = \langle[\ell_{31}', \ell_{32}'], [\ell_o, \ell_o]\rangle$.
If $g_1 = \ell_1$ and $g_2 = \ell_2$, then $\ell_{32}' = \ell_1 \curlyvee \ell_2$, $\ell_{22}' = \ell_2$ and $\ell_{12}' = \ell_1$. Also $\ell_{31}' = \ell_1 \curlyvee \ell_2$, $\ell_{21}' = \ell_2$ and $\ell_{11}' = \ell_1$.
If $g_1 = ?$ or $g_2 = \ell_2$ (the other case is analogous) then $\ell_{32}' = \ell_o$ and, $\ell_{12}' = \ell_o$ and $\ell_{22}' = \ell_2$ such that $\ell_2 \preccurlyeq \ell_o$. Also $\ell_{11}' = \bot$, $\ell_{21}' = \ell_2$, but $\ell_{31}' = \bot$. Therefore $\ell_{32}' = \ell_{12}' \curlyvee \ell_{22}'$ and $\ell_{31}' \preccurlyeq \ell_{11}' \curlyvee \ell_{21}'$.
We know that

(1) $\ell_{13} \curlyvee \ell'_{11} \preccurlyeq \ell_{14} \curlywedge \ell'_{12}$,
(2) $\ell_{11} \preccurlyeq \ell_{14} \curlywedge \ell'_{12}$, or
(3) $\ell_{13} \curlyvee \ell'_{11} \preccurlyeq \ell_o$ or
(4) $\ell_{11} \preccurlyeq \ell_o$.
(5) $\ell_{23} \curlyvee \ell'_{21} \preccurlyeq \ell_{24} \curlywedge \ell'_{22}$,
(6) $\ell_{21} \preccurlyeq \ell_{24} \curlywedge \ell'_{22}$, or
(7) $\ell_{23} \curlyvee \ell'_{21} \preccurlyeq \ell_o$ or
(8) $\ell_{21} \preccurlyeq \ell_o$.

We have to prove

(10) $(\ell_{13} \curlyvee \ell_{23}) \curlyvee \ell'_{31} \preccurlyeq (\ell_{14} \curlyvee \ell_{24}) \curlywedge \ell'_{32}$,
(11) $(\ell_{11} \curlyvee \ell_{21}) \preccurlyeq (\ell_{14} \curlyvee \ell_{24}) \curlywedge \ell'_{32}$, or
(12) $(\ell_{13} \curlyvee \ell_{23}) \curlyvee \ell'_{31} \preccurlyeq \ell_o$ or
(13) $(\ell_{11} \curlyvee \ell_{21}) \preccurlyeq \ell_o$.

(13) follows directly by (4) and (8).
(12) follows from (3) and (7) and monotonicity of the join.
By definition of evidence and interior, $\ell'_{32} \preccurlyeq \ell_o$ and $\ell'_{31} \preccurlyeq \ell'_{32}$. Therefore, from (1) $\ell_{13} \preccurlyeq \ell_{14}$, from (5) $\ell_{23} \preccurlyeq \ell_{24}$ and therefore $\ell_{13} \curlyvee \ell_{23} \preccurlyeq \ell_{14} \curlyvee \ell_{24}$. Also as $\ell_{13} \preccurlyeq \ell'_{12}$ and $\ell_{23} \preccurlyeq \ell'_{12}$, then $\ell_{13} \curlyvee \ell_{23} \preccurlyeq \ell'_{12} \curlyvee \ell'_{22} = \ell'_{32}$. By similar argument $\ell'_{31} \preccurlyeq (\ell_{14} \curlyvee \ell_{24})$, and also $\ell'_{11} \curlyvee \ell'_{21} \preccurlyeq \ell'_{32}$. But then $\ell'_{31} \preccurlyeq \ell'_{11} \curlyvee \ell'_{21} \preccurlyeq \ell'_{32}$ and (10) holds. □

Lemma 6.49. *Consider* $\varepsilon_1 \vdash g_1 \widetilde{\preccurlyeq} g_2$, $\varepsilon_2 \vdash g_2 \widetilde{\preccurlyeq} g_3$, *and* $\varepsilon_3 = \varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ *such that* $\varepsilon_3 \vdash g_1 \widetilde{\preccurlyeq} g_3$. *Then* $\mathrm{obs}_{\ell_o}(\varepsilon_3(g_3)) \Rightarrow (\mathrm{obs}_{\ell_o}(\varepsilon_1 g_2) \wedge \mathrm{obs}_{\ell_o}(\varepsilon_2 g_3))$.

Proof. Suppose $\varepsilon_1 = \langle[\ell_1, \ell_2], [\ell_3, \ell_4]\rangle, \varepsilon_2 = \langle[\ell_5, \ell_6], [\ell_7, \ell_8]\rangle$.
$\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 = \Delta^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3 \curlyvee \ell_5, \ell_4 \curlywedge \ell_6], [\ell_7, \ell_8]) = \langle[\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7, \ell_8]\rangle$

Notice that as $\ell_3 \preccurlyeq \ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7$ then $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_3$, and as $\ell_7 \preccurlyeq \ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7$ then $\varepsilon_2 \lfloor\leq\rfloor \varepsilon_3$. What we have to prove is equivalent to prove that

$$(\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_2) \vee \neg\mathrm{obs}_{\ell_o}(\varepsilon_2 g_3)) \Rightarrow \neg\mathrm{obs}_{\ell_o}(\varepsilon_3(g_3))$$

If $\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_2)$ and as $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_3$, then by Lemma 6.47 $\neg\mathrm{obs}_{\ell_o}(\varepsilon_3(g_3))$ and the result holds. Similarly, if $\neg\mathrm{obs}_{\ell_o}(\varepsilon_2 g_3)$ and as $\varepsilon_2 \lfloor\leq\rfloor \varepsilon_3$, then by Lemma 6.47 $\neg\mathrm{obs}_{\ell_o}(\varepsilon_3(g_3))$ and the result holds.

□

Lemma 6.50. *Consider* $\varepsilon_1 \vdash g_1 \widetilde{\preccurlyeq} g_2$, $\varepsilon_2 \vdash g_2 \widetilde{\preccurlyeq} g_3$, *and* $\varepsilon_3 = \varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ *such that* $\varepsilon_3 \vdash g_1 \widetilde{\preccurlyeq} g_3$. *Then* $(\mathrm{obs}_{\ell_o}(\varepsilon_1 g_2) \wedge \mathrm{obs}_{\ell_o}(\varepsilon_2 g_3)) \Rightarrow \mathrm{obs}_{\ell_o}(\varepsilon_3(g_3))$.

Proof. Suppose $\varepsilon_1 = \langle[\ell_1, \ell_2], [\ell_3, \ell_4]\rangle, \varepsilon_2 = \langle[\ell_5, \ell_6], [\ell_7, \ell_8]\rangle$.
$\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2 = \Delta^{\preccurlyeq}([\ell_1, \ell_2], [\ell_3 \curlyvee \ell_5, \ell_4 \curlywedge \ell_6], [\ell_7, \ell_8]) = \langle[\ell_1, \ell_2 \curlywedge \ell_4 \curlywedge \ell_6 \curlywedge \ell_8], [\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7, \ell_8]\rangle$

By definition of the transitivity operator, $\ell_1 \preccurlyeq \ell_8$, $\ell_1 \preccurlyeq \ell_4 \curlywedge \ell_6$, and $\ell_3 \curlyvee \ell_5 \preccurlyeq \ell_8$.
Let us consider $\varepsilon'_1 = \mathcal{I}_{\preccurlyeq}(g_2, \ell_o) = \langle[\ell'_1, \ell'_2], [\ell_o, \ell_o]\rangle$, $\varepsilon'_2 = \varepsilon'_3 = \mathcal{I}_{\preccurlyeq}(g_3, \ell_o) = \langle[\ell'_5, \ell'_6], [\ell_o, \ell_o]\rangle$ We know that

(1) $\ell_3 \curlyvee \ell'_1 \preccurlyeq \ell_4 \curlywedge \ell'_2$,
(2) $\ell_1 \preccurlyeq \ell_4 \curlywedge \ell'_2$, or
(3) $\ell_3 \curlyvee \ell'_1 \preccurlyeq \ell_o$ or
(4) $\ell_1 \preccurlyeq \ell_o$.
(5) $\ell_7 \curlyvee \ell'_5 \preccurlyeq \ell_8 \curlywedge \ell'_6$,

(6) $\ell_5 \preccurlyeq \ell_8 \curlywedge \ell_6'$, or
(7) $\ell_7 \curlyvee \ell_5' \preccurlyeq \ell_o$ or
(8) $\ell_5 \preccurlyeq \ell_o$.

We have to prove
(10) $(\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7) \curlyvee \ell_5' \preccurlyeq \ell_8 \curlywedge \ell_6'$,
(11) $\ell_1 \preccurlyeq \ell_8 \curlywedge \ell_6'$, or
(12) $(\ell_1 \curlyvee \ell_3 \curlyvee \ell_5 \curlyvee \ell_7) \curlyvee \ell_5' \preccurlyeq \ell_o$ or
(13) $\ell_1 \preccurlyeq \ell_o$.

Notice that if $g_3 = ?$ then $\ell_6' = \ell_o$ and therefore by (4) $\ell_1 \preccurlyeq \ell_6'$, and by (3), $\ell_3 \preccurlyeq \ell_6'$. Also $\ell_5' = \bot$ and therefore $\ell_5' \preccurlyeq \ell_7 \preccurlyeq \ell_8$. If $g_3 = \ell$, then $\ell_5' = \ell_6' = \ell$ and $\ell_7 = \ell_8 = \ell$, but we know that $\ell_1 \preccurlyeq \ell_8$, and therefore $\ell_1 \preccurlyeq \ell_6'$ and $\ell_5' \preccurlyeq \ell_8$. Also as $\ell_3 \preccurlyeq \ell_8$ then $\ell_3 \preccurlyeq \ell_6'$.

