Gradual Union Types

Complete Definition and Proofs

Technical Report TR/DCC-2017-1
University of Chile
June 2017

Matías Toro and Éric Tanter

PLEIAD Laboratory
Computer Science Department (DCC)
University of Chile

Abstract. This report presents the complete definitions and proofs of GTFL$^{\oplus}$, a gradual language that combines both gradual unions and the traditional, totally-unknown type. Gradual union types combine the benefits of both tagged and untagged unions, with added static flexibility backed by runtime checks. Compared to a standard gradually-typed language with only the totally-unknown type ?, the resulting design is stricter, allowing more blatantly wrong programs to be statically rejected. We also present a correct compilation scheme to GTFL$^{\oplus}_{\Rightarrow}$, an internal language with support to threesomes, a space-efficient representation for casts.

Table of Contents

1 Overview

In this document we present the full definitions and proofs of the static language STFL, the gradual language GTFL$^{\oplus}$ and the intermediate language GTFL$^{\oplus}_{\Rightarrow}$. Section 2 presents the full definition and proofs of the static language STFL: the static semantics, the dynamic semantics and its properties. Section 3 presents the full definition and proofs of GTFL$^{\oplus}$: the Galois connections are presented in Section 3.1 , the static semantics in Section 3.2, and the dynamic semantics in Section 3.3 Section 4 presents properties of GTFL$^{\oplus}$ and their proofs: the static gradual guarantee in Section 4.1, type safety in Section 4.2 and finally the dynamic gradual guarantee in Section 4.3. Section 5 presents the full definition and proofs of the compilation of GTFL$^{\oplus}$ to a threesome calculus. First we present the intermediate language GTFL$^{\oplus}_{\Rightarrow}$ in Section 5.1. Section 5.2 presents the cast insertion rules, and Section 5.3 presents the full formalization of the correctness of the translational semantics.

2 The Static Language: STFL

In this section we present the full definition of STFL. Figure 1 presents the syntax and type system, and Figure 2 presents the dynamic semantics. The rest of this section present the type safety proof of STFL.

Lemma 1 (Substitution). *If $\Gamma, x : T_1 \vdash t : T$ and $\Gamma \vdash v : T_1'$ such that $T_1' = T_1$, then $\Gamma \vdash [v/x]t : T'$ such that $T' = T$.*

Proof. By induction on the derivation of $\Gamma, x : T_1 \vdash t : T$.

Proposition 1 ($\longrightarrow$ is well defined). *If $\Gamma \vdash t : T$, $t \longrightarrow t'$ then, $\Gamma \vdash t' : T'$, where $T' = T$.*

Proof. Case (T+). Then $t = b_1 + b_2$ and

$$(\text{T+})\frac{(\text{T+})\ \dfrac{\mathcal{D}_1}{\Gamma \vdash n_1 : T_1} \qquad (\text{T+})\ \dfrac{\mathcal{D}_2}{\Gamma \vdash n_2 : T_2} \qquad T_1 = \mathsf{Int} \qquad T_2 = \mathsf{Int}}{\Gamma \vdash n_1 + n_2 : \mathsf{Int}}$$

Then

$$n_1 + n_2 \longrightarrow (n_1 \,[\![+]\!]\, n_2)$$

But $\Gamma \vdash (n_1 \,[\![+]\!]\, n_2) : \mathsf{Int}$ and the result holds.

$$T \in \textsc{Type}, \quad x \in \textsc{Var}, \quad t \in \textsc{Term}, \quad \Gamma \in \textsc{Var} \xrightarrow{\text{fin}} \textsc{Type}$$

$$\begin{array}{lll} T ::= \mathsf{Bool} \mid \mathsf{Int} \mid T \to T & & \text{(types)} \\ v ::= n \mid b \mid (\lambda x : T.t) & & \text{(values)} \\ t ::= v \mid x \mid t\ t \mid t + t \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t \mid t :: T & & \text{(terms)} \end{array}$$

$$(\text{Tx})\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \qquad (\text{T}b)\frac{}{\Gamma \vdash b : \mathsf{Bool}} \qquad (\text{T}n)\frac{}{\Gamma \vdash n : \mathsf{Int}}$$

$$(\text{T}\lambda)\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma; \Sigma \vdash (\lambda x : T_1.t) : T_1 \to T_2} \qquad (\text{T::})\frac{\Gamma \vdash t : T \quad T = T_1}{\Gamma \vdash (t :: T_1) : T_1}$$

$$(\text{Tapp})\frac{\begin{array}{c}\Gamma \vdash t_1 : T_1 \\ \Gamma \vdash t_2 : T_2 \quad T_2 = dom(T_1)\end{array}}{\Gamma \vdash t_1\ t_2 : cod(T_1)} \qquad (\text{T}+)\frac{\begin{array}{cc}\Gamma \vdash t_1 : T_1 & \Gamma \vdash t_2 : T_2 \\ T_1 = \mathsf{Int} & T_2 = \mathsf{Int}\end{array}}{\Gamma \vdash t_1 + t_2 : \mathsf{Int}}$$

$$(\text{Tif})\frac{\Gamma \vdash t_1 : T_1 \quad T_1 = \mathsf{Bool} \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3}{\Gamma \vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : equate(T_2, T_3)}$$

$dom : \textsc{Type} \rightharpoonup \textsc{Type}$	$cod : \textsc{Type} \rightharpoonup \textsc{Type}$	$equate : \textsc{Type}^2 \rightharpoonup \textsc{Type}$
$dom(T_1 \to T_2) = T_1$	$cod(T_1 \to T_2) = T_2$	$equate(T, T) = T$
$dom(T)$ undefined o.w.	$cod(T)$ undefined o.w.	$equate(T_1, T_2)$ undefined o.w.

Fig. 1. STFL: Syntax and Type System

Case (Tapp). Then $t = (\lambda x : T_{11}.t_1)\ v$, suppose $\Gamma \vdash (\lambda x : T_{11}.t_1) : T_1$, and $dom(T_1) = T_{11}$ and $cod(T_1) = T_{12}$. Therefore

$$(\text{Tapp})\frac{(\text{T}\lambda)\ \dfrac{\dfrac{\mathcal{D}_1}{\Gamma, x : T_{11} \vdash t : T_{12}}}{\Gamma \vdash (\lambda x : T_{11}.t_1) : T_{11} \longrightarrow T_{12}} \qquad \dfrac{\dfrac{\mathcal{D}_2}{\Gamma \vdash v : T_2} \qquad T_2 = T_{11}}{}}{\Gamma \vdash (\lambda x : T_{11}.t_1)\ v : T_{12}}$$

Then

$$(\lambda x : T_{11}.t_1)\ v \longrightarrow [v/x]t_1$$

By Lemma 1, $\Gamma \vdash [v/x]t_1 : T'_{12}$, where $T'_{12} = T_{12}$, and the result holds.

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

$$(\text{Sif})\frac{\dfrac{\mathcal{D}_0}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}} \qquad \dfrac{\mathcal{D}_1}{\Gamma \vdash t_1 : T_1} \qquad \dfrac{\mathcal{D}_2}{\Gamma \vdash t_2 : T_2}}{\Gamma \vdash \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 : equate(T_1, T_2)}$$

Then if

$$\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \longrightarrow t_1$$

$$
\begin{array}{lll}
v ::= n \mid \lambda x.t \mid \mathsf{true} \mid \mathsf{false} \mid v :: T & & \text{(values)} \\
f ::= \Box + t \mid v + \Box \mid \Box\ t \mid v\ \Box \mid \Box :: T & & \text{(frames)} \\
\quad \mathsf{if}\ \Box\ \mathsf{then}\ t\ \mathsf{else}\ t &&
\end{array}
$$

$\boxed{t \longrightarrow t}$ **Notions of Reduction**

$$
\begin{array}{l}
n_1 + n_2 \longrightarrow n_3 \text{ where } n_3 = n_1 \,[\![+]\!]\, n_2 \\
(\lambda x.t)\ v \longrightarrow ([v/x]t) \\
\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \longrightarrow t_1 \\
\mathsf{if}\ \mathsf{false}\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \longrightarrow t_2 \\
v :: T \longrightarrow v
\end{array}
$$

$\boxed{t \longmapsto t}$ **Reduction**

$$
\frac{t_1 \longrightarrow t_2}{t_1 \longmapsto t_2} \qquad\qquad \frac{t_1 \longmapsto t_2}{f[t_1] \longmapsto f[t_2]}
$$

Fig. 2. STFL: Dynamic Semantics

But

$$
\frac{\mathcal{D}_1}{\Gamma \vdash t_1 : T_1}
$$

and by definition of the *equate* operator, $T_1 = \mathit{equate}(T_1, T_2)$ and the result holds.

Case (Tif-false*).* Analogous to case (if-true).

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

$$
(\mathrm{T::})\frac{\dfrac{\mathcal{D}}{\Gamma \vdash v : T_1} \qquad T_1 = T}{\Gamma \vdash v :: T : T}
$$

Then

$$
v :: T \ \longrightarrow\ v
$$

But $T_1 = T$ and the result holds.

Proposition 2 (Canonical forms). *Consider a value v such that $\cdot \vdash v : T$. Then:*

1. *If $T = \mathsf{Bool}$ then $v = b$ for some b.*
2. *If $T = \mathsf{Int}$ then $v = n$ for some n.*
3. *If $T = T_1 \longrightarrow T_2$ then $v = (\lambda x : T_1.t_2)$ for some t_2.*

Proof. By inspection of the type derivation rules.

Lemma 2. *Consider frame f, and term t_1, such as $\Gamma \vdash t_1 : T_1$ and $\Gamma \vdash f[t_1] : T$. Consider term t_1', such that $\Gamma \vdash t_1' : T_1'$ and $T_1' = T_1$. Then $\Gamma \vdash f[t_1'] : T'$ such that $T' = T$.*

Proof. By induction on the derivation of $f[t_1]$.

Case ($\Box\ t$). Then $f = \Box\ t_2$, $f[t_1] = t_1\ t_2$ and

$$(\text{Tapp})\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad T_2 = dom(T_1)}{\Gamma \vdash t_1\ t_2 : cod(T_1)}$$

then $f[t_1'] = t_1'\ t_2$. But as $T_1' = T_1$ then $cod(T_1') = cod(T_1)$ and $dom(T_1) = dom(T_1')$. Therefore

$$(\text{Tapp})\frac{\Gamma \vdash t_1 : T_1' \quad \Gamma \vdash t_2 : T_2 \quad T_2 = dom(T_1')}{\Gamma \vdash t_1'\ t_2 : cod(T_1')}$$

and the result holds.

Case ($v\ \Box, \Box + t, v + \Box, \mathsf{if}\ \Box\ \mathsf{then}\ t\ \mathsf{else}\ t$). Analogous to ($\Box\ t$)

Proposition 3 ($\longmapsto$ is well defined). *If $\Gamma \vdash t : T$, $t \longmapsto t'$ then, $\Gamma \vdash t' : T'$, where $T' = T$.*

Proof. By induction on the structure of a derivation of $t \longmapsto t'$.

Case ($\longrightarrow$). Then $t \longrightarrow t'$. By well-definedeness of $\longrightarrow$ (Prop 1), $\Gamma \vdash t' : T'$, where $T' = T$, and the result holds immediately.

Case ($f\Box$). Then $t = f[t_1]$, $\Gamma \vdash t_1 : T_1$ and $t_1 \longrightarrow t_2$. By induction hypothesis $\Gamma \vdash t_2 : T_2$ where $T_2 = T_1$. By Lemma 2, $\Gamma \vdash f[t_2] : T'$ such that $T' = T$, and the result holds immediatly.

Proposition 4 (Safety). *If $\Gamma \vdash t : T$, then one of the following is true:*

- *t is a value v;*
- *$t \longmapsto t'$, and $\Gamma \vdash t' : T'$ where $T' = T$,*

Proof. By induction on the structure of t.

Case (Tb, Tn, Tλ, Tl). t is a value.

Case (T+). Then $t = t_1 + t_2$ and

$$(\text{T+})\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad T_1 = \mathsf{Int} \quad T_2 = \mathsf{Int}}{\Gamma \vdash t_1 + t_2 : \mathsf{Int}}$$

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 2)

$$(\mathrm{R}\rightarrow)\frac{t \longrightarrow t'}{t \longmapsto t'}$$

and by Prop 1, $\Gamma \vdash t' : T'$, where $T' = T$, and therefore the result holds.

(b) $t_2 \longmapsto t_2'$. Then by induction hypothesis, $\Gamma \vdash t_2 : T_2'$, where $T_2' = T_2$. Then by (Tf), using $f = v + \Box$, $t \longmapsto t_1 + t_2'$ and by Lemma 2, $\Gamma \vdash t_1 + t_2' : \mathsf{Int}$ and the result holds.

2. $t_1 \longmapsto t_1'$. Then by induction hypotheses, $\Gamma' \vdash t_1' : \mathsf{Int}$. Then by (T$f$), using $f = \Box + t_2$, $t \longmapsto t_1' + t_2$ and by Lemma 2, $\Gamma \vdash t_1' + t_2 : \mathsf{Int}$ and the result holds.

Case (Sapp). Then $t = t_1\ t_2$, $T = T_{12}$ and

$$(\mathrm{Tapp})\frac{\Gamma \vdash t_1 : T_1 \qquad \Gamma \vdash t_2 : T_2 \qquad T_2 = dom(T_1)}{\Gamma \vdash t_1\ t_2 : cod(T_1)}$$

By induction hypotheses, one of the following holds:

1. t_1 is a value. Then by Canonical Forms (Lemma 2), and induction on t_2 one of the following holds:
   (a) t_2 is a value. Then by Canonical Forms (Lemma 2)

$$(\mathrm{R}\rightarrow)\frac{t \longrightarrow t'}{t \longmapsto t'}$$

   and by Prop 1 $\Gamma \vdash t' : T'$, where $T' = T$, and therefore the result holds.

   (b) $t_2 \longmapsto t_2'$. Then by induction hypothesis, $\Gamma \vdash t_2 : T_2'$, where $T_2' = T_2$. Then by (Tf), using $f = v\ \Box$, $t \longmapsto t_1\ t_2'$ and by Lemma 2, $\Gamma \vdash t_1\ t_2' : T_{12}'$ where $T_{12}' = cod(T_1)$ and the result holds.
2. $t_1 \longmapsto t_1'$. Then by induction hypotheses, $\Gamma' \vdash t_1' : T_{11}' \longrightarrow T_{12}'$ where $T_{11}' \longrightarrow T_{12}' = T_1$, . Then by (T$f$), using $f = \Box\ t_2$, $t \longmapsto t_1'\ t_2$ and by Lemma 2, $\Gamma \vdash t_1'\ t_2 : T_{12}'$ where $T_{12}' = cod(T_1)$ and the result holds.

Case (Tif). Then $t = \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$ and

$$(\mathrm{Tif})\frac{\Gamma \vdash t_1 : T_1 \qquad T_1 = \mathsf{Bool} \qquad \Gamma \vdash t_2 : T_2 \qquad \Gamma \vdash t_3 : T_3}{\Gamma \vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : equate(T_2, T_3)}$$

By induction hypotheses, one of the following holds:

1. t_1 is a value. Then by Canonical Forms (Lemma 2)

$$(\mathrm{R}\rightarrow)\frac{t \longrightarrow t'}{t \longmapsto t'}$$

   and by Prop 1, $\Gamma \vdash t' : T'$, where $T' = T$, and therefore the result holds.
2. $t_1 \longmapsto t_1'$. Then by induction hypothesis, $\Gamma \vdash t_1' : T_1'$, where $T_1' = T_1$. Therefore $T_1' = \mathsf{Bool}$. Then by (Tf), using $f = \mathsf{if}\ \Box\ \mathsf{then}\ t\ \mathsf{else}\ t$, $t \longmapsto \mathsf{if}\ t_1'\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$ and by Lemma 2, $\Gamma \vdash \mathsf{if}\ t_1'\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : T'$ where $T' = equate(T_2, T_3)$ and the result holds.

Case (T::). Then $t = t_1 :: T_2$ and

$$(\text{T::})\frac{\Gamma \vdash t_1 : T_1 \qquad T_1 = T_2}{\Gamma \vdash t_1 :: T_2 : T_2}$$

By induction hypotheses, one of the following holds:

1. t_1 is a value. Then

$$(\text{R}\rightarrow)\frac{t \;\longrightarrow\; t'}{t \longmapsto t'}$$

   and by Prop 1, $\Gamma \vdash t' : T'$, where $T' = T$, and therefore the result holds.
2. $t_1 \longmapsto t_1'$. Then by induction hypothesis, $\Gamma \vdash t_1' : T_1'$, where $T_1' = T_1$. Then by (Tf), using $f = \square :: T$, $t \longmapsto t_1' :: T_2$, but $\Gamma \vdash t_1' :: T_2 : T_2$ and the result holds.

3 GTFL$^{\oplus}$: Definitions

Section 3.1 presents the Galois connections along its soundness and optimality proofs. Section 3.2 presents the static semantics of GTFL$^{\oplus}$. Finally, Section 3.3 presents the dynamic semantics of GTFL$^{\oplus}$.

3.1 Meaning of Gradual Unions

This section presents the different steps taken to derive the Galois connection. Section 3.1.1 presents the classic interpretation which interprets the unknown type ?. Section 3.1.2 presents the powerset lifting of the classic interpretation. Section 3.1.3 presents the union interpretation which add support for gradual unions among gradual types that include the unknown type. Finally in Section 3.1.4 we combine the classic set interpretation and the union interpretation, named the stratified interpretation.

3.1.1 Step 1: The Classic Interpretation We start by presenting the Galois connection for gradual types made up with the (nullary constructor) ?, here denoted GTYPE.

$$G \in \text{GTYPE}$$
$$G ::= \; ? \;\mid\; \mathsf{Bool} \;\mid\; \mathsf{Int} \;\mid\; G \rightarrow G$$

The meaning of these gradual types is standard, and defined through concretization by Garcia *et al.* [2] as follows:

Definition 1 (GType Concretization). $\gamma_? : \text{GTYPE} \rightarrow \mathcal{P}(\text{TYPE})$

$$\gamma_?(\mathsf{Int}) = \{\, \mathsf{Int} \,\} \qquad \gamma_?(\mathsf{Bool}) = \{\, \mathsf{Bool} \,\} \qquad \gamma_?(?) = \text{TYPE}$$

$$\gamma_?(G_1 \rightarrow G_2) = \{\, T_1 \rightarrow T_2 \mid T_1 \in \gamma_?(G_1) \wedge T_2 \in \gamma_?(G_2) \,\}$$

Definition 2 (GType Precision). *G_1 is less imprecise than G_2, notation $G_1 \sqsubseteq G_2$, if and only if $\gamma_?(G_1) \subseteq \gamma_?(G_2)$.*

Definition 3 (GType Abstraction). $\alpha_? : \mathcal{P}(\textsc{Type}) \rightharpoonup \textsc{GType}$

$$\alpha_?(\{\, T \,\}) = T \qquad \alpha_?(\widehat{T_1 \to T_2}) = \alpha_?(\widehat{T}_1) \to \alpha_?(\widehat{T}_2) \qquad \alpha_?(\emptyset) = \textit{undefined}$$

$$\alpha_?(\widehat{T}) = ? \ \textit{otherwise}$$

Proposition 5 ($\alpha_?$ is Sound and Optimal). *If $\widehat{T}$ is not empty, then*

$$(a)\ \widehat{T} \subseteq \gamma_?(\alpha_?(\widehat{T})). \qquad (b)\ \widehat{T} \subseteq \gamma_?(G) \Rightarrow \alpha_?(\widehat{T}) \sqsubseteq G.$$

Proof. We proceed by induction on the structure of U. Let us start by proving a).

Case ($\{\, \mathsf{Int} \,\}$). Then $\alpha_?(\{\, \mathsf{Int} \,\}) = \mathsf{Int}$. But $\gamma_?(\mathsf{Int}) = \{\, \mathsf{Int} \,\}$ and the result holds.

Case ($\{\, \mathsf{Bool} \,\}$). Analogous to case $\{\, \mathsf{Int} \,\}$.

Case ($\widehat{T}_1 \mathbin{\widehat{\to}} \widehat{T}_2$). Then $\alpha_?(\widehat{T}_1 \mathbin{\widehat{\to}} \widehat{T}_2) = \alpha_?(\widehat{T}_1) \to \alpha_?(\widehat{T}_2)$. But by definition of $\gamma_?$, $\gamma_?(\alpha_?(\widehat{T}_1) \to \alpha_?(\widehat{T}_2)) = \gamma_?(\alpha_?(\widehat{T}_1)) \mathbin{\widehat{\to}} \gamma_?(\alpha_?(\widehat{T}_2))$. By induction hypotheses, $\widehat{T}_1 \subseteq \gamma_?(\alpha_?(\widehat{T}_1))$ and $\widehat{T}_2 \subseteq \gamma_?(\alpha_?(\widehat{T}_2))$, therefore $\widehat{T}_1 \mathbin{\widehat{\to}} \widehat{T}_2 \subseteq \gamma_?(\alpha_?(\widehat{T}_1) \to \alpha_?(\widehat{T}_2))$ and the result holds.