We also know that $\ell_3 \curlyvee \ell_5 \preccurlyeq \ell_8$ and by definition of intervals $\ell_7 \preccurlyeq \ell_8$. We know that $\ell_1 \preccurlyeq \ell_6'$. By (5) $\ell_7 \curlyvee \ell_5' \preccurlyeq \ell_6'$. By (6) $\ell_5 \preccurlyeq \ell_6'$. Also $\ell_3 \preccurlyeq \ell_6'$ and (10) follows.

We know that $\ell_1 \preccurlyeq \ell_8$ and that $\ell_1 \preccurlyeq \ell_6'$ therefore (11) holds. By (4), (3), (7), (8) and because $\ell_5' \preccurlyeq \ell_o$ by definition of interior, (12) holds. Finally (13) holds by (4). □

Lemma 6.51. *Consider* $\varepsilon_1 \vdash g_1 \widetilde{\lesssim} g_2$, $\varepsilon_2 \vdash g_2 \widetilde{\lesssim} g_3$, *and* $\varepsilon_3 = \varepsilon_1 \circ^{\preccurlyeq} \varepsilon_2$ *such that* $\varepsilon_3 \vdash g_1 \widetilde{\lesssim} g_3$. *Then* $(\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_2) \vee \neg\mathrm{obs}_{\ell_o}(\varepsilon_2 g_3)) \iff \neg\mathrm{obs}_{\ell_o}(\varepsilon_3(g_3))$.

Proof. Direct by Lemmas 6.49 and 6.50 . □

Lemma 6.52. *Consider* ε_1 *and* $\varepsilon_1' = \varepsilon_2 \,\widetilde{\curlyvee}\, (\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3)$, *for some* ε_2 *and* ε_3. *Then* $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_1'$

Proof. Suppose $\varepsilon_2 = \langle[\ell_1, \ell_2], [\ell_3, \ell_4]\rangle$, $\varepsilon_1 = \langle[\ell_5, \ell_2], [\ell_7, \ell_8]\rangle$, and $\varepsilon_3 = \langle[\ell_9, \ell_{10}], [\ell_{11}, \ell_{12}]\rangle$.
$\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3 = \Delta^{\preccurlyeq}([\ell_5, \ell_6], [\ell_7 \curlyvee \ell_9, \ell_8 \curlywedge \ell_{10}], [\ell_{11}, \ell_{12}]) = \langle[\ell_5, \ell_6 \curlywedge \ell_8 \curlywedge \ell_{10} \curlywedge \ell_{12}], [\ell_5 \curlyvee \ell_7 \curlyvee \ell_9 \curlyvee \ell_{11}, \ell_{12}]\rangle$
$\varepsilon_2 \,\widetilde{\curlyvee}\, (\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3) = \langle[\ell_1 \curlyvee \ell_5, \ell_2 \curlyvee (\ell_6 \curlywedge \ell_8 \curlywedge \ell_{10} \curlywedge \ell_{12})], [\ell_3 \curlyvee \ell_5 \curlyvee \ell_7 \curlyvee \ell_9 \curlyvee \ell_{11}, \ell_4 \curlyvee \ell_{12}]\rangle$.

But $\ell_7 \preccurlyeq \ell_3 \curlyvee \ell_5 \curlyvee \ell_7 \curlyvee \ell_9 \curlyvee \ell_{11}$ and therefore, $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_1'$. □

Lemma 6.53. *Consider* $\varepsilon_1 \vdash g_1' \widetilde{\lesssim} g_1$ *and* $\varepsilon_1' = \varepsilon_2 \,\widetilde{\curlyvee}\, (\varepsilon_1 \circ^{\preccurlyeq} \varepsilon_3)$ *such that* $\varepsilon_1' \vdash g_2' \widetilde{\lesssim} g_2$. *Then* $\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_1) \Rightarrow \neg\mathrm{obs}_{\ell_o}(\varepsilon_1' g_2)$.

Proof. By Lemma 6.52 and Lemma 6.47 the result holds immediately. □

Lemma 6.54. *Consider* $\varepsilon_1 \vdash g_1' \widetilde{\lesssim} g_1$, $\varepsilon_2 \vdash g_2' \widetilde{\lesssim} g_2$, *and* $\varepsilon_3 = \varepsilon_1 \,\widetilde{\curlyvee}\, \varepsilon_2$ *such that* $\varepsilon_3 \vdash \widetilde{g_1' \curlyvee g_2' \preccurlyeq g_1 \curlyvee g_2}$. *Then* $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_3$.

Proof. Suppose $\varepsilon_1 = \langle[\ell_1, \ell_2], [\ell_3, \ell_4]\rangle$, $\varepsilon_2 = \langle[\ell_5, \ell_6], [\ell_7, \ell_8]\rangle$, then
$\varepsilon_3 = \langle[\ell_1 \curlyvee \ell_5, \ell_2 \curlyvee \ell_6], [\ell_3 \curlyvee \ell_7, \ell_4 \curlyvee \ell_8]\rangle$.
As $\ell_3 \preccurlyeq \ell_3 \curlyvee \ell_3 \curlyvee \ell_7$ therefore, $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_3$ and the result holds. □

Lemma 6.55. *Consider* $\varepsilon_1 \vdash g_1' \widetilde{\lesssim} g_1$, $\varepsilon_2 \vdash g_2' \widetilde{\lesssim} g_2$, *and* $\varepsilon_3 = \varepsilon_1 \,\widetilde{\curlyvee}\, \varepsilon_2$ *such that* $\varepsilon_3 \vdash \widetilde{g_1' \curlyvee g_2' \preccurlyeq g_1 \curlyvee g_2}$. *Then* $(\neg\mathrm{obs}_{\ell_o}(\varepsilon_1 g_1) \vee \neg\mathrm{obs}_{\ell_o}(\varepsilon_2 g_2)) \iff \neg\mathrm{obs}_{\ell_o}(\varepsilon_3(g_1 \,\widetilde{\curlyvee}\, g_2))$.

Proof. First we prove the $\Rightarrow$ direction. By Lemma 6.54, $\varepsilon_1 \lfloor\leq\rfloor \varepsilon_3$. Suppose $\mathrm{obs}_{\ell_o}(\varepsilon_1 g_1)$ does not hold (the other case is analogous). Then by Lemma 6.47 the result holds immediately. Then for the $\Leftarrow$ we use Lemma 6.48 and the result holds immediately. □

Lemma 6.56. *Consider $\phi' \triangleright t^U \in \mathbb{T}[U]$, and μ, such that $t^U \vdash \mu$ and $\neg\mathrm{obs}_{\ell_o}(\phi')$, and $\forall k > 0$, such that $t^U \mid \mu \overset{\phi'}{\longmapsto}{}^k t'^U \mid \mu'$, then $\forall \phi$,*

(1) $\forall o^{U'} \in dom(\mu') \backslash dom(\mu)$, $\neg\mathrm{obs}_{\ell_o}(\phi \triangleright \mu'(o^{U'}))$.
(2) $\forall o^{U'} \in dom(\mu') \cap dom(\mu) \wedge \mu'(o^{U'}) \neq \mu(o^{U'})$,
  (a) $\neg\mathrm{obs}_{\ell_o}(\phi \triangleright \mu(o^{U'}))$, and
  (b) $\neg\mathrm{obs}_{\ell_o}(\phi \triangleright \mu'(o^{U'}))$.

Proof. We use induction on the derivation of t^U. The interest cases are the last step of reduction rules for references and assignments.

Case ($t = \varepsilon_1 o^U_{g'} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u$). We are only updating the heap so we only have to prove (a) and (b). Then

$$\varepsilon_1 o^U_{g'} \overset{g,U_1}{:=}_{\varepsilon_\ell} \varepsilon_2 u \overset{\phi'}{\longmapsto} \mathsf{unit}_\perp \mid \mu[o^U \mapsto \varepsilon'(u \,\tilde{\curlyvee}\, (\phi'.g_c \,\tilde{\curlyvee}\, g')) :: U']$$

where $\varepsilon' = (\varepsilon_2 \circ^{<:} iref(\varepsilon_1)) \,\tilde{\curlyvee}\, ((\phi'.\varepsilon \,\tilde{\curlyvee}\, ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_3 \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1))$ and if $\mu(o^{U'}) = \varepsilon u :: U'$, then $\phi'.\varepsilon \lfloor\leq\rfloor \varepsilon$. For simplicity let us call $\varepsilon_2' = (\varepsilon_2 \circ^{<:} iref(\varepsilon_1))$ and $\varepsilon_3' = \varepsilon_3 \circ^{\preccurlyeq} ilbl(iref(\varepsilon_1))$. We have to prove that (b) $\neg(\mathrm{obs}_{\ell_o}(\varepsilon' \widetilde{label}(U'))$. As $\neg\mathrm{obs}_{\ell_o}(\phi')$, by Lemma 6.55, $\neg\mathrm{obs}_{\ell_o}((\phi'.\varepsilon \tilde{\curlyvee} ilbl(\varepsilon_1))(\phi'.g_c \tilde{\curlyvee} g))$. Then by Lemma 6.53, $\neg(\mathrm{obs}_{\ell_o}(\varepsilon' \widetilde{label}(U'))$. Next we have to prove that (a) $\mathrm{obs}_{\ell_o}(\phi \triangleright \mu(o^{U'}))$ is not defined. Consider that $\mu(o^{U'}) = \varepsilon u :: U'$. We know that $\mathrm{obs}_{\ell_o}(\phi'.\varepsilon\phi'.g_c)$ is not defined, and that $\phi'.\varepsilon \lfloor\leq\rfloor \varepsilon$, therefore by Lemma 6.47, $\mathrm{obs}_{\ell_o}(\varepsilon U')$ is not defined, concluding that $\mathrm{obs}_{\ell_o}(\phi \triangleright \mu(o^{U'}))$ is not defined as well and the result holds.