Case ($\emptyset$). This case cannot happen because $\alpha_?$ is restricted to non-empty sets of gradual intermediate types.

Case ($\widehat{T}$). Then $\alpha_?(\widehat{T}) = ?$ and threfore $\gamma_?(\alpha_?(\widehat{T})) = \textsc{Type}$, but $\widehat{T} \subseteq \textsc{Type}$ and the result holds.

Now let us prove b).

Case (Int). Trivial because $\gamma_?(\mathsf{Int}) = \{\, \mathsf{Int} \,\}$, and $\alpha_\oplus(\{\, \mathsf{Int} \,\}) = \mathsf{Int}$.

Case (Bool). Analogous to case Int.

Case ($G_1 \to G_2$). We have to prove that $\gamma_?(\alpha_?(\widehat{T})) \subseteq \gamma_?(G_1 \to G_2)$. But we know that $\widehat{T} \subseteq \gamma_?(G_1 \to G_2) = \gamma_?(G_1) \mathbin{\widehat{\to}} \gamma_?(G_2)$, therefore $\widehat{T}$ has the form $\{\, \overline{T_{i1} \to T_{i2}} \,\}$, for some $\{\, \overline{T_{i1}} \,\} \subseteq \gamma_?(G_1)$ and $\{\, \overline{T_{i2}} \,\} \subseteq \gamma_?(G_2)$. But by definition of $\alpha_?$, $\alpha_?(\{\, \overline{T_{i1} \to T_{i2}} \,\}) = \alpha_?(\{\, \overline{T_{i1}} \,\}) \to \alpha_?(\{\, \overline{T_{i2}} \,\})$ and therefore $\gamma_?(\alpha_?(\{\, \overline{T_{i1}} \,\}) \to \alpha_?(\{\, \overline{T_{i2}} \,\})) = \gamma_?(\alpha_?(\{\, \overline{T_{i1}} \,\})) \mathbin{\widehat{\to}} \gamma_?\alpha_?(\{\, \overline{T_{i2}} \,\})$. But by induction hypotheses $\gamma_?(\alpha_?(\{\, \overline{T_{i1}} \,\})) \subseteq \gamma_?(G_1)$ and $\gamma_?(\alpha_?(\{\, \overline{T_{i2}} \,\})) \subseteq \gamma_?(G_2)$ and the result holds.

Case ($?$). Then we have to prove that $\gamma_?(\alpha_?(\widehat{T})) \subseteq \gamma_?(?) = \textsc{Type}$, but this is always true and the result holds immediately.

3.1.2 Step 2: The Classic Set Interpretation The *powerset lifting* of $\gamma_?$, denoted $\widehat{\gamma_?}$, is simply the piecewise application of $\gamma_?$:

Definition 4 ($\mathcal{P}_{fin}$(GType) Concretization). $\widehat{\gamma_?} : \mathcal{P}_{fin}(\text{GTYPE}) \to \mathcal{P}_{fin}(\mathcal{P}(\text{TYPE}))$

$$\widehat{\gamma_?}(\widehat{G}) = \{\, \gamma_?(G) \mid G \in \widehat{G} \,\}$$

Similarly, the powerset lifting of the abstraction function $\alpha_?$, denoted $\widehat{\alpha_?}$, is the union of the piecewise application of $\alpha_?$:

Definition 5 ($\mathcal{P}_{fin}$(GType) Abstraction). $\widehat{\alpha_?} : \mathcal{P}_{fin}(\mathcal{P}(\text{TYPE})) \rightharpoonup \mathcal{P}_{fin}(\text{GTYPE})$

$$\widehat{\alpha_?}(\emptyset) = \textit{undefined} \qquad\qquad \widehat{\alpha_?}(\widehat{\widehat{T}}) = \bigcup_{\widehat{T} \in \widehat{\widehat{T}}} \alpha_?(\widehat{T})$$

Proposition 6 ($\widehat{\alpha_?}$ is Sound and Optimal). *If $\widehat{\widehat{T}}$ is not empty, then*

$$a)\ \widehat{\widehat{T}} \subseteq \widehat{\gamma_?}(\widehat{\alpha_?}(\widehat{\widehat{T}})). \qquad\qquad b)\ \widehat{\widehat{T}} \subseteq \widehat{\gamma_?}(\widehat{G}) \Rightarrow \widehat{\alpha_?}(\widehat{\widehat{T}}) \sqsubseteq \widehat{G}.$$

Proof. We start proving a). By definition of $\widehat{\alpha_?}$, $\widehat{\alpha_?}(\widehat{\widehat{T}}) = \bigcup_{\widehat{T} \in \widehat{\widehat{T}}} \alpha_?(\widehat{T})$. And by definition of $\widehat{\gamma_?}$, $\widehat{\gamma_?}(\widehat{\alpha_?}(\widehat{\widehat{T}})) = \{\, \gamma_?(\alpha_?(\widehat{T})) \mid \widehat{T} \in \widehat{\widehat{T}} \,\}$. We have to prove that $\forall \widehat{T} \in \widehat{\widehat{T}}, \exists \widehat{T}' \in \widehat{\gamma_?}(\widehat{\alpha_?}(\widehat{\widehat{T}}))$ such that $\widehat{T} \subseteq \widehat{T}'$. But we now that $\alpha_?$ is sound therefore $\forall \widehat{T} \in \widehat{\widehat{T}}, \widehat{T} \subseteq \gamma_?(\alpha_?(\widehat{T}))$, and the result holds immediately.

Now we prove b). We know that $\forall \widehat{T} \in \widehat{\widehat{T}}, \exists \widehat{T}' \in \widehat{\gamma_?}(\widehat{G})$ such that $\widehat{T} \subseteq \widehat{T}'$. We have to prove that $\widehat{\gamma_?}(\widehat{\alpha_?}(\widehat{\widehat{T}})) \subseteq \widehat{\gamma_?}(\widehat{G})$. But $\widehat{\gamma_?}(\widehat{\alpha_?}(\widehat{\widehat{T}})) = \{\, \gamma_?(\alpha_?(\widehat{T})) \mid \widehat{T} \in \widehat{\widehat{T}} \,\}$. But as $\alpha_?$ is optimal, $\forall$ non empty $\widehat{T}$, if $\widehat{T} \subseteq G$ then $\gamma_?(\alpha_?(\widehat{T})) \subseteq \gamma_?(G)$. Then we know that $\forall \widehat{T} \in \widehat{\widehat{T}}, \gamma_?(\alpha_?(\widehat{T})) \subseteq \widehat{T}'$, but $\widehat{T}' \in \gamma_?(\widehat{G})$ and the result holds.

3.1.3 Step 3: The Union Interpretation Now that we have defined the meaning of gradual types formed with the unknown type ?, as well as the meaning of gradual types formed with gradual unions $\oplus$, we turn to defining the meaning of gradual types in GTFL$^{\oplus}$, which combine both constructors, denoted UTYPE:

$$\begin{array}{l} U \in \text{UTYPE} \\ U ::= \boxed{\,?\,} \mid U \boxed{\,\oplus\,} U \mid \mathsf{Bool} \mid \mathsf{Int} \mid U \to U \quad \text{(gradual types)} \end{array}$$

We define a Galois connection between UTYPE and $\mathcal{P}_{fin}$(GTYPE) as follows:

Definition 6 (UType Concretization). $\gamma_{\oplus} : \text{UTYPE} \to \mathcal{P}_{fin}(\text{GTYPE})$

$$\gamma_{\oplus}(\mathsf{Int}) = \{\, \mathsf{Int} \,\} \qquad \gamma_{\oplus}(\mathsf{Bool}) = \{\, \mathsf{Bool} \,\} \qquad \boxed{\gamma_{\oplus}(?) = \{\, ? \,\}}$$

$$\gamma_{\oplus}(U_1 \to U_2) = \{\, T_1 \to T_2 \mid T_1 \in \gamma_{\oplus}(U_1) \wedge T_2 \in \gamma_{\oplus}(U_2) \,\}$$

$$\gamma_{\oplus}(U_1 \oplus U_2) = \gamma_{\oplus}(U_1) \cup \gamma_{\oplus}(U_2)$$

Definition 7 (UType Abstraction). $\alpha_\oplus : \mathcal{P}_{fin}(\text{GType}) \rightharpoonup \text{UType}$

$$\alpha_\oplus(\widehat{G}) = \oplus\widehat{G} \qquad \text{if } \widehat{G} \neq \emptyset$$

Proposition 7 ($\alpha_\oplus$ is Sound and Optimal). *If $\widehat{G}$ is not empty, then*

$$a)\ \widehat{G} \subseteq \gamma_\oplus(\alpha_\oplus(\widehat{G})). \qquad\qquad b)\ \widehat{G} \subseteq \gamma_\oplus(U) \Rightarrow \alpha_\oplus(\widehat{G}) \sqsubseteq U.$$

Proof. We proceed by induction on the structure of U. Let us start by proving a).

Case ($\{\,\mathsf{Int}\,\}$). Then $\alpha_\oplus(\{\,\mathsf{Int}\,\}) = \mathsf{Int}$. But $\gamma_\oplus(\mathsf{Int}) = \{\,\mathsf{Int}\,\}$ and the result holds.

Case ($\{\,\mathsf{Bool}\,\}$). Analogous to case $\{\,\mathsf{Int}\,\}$.

Case ($\{\,?\,\}$). Analogous to case $\{\,\mathsf{Int}\,\}$.

Case ($\emptyset$). This case cannot happen because $\alpha_\oplus$ is restricted to non-empty sets of gradual intermediate types.