Case ($t = \mathsf{ref}^{U'}_{\varepsilon_\ell}\ \varepsilon_s u$). We are extending the heap, so we need to only prove (1). Then

$$\mathsf{ref}^{U}_{\varepsilon_\ell}\ \varepsilon_s u \mid \mu \overset{\phi'}{\longmapsto} o^{U'}_\perp \mid \mu[o^{U'} \mapsto \varepsilon'(u \,\tilde{\curlyvee}\, \phi'.g_c) :: U']$$

where $o^{U'} \notin dom(\mu)$, $\varepsilon' = \varepsilon_s \,\tilde{\curlyvee}\, (\phi'.g_c \circ^{\preccurlyeq} \varepsilon_\ell)$. We need to prove that $\mathrm{obs}_{\ell_o}(\phi \triangleright \varepsilon'(u \,\tilde{\curlyvee}\, \phi'.g_c) :: U')$ does not hold. In order to do so, we will show that $\mathrm{obs}_{\ell_o}(ilbl(\varepsilon')\widetilde{label}(U'))$ does not holds, which follows directly by Lemma 6.53.

□

Lemma 6.57. *Consider ϕ', such that $\mathrm{obs}_{\ell_o}(\phi'.\varepsilon\phi'.g_c)$ does not hold, then then $\forall k > 0$, such that $t_i^U \mid \mu_i \overset{\phi'}{\longmapsto}{}^k t_i'^U \mid \mu_i'$, then if $\mu_1 \approx^k_{\ell_o} \mu_2$, then $\mu_1' \approx^k_{\ell_o} \mu_2'$*

Proof. By Lemma 6.56 we know three things:

(1) $\forall o^{U'} \in dom(\mu_i') \backslash dom(\mu_i)$, $\mathrm{obs}_{\ell_o}(\phi \triangleright \mu_i'(o^{U'}))$ does not hold, i.e. new locations are not observable.
(2) $\forall o^{U'} \in dom(\mu_i') \cap dom(\mu_i) \wedge \mu_i'(o^{U'}) \neq \mu(o^{U'})$,
  (a) $\mathrm{obs}_{\ell_o}(\phi \triangleright \mu_i(o^{U'}))$ does not hold, and
  (b) $\mathrm{obs}_{\ell_o}(\phi \triangleright \mu_i'(o^{U'}))$ does not hold.
  i.e. for all updated references they have to be previously not observable, and by definition therefore related, and second they are still non observable after the update, and by definition those locations are still related under ϕ.

Therefore $\mu_1' \approx^k_{\ell_o} \mu_2'$ and the result holds. □

LEMMA 6.58. *Consider simple values* $u_i \in \mathbb{T}[U_i]$ *and* $\langle \phi_1, \varepsilon_1 u_1 :: U, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \varepsilon_2 u_2 :: U, \mu_2 \rangle : U$.
If $\varepsilon_i \vdash g \lesssim U$, *then*

$$\langle \phi_1, (\varepsilon_1' \mathbin{\widetilde{\curlyvee}} \varepsilon_1)(u_1 \mathbin{\widetilde{\curlyvee}} g) :: U, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, (\varepsilon_2' \mathbin{\widetilde{\curlyvee}} \varepsilon_2))(u_2 \mathbin{\widetilde{\curlyvee}} g) :: U, \mu_2 \rangle : U$$

PROOF. By definition of related values and Lemma 6.55 (observational-monotonicity of the join), considering that the label stamping can make the new values non observable and that join of evidences does not introduce imprecision. □

LEMMA 6.59. *Suppose that* $\phi_i \leq_{\ell_o} \phi_i'$, $\phi_i' \triangleright \mathsf{prot}^{g,U}_{\varepsilon_i' g_i'} \phi_i''(\varepsilon_i t^{U_i}) \in \mathbb{T}[U \mathbin{\widetilde{\curlyvee}} g]$, *for* $i \in \{1,2\}$, *where* $\mathsf{obs}_{\ell_o}(\phi_i''.\varepsilon\phi_i''.g_c)$ *does not hold, and either* $\mathsf{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ *or* $\mathsf{obs}_{\ell_o}(\varepsilon_i' g)$ *does not hold. Also consider two stores* μ_i *such that* $\mu_1 \approx^k_{\ell_o} \mu_2$.
Then $\langle \phi_1, \mathsf{prot}^{g,U}_{\varepsilon_1' g_1'} \phi_1'(\varepsilon_1 t^{U_1}) s, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \mathsf{prot}^{g,U}_{\varepsilon_2' g_2'} \phi_2'(\varepsilon_2 t^{U_2}), \mu_2 \rangle$

PROOF. Suppose that after at least j more steps, where $j < k$, both subterms reduce to a value (let us assume no cast errors are produced, otherwise the lemma vacuously holds):

$$t^{U_i} \mid \mu_i \quad \overset{\phi_i''}{\longmapsto}{}^{j} \quad \varepsilon_i' v_i \mid \mu_i'$$

Therefore:

$$\begin{aligned} & \mathsf{prot}^{g,U}_{\varepsilon_i' g_i'} \phi_i''(\varepsilon_i t^{U_i}) \mid \mu_i' \\ \overset{\phi_i'}{\longmapsto}{}^{j} \quad & \mathsf{prot}^{g,U}_{\varepsilon_i' g_i'} \phi_i''(\varepsilon_i'' u_i) \mid \mu_i' \\ \overset{\phi_i'}{\longmapsto}{}^{1} \quad & (\varepsilon_i'' \mathbin{\widetilde{\curlyvee}} \varepsilon_i')(u_i \mathbin{\widetilde{\curlyvee}} g_i') :: U \mathbin{\widetilde{\curlyvee}} g \mid \mu_i' \end{aligned}$$

As the values can be radically different we have to make sure that both values are not observables. If $\mathsf{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ does not hold then the values are not observables because the security context is not observable. Let us assume that $\mathsf{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ holds, but $\mathsf{obs}_{\ell_o}(\varepsilon_i' g)$ not. Then by Lemma 6.55, $\mathsf{obs}_{\ell_o}((\varepsilon_i'' \mathbin{\widetilde{\curlyvee}} \varepsilon_i')(\widetilde{label}(U) \mathbin{\widetilde{\curlyvee}} g))$ does not hold, and therefore $\mathsf{obs}_{\ell_o}(\phi_i \triangleright (\varepsilon_i'' \mathbin{\widetilde{\curlyvee}} \varepsilon_i')(u_i \mathbin{\widetilde{\curlyvee}} g_i') :: U \mathbin{\widetilde{\curlyvee}} g)$ does not hold.

Now we have to prove that the resulting stores are related. But by Lemma 6.57 the result immediately.

□

LEMMA 6.60. *Suppose that* $\phi_i \leq_{\ell_o} \phi_i'$, $\phi_i \leq_{\ell_o} \phi_i''$, $\langle \phi_1, t_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, t_2, \mu_2 \rangle : \mathcal{C}(U')$, *and that* $\phi_i' \triangleright \mathsf{prot}^{g,U}_{\varepsilon_i' g_i'} \phi_i''(\varepsilon_i t^{U_i}) \in \mathbb{T}[U \mathbin{\widetilde{\curlyvee}} g]$, *for* $i \in \{1,2\}$. *If* $\phi_1 \approx^k_{\ell_o} \phi_2$, *and both* $\mathsf{obs}_{\ell_o}(\varepsilon_i' g_i)$ *hold or does not hold,*
then $\langle \phi_1, \mathsf{prot}^{g,U}_{\varepsilon_1' g_1'} \phi_1''(\varepsilon_1 t_1^{U'}), \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \mathsf{prot}^{g,U}_{\varepsilon_2' g_2'} \phi_2''(\varepsilon_2 t_2^{U'}), \mu_2 \rangle : \mathcal{C}(U \mathbin{\widetilde{\curlyvee}} g)$

PROOF. In case that combining evidence may fail, then the Lemma vacuously holds. Let us assume that combining evidence always successes. Consider $j < k$, we know by definition of related computations that

$$t_i^{U'} \mid \mu_i \overset{\phi_i''}{\longmapsto}{}^{j} t_i'^{U'} \mid \mu_i'$$

then $\mu_1' \approx^j_{\ell_o} \mu_2'$, and by Lemma 6.61 $\mu_i \twoheadrightarrow \mu_i'$. If $t_i'^{U}$ are reducible after $k-1$ steps, then the result holds immediately by (Rprot()). The interest case if $t_i'^{U}$ are irreducible after $j < k$ steps:

Suppose that after j steps $t_i'^{U'} = v_i$, then $\langle \phi_1, v_1, \mu_1' \rangle \approx^{k-j}_{\ell_o} \langle \phi_2, v_2, \mu_2' \rangle : U'$.

Therefore:

$$
\begin{array}{rl}
& \mathsf{prot}^{g,U}_{\varepsilon'_i g'_i} \phi''_i(\varepsilon_i t_i^{U'}) \mid \mu'_i \\
\xrightarrow{\phi'_i}{}^{j} & \mathsf{prot}^{g,U}_{\varepsilon'_i g'_i} \phi''_i(\varepsilon''_i u_i) \mid \mu'_i \\
\xrightarrow{\phi'_i}{}^{1} & (\varepsilon''_i \mathbin{\widetilde{\curlyvee}} \varepsilon'_i)(u_i \mathbin{\widetilde{\curlyvee}} g'_i) :: U \mathbin{\widetilde{\curlyvee}} g \mid \mu'_i
\end{array}
$$

If both $\mathrm{obs}_{\ell_o}(\phi_i \triangleright v_i)$ do not hold, then by Lemma 6.63, $\mathrm{obs}_{\ell_o}(\varepsilon''_i u_i :: U')$ also does not hold. Finally by Lemma 6.55 $\mathrm{obs}_{\ell_o}((\varepsilon''_i \mathbin{\widetilde{\curlyvee}} \varepsilon'_i)(\widetilde{label}(U) \mathbin{\widetilde{\curlyvee}} g))$ does not hold and therefore the values are related.