Case ($\widehat{T_1} \cup \widehat{T_2}$). Then $\alpha_\oplus(\widehat{T_1} \cup \widehat{T_2}) = \alpha_\oplus(\widehat{T_1}) \oplus \alpha_\oplus(\widehat{T_2})$. But by definition of $\gamma_\oplus$, $\gamma_\oplus(\alpha_\oplus(\widehat{T_1}) \oplus \alpha_\oplus(\widehat{T_2})) = \gamma_\oplus(\alpha_\oplus(\widehat{T_1})) \cup \gamma_\oplus(\alpha_\oplus(\widehat{T_2}))$. By induction hypotheses, $\widehat{T_1} \subseteq \gamma_\oplus(\alpha_\oplus(\widehat{T_1}))$ and $\widehat{T_2} \subseteq \gamma_\oplus(\alpha_\oplus(\widehat{T_2}))$, therefore $\widehat{T_1} \cup \widehat{T_2} \subseteq \gamma_\oplus(\alpha_\oplus(\widehat{T_1}) \oplus \alpha_\oplus(\widehat{T_2}))$ and the result holds.

Now let us prove b).

Case (Int). Trivial because $\gamma_\oplus(\mathsf{Int}) = \{\,\mathsf{Int}\,\}$, and $\alpha_\oplus(\{\,\mathsf{Int}\,\}) = \mathsf{Int}$.

Case (Bool). Analogous to case Int.

Case ($?$). Analogous to case $?$.

Case ($U_1 \to U_2$). We have to prove that $\gamma_\oplus(\alpha_\oplus(\widehat{G})) \subseteq \gamma_\oplus(U_1 \to U_2)$. But $\gamma_\oplus(\alpha_\oplus(\widehat{G})) = \gamma_\oplus(\oplus\widehat{G}) = \widehat{G}$ and the result follows.

Case ($U_1 \oplus U_2$). Then $\gamma_\oplus(U_1 \oplus U_2) = \gamma_\oplus(U_1) \cup \gamma_\oplus(U_2)$ and $G = G_1 \cup G_2$ for some G_1 and G_2 such that $G_1 \subseteq \gamma_\oplus(U_1)$ and $G_2 \subseteq \gamma_\oplus(U_2)$. By definition of $\alpha_\oplus$, $\alpha_\oplus(G) = \alpha_\oplus(G_1) \cup \alpha_\oplus(G_2)$. By induction hypotheses, $\gamma_\oplus(\alpha_\oplus(G_1)) \subseteq \gamma_\oplus(U_1)$ and $\gamma_\oplus(\alpha_\oplus(G_2)) \subseteq \gamma_\oplus(U_2)$, therefore as $\gamma_\oplus(G) = \gamma_\oplus(\alpha_\oplus(G_1)) \cup \gamma_\oplus(\alpha_\oplus(G_2))$, then $\gamma_\oplus(G) \subseteq \gamma_\oplus(U_1) \cup \gamma_\oplus(U_2) = \gamma_\oplus(U)$ and the result holds.

3.1.4 Step 4: The Stratified Interpretation

We can now compose the two Galois connections in order to define a stratified interpretation for UType in terms of sets of sets of static types.

Definition 8 (Concretization). $\gamma : \text{UType} \to \mathcal{P}_{fin}(\mathcal{P}(\text{Type}))$, $\gamma = \widehat{\gamma_?} \circ \gamma_\oplus$

Definition 9 (Abstraction). $\alpha : \mathcal{P}_{fin}(\mathcal{P}(\text{Type})) \rightharpoonup \text{UType}$, $\alpha = \alpha_\oplus \circ \widehat{\alpha_?}$

$$U \in \text{UType}, \quad x \in \text{Var}, \quad \widetilde{t} \in \text{UTerm}, \Gamma \in \text{Var} \xrightarrow{\text{fin}} \text{UType}$$
$$U ::= U \oplus U \mid \mathsf{Int} \mid \mathsf{Bool} \mid U \to U \qquad \text{(types)}$$
$$v ::= n \mid \mathsf{true} \mid \mathsf{false} \mid (\lambda x : U.\widetilde{t}) \qquad \text{(values)}$$
$$\widetilde{t} ::= v \mid x \mid \widetilde{t}\,\widetilde{t} \mid \widetilde{t} + \widetilde{t} \mid \mathsf{if}\ \widetilde{t}\ \mathsf{then}\ \widetilde{t}\ \mathsf{else}\ \widetilde{t} \mid \widetilde{t} :: U \qquad \text{(terms)}$$

$$(U\text{x})\frac{x : U \in \Gamma}{\Gamma \vdash x : U} \qquad (U\text{b})\frac{}{\Gamma \vdash b : \mathsf{Bool}} \qquad (U\text{n})\frac{}{\Gamma \vdash n : \mathsf{Int}}$$

$$(U\lambda)\frac{\Gamma, x : U_1 \vdash \widetilde{t} : U_2}{\Gamma \vdash (\lambda x : U_1.\widetilde{t}) : U_1 \to U_2} \qquad (U::)\frac{\Gamma \vdash \widetilde{t} : U \quad U \sim U_1}{\Gamma \vdash (\widetilde{t} :: U_1) : U_1}$$

$$(U\text{app})\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \quad \Gamma \vdash \widetilde{t}_2 : U_2 \quad U_2 \sim \widetilde{dom}(U_1)}{\Gamma \vdash \widetilde{t}_1\ \widetilde{t}_2 : \widetilde{cod}(U_1)} \qquad (U+)\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \quad \Gamma \vdash \widetilde{t}_2 : U_2 \quad U_1 \sim \mathsf{Int} \quad U_2 \sim \mathsf{Int}}{\Gamma \vdash \widetilde{t}_1 + \widetilde{t}_2 : \mathsf{Int}}$$

$$(U\text{if})\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \quad U_1 \sim \mathsf{Bool} \quad \Gamma \vdash \widetilde{t}_2 : U_2 \quad \Gamma \vdash \widetilde{t}_3 : U_3}{\Gamma \vdash \mathsf{if}\ \widetilde{t}_1\ \mathsf{then}\ \widetilde{t}_2\ \mathsf{else}\ \widetilde{t}_3 : U_2 \sqcap U_3}$$

$$\widetilde{dom} : \text{UType} \rightharpoonup \text{UType} \qquad \widetilde{cod} : \text{UType} \rightharpoonup \text{UType}$$
$$\widetilde{dom}(U) = \alpha(\widehat{dom}(\gamma(U))) \qquad \widetilde{cod}(U) = \alpha(\widehat{cod}(\gamma(U)))$$

Fig. 3. GTFL$^{\oplus}$: Syntax and Type System

Because the composition of two Galois connection is a Galois connection, the stratified interpretation $\langle \gamma, \alpha \rangle$ is a Galois connection.

Proposition 8 (α is Sound and Optimal). *If $\widehat{\widehat{T}}$ is not empty, then*

$$a)\ \widehat{\widehat{T}} \subseteq \gamma(\alpha(\widehat{\widehat{T}})). \qquad b)\ \widehat{\widehat{T}} \subseteq \gamma(U) \Rightarrow \alpha(\widehat{\widehat{T}}) \sqsubseteq U.$$

Proof. By propositions 7 and 6 and composition of sound and optimal abstractions.

3.2 Static Semantics of GTFL$^{\oplus}$

This section presents the full static semantics of GTFL$^{\oplus}$ in Figure 3. Section 3.2.1, formally justifies the compositional lifting of predicates and functions. Section 3.2.2 presents the inductive definitions. Finally, Section 3.2.3 presents some type derivations examples.

3.2.1 Consistent Predicates and Functions We can base our liftings of predicates and types on inclusion and pointwise application that are extended to sets of sets.

Definition 10 (Predicate Lifting). $\widetilde{P}(U_1, U_2) \iff \exists T_1 \mathbin{\widehat{\in}} \gamma(U_1), T_2 \mathbin{\widehat{\in}} \gamma(U_2), P(T_1, T_2)$
where $\widehat{\in}$ *is the existential lifting of* $\in$ *to powersets:* $T \mathbin{\widehat{\in}} \widehat{\widehat{T}} \iff \exists \widehat{T} \in \widehat{\widehat{T}}, T \in \widehat{T}$

Equivalently: $\widetilde{P}(U_1, U_2) \iff \exists \widehat{T_1} \in \gamma(U_1), \exists \widehat{T_2} \in \gamma(U_2), \exists T_1 \in \widehat{T_1}, \exists T_2 \in \widehat{T_2}, P(T_1, T_2)$

The lifting of a predicate can also be defined in terms of each of the composed interpretations:

The lifting of a type function f uses the pointwise application of f to all elements of each subset of a powerset, which we note $\widehat{\widehat{f}}$.

Definition 11 (Function Lifting). $\widetilde{f} = \alpha \circ \widehat{\widehat{f}} \circ \gamma$

Definition 12 (Consistency). $U_1 \sim U_2$ *if and only if* $\exists \widehat{T_1} \in \gamma(U_1), \exists T_1 \in \widehat{T_1}, \exists \widehat{T_2} \in \gamma(U_2), \exists T_2 \in \widehat{T_2}, T_1 = T_2$.

Proposition 9. $\widetilde{P}(U_1, U_2) \iff \exists G_1 \in \gamma_\oplus(U_1), \exists G_2 \in \gamma_\oplus(U_2), \widetilde{P}_?(G_1, G_2)$
where $\widetilde{P}_?$ *is the predicate* P *lifted with* $\gamma_?$.

Proof. By definition of lifting predicates, $\widetilde{P}(U_1, U_2) = \exists \widehat{T_1} \in \gamma(U_1), \exists T_1 \in \widehat{T_1}, \exists \widehat{T_2} \in \gamma(U_2), \exists T_2 \in \widehat{T_2}.P(T_1, T_2)$.

If we unfold $\widetilde{P}_?(G_1, G_2)$, then $\exists G_1 \in \gamma_\oplus(U_1), \exists G_2 \in \gamma_\oplus(U_2), \widetilde{P}_?(G_1, G_2) \iff \exists G_1 \in \gamma_\oplus(U_1), \exists G_2 \in \gamma_\oplus(U_2), \exists T_1 \in \gamma_?(G_1), \exists T_2 \in \gamma_?(G_2).P(T_1, T_2)$.

We start with direction $\Rightarrow$. As $\gamma = \widehat{\gamma_?} \circ \gamma_\oplus$, then $\gamma(U_1) = \widehat{\gamma_?}(\widehat{G_1})$ and $\gamma(U_2) = \widehat{\gamma_?}(\widehat{G_2})$. As $\widehat{T_1} \in \widehat{\gamma_?}(\widehat{G_1})$ then $\widehat{T_1} = \gamma_?(G_1')$ for some $G_1' \in \widehat{G_1}$, and also $\widehat{T_2} \in \widehat{\gamma_?}(\widehat{G_2})$ then $\widehat{T_2} = \gamma_?(G_2')$ for some $G_2' \in \widehat{G_2}$. Then we can always choose $G_1 = G_1'$ and $G_2 = G_2'$ and the result holds.