Let us consider that $\mathrm{obs}_{\ell_o}(\phi \triangleright v_i)$ holds and that $\mathrm{obs}_{\ell_o}(\phi \triangleright \varepsilon''_i u_i :: U')$ holds (otherwise we follow by the previous argument). If both $\mathrm{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ do not hold, then the values are not observables because the security contexts are not observable.

Let us assume that both $\mathrm{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ hold, but $\mathrm{obs}_{\ell_o}(\varepsilon'_i g)$ not. Then by Lemma 6.55, $\mathrm{obs}_{\ell_o}((\varepsilon''_i \mathbin{\widetilde{\curlyvee}} \varepsilon'_i)(\widetilde{label}(U) \mathbin{\widetilde{\curlyvee}} g))$ do not hold, and therefore $\mathrm{obs}_{\ell_o}(\phi_i \triangleright (\varepsilon''_i \mathbin{\widetilde{\curlyvee}} \varepsilon'_i)(u_i \mathbin{\widetilde{\curlyvee}} g'_i) :: U \mathbin{\widetilde{\curlyvee}} g)$ do not hold.

If $\mathrm{obs}_{\ell_o}(\phi_i.\varepsilon\phi_i.g_c)$ and $\mathrm{obs}_{\ell_o}((\varepsilon''_i \mathbin{\widetilde{\curlyvee}} \varepsilon'_i)(\widetilde{label}(U) \mathbin{\widetilde{\curlyvee}} g))$ hold, then the result follows by Lemma 6.58, and by backward preservation of the relations (Lemma 6.43).

□

Lemma 6.61. *Consider term $\phi \triangleright t^U \in \mathbb{T}[U]$, store μ and $j > 0$, such that $t^U \mid \mu \xrightarrow{\phi}{}^{j} t'^U \mid \mu'$. Then $\mu \twoheadrightarrow \mu'$.*

Proof. Trivial by induction on the derivation of t^U. The only rules that change the store are the ones for reference and assignment, neither of which remove locations. □

Lemma 6.62. *If $\phi \leq_{\ell_o} \phi'$ and $\phi' \leq_{\ell_o} \phi''$, then $\phi \leq_{\ell_o} \phi''$.*

Proof. Trivial because if ϕ is not observable, then ϕ' is not observable as well by definition of $\leq_{\ell_o}$, and therefore ϕ'' must also be not observable. □

Lemma 6.63. *Consider $\phi_i \triangleright v \in \mathbb{T}[U]$, and $\varepsilon \vdash U \lesssim U'$. Suppose $\varepsilon v :: U' \xrightarrow{i} \varepsilon' u :: U'$. If $\neg\mathrm{obs}_{\ell_o}(\phi_i \triangleright v) \vee \neg\mathrm{obs}_{\ell_o}(\varepsilon U') \iff \neg\mathrm{obs}_{\ell_o}(\phi_i \triangleright \varepsilon' u :: U')$.*

Proof. Direct by Lemma 6.51. □

Next, we present the Noninterference proposition, which naturally implies the Security Type Soundness proposition (Prop 2.24) presented in the paper.

Proposition 6.64 (Noninterference). *If $\phi'_i \triangleright \check{t} \in \mathbb{T}[U]$, $\mu_i \in$ Store, $\check{t} \vdash \mu_i$, $\Gamma = FV(\check{t})$, and $\forall k \geq 0, \phi_i \leq_{\ell_o} \phi'_i, \Gamma \vdash \langle \phi_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2, \mu_2 \rangle$, then $\langle \phi_1, \rho_1(\check{t}), \mu_1, \approx_{\ell_o} \rangle\, [\ell_o]^k \,\langle \phi_2, \rho_2(\check{t}), \mu_2, :\rangle\, \mathcal{C}(U)$.*

Proof. By induction on the derivation of term $\check{t} \in \mathbb{T}[U]$. Let us take an arbitrary index $k \geq 0$.

Case (x). $\check{t} = x^U$ so $\Gamma = \{x^U\}$. $\Gamma \vdash \langle \phi_1, \rho_1, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2, \mu_2 \rangle$ implies by definition that $\langle \phi_1, \rho_1(x^U), \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2(x^U), \mu_2 \rangle : U$, and the result holds immediately.

———

Case (b). $\check{t} = b_g$. By definition of substitution, $\rho_1(b_g) = \rho_2(b_g) = b_g$. By definition, $\langle \phi_1, b_g, \mu_1 \rangle \approx^k_{\ell_o} \langle \phi_2, b_g, \mu_2 \rangle : \mathsf{Bool}_g$ as required.

———

Case (o). $\check{t} = o^{U_1}_{g_1}$ where $U = \mathsf{Ref}_{g_1}\ U_1$. By definition of substitution, $\rho_1(o^{U_1}_{g_1}) = \rho_2(o^{U_1}_{g_1}) = o^{U_1}_{g_1}$. We know that $\phi_i \triangleright o^{U_1}_{g_1} \in \mathbb{T}[\mathsf{Ref}_{g_1}\ U_1]$. By definition of related stores, $\langle \phi_1, o^{U_1}_{g_1}, \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, o^{U_1}_{g_1}, \mu_2\rangle$: $\mathsf{Ref}_{g_1}\ U_1$ as required, and the result holds.

———

Case (λ). $t^U = (\lambda^{g'_c} x^{U_1}.t^{U_2})_g$. Then $U = U_1 \xrightarrow{g'_c}_g U_2$.
By definition of substitution, assuming $x^{U_1} \notin dom(\rho_i)$, and Lemma 6.42:

$$\phi'_i \triangleright \rho_i(t^U) = \phi'_i \triangleright (\lambda^{g'_c} x^{U_1}.\rho_i(t^{U_2}))_g \in \mathbb{T}[U]$$

Consider $j \leq k$, μ'_1, μ'_2 such that $\mu_i \twoheadrightarrow \mu'_i$ and $\mu'_1 \approx^j_{\ell_o} \mu'_2$, and assume two values v_1 and v_2 such that $\langle \phi_1, v_1, \mu'_1\rangle \approx^j_{\ell_o} \langle \phi_2, v_2, \mu'_2\rangle : U'_1$. Consider $U' = U''_1 \xrightarrow{g''_c}_{g''} U''_2$ and $\varepsilon_1, \varepsilon_2, \varepsilon_\ell$, such that $\varepsilon_1 \vdash U_1 \xrightarrow{g'_c}_g U_2 \lesssim U'$, that $\varepsilon_2 \vdash U'_1 \lesssim U''_1$, and that $\varepsilon_\ell \vdash \widetilde{\phi'.g_c \vee g''} \preccurlyeq g''$
For simplicity, let us annotate $U'_2 = U''_2 \,\widetilde{\curlyvee}\, g''$. We need to show that:

$$\begin{array}{ll} & \langle \phi_1, \varepsilon_1(\lambda^{g'_c} x^{U_1}.\rho_1(t^{U_2}))_g \mathbin{@}^{U'}_{\varepsilon_\ell} \varepsilon_2 v_1, \mu'_1\rangle \\ \approx^j_{\ell_o} & \langle \phi_2, \varepsilon_1(\lambda^{g'_c} x^{U_1}.\rho_2(t^{U_2}))_g \mathbin{@}^{U'}_{\varepsilon_\ell} \varepsilon_2 v_2, \mu'_2\rangle \quad : \mathcal{C}(U'_2) \end{array}$$

Each v_i is either a bare value u_i or a casted value $\varepsilon_{2i} u_i :: U'_1$. In the latter case, the application expression combines evidence, which may fail with **error**. If it succeeds, we call the combined evidence ε'_{2i}. The application rule then applies: it may fail with **error** if the evidence ε'_{2i} cannot be combined with the evidence for the function parameter. Every time a failure is produced product of evidence combination, then the relation vacuously holds. We therefore consider the only interesting case, where reductions always succeed. Then:

$$\begin{array}{ll} & \varepsilon_1(\lambda^{g'_c} x^{U_1}.\rho_i(t^{U_2}))_g \mathbin{@}^{U'} \varepsilon'_{2i} u_i \mid \mu'_i \\ \xmapsto{\phi'_i} & \mathsf{prot}^{g'',U''_2}_{\varepsilon_\ell g}\ \phi''_i(\varepsilon_p([\varepsilon_{ai} u_i :: U_1/x^{U_1}]\rho_i(t^{U_2}))) \mid \mu'_i \\ \xmapsto{\phi'_i}{}^{*} & \mathsf{prot}^{g'',U''_2}_{\varepsilon_\ell g}\ \phi''_i(\varepsilon_p([\varepsilon_{ai} u_i :: U_1/x^{U_1}]\rho_i(t^{U_2}))) \mid \mu'_i \end{array}$$

where $\phi'' = \langle \varepsilon'_r(\phi'.g_c \,\widetilde{\curlyvee}\, g), g'_c\rangle$, $\varepsilon'_r = (\phi'.\varepsilon \,\widetilde{\curlyvee}\, ilbl(\varepsilon_1)) \circ^{\preccurlyeq} \varepsilon_\ell \circ^{\preccurlyeq} ilat(\varepsilon_1)$. If $\mathsf{obs}_{\ell_o}(\phi'_i)$ do not hold, then by Lemma 6.55, $\mathsf{obs}_{\ell_o}(\phi''_i)$ do not hold. Then $\phi'_i \leq_{\ell_o} \phi''$, and by Lemma 6.62, $\phi_i \leq_{\ell_o} \phi''_i$.