Then we prove direction $\Leftarrow$. As $G_1 \in \gamma_\oplus(U_1)$ and $T_1 \in \gamma_?(G_1)$, then its easy to see that $T_1 \in (\widehat{\gamma_?} \circ \gamma_\oplus)(U_1)$ as we know that $G_1 \in \gamma_\oplus(U_1)$ and $(\widehat{\gamma_?} \circ \gamma_\oplus)(U_1) = \bigcup_{G \in \gamma_\oplus(U_1)} \gamma_?(G)$. Analogous $T_2 \in (\widehat{\gamma_?} \circ \gamma_\oplus)(U_2)$, but $\gamma = \widehat{\gamma_?} \circ \gamma_\oplus$ and the result holds.

Proposition 10. $U_1 \sim U_2 \iff \exists G_1 \in \gamma_\oplus(U_1), \exists G_2 \in \gamma_\oplus(U_2), G_1 \sim_? G_2$ *where* $\sim_?$ *is the classic consistency operator defined in [2].*

Proof. Direct by proposition 9.

When a function f receives more than one argument, then the lifting is the pointwise application of f to every combinations of elements of each set of each powerset. Formally:

Definition 13 (Lifting of type functions). *Let* $f : \textsc{Type}^n \rightharpoonup \textsc{Type}$, *then* $\widetilde{f} : \textsc{UType}^n \rightharpoonup \textsc{UType}$ *is defined as:*

$$\widetilde{f}(U_1, ..., U_2) = \alpha\Big(\bigcup_{G_1 \in \gamma_\oplus(U_1)} \dots \bigcup_{G_n \in \gamma_\oplus(U_n)} \bigcup_{T_1 \in \gamma_?(G_1)} \dots \bigcup_{T_n \in \gamma_?(G_n)} f(T_1, ..., T_n) \Big)$$

$$\widetilde{f}(U_1, ..., U_2) = \alpha\big(\{\{\, f(T_1, ..., T_n) \mid T_1 \in \gamma_?(G_1), ..., T_n \in \gamma_?(G_n) \,\} \mid G_1 \in \gamma_\oplus(U_1), ..., G_n \in \gamma_\oplus(U_n)\}\big)$$

Note that we can also define the lifting of a function using its intermediate lifting:

Proposition 11. *Let* $\widetilde{f} : \textsc{UType}^n \rightharpoonup \textsc{UType}$ *and* $\widetilde{f_?} : \textsc{UType}^n \rightharpoonup \textsc{UType}$ *then,*

$$\widetilde{f}(U_1, ..., U_2) \iff$$
$$\alpha^n_\oplus \circ \widehat{\widetilde{f_?}} \circ \gamma^n_\oplus(U_1, ..., U_n) \equiv \alpha^n_\oplus\big(\{\, \widetilde{f_?}(G_1, ..., G_n) \mid G_1 \in \gamma_\oplus(U_1), ..., G_n \in \gamma_\oplus(U_n) \,\} \big)$$

where $\widehat{\widetilde{f_?}}$ is the pointwise application of $\widetilde{f_?}$, and $\widetilde{f_?}$ is the lifting of f in the intermediate abstraction.

Proof. For simplicity, we will prove it for a function that receives one argument, the general case is analogous.

$$\begin{aligned}
\widetilde{f}(U) &= \alpha \circ \widehat{\widetilde{f}} \circ \gamma(U) \\
&= \alpha_\oplus \circ \widehat{\alpha_?} \circ \widehat{\widetilde{f}} \circ \widehat{\gamma_?} \circ \gamma_\oplus(U) \\
&= \alpha_\oplus \circ \widehat{\alpha_?} \circ \widehat{\widetilde{f}} \circ \widehat{\gamma_?}(\{G \mid G \in \gamma_\oplus(U)\}) \\
&= \alpha_\oplus \circ \widehat{\alpha_?} \circ \widehat{\widetilde{f}}(\{\{\, T \mid T \in \gamma_?(G) \,\} \mid G \in \gamma_\oplus(U)\}) \\
&= \alpha_\oplus \circ \widehat{\alpha_?}(\{\{\, f(T) \mid T \in \gamma_?(G) \,\} \mid G \in \gamma_\oplus(U)\}) \\
&= \alpha_\oplus(\{\alpha_?(\{\, f(T) \mid T \in \gamma_?(G) \,\}) \mid G \in \gamma_\oplus(U)\}) \\
&= \alpha_\oplus(\{\widetilde{f}(G) \mid G \in \gamma_\oplus(U)\}) \\
&= \alpha_\oplus \circ \widehat{\widetilde{f_?}} \circ \gamma_\oplus(U)
\end{aligned}$$

Proposition 12. *Let* $F : \textsc{Type} \rightharpoonup \textsc{Type}$ *be a partial function, and define the predicate* $P(T_1, T_2) \equiv T_1 = F(T_2)$*. Then* $\widetilde{P}(U_1, U_2)$ *implies* $U_1 \sim \widetilde{F}(U_2)$*.*

Proof. Suppose $\widetilde{P}(U_1, U_2)$. Then $T_1 = F(T_2)$ for some $T_1 \in \widehat{T_1}, \widehat{T_1} \in \gamma(U_1)$ and $T_2 \in \widehat{T_2}, \widehat{T_2} \in \gamma(U_2)$. Therefore $F(T_2) \in \widehat{T_F}, \widehat{T_F} \in \widehat{\widetilde{F}}(\gamma(U_2))$ and by Prop 8 $\widehat{\widetilde{F}}(\gamma(U_2)) \subseteq \gamma(\alpha(\widehat{\widetilde{F}}(\gamma(U_2))))$. But $\widetilde{F}(U_2) = \alpha(\widehat{\widetilde{F}}(\gamma(U_2)))$, so $F(T_2) \in \widehat{T_F}, \widehat{T_F} \in \gamma(\alpha(\widehat{\widetilde{F}}(\gamma(U_2)))) = \gamma(\widetilde{F}(U_2))$. Then by definition of consistency, we can choose $(T_1, F(T_2)) \in (\widehat{T_1}, \widehat{T_F}), (\widehat{T_1}, \widehat{T_F}), \in \gamma(U_1, \widetilde{F}(U_2))$ such that $T_1 = F(T_2)$, therefore $U_1 \sim \widetilde{F}(U_2)$.

Proposition 13. *Let* $P(T_1, T_2) \equiv T_1 = dom(T_2)$*. Then* $U_1 \sim \widetilde{dom}(U_2)$ *implies* $\widetilde{P}(U_1, U_2)$*.*

Proof. Suppose $U_1 \sim \widetilde{dom}(U_2)$. Then exists $T_1 \in \widehat{T_1}, \widehat{T_1} \in \gamma(U_1)$ and $T_2 \in \widehat{T_2}, \widehat{T_2} \in \gamma(\widetilde{dom}(U_2))$ such that $T_1 = T_2$, which implies that $\exists T'_2 \in \widehat{T'_2}, \widehat{T'_2} \in \gamma(G_2)$, such that $T_2 = dom(T'_2)$, which is by definition $\widetilde{P}(U_1, U_2)$.

Proposition 14. *Let* $P(T_1, T_2) \equiv T_1 = dom(T_2)$. *Then* $U_1 \sim \widetilde{dom}(U_2)$ *if and only if* $\widetilde{P}(U_1, U_2)$.

Proof. Direct consequence of Prop. 12 and 13.

3.2.2 Inductive definitions This section presents inductive definitions of some of the metafunctions presented in the paper.

Proposition 15.

$$\frac{U \sim U_1}{U \sim U_1 \oplus U_2} \qquad \frac{U \sim U_2}{U \sim U_1 \oplus U_2} \qquad \frac{U_1 \sim U}{U_1 \oplus U_2 \sim U} \qquad \frac{U_2 \sim U}{U_1 \oplus U_2 \sim U}$$

$$\frac{}{U \sim U} \qquad \frac{}{? \sim U} \qquad \frac{}{U \sim ?} \qquad \frac{U_{21} \sim U_{11} \quad U_{12} \sim U_{22}}{U_{11} \to U_{12} \sim U_{21} \to U_{22}}$$

Proof. Straightforward from the definition of consistency.

Definition 14 (Gradual Meet). *Let* $\sqcap : \text{UType} \rightharpoonup \text{UType}$ *be defined as:*

1. $U \sqcap U = U$
2. $? \sqcap U = U \sqcap ? = U$
3. $U \sqcap (U_1 \oplus U_2) = (U_1 \oplus U_2) \sqcap U = \begin{cases} U \sqcap U_1 & \text{if } U \sqcap U_2 \text{ is undefined} \\ U \sqcap U_2 & \text{if } U \sqcap U_1 \text{ is undefined} \\ (U \sqcap U_1) \oplus (U \sqcap U_2) & \text{otherwise} \end{cases}$.
4. $(U_{11} \to U_{12}) \sqcap (U_{21} \to U_{22}) = (U_{11} \sqcap U_{21}) \to (U_{12} \sqcap U_{22})$
5. $U_1 \sqcap U_2$ *is undefined otherwise.*

Definition 15 (Equate lifting).

$$\widetilde{equate}(U_1, U_2) = U_1 \sqcap U_2 = \alpha\big(\{\widehat{T}_1 \cap \widehat{T}_2 \mid \widehat{T}_1 \in \gamma(U_1), \widehat{T}_2 \in \gamma(U_2)\}\big) = \alpha_\oplus\big(\{\widetilde{equate}_?(G_1, G_2) \mid G_1 \in \gamma_\oplus(U_1), G_2 \in \gamma_\oplus(U_2)\}\big)$$

where $\widetilde{equate}_?$ is the lifting of the *equate* function in the intermediate abstraction (defined in [2]).

Proposition 16. $\sqcap = \alpha \circ \widetilde{\widetilde{equate}} \circ \gamma$

Proof. Direct using induction and the definition of meet as intersection of sets of sets.