ε_ℓ, ε_p and ε_{ai} are the new evidences for the label, return value and argument, respectively. We then extend the substitutions to map x^{U_1} to the casted arguments:

$$\rho'_i = \rho_i\{x^{U_1} \mapsto \varepsilon_{ai} u_i :: U_1\}$$

We know that $\langle \phi_1, v_1, \mu'_1\rangle \approx^j_{\ell_o} \langle \phi_2, v_2, \mu'_2\rangle$ and consider $\phi \triangleright u_i \in \mathbb{T}[U_{ui}]$ then $\varepsilon_{ai} \vdash U_{ui} \lesssim U_1$ and $\varepsilon_{ai} = (\varepsilon_{2i} \circ^{<:} \varepsilon_2) \circ^{<:} idom(\varepsilon_1)$, therefore using Lemma 6.45 $\langle \phi_1, (\varepsilon_{a1} u_1 :: U_1), \mu'_1\rangle \approx^j_{\ell_o} \langle \phi_2, (\varepsilon_{a2} u_2 :: U_1), \mu'_2\rangle : U_1$

So as $\mu_i \twoheadrightarrow \mu'_i$ then by Lemma 6.41, $\Gamma, x^{U_1} \vdash \langle \phi_1, \rho'_1, \mu'_1\rangle \approx^j_{\ell_o} \langle \phi_2, \rho'_2, \mu'_2\rangle$.
We also know that $\phi''_i \triangleright \rho_i(t^{U_2}) \in \mathbb{T}[U_2]$. Then by induction hypothesis:

$$\langle \phi_1, \rho'_1(t^{U_2}), \mu'_1\rangle \approx^{j-1}_{\ell_o} \langle \phi_2, \rho'_2(t^{U_2}), \mu'_2\rangle : \mathcal{C}(U_2)$$

Finally, by Lemma 6.60:

$$\begin{array}{ll} & \langle \phi_1, \mathsf{prot}^{g'',U''_2}_{\varepsilon_\ell g}\ \phi''(\varepsilon_p \rho'_1(t^{U_2})), \mu'_1\rangle \\ \approx^j_{\ell_o} & \langle \phi_2, \mathsf{prot}^{g'',U''_2}_{\varepsilon_\ell g}\ \phi''(\varepsilon_p \rho'_2(t^{U_2})), \mu'_2\rangle : \mathcal{C}(U'_2) \end{array}$$

and finally the result holds by backward preservation of the relations (Lemma 6.43).

———

Case (!). $t^U = !^{\mathsf{Ref}_g\ U_1}\varepsilon t^{U'_1}$. Then $U = U_1 \mathbin{\widetilde{\curlyvee}} g$.
By definition of substitution:

$$\rho_i(t^U) = !^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1})$$

We have to show that

$$\begin{array}{rl} & \langle \phi_1, !^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1}), \mu_1\rangle \\ \approx^k_{\ell_o} & \langle \phi_2, !^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1}), \mu_2\rangle : \mathcal{C}(U_1 \mathbin{\widetilde{\curlyvee}} g) \end{array}$$

By Lemma 6.42:

$$\phi'_i \triangleright !^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1}) \in \mathbb{T}[U_1 \mathbin{\widetilde{\curlyvee}} g]$$

By induction hypotheses on the subterm:

$$\langle \phi_1, \rho_1(t^{U'_1}), \mu_1\rangle \approx^k_{\ell_o} \langle \phi_2, \rho_2(t^{U'_1}), \mu_2\rangle : \mathcal{C}(U'_1)$$

Consider $j < k$, then by definition of related computations

$$\rho_i(t^{U'_1}) \mid \mu_i \xrightarrow{\phi'_i}{}^j t_i^{U'_1} \mid \mu'_i \implies \mu'_1 \approx^{k-j}_{\ell_o} \mu'_2 \wedge (irred(t_i^{U'_1}) \Rightarrow \langle \phi_1, t_1^{U'_1}, \mu'_1\rangle \approx^{k-j}_{\ell_o} \langle \phi_2, t_2^{U'_1}, \mu'_2\rangle : U'_1)$$

Where $U'_1 = \mathsf{Ref}_{g'}\ U''_1$. If terms $t_i^{U'_1}$ are reducible after $j = k-1$ steps, then $!^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1}) \mid \mu_i \xrightarrow{\phi'_i}{}^j !^{\mathsf{Ref}_g\ U_1}\varepsilon t_i'^{U'_1} \mid \mu'_i$ and the result holds.
If after at most j steps $t_i^{U'_1}$ is irreducible it means that for some $j' \leq j$, $!^{\mathsf{Ref}_g\ U_1}\varepsilon\rho_i(t^{U'_1}) \mid \mu_i \xrightarrow{\phi'_i}{}^{j'} !^{\mathsf{Ref}_g\ U_1}\varepsilon v_i \mid \mu'_i$. If $j' = j$ then we use the same same argument for reducible terms and the result holds.
Let us consider now $j' < j$. Then $\langle \phi_1, v_1, \mu'_1\rangle \approx^{k-j'}_{\ell_o} \langle \phi_2, v_2, \mu'_2\rangle : U'_1$. By Lemma 6.10, each v_i is either a location $(o_{i_{g'_i}}^{U''_i})$ or a casted location $\varepsilon_i(o_{i_{g''_i}}^{U'''_i}) :: U'_1$. Let us assume they both are a casted location (the other cases are analogous). In case a value v_{ij} is a casted value, then the whole term $\rho_i(t^U)$ can take a step by (Rg), combining ε with ε_i. Such a step either fails, or succeeds with a new combined evidence. Therefore, either:

$$\rho_i(t^U) \mid \mu_i \xrightarrow{\phi'_i}{}^{j'} \mathbf{error}$$

in which case we do not care since we only consider termination-insensitive noninterference, or:

$$\begin{array}{rll} \rho_i(t^U) \mid \mu & \xrightarrow{\phi'_i}{}^{j'+1} & !^{\mathsf{Ref}_g\ U_1}\varepsilon'_i o_{i_{g''_i}}^{U'''_i} \mid \mu'_i \\ & \xrightarrow{\phi'_i}{}^{1} & \mathsf{prot}^{g,U_1}_{ilbl(\varepsilon'_i)g''_i}\phi''_i(iref(\varepsilon'_i)v'_i) \mid \mu'_i \end{array}$$

with $v'_i = \mu'_i(o_{i_{g''}}^{U'''_i}) = \varepsilon_{ui}u'_i :: U'''_i$, $\phi''_i = \langle(\phi'_i\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon'_i))(\phi'_i g_c \mathbin{\widetilde{\curlyvee}} g''_i), \phi'_i g_c \mathbin{\widetilde{\curlyvee}} g\rangle$. By Lemma 6.55, if $\neg\mathsf{obs}_{\ell_o}(\phi'_i)$ then $\neg\mathsf{obs}_{\ell_o}(\phi''_i)$. Then by Lemma 6.62, $\phi_i \leq_{\ell_o} \phi''_i$. As $\langle \phi_1, v_1, \mu'_1\rangle \approx^{k-j'}_{\ell_o} \langle \phi_2, v_2, \mu'_2\rangle : U'_1$, then by Lemma 6.51 either both $\mathsf{obs}_{\ell_o}(ilbl(\varepsilon'_i)(\widetilde{label}(U) \mathbin{\widetilde{\curlyvee}} g))$ holds or do not hold. Finally as $\langle \phi_1, v'_1, \mu'_1\rangle \approx^{k-j'}_{\ell_o} \langle \phi_2, v'_2, \mu'_2\rangle : U'_1$, by Lemma 6.59,

$$\begin{array}{rl} & \langle \phi_1, \mathsf{prot}^{g,U_1}_{ilbl(\varepsilon'_i)g''_1}\phi''_1(iref(\varepsilon'_1)v'_1), \mu'_1\rangle \\ \approx^j_{\ell_o} & \langle \phi_2, \mathsf{prot}^{g,U_1}_{ilbl(\varepsilon'_i)g''_2}\phi''_2(iref(\varepsilon'_2)v'_2), \mu'_2\rangle : \mathcal{C}(U'_2) \end{array}$$

and finally the result holds by backward preservation of the relations (Lemma 6.43).