Proposition 17.

$$\frac{}{U \sqsubseteq U} \qquad \frac{}{? \sqsubseteq U} \qquad \frac{U_1 \sqsubseteq U_3 \quad U_2 \sqsubseteq U_4}{U_1 \to U_2 \sqsubseteq U_3 \to U_4}$$

$$\frac{U_1 \sqsubseteq U_2 \quad U_1 \sqsubseteq U_3}{U_1 \sqsubseteq U_2 \oplus U_3} \qquad \frac{U_1 \sqsubseteq U_3}{U_1 \oplus U_2 \sqsubseteq U_3} \qquad \frac{U_2 \sqsubseteq U_3}{U_1 \oplus U_2 \sqsubseteq U_3}$$

Proof. Direct using induction and the definition of the concretization function.

$$\dfrac{\dfrac{x : \mathsf{Int} \vdash x : \mathsf{Int} \oplus \mathsf{Bool} \quad x : \mathsf{Int} \vdash 1 : \mathsf{Int} \quad \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \quad \mathsf{Int} \sim \mathsf{Int}}{x : \mathsf{Int} \vdash (x+1) : \mathsf{Int}}}{\cdot \vdash (\lambda x : \mathsf{Int}.(x+1)) : (\mathsf{Int} \oplus \mathsf{Bool}) \to \mathsf{Int}}$$

$$\dfrac{\dfrac{\dfrac{x : \mathsf{Int} \vdash x : \mathsf{Int} \oplus \mathsf{Bool} \quad x : \mathsf{Int} \vdash 1 : \mathsf{Int}}{\mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \qquad \mathsf{Int} \sim \mathsf{Int}}}{x : \mathsf{Int} \vdash (x+1) : \mathsf{Int}}}{\cdot \vdash (\lambda x : \mathsf{Int}.(x+1)) : (\mathsf{Int} \oplus \mathsf{Bool}) \to \mathsf{Int}} \qquad \cdot \vdash 1 : \mathsf{Int} \qquad \mathsf{Int} \sim \mathsf{Int} \oplus \mathsf{Bool}}{((\lambda x : \mathsf{Int}.(x+1))1) : \mathsf{Int}}$$

$$\dfrac{\dfrac{\dfrac{x : \mathsf{Int} \vdash x : \mathsf{Int} \oplus \mathsf{Bool} \quad x : \mathsf{Int} \vdash 1 : \mathsf{Int}}{\mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \qquad \mathsf{Int} \sim \mathsf{Int}}}{x : \mathsf{Int} \vdash (x+1) : \mathsf{Int}}}{\cdot \vdash (\lambda x : \mathsf{Int}.(x+1)) : (\mathsf{Int} \oplus \mathsf{Bool}) \to \mathsf{Int}} \qquad \cdot \vdash \mathsf{true} : \mathsf{Bool} \qquad \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}}{((\lambda x : \mathsf{Int}.(x+1))\mathsf{true}) : \mathsf{Int}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{x : \mathsf{Bool} \vdash 1 : \mathsf{Int} \quad \mathsf{Int} \sim \mathsf{Int} \oplus \mathsf{Bool}}{(x : \mathsf{Bool} \vdash 1 :: \mathsf{Int} \oplus \mathsf{Bool}) : \mathsf{Int} \oplus \mathsf{Bool}} \qquad \dfrac{x : \mathsf{Bool} \vdash \mathsf{false} : \mathsf{Bool} \quad \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}}{(x : \mathsf{Bool} \vdash \mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool}) : \mathsf{Int} \oplus \mathsf{Bool}}}{x : \mathsf{Bool} \vdash x : \mathsf{Bool} \quad \mathsf{Bool} \sim \mathsf{Bool} \quad \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool} \quad \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}}}{x : \mathsf{Bool} \vdash (\mathsf{if}\ x\ \mathsf{then}\ (1 :: \mathsf{Int} \oplus \mathsf{Bool})\ \mathsf{else}\ (\mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool})) : \mathsf{Int} \oplus \mathsf{Bool}}}{\cdot \vdash (\lambda x : \mathsf{Bool}.(\mathsf{if}\ x\ \mathsf{then}\ (1 :: \mathsf{Int} \oplus \mathsf{Bool})\ \mathsf{else}\ (\mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool}))) : \mathsf{Bool} \to (\mathsf{Int} \oplus \mathsf{Bool})}$$

Fig. 4. Examples of intrinsic type derivations

3.2.3 Examples of type derivations In this section we present some type derivations examples in Figure 4.

3.3 Dynamic Semantics of GTFL$^{\oplus}$

One of the salient features of the AGT methodology is that it provides a *direct* dynamic semantics for gradual programs [2], instead of the typical translational semantics through an intermediate cast calculus [5]. The key idea is to apply *proof reduction* on gradual typing derivations [3]; by the Curry-Howard correspondence, this gives a notion of relation for gradual terms. We call such semantics the *reference semantics*.

The main insight of AGT is that gradual typing derivations need to be augmented with *evidence* to support consistent judgments. Evidence reflects the justification of *why* a given consistent judgment holds. Therefore, the dynamic

semantics mirrors the type preservation argument of the static language, combining evidences at each reduction step in order to determine whether the program can reduce or should halt with a runtime error.

A consistency judgment $U_1 \sim U_2$ is supported by an evidence ε that denotes the most precise knowledge about U_1 and U_2 gained by knowing that they are related by consistency. It is written $\varepsilon \vdash U_1 \sim U_2$. In the case of consistency, which is symmetric (as opposed to, say, consistent subtyping), evidence boils down to a single gradual type , i.e. $\varepsilon \in \textsc{UType}$, which is precisely the least upper bound of both types, *i.e.* $U_1 \sqcap U_2$ [2]. For instance, $\mathsf{Int} \vdash \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int}$.

Consider the simple program: $(\lambda x : \mathsf{Int} \oplus \mathsf{Bool}.x + 1)\ \mathsf{true}$. It is a well-typed gradual program, and its typing derivation includes two consistent judgments: $\mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}$ to accept passing true as argument, and $\mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int}$ to accept using x in the addition. When simplifying the typing derivation itself (replacing the use of the x hypothesis with the typing derivation of the argument) it becomes necessary to combine the two consistent judgments in order to justify the reduction. In general, in the safety proof of the static language, this corresponds to a use of the fact that type equality is transitive. Here, transitivity demands that the respective evidences for the consistent judgments can be combined. In the example, Int and Bool cannot be combined (their meet is undefined), therefore the program halts with **error**.

To formalize this approach while avoiding writing down reduction rules on actual (bi-dimensional) derivation trees, Garcia *et al.* adopt *intrinsic terms* [1], which are a flat notation that is isomorphic to typing derivations. Specifically, the typing derivation for the judgment $\Gamma \vdash \widetilde{t} : U$ is represented by an intrinsic term $\check{t} \in \textsc{Term}_U$. (The reversed hat on $\check{t}$ is meant to suggest the derivation tree.)

To illustrate how intrinsic terms are formed, consider addition and ascription:

$$\frac{\check{t}_1 \in \textsc{Term}_{U_1} \qquad \varepsilon_1 \vdash U_1 \sim \mathsf{Int} \qquad \check{t}_2 \in \textsc{Term}_{U_2} \qquad \varepsilon_2 \vdash U_2 \sim \mathsf{Int}}{\varepsilon_1 \check{t}_1 + \varepsilon_2 \check{t}_2 \in \textsc{Term}_{\mathsf{Int}}}$$

$$\frac{\check{t} \in \textsc{Term}_U \qquad \varepsilon \vdash U \sim U'}{\varepsilon \check{t} :: U' \in \textsc{Term}_{U'}}$$

The rules describe how a derivation tree for a compound expression is formed from the sub-derivations of the subterms together with the evidences that support the consistency judgments. Note how the involved evidences show up in the term representation. The syntax of intrinsic terms follows the same pattern as that illustrated with addition and ascription. Also, intrinsic values $\check{v}$ can either be simple values $\check{u}$ or ascribed values $\varepsilon\check{u} :: U$. With this notational device, the reduction rules on derivation trees can be written as reduction rules on intrinsic terms, possibly failing with an **error** when combining evidences, for instance:

$$\varepsilon_1(\varepsilon_2 \check{v} :: U) \longrightarrow_c \begin{cases} (\varepsilon_2 \circ^{=} \varepsilon_1)v \\ \textbf{error} \qquad \text{if } (\varepsilon_2 \circ^{=} \varepsilon_1) \text{ is not defined} \end{cases}$$

$$(\mathrm{I}U\mathrm{x})\frac{}{x^{U} \in \textsc{Term}_{U}} \qquad (\mathrm{I}U\mathrm{b})\frac{}{b^{\mathsf{Bool}} \in \textsc{Term}_{\mathsf{Bool}}} \qquad (\mathrm{I}U\mathrm{n})\frac{}{n^{\mathsf{Int}} \in \textsc{Term}_{\mathsf{Int}}}$$

$$(\mathrm{I}U\lambda)\frac{t^{U_2} \in \textsc{Term}_{U_2}}{(\lambda x^{U_1}.t^{U_2}) \in \textsc{Term}_{U_1 \to U_2}} \qquad (\mathrm{I}U{::})\frac{t^{U_1} \in \textsc{Term}_{U_1} \quad \varepsilon \vdash U_1 \sim U}{\varepsilon t^{U_1} :: U \in \textsc{Term}_{U}}$$

$$(\mathrm{I}U\mathrm{app})\frac{t^{U_1} \in \textsc{Term}_{U_1} \quad \varepsilon_1 \vdash U_1 \sim U_{11} \to U_{12} \quad t^{U_2} \in \textsc{Term}_{U_2} \quad \varepsilon_2 \vdash U_2 \sim U_{11}}{(\varepsilon_1 t^{U_1})\ @^{U_{11} \to U_{12}}\ (\varepsilon_2 t^{U_2}) \in \textsc{Term}_{U_{12}}}$$

$$(\mathrm{I}U{+})\frac{t^{U_1} \in \textsc{Term}_{U_1} \quad \varepsilon_1 \vdash U_1 \sim \mathsf{Int} \quad t^{U_2} \in \textsc{Term}_{U_2} \quad \varepsilon_2 \vdash U_2 \sim \mathsf{Int}}{\varepsilon_1 t^{U_1} + \varepsilon_2 t^{U_2} \in \textsc{Term}_{\mathsf{Int}}}$$

$$(\mathrm{I}U\mathrm{if})\frac{\begin{array}{ccc} t^{U_1} \in \textsc{Term}_{U_1} & \varepsilon_1 \vdash U_1 \sim \mathsf{Bool} & U = (U_2 \sqcap U_3) \\ t^{U_2} \in \textsc{Term}_{U_2} & \varepsilon_2 \vdash U_2 \sim U & t^{U_3} \in \textsc{Term}_{U_3} \quad \varepsilon_3 \vdash U_3 \sim U \end{array}}{\mathsf{if}\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \in \textsc{Term}_{U}}$$

Fig. 5. Gradual Intrinsic Terms for GTFL$^{\oplus}$

The definition of consistent transitivity for a type predicate P, $\circ^{P}$, is given by the abstract interpretation framework [2]; in particular, for type equality, $\circ^{=}$ corresponds to the meet of gradual types $\sqcap$.