———

Case (:=). $t^U = \varepsilon_1 t_1^{U_1} \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 t_2^{U_2}$. Then $U = \mathsf{Unit}_\perp$.
By definition of substitution:

$$\rho_i(t^U) = \varepsilon_1 \rho_i(t^{U_1}) \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 \rho_i(t^{U_2})$$

and Lemma 6.42:

$$\phi_i' \triangleright \varepsilon_1 \rho_i(t^{U_1}) \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 \rho_i(t^{U_2}) \in \mathbb{T}[\mathsf{Unit}_\perp]$$

We have to show that

$$\begin{array}{ll} & \langle \phi_1, \varepsilon_1 \rho_1(t^{U_1}) \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 \rho_1(t^{U_2}), \mu_1 \rangle \\ \approx_{\ell_o}^{k} & \langle \phi_2, \varepsilon_1 \rho_2(t^{U_1}) \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 \rho_2(t^{U_2}), \mu_2 \rangle : \mathcal{C}(U) \end{array}$$

By induction hypotheses

$$\langle \phi_1, \rho_1(t^{U_1}), \mu_1 \rangle \approx_{\ell_o}^{k} \langle \phi_2, \rho_2(t^{U_1}), \mu_2 \rangle : \mathcal{C}(U_1)$$

Suppose $j_1 < k$, and that $\rho_i(t^{U_1})$ are irreducible after j_1 steps (otherwise, similar to case !, the result holds immediately). Then by definition of related computations:

$$\rho_i(t^{U_1}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1} v_i \mid \mu_i' \implies \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \langle \phi_1, v_1, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \phi_2, v_2, \mu_2' \rangle : U_1$$

By Lemma 6.61 $\mu_i \twoheadrightarrow \mu_i'$, and $\mu_1' \approx_{\ell_o}^{k-j_1} \mu_2'$ then by Lemma 6.41, $\langle \phi_1, \rho_1, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \phi_2, \rho_2, \mu_2' \rangle$. By induction hypotheses:

$$\langle \phi_1, \rho_1(t^{U_2}), \mu_1' \rangle \approx_{\ell_o}^{k} \langle \phi_2, \rho_2(t^{U_2}), \mu_2' \rangle : \mathcal{C}(U_2)$$

Again, consider $j_2 = k - j_1$, if after j_2 steps $\rho_1(t^{U_2})$ is reducible or is a value, the result holds immediately. The interest case if after $j_2' < j_2$ steps $\rho_1(t^{U_2})$ reduces to values v_i':

$$\rho_i(t^{U_2}) \mid \mu_i' \overset{\phi_i'}{\longmapsto}{}^{j_2'} v_i' \mid \mu_i'' \implies \mu_1'' \approx_{\ell_o}^{k-j_1-j_2'} \mu_2'' \wedge \langle \phi_1, v_1', \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2'} \langle \phi_2, v_2', \mu_2'' \rangle : U_2$$

Then

$$\rho_i(t^U) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1+j_2'} \varepsilon_1 v_i \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 v_i' \mid \mu_i'' \wedge \mu_1'' \approx_{\ell_o}^{k-j_1-j_2'} \mu_2''$$

Now v_i and v_i' can be bare values or casted values. In the case of casted values we can combine evidence, which may fail with **error**. We assume that all evidence combinations succeed, otherwise the relation vacuously holds. As both values v_i are related at some reference type, then by canonical forms (Lemma 6.10) they both must be locations $o^{U_1'}$ for some $U_1' \lesssim U_1$. We consider when the values are observable and the locations are identical (otherwise the result is trivial):

$$\begin{array}{ll} & \varepsilon_1 v_i \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_2 v_i' \mid \mu_i'' \\ \overset{\phi_i'}{\longmapsto}{}^{2} & \varepsilon_{1i}' o_g^{U_1'} \overset{g,U_1'}{:=}_{\varepsilon_\ell} \varepsilon_{2i}' u_i' \mid \mu_i'' \\ \overset{\phi_i'}{\longmapsto}{}^{1} & \mathsf{unit}_\perp \mid \mu_i''' \end{array}$$

Where $\mu_i''' = \mu_i''[o^{U_1'} \mapsto \varepsilon_{1i}''(u_i' \mathbin{\widetilde{\curlyvee}} (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U_1']$. As
$\langle \phi_1, v_1', \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2'} \langle \phi_2, v_2', \mu_2'' \rangle : U_2$ then by Lemma 6.45,
$\langle \phi_1, \varepsilon_{21}'' u_1' :: U_1', \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2'} \langle \phi_2, \varepsilon_{22}'' u_2' :: U_1', \mu_1'' \rangle : U_1'$. As $\varepsilon_i' \vdash \phi'.\widetilde{g_c \curlyvee g \preccurlyeq \mathit{label}(U_1')}$ and $\varepsilon_{1i}'' = \varepsilon_{2i}'' \mathbin{\widetilde{\curlyvee}} \varepsilon_i'$, by Lemma 6.58,

$$\begin{array}{ll} & \langle \phi_1, \varepsilon_{11}''(u_1' \mathbin{\widetilde{\curlyvee}} (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U_1', \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2'} & \langle \phi_2, \varepsilon_{12}''(u_2' \mathbin{\widetilde{\curlyvee}} (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U_1', \mu_1'' \rangle \end{array}$$

Also if $\neg\mathsf{obs}_{\ell_o}(\phi_i) \Rightarrow \neg\mathsf{obs}_{\ell_o}(\phi_i')$ and therefore by monotonicity of the join $\neg\mathsf{obs}_{\ell_o}(\varepsilon_{1i}'' \widetilde{label}(U_1'))$. Therefore if the values where different but context not observables, now the new values are going to be not observable as well, independently of the context. Then $\forall, \phi_1'' \approx_{\ell_o}^{k} \phi_2''$,

$$\begin{array}{ll} & \langle \phi_1'', \varepsilon_{11}''(u_1' \mathbin{\widetilde{\curlyvee}} (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U_1', \mu_1'' \rangle \\ \approx_{\ell_o}^{k-j_1-j_2'} & \langle \phi_2'', \varepsilon_{12}''(u_2' \mathbin{\widetilde{\curlyvee}} (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)) :: U_1', \mu_1'' \rangle \end{array}$$

As every values are related at type Unit, we only have to prove that $\mu_1''' \approx_{\ell_o}^{k-j_1-j_2'-3} \mu_1'''$, but using monotonicity (Lemma 6.46), it is trivial to prove that because either both both stores update the same location $o^{U_1'}$ to values that are related, therefore the result holds.

———

Case (ref). $t^U = \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon t^{U_1'}$. Then $U = \mathsf{Ref}_\perp\ U_1$.
By definition of substitution:

$$\rho_i(t^U) = \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_i(t^{U_1'})$$

and Lemma 6.42:

$$\phi_i' \triangleright \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_i(t^{U_1'}) \in \mathbb{T}[\mathsf{Ref}_\perp\ U_1]$$

We have to show that

$$\begin{array}{ll} & \langle \phi_1, \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_1(t^{U_1'}), \mu_1 \rangle \\ \approx_{\ell_o}^{k} & \langle \phi_2, \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_2(t^{U_1'}), \mu_2 \rangle : \mathcal{C}(\mathsf{Ref}_\perp\ U_1) \end{array}$$

By induction hypotheses:

$$\langle \phi_1, \rho_1(t^{U_1'}), \mu \rangle \approx_{\ell_o}^{k} \langle \phi_2, \rho_2(t^{U_1'}), \mu \rangle : \mathcal{C}(U_1')$$

Consider $j < k$, by definition of related computations

$$\rho_i(t^{U_1'}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j} t_i^{U_1'} \mid \mu_i' \implies \mu_1' \approx_{\ell_o}^{k-j} \mu_2' \wedge (irred(t_i^{U_1'}) \Rightarrow \langle \phi_1, t_1^{U_1'}, \mu_1' \rangle \approx_{\ell_o}^{k-j} \langle \phi_2, t_2^{U_1'}, \mu_2' \rangle : U_1')$$

If terms $t_i^{U_1'}$ are reducible after $j = k - 1$ steps, then
$\mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_i(t^{U_1'}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j} \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon t_i'^{U_1'} \mid \mu_i'$ and the result holds.
If after at most j steps $t_i^{U_1'}$ is irreducible, it means that for some $j' \leq j$ $\mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon \rho_i(t^{U_1'}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j'} \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon v_i \mid \mu_i'$. If $j' = j$ then we use the same same argument for reducible terms and the result holds.
Let us consider now $j' < j$. By Lemma 6.10, each v_i is either a base value u_i or a casted base value $\varepsilon_i u_i :: U_1'$. In case a value v_{ij} is a casted value, then the whole term $\rho_i(t^U)$ can take a step by (Rg), combining ε with ε_i. Such a step either fails, or succeeds with a new combined evidence. Therefore, either:

$$\rho_i(t^U) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j'} \mathbf{error}$$

in which case we do not care since we only consider termination-insensitive noninterference, or:

$$\begin{array}{lll} \rho_i(t^U) \mid \mu & \overset{\phi_i'}{\longmapsto}{}^{j'+1} & \mathsf{ref}_{\varepsilon_\ell}^{U_1}\ \varepsilon_i' u_i \mid \mu_i' \\ & \overset{\phi_i'}{\longmapsto}{}^{1} & o_\perp^{U_1} \mid \mu_i'' \end{array}$$

with, $\mu_i'' = \mu_i'[o^{U_1} \mapsto \varepsilon_i'(u_i \mathbin{\widetilde{\curlyvee}} \phi_i' g_c) :: U_1]$. Where $\varepsilon_i'' = \varepsilon_i' \mathbin{\widetilde{\curlyvee}} (\phi'.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell)$. We know that if $u_i \in \mathbb{T}[U_i]$, then $\varepsilon_i \vdash U_i \lesssim U_1$. Also, as $\langle \phi_1, v_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'} \langle \phi_2, v_2, \mu_2' \rangle : U_1'$ then by Lemma 6.45,
$\langle \phi_1, \varepsilon_1' u_1 :: U_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'} \langle \phi_2, \varepsilon_2' u_2 :: U_1, \mu_2' \rangle : U_1'$ and as $(\phi'.\varepsilon \circ^{\preccurlyeq} \varepsilon_\ell) \vdash \phi'.g_c \mathrel{\widetilde{\preccurlyeq}} label(U_1)$, then by

Lemma 6.58, Lemma 6.53, and Lemma 6.46,
$\langle \phi_1, \varepsilon_1''(u_1 \mathbin{\widetilde{\curlyvee}} \phi_1' g_c) :: U_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'-2} \langle \phi_2, \varepsilon_2''(u_2 \mathbin{\widetilde{\curlyvee}} \phi_2' g_c) :: U_1, \mu_2' \rangle : U_1'$.

Also if $\neg\mathrm{obs}_{\ell_o}(\phi_i) \Rightarrow \neg\mathrm{obs}_{\ell_o}(\phi_i')$ and therefore by monotonicity of the join $\neg\mathrm{obs}_{\ell_o}(\varepsilon_i''\widetilde{label}(U_1))$. Therefore if the values where different but context not observables, now the new values are going to be not observable as well, independently of the context. Then
$\forall, \phi_1'' \approx_{\ell_o}^{k} \phi_2'', \langle \phi_1'', \varepsilon_1''(u_1 \mathbin{\widetilde{\curlyvee}} \phi_1' g_c) :: U_1, \mu_1' \rangle \approx_{\ell_o}^{k-j'-2} \langle \phi_2'', \varepsilon_2''(u_2 \mathbin{\widetilde{\curlyvee}} \phi_2' g_c) :: U_1, \mu_2' \rangle : U_1'$.

By definition of related stores $\mu_1'' \approx_{\ell_o}^{k-j'} \mu_2''$. Then by Monotonicity of the relation (Lemma 6.46) $\mu_1'' \approx_{\ell_o}^{k-j'-2} \mu_2''$ and the result holds.

———

Case ($\oplus$). $t^U = \varepsilon_1 t^{U_1} \ \oplus^g \ \varepsilon_2 t^{U_2}$

By definition of substitution:

$$\rho_i(t^U) = \varepsilon_1 \rho_i(t^{U_1}) \ \oplus^g \ \varepsilon_2 \rho_i(t^{U_2})$$

and Lemma 6.42:

$$\phi_i' \triangleright \varepsilon_1 \rho_i(t^{U_1}) \ \oplus^g \ \varepsilon_2 \rho_i(t^{U_2}) \in \mathbb{T}[U]$$

We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k - 3$ where:

$$\rho_i(t^{U_1}) \mid \mu_i \xrightarrow{\phi_i'}{}^{j_1} v_{i1} \mid \mu_i' \implies \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \langle \phi_1, v_{11}, \mu_1' \rangle \approx_{\ell_o}^{k-j_1} \langle \phi_2, v_{21}, \mu_2' \rangle : U_1$$

$$\rho_i(t^{U_2}) \mid \mu_i' \xrightarrow{\phi_i'}{}^{j_2} v_{i2} \mid \mu_i'' \implies \mu_1'' \approx_{\ell_o}^{k-j_1-j_2} \mu_2'' \wedge \langle \phi_1, v_{12}, \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2} \langle \phi_2, v_{22}, \mu_2'' \rangle : U_2$$

By Lemma 6.10, each v_{ij} is either a boolean $(b_{ij})_{g_{ij}}$ or a casted boolean $\varepsilon_{ij}(b_{ij})_{g_{ij}'} :: U_j$. In case a value v_{ij} is a casted value, then the whole term $\rho_i(t^U)$ can take a step by (Rg), combining ε_i with ε_{ij}. Such a step either fails, or succeeds with a new combined evidence. Therefore, either:

$$\rho_i(t^U) \mid \mu_i \xrightarrow{\phi_i'}{}^{j_1+j_2} \mathbf{error}$$

in which case we do not care since we only consider termination-insensitive noninterference, or:

$$\begin{aligned} & \rho_i(t^U) \mid \mu_i'' \\ \longmapsto^{j_1+j_2+2} \ & \varepsilon_{i1}'(b_{i1})_{g_{i1}'} \ \oplus^g \ \varepsilon_{i2}'(b_{i2})_{g_{i2}'} \mid \mu_i'' \\ \longmapsto^{1} \ & \varepsilon_i'(b_i)_{g_i'} :: \mathsf{Bool}_g \mid \mu_i'' \end{aligned}$$

with $b_i = b_{i1} [\![\oplus]\!] b_{i2}$, $\varepsilon_i' = \varepsilon_{i1}' \mathbin{\widetilde{\curlyvee}} \varepsilon_{i2}'$, and $g_i' = g_{i1}' \mathbin{\widetilde{\curlyvee}} g_{i2}'$. It remains to show that:

$$\langle \phi_1, \varepsilon_1'(b_1)_{g_1'} :: \mathsf{Bool}_g, \mu_1'' \rangle \approx_{\ell_o}^{k-j_1-j_2-3} \langle \phi_2, \varepsilon_2'(b_2)_{g_2'} :: \mathsf{Bool}_g, \mu_2'' \rangle : \mathsf{Bool}_g$$

If $\neg\mathrm{obs}_{\ell_o}(\phi_i)$, then the result is trivial because the resulting booleans are also related as they are not observable.

If $\mathrm{obs}_{\ell_o}(\phi_i)$, then by Lemma 6.45, $\langle \phi_1, \varepsilon_{11}'(b_{11})_{g_{11}'} :: \mathsf{Bool}_g, \mu_1'' \rangle \approx_{\ell_o}^{k} \langle \phi_2, \varepsilon_{21}'(b_{21})_{g_{21}'} :: \mathsf{Bool}_g, \mu_2'' \rangle$. If $\neg\mathrm{obs}_{\ell_o}(ilbl(\varepsilon_{i1}')g)$ or $\neg\mathrm{obs}_{\ell_o}(ilbl(\varepsilon_{i2}')g)$, then by Lemma 6.55, $\neg\mathrm{obs}_{\ell_o}(\varepsilon_i' g)$ and the result holds. If both $\mathrm{obs}_{\ell_o}(ilbl(\varepsilon_{i2}')g)$ then $b_{11} = b_{21}$ and $b_{12} = b_{22}$, so $b_1 = b_2$, and the result holds.

———

Case (app). $t^U = \varepsilon_1 t^{U_1} \mathbin{@^{U_{11} \xrightarrow{g_c'}_g U_{12}}_{\varepsilon_\ell}} \varepsilon_2 t^{U_2}$
with $\varepsilon_1 \vdash U_1 \lesssim S_{11} \rightarrow_g S_{12}$, $\varepsilon_2 \vdash U_2 \lesssim U_{11}$, and $U = U_{12} \mathbin{\widetilde{\curlyvee}} g$.
We omit the $@^{U_{11} \xrightarrow{g_c'}_g U_{12}}_{\varepsilon_\ell}$ operator in applications below.

By definition of substitution:

$$\rho_i(t^U) = \varepsilon_1 \rho_i(t^{U_1})\ \varepsilon_2 \rho_i(t^{U_2})$$

and Lemma 6.42:

$$\phi_i' \triangleright \varepsilon_1 \rho_i(t^{U_1})\ \varepsilon_2 \rho_i(t^{U_2}) \in \mathbb{T}[U]$$

We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k$ where by induction hypotheses and the definition of related computations:

$$\rho_i(t^{U_1}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1} v_{i1} \mid \mu_i' \implies \mu_1' \approx^{k-j_1}_{\ell_o} \mu_2' \wedge \langle \phi_1, v_{11}, \mu_1' \rangle \approx^{k-j_1}_{\ell_o} \langle \phi_2, v_{21}, \mu_2' \rangle : U_1$$

$$\rho_i(t^{U_2}) \mid \mu_i' \overset{\phi_i'}{\longmapsto}{}^{j_2} v_{i2} \mid \mu_i'' \implies \mu_1'' \approx^{k-j_1-j_2}_{\ell_o} \mu_2'' \wedge \langle \phi_1, v_{12}, \mu_1'' \rangle \approx^{k-j_1-j_2}_{\ell_o} \langle \phi_2, v_{22}, \mu_2'' \rangle : U_2$$

Then

$$\rho_i(t^U) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1+j_2} \varepsilon_1 v_{11}\ \varepsilon_2 v_{12} \mid \mu_i''$$

If $\mathsf{obs}_{\ell_o}(\phi_i \triangleright v_{i1})$ then, by definition of $\approx_{\ell_o}$ at values of function type, using ε_1 and ε_2 to justify the subtyping relations, we have:

$$\begin{array}{rl} & \langle \phi_1, (\varepsilon_1 v_{11}\ \varepsilon_2 v_{12}), \mu_1'' \rangle \\ \approx^{k-j_1-j_2}_{\ell_o} & \langle \phi_2, (\varepsilon_1 v_{21}\ \varepsilon_2 v_{22}), \mu_2'' \rangle : \mathcal{C}(U_{12} \mathbin{\widetilde{\curlyvee}} g) \end{array}$$

Finally, by backward preservation of the relations (Lemma 6.43) the result holds.

If $\neg\mathsf{obs}_{\ell_o}(\phi_i \triangleright v_{i1})$, and we assume by canonical forms that $v_{i1} = \varepsilon_{i1}(\lambda^{g_i'} x.t_i)_{g_i} :: U_1$ and that $v_{i2} = \varepsilon_{i2} u_{i2} :: U_2$ (and that evidence combination always succeed or the result holds immediately), then,

$$\begin{array}{rl} & (\varepsilon_1 v_{i1}\ \varepsilon_2 v_{i2}) \mid \mu_1'' \\ \overset{\phi_i'}{\longmapsto}{}^{1} & (\varepsilon_{i1}'(\lambda^{g_i'} x.t_i)_{g_i}\ \varepsilon_{i2}' u_{i2}) \mid \mu_1'' \\ \overset{\phi_i'}{\longmapsto}{}^{1} & \mathsf{prot}^{g_c', U_{12}}_{ilbl(\varepsilon_{i1}') g_i} \phi_i''(icod(\varepsilon_{i1}') t_i') \mid \mu_1'' \end{array}$$

Where $\varepsilon_{i1}' = \varepsilon_{i1} \circ^{\preccurlyeq} \varepsilon_1$, $\varepsilon_{i2}' = \varepsilon_{i2} \circ^{\preccurlyeq} \varepsilon_2$, and $\phi_i'' = \langle \varepsilon_i''(\phi_i' g_c \mathbin{\widetilde{\curlyvee}} g_i), g_i' \rangle$, $\varepsilon_i'' = (\phi_i' \varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{i1}')) \circ^{\preccurlyeq} \varepsilon_\ell \circ^{\preccurlyeq} ilat(\varepsilon_{i1}'))$.