Notice that for convenience, from this point forward we use the notation $t^{U} \in \textsc{Term}_{U}$ to refer to an intrinsic term $\check{t} \in \textsc{Term}_{U}$, u to refer to an $\check{u}$, v to refer to an $\check{v}$, and finally we sometimes omit the type notation in t^{U} when the type is not important in that context. Figure 5 presents the gradual intrinsic terms of GTFL$^{\oplus}$. Figures 6 and 7 present some intrinsic type derivations of the examples presented in Section 3.2.3. Figure 8 presents the syntax and notions of reductions. Figure 9 presents the intrinsic reduction of GTFL$^{\oplus}$.

4 Properties of GTFL$^{\oplus}$

This section presents the proof of the static gradual guarantee in Section 2. Section 4.2 presents the proof of type safety. Finally, Section 4.3 presents the proof of the dynamic gradual guarantee.

4.1 Static Gradual Guarantee

Proposition 18 (Equivalence for fully-annotated terms).
For any $t \in \textsc{Term}$, $. \vdash_{S} t : T$ if and only if $. \vdash t : T$

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

$$\dfrac{\dfrac{x \in \textsc{Term}_{\mathsf{Int}\oplus\mathsf{Bool}} \quad 1 \in \textsc{Term}_{\mathsf{Int}} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int}\oplus\mathsf{Bool} \sim \mathsf{Int} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int} \sim \mathsf{Int}}{(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1) \in \textsc{Term}_{\mathsf{Int}}}}{(\lambda x.(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1)) \in \textsc{Term}_{(\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}}}$$

$$\dfrac{\dfrac{\dfrac{x \in \textsc{Term}_{\mathsf{Int}\oplus\mathsf{Bool}} \quad 1 \in \textsc{Term}_{\mathsf{Int}} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int}\oplus\mathsf{Bool} \sim \mathsf{Int} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int} \sim \mathsf{Int}}{(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1) \in \textsc{Term}_{\mathsf{Int}}}}{(\lambda x.(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1)) \in \textsc{Term}_{(\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}}} \quad \langle (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}\rangle \vdash (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int} \sim (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int} \quad 1 \in \textsc{Term}_{\mathsf{Int}} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int} \sim \mathsf{Int}\oplus\mathsf{Bool}}{(\langle (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}\rangle(\lambda x.(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1))@^{(\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}}\langle \mathsf{Int}\rangle 1) \in \textsc{Term}_{\mathsf{Int}}}$$

$$\dfrac{\dfrac{\dfrac{x \in \textsc{Term}_{\mathsf{Int}\oplus\mathsf{Bool}} \quad 1 \in \textsc{Term}_{\mathsf{Int}} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int}\oplus\mathsf{Bool} \sim \mathsf{Int} \quad \langle \mathsf{Int}\rangle \vdash \mathsf{Int} \sim \mathsf{Int}}{(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1) \in \textsc{Term}_{\mathsf{Int}}}}{(\lambda x.(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1)) \in \textsc{Term}_{(\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}}} \quad \mathsf{true} \in \textsc{Term}_{\mathsf{Bool}} \quad \langle (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}\rangle \vdash (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int} \sim (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int} \quad \langle \mathsf{Bool}\rangle \vdash \mathsf{Bool} \sim \mathsf{Int}\oplus\mathsf{Bool}}{(\langle (\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}\rangle(\lambda x.(\langle \mathsf{Int}\rangle x + \langle \mathsf{Int}\rangle 1))@^{(\mathsf{Int}\oplus\mathsf{Bool})\to\mathsf{Int}}\langle \mathsf{Bool}\rangle \mathsf{true}) \in \textsc{Term}_{\mathsf{Int}}}$$

Fig. 6. Examples of intrinsic type derivations (part 1)

Definition 16 (Term precision).

$$(Px)\dfrac{}{x \sqsubseteq x} \qquad (Pb)\dfrac{}{b \sqsubseteq b} \qquad (Pn)\dfrac{}{n \sqsubseteq n}$$

$$(P\lambda)\dfrac{\widetilde{t} \sqsubseteq \widetilde{t'} \quad U_1 \sqsubseteq U_1'}{(\lambda x : U_1.\widetilde{t}) \sqsubseteq (\lambda x : U_1'.\widetilde{t'})} \qquad (P+)\dfrac{\widetilde{t}_1 \sqsubseteq \widetilde{t}_1' \quad \widetilde{t}_2 \sqsubseteq \widetilde{t}_2'}{\widetilde{t}_1 + \widetilde{t}_2 \sqsubseteq \widetilde{t}_1' + \widetilde{t}_2'}$$

$$(Papp)\dfrac{\widetilde{t}_1 \sqsubseteq \widetilde{t}_1' \quad \widetilde{t}_2 \sqsubseteq \widetilde{t}_2'}{\widetilde{t}_1\ \widetilde{t}_2 \sqsubseteq \widetilde{t}_1'\ \widetilde{t}_2'} \qquad (Pif)\dfrac{\widetilde{t} \sqsubseteq \widetilde{t} \quad \widetilde{t}_1 \sqsubseteq \widetilde{t}_1' \quad \widetilde{t}_2 \sqsubseteq \widetilde{t}_2'}{\mathsf{if}\ \widetilde{t}\ \mathsf{then}\ \widetilde{t}_1\ \mathsf{else}\ \widetilde{t}_2 \sqsubseteq \mathsf{if}\ \widetilde{t'}\ \mathsf{then}\ \widetilde{t}_1'\ \mathsf{else}\ \widetilde{t}_2'}$$

$$(P::)\dfrac{\widetilde{t} \sqsubseteq \widetilde{t'} \quad U \sqsubseteq U'}{\widetilde{t} :: U \sqsubseteq \widetilde{t'} :: U'}$$

Definition 17 (Type environment precision).

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

Lemma 3. *If $\Gamma \vdash \widetilde{t} : U$ and $\Gamma \sqsubseteq \Gamma'$, then $\Gamma' \vdash \widetilde{t} : U'$ for some $U \sqsubseteq U'$.*

Proof. Simple induction on typing derivations.

Lemma 4. *If $U_1 \sim U_2$ and $U_1 \sqsubseteq U_1'$ and $U_2 \sqsubseteq U_2'$ then $U_1' \sim U_2'$.*

$$\dfrac{\dfrac{1 \in \textsc{Term}_{\mathsf{Int}} \quad \langle \mathsf{Int} \rangle \vdash \mathsf{Int} \sim \mathsf{Int} \oplus \mathsf{Bool}}{(\langle \mathsf{Int} \rangle 1 :: \mathsf{Int} \oplus \mathsf{Bool}) \in \textsc{Term}_{\mathsf{Int} \oplus \mathsf{Bool}}} \quad \dfrac{\mathsf{false} \in \textsc{Term}_{\mathsf{Bool}} \quad \langle \mathsf{Bool} \rangle \vdash \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}}{(\langle \mathsf{Bool} \rangle \mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool}) \in \textsc{Term}_{\mathsf{Int} \oplus \mathsf{Bool}}} \quad x \in \textsc{Term}_{\mathsf{Bool}} \quad \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle \vdash \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool} \quad \langle \mathsf{Bool} \rangle \vdash \mathsf{Bool} \sim \mathsf{Bool} \quad \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle \vdash \mathsf{Int} \oplus \mathsf{Bool} \sim \mathsf{Int} \oplus \mathsf{Bool}}{\begin{array}{c}\mathsf{if}\ \langle \mathsf{Bool} \rangle x\ \mathsf{then}\ \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle (\langle \mathsf{Int} \rangle 1 :: \mathsf{Int} \oplus \mathsf{Bool}) \\ \mathsf{else}\ \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle (\langle \mathsf{Bool} \rangle \mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool})) \in \textsc{Term}_{\mathsf{Int} \oplus \mathsf{Bool}}\end{array}}}{\begin{array}{c}(\lambda x.\mathsf{if}\ \langle \mathsf{Bool} \rangle x\ \mathsf{then}\ \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle (\langle \mathsf{Int} \rangle 1 :: \mathsf{Int} \oplus \mathsf{Bool}) \\ \mathsf{else}\ \langle \mathsf{Int} \oplus \mathsf{Bool} \rangle (\langle \mathsf{Bool} \rangle \mathsf{false} :: \mathsf{Int} \oplus \mathsf{Bool})) \in \textsc{Term}_{\mathsf{Bool} \to (\mathsf{Int} \oplus \mathsf{Bool})}\end{array}}$$

Fig. 7. Examples of intrinsic type derivations (part 2)

Proof. By definition of $\sim$, there exists $\langle T_1, T_2 \rangle \in \langle \widehat{T}_1, \widehat{T}_2 \rangle \in \gamma^2(U_1, U_2)$ such that $T_1 = T_2$. $U_1 \sqsubseteq U_1'$ and $U_2 \sqsubseteq U_2'$ mean that $\gamma(U_1) \subseteq \gamma(U_1')$ and $\gamma(U_2) \subseteq \gamma(U_2')$, therefore $\langle T_1, T_2 \rangle \in \langle \widehat{T}_1, \widehat{T}_2 \rangle \in \gamma^2(U_1', U_2')$.

Proposition 19 (Static gradual guarantee). *If* $. \vdash \widetilde{t}_1 : U_1$ *and* $\widetilde{t}_1 \sqsubseteq \widetilde{t}_2$*, then* $. \vdash \widetilde{t}_2 : U_2$*, for some* U_2 *such that* $U_1 \sqsubseteq U_2$*.*

Proof. We prove the property on opens terms instead of closed terms: If $\Gamma \vdash \widetilde{t}_1 : U_1$ and $\widetilde{t}_1 \sqsubseteq \widetilde{t}_2$ then $\Gamma \vdash \widetilde{t}_2 : U_2$ and $U_1 \sqsubseteq U_2$.

The proof proceed by induction on the typing derivation.