If $\neg\mathsf{obs}_{\ell_o}(\phi_i)$ then $\neg\mathsf{obs}_{\ell_o}(\phi_i')$ and by Lemma 6.55 and 6.53, $\neg\mathsf{obs}_{\ell_o}(\phi_i'')$. As $\varepsilon_{i1}' = \varepsilon_{i1} \circ^{\preccurlyeq} \varepsilon_1$, by Lemma 6.51, either both $ilbl(\varepsilon_{i1}')$ are observable or not (the latter when $\neg\mathsf{obs}_{\ell_o}(ilbl(\varepsilon_{i1})\widetilde{label}(U_1))$). If $\neg\mathsf{obs}_{\ell_o}(ilbl(\varepsilon_{i1})\widetilde{label}(U_1))$ then similar to the context case, $\neg\mathsf{obs}_{\ell_o}(\phi_i'')$.

Finally by Lemma 6.59,

$$\begin{array}{rl} & \langle \phi_1, \mathsf{prot}^{g_c', U_{12}}_{ilbl(\varepsilon_{11}') g_1} \phi_1''(icod(\varepsilon_{11}') t_1'), \mu_1'' \rangle \\ \approx^{k-j_1-j_2}_{\ell_o} & \langle \phi_2, \mathsf{prot}^{g_c', U_{12}}_{ilbl(\varepsilon_{21}') g_2} \phi_2''(icod(\varepsilon_{21}') t_2'), \mu_2'' \rangle : \mathcal{C}(U_{12} \mathbin{\widetilde{\curlyvee}} g) \end{array}$$

Finally, by backward preservation of the relations (Lemma 6.43) the result holds.

———

Case (if). $t^U = \mathsf{if}^g\ \varepsilon_1 t^{U_1}$ then $\varepsilon_2 t^{U_2}$ else $\varepsilon_3 t^{U_3}$, with $\phi_i' \triangleright t^{U_1} \in \mathbb{T}[U_1]$, $g' = label(U_1)$, $\varepsilon_{ri}' = (\phi_i.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_1))$, $\phi_i'' = \langle \varepsilon_r'(\phi'.g_c \mathbin{\widetilde{\curlyvee}} g'), (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)\rangle$ $\phi_i'' \triangleright t^{U_2} \in \mathbb{T}[U_2]$, $\phi_i'' \triangleright t^{U_3} \in \mathbb{T}[U_3]$ $\varepsilon_1 \vdash U_1 \lesssim \mathsf{Bool}_g$ and $U = (U_2 \mathbin{\widetilde{\curlyvee}} U_3) \mathbin{\widetilde{\curlyvee}} g$

By definition of substitution:

$$\rho_i(t^U) = \mathsf{if}^g\ \varepsilon_1 \rho_i(t^{U_1}) \text{ then } \varepsilon_2 \rho_i(t^{U_2}) \text{ else } \varepsilon_3 \rho_i(t^{U_3})$$

We use a similar argument to case := for reducible terms. The interest case is when we suppose some j_1 and j_2 such that $j_1 + j_2 < k$ where by induction hypotheses and related computations we have that:

$$\rho_i(t^{U_1}) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1} v_{i1} \mid \mu_i' \implies \mu_1' \approx_{\ell_o}^{k-j_1} \mu_2' \wedge \langle \phi_1 \triangleright v_{11}, \mu_1'\rangle \approx_{\ell_o}^{k-j_1} \langle \phi_2 \triangleright v_{21}, \mu_2'\rangle : U_1$$

By Lemma 6.10, each v_{i1} is either a boolean $(b_{i1})_{g_{i1}}$ or a casted boolean $\varepsilon_{i1}(b_{i1})_{g_{i1}'} :: U_1$. In either case, $U_1 \lesssim \mathsf{Bool}_{g_1}$ implies $U_1 = \mathsf{Bool}_{g_1'}$. In case a value v_{i1} is a casted value, then the whole term $\rho_i(t^U)$ can take a step by (Rg), combining ε_i with ε_{i1}. Such a step either fails, or succeeds with a new combined evidence. Therefore, either:

$$\rho_i(t^U) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1+1} \mathbf{error}$$

in which case we do not care since we only consider termination-insensitive noninterference, or:

$$\rho_i(t^U) \mid \mu_i \overset{\phi_i'}{\longmapsto}{}^{j_1+1} \mathsf{if}^g\ \varepsilon_{i1}'(b_{i1})_{g_{i1}'} \text{ then } \varepsilon_2 \rho_i(t^{U_2}) \text{ else } \varepsilon_3 \rho_i(t^{U_3}) \mid \mu_i'$$

If $\mathsf{obs}_{\ell_o}(\phi_i \triangleright v_{i1})$ does not hold, then by Lemma 6.63 $\mathsf{obs}_{\ell_o}(\phi_i \triangleright \varepsilon_{i1}' b_{i1} :: \mathsf{Bool}_{g_1'})$ is not observable. Let us assume the worst case scenario and that both execution reduce via different branches of the conditional.

Consider $\phi_i'' = \langle (\phi'.\varepsilon \mathbin{\widetilde{\curlyvee}} ilbl(\varepsilon_{i1}'))(\phi'.g_c \mathbin{\widetilde{\curlyvee}} g_{i1}'), (\phi'.g_c \mathbin{\widetilde{\curlyvee}} g)\rangle$. It is easy to see that if ϕ_i is not observable, then as $\leq_{\ell_o}$ ϕ_i' is not observable, and therefore by Lemma 6.55, $\mathsf{obs}_{\ell_o}(\phi_i''.\varepsilon\phi_i''.g_c)$ does not hold. Therefore $\phi_i \leq_{\ell_o} \phi_i''$. If $\mathsf{obs}_{\ell_o}(\varepsilon_{i1}'\mathsf{Bool}_{g_1'})$ does not hold, then also by Lemma 6.55, $\mathsf{obs}_{\ell_o}(\phi_i''.\varepsilon\phi_i''.g_c)$ does not hold as well. Then

$$\rho_1(t^U) \mid \mu_1 \overset{\phi_i'}{\longmapsto}{}^{j_1+2} \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{11}')g_{11}'} \phi_1''(\varepsilon_2 \rho_1(t^{U_2})) \mid \mu_1'$$

$$\rho_2(t^U) \mid \mu_2 \overset{\phi_i'}{\longmapsto}{}^{j_1+2} \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{21}')g_{21}'} \phi_2''(\varepsilon_3 \rho_2(t^{U_3})) \mid \mu_2'$$

But because $\mathsf{obs}_{\ell_o}(\phi \triangleright \varepsilon_{i1}' b_{i1} :: \mathsf{Bool}_{g_1'})$ does not hold then either $\mathsf{obs}_{\ell_o}(\phi.\varepsilon\phi.g_c)$ or $\mathsf{obs}_{\ell_o}(ilbl(\varepsilon_{i1}'g))$ does not hold. Then as $\phi_i \leq_{\ell_o} \phi_i''$ by Lemma 6.59,

$$\langle \phi_1, \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{11}')g_{11}'} \phi_1''(\varepsilon_2 \rho_1(t^{U_2})), \mu_1'\rangle \approx_{\ell_o}^k \langle \phi_2, \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{21}')g_{21}'} \phi_2''(\varepsilon_3 \rho_2(t^{U_3})), \mu_2'\rangle$$

and the result holds by backward preservation of the relations (Lemma 6.43).

Now consider if $\mathsf{obs}_{\ell_o}(\phi \triangleright v_{i1})$ holds, then $\mathsf{obs}_{\ell_o}(\phi \triangleright \varepsilon_{i1}' b_{i1} :: \mathsf{Bool}_{g_1'})$ may hold or not. If its not observable we proceed like we just did for the non-observable case. Let us consider that $\mathsf{obs}_{\ell_o}(\phi \triangleright \varepsilon_{i1}' b_{i1} :: \mathsf{Bool}_{g_1'})$ holds.

Then by definition of $\approx_{\ell_o}$ on boolean values, $b_{11} = b_{21}$ Because $b_{11} = b_{21}$, both $\rho_1(t^U)$ and $\rho_2(t^U)$ step into the same branch of the conditional. Let us assume the condition is true (the other case is similar):

Then by induction hypotheses $\langle \phi_1, \rho_1(t^{U_2}), \mu_1'\rangle \approx_{\ell_o}^k \langle \phi_2, \rho_2(t^{U_2}), \mu_2'\rangle$, then as $\phi_i \leq_{\ell_o} \phi_i''$, by Lemma 6.60,

$$\langle \phi_1, \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{11}')g_{11}'} \phi_1''(\varepsilon_2 \rho_1(t^{U_2})), \mu_1'\rangle \approx_{\ell_o}^k \langle \phi_2, \mathsf{prot}^{g,U}_{ilbl(\varepsilon_{21}')g_{21}'} \phi_2''(\varepsilon_3 \rho_2(t^{U_2})), \mu_2'\rangle$$

and the result holds by backward preservation of the relations (Lemma 6.43).

Case (prot()). Direct by using Lemma 6.60.

□

REFERENCES

Alonzo Church. 1940. A Formulation of the Simple Theory of Types. *J. Symbolic Logic* 5, 2 (06 1940), 56–68.

Ronald Garcia, Alison M. Clark, and Éric Tanter. 2016. Abstracting Gradual Typing. In *43rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2016)*. ACM Press, St Petersburg, FL, USA, 429–442.

Jeremy G. Siek and Walid Taha. 2006. Gradual Typing for Functional Languages. In *Scheme and Functional Programming Workshop*. 81–92.

Steve Zdancewic. 2002. *Programming Languages for Information Security*. Ph.D. Dissertation. Cornell University.