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

Case (Uλ). Then $\widetilde{t}_1 = (\lambda x : U_1.\widetilde{t})$ and $U_1 = U_1' \to U_2'$. By $(U\lambda)$ we know that:

$$(U\lambda)\frac{\Gamma, x : U_1' \vdash \widetilde{t} : U_2'}{\Gamma \vdash (\lambda x : U_1'.\widetilde{t}) : U_1' \to U_2'} \tag{1}$$

Consider $\widetilde{t}_2$ such that $\widetilde{t}_1 \sqsubseteq \widetilde{t}_2$. By definition of term precision $\widetilde{t}_2$ must have the form $\widetilde{t}_2 = (\lambda x : U_1''.\widetilde{t'})'$ and therefore

$$(U\lambda)\frac{\widetilde{t} \sqsubseteq \widetilde{t'} \qquad U_1' \sqsubseteq U_1''}{(\lambda x : U_1'.\widetilde{t}) \sqsubseteq (\lambda x : U_1''.\widetilde{t'})} \tag{2}$$

Using induction hypotheses on the premise of 1, $\Gamma, x : U_1' \vdash \widetilde{t'} : U_2''$ with $U_2' \sqsubseteq U_2''$. By Lemma 3, $\Gamma, x : U_1'' \vdash \widetilde{t'} : U_2'''$ where $U_2'' \sqsubseteq U_2'''$. Then we can use rule $(U\lambda)$ to derive:

$$(U\lambda)\frac{\Gamma, x : U_1''; . \vdash \widetilde{t'} : U_2'''}{\Gamma \vdash (\lambda x : U_1'.\widetilde{t'}) : U_1'' \to U_2'''}$$

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

$$U_1' \to U_2' \sqsubseteq U_1'' \to U_2'''$$

and the result holds.

$\varepsilon \in \textsc{Evidence}, \quad et \in \textsc{EvTerm}, \quad ev \in \textsc{EvValue}, \quad t \in \textsc{Term}_*,$
$v \in \textsc{Value}, \quad u \in \textsc{SimpleValue}, g \in \textsc{EvFrame}, \quad f \in \textsc{TmFrame}$

$et ::= \varepsilon t$

$ev ::= \varepsilon u$

$u ::= x \mid n \mid b \mid \lambda x.t$

$v ::= u \mid \varepsilon u :: U$

$g ::= \Box + et \mid ev + \Box \mid \Box \mathbin{@^U} et \mid ev \mathbin{@^U} \Box \mid \Box :: U \mid \mathsf{if}\ \Box\ \mathsf{then}\ et\ \mathsf{else}\ et$

$f ::= g[\varepsilon\Box]$

Notions of Reduction

$\longrightarrow$: $\textsc{Term}_U \times (\textsc{Term}_U \cup \{\,\mathbf{error}\,\})$
$\longrightarrow_c$: $\textsc{EvTerm} \times (\textsc{EvTerm} \cup \{\,\mathbf{error}\,\})$

$$\varepsilon_1 n_1 + \varepsilon_2 n_2 \longrightarrow n_3 \text{ where } n_3 = n_1 \,[\![+]\!]\, n_2$$

$$\varepsilon_1(\lambda x^{U_{11}}.t) \mathbin{@^{U_1 \longrightarrow U_2}} \varepsilon_2 u \longrightarrow \begin{cases} icod(\varepsilon_1)([(\varepsilon_2 \circ^{=} idom(\varepsilon_1))u :: U_{11})/x^{U_{11}}]t) :: U_2 \\ \mathbf{error} \qquad \text{if not defined} \end{cases}$$

$$\mathsf{if}\ \varepsilon_1 b\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \longrightarrow \begin{cases} \varepsilon_2 t^{U_2} :: U_2 \sqcap U_3 & b = \mathsf{true} \\ \varepsilon_3 t^{U_3} :: U_2 \sqcap U_3 & b = \mathsf{false} \end{cases}$$

$$\varepsilon_1(\varepsilon_2 v :: U) \longrightarrow_c \begin{cases} (\varepsilon_2 \circ^{=} \varepsilon_1)v \\ \mathbf{error} \qquad \text{if not defined} \end{cases}$$

Fig. 8. Syntax and notions of Reduction

Case (U+). Then $\widetilde{t}_1 = \widetilde{t'_1} + \widetilde{t'_2}$ and $U_1 = \mathsf{Int}$. By $(U+)$ we know that:

$$(\mathrm{T}+)\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \qquad \Gamma \vdash \widetilde{t}_2 : U_2 \qquad U_1 \sim \mathsf{Int} \qquad U_2 \sim \mathsf{Int}}{\Gamma \vdash \widetilde{t}_1 + \widetilde{t}_2 : \mathsf{Int}} \tag{3}$$

Consider $\widetilde{t}_2$ such that $\widetilde{t}_1 \sqsubseteq \widetilde{t}_2$. By definition of term precision $\widetilde{t}_2$ must have the form $\widetilde{t}_2 = \widetilde{t''}_1 + \widetilde{t''}_2$ and therefore

$$(\mathrm{P}+)\frac{\widetilde{t'_1} \sqsubseteq \widetilde{t''}_1 \qquad \widetilde{t'_2} \sqsubseteq \widetilde{t''}_2}{\widetilde{t'_1} + \widetilde{t'_2} \sqsubseteq \widetilde{t''}_1 + \widetilde{t''}_2} \tag{4}$$

Using induction hypotheses on the premises of 3, $\Gamma \vdash \widetilde{t''_1} : U'_1$ and $\Gamma \vdash \widetilde{t''_2} : U'_2$, where $U_1 \sqsubseteq U'_1$ and $U_2 \sqsubseteq U'_2$. By Lemma 4, $U'_1 \sim \mathsf{Int}$ and $U'_2 \sim \mathsf{Int}$. Therefore we can use rule $(U+)$ to derive:

$$(\mathrm{T}+)\frac{\Gamma \vdash \widetilde{t'_1} : U'_1 \qquad \Gamma \vdash \widetilde{t'_2} : U'_2 \qquad U'_1 \sim \mathsf{Int} \qquad U'_2 \sim \mathsf{Int}}{\Gamma \vdash \widetilde{t'_1} + \widetilde{t'_2} : \mathsf{Int}}$$

and the result holds.

$\boxed{\longmapsto : \textsc{Term}_U \times (\textsc{Term}_U \cup \{\,\mathbf{error}\,\})}$ **Reduction**

$$(\mathrm{R}\longrightarrow)\frac{t^U \longrightarrow r \qquad r \in (\textsc{Term}_U \cup \{\,\mathbf{error}\,\})}{t^U \longmapsto r} \qquad (\mathrm{R}g)\frac{et \longrightarrow_c et'}{g[et] \longmapsto g[et']}$$

$$(\mathrm{R}g\mathrm{err})\frac{et \longrightarrow_c \mathbf{error}}{g[et] \longmapsto \mathbf{error}} \qquad (\mathrm{R}f)\frac{t_1^U \longmapsto t_2^U}{f[t_1^U] \longmapsto f[t_2^U]} \qquad (\mathrm{R}f\mathrm{err})\frac{t_1^U \longmapsto \mathbf{error}}{f[t_1^U] \longmapsto \mathbf{error}}$$

Fig. 9. Intrinsic Reduction

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

$$(U\mathrm{app})\frac{\Gamma \vdash \widetilde{t'_1} : U'_1 \qquad \Gamma \vdash \widetilde{t'_2} : U'_2 \qquad U'_2 \sim \widetilde{dom}(U'_1)}{\Gamma \vdash \widetilde{t'_1}\ \widetilde{t'_2} : \widetilde{cod}(U'_1)} \tag{5}$$

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

$$(\mathrm{Papp})\frac{\widetilde{t'_1} \sqsubseteq \widetilde{t''_1} \qquad \widetilde{t'_2} \sqsubseteq \widetilde{t''_2}}{\widetilde{t'_1}\ \widetilde{t'_2} \sqsubseteq \widetilde{t''_1}\ \widetilde{t''_2}} \tag{6}$$

Using induction hypotheses on the premises of 5, $\Gamma \vdash \widetilde{t''}_1 : U''_1$ and $\Gamma \vdash \widetilde{t''}_2 : U''_2$, where $U'_1 \sqsubseteq U''_1$ and $U'_2 \sqsubseteq U''_2$. By definition precision (Def. 2) and the definition of $\widetilde{dom}$, $\widetilde{dom}(U'_1) \sqsubseteq \widetilde{dom}(U''_1)$ and, therefore by Lemma 4, $U''_2 \sim \widetilde{dom}(U''_1)$. Also, by the previous argument $\widetilde{cod}(U'_1) \sqsubseteq \widetilde{cod}(U''_1)$ Then we can use rule (Uapp) to derive:

$$(U\mathrm{app})\frac{\Gamma \vdash \widetilde{t''_1} : U''_1 \qquad \Gamma \vdash \widetilde{t''_2} : U''_2 \qquad U''_2 \sim \widetilde{dom}(U''_1)}{\Gamma \vdash \widetilde{t''_1}\ \widetilde{t''_2} : \widetilde{cod}(U''_1)}$$

and the result holds.

Case (U if). Then $\widetilde{t}_1 = \mathsf{if}\ \widetilde{t'_1}\ \mathsf{then}\ \widetilde{t'_2}\ \mathsf{else}\ \widetilde{t'_3}$ and $U_1 = (U'_2 \sqcap U'_3)$. By (Uif) we know that:

$$(U\mathrm{if})\frac{\Gamma \vdash \widetilde{t'_1} : U'_1 \qquad \Gamma \vdash \widetilde{t'_2} : U'_2 \qquad \Gamma \vdash \widetilde{t'_3} : U'_3}{\Gamma \vdash \mathsf{if}\ \widetilde{t'_1}\ \mathsf{then}\ \widetilde{t'_2}\ \mathsf{else}\ \widetilde{t'_3} : (U'_2 \sqcap U'_3)} \tag{7}$$

Consider $\widetilde{t_2}$ such that $\widetilde{t}_1 \sqsubseteq \widetilde{t}_2$. By definition of term precision $\widetilde{t}_2$ must have the form $\widetilde{t}_2 = \mathsf{if}\ \widetilde{t''_1}\ \mathsf{then}\ \widetilde{t''_2}\ \mathsf{else}\ \widetilde{t''_3}$ and therefore

$$(\mathrm{Pif})\frac{\widetilde{t'_1} \sqsubseteq \widetilde{t''_1} \qquad \widetilde{t'_1} \sqsubseteq \widetilde{t''_1} \qquad \widetilde{t'_2} \sqsubseteq \widetilde{t''_2}}{\mathsf{if}\ \widetilde{t'_1}\ \mathsf{then}\ \widetilde{t'_2}\ \mathsf{else}\ \widetilde{t'_3} \sqsubseteq \mathsf{if}\ \widetilde{t''_1}\ \mathsf{then}\ \widetilde{t''_2}\ \mathsf{else}\ \widetilde{t''_3}} \tag{8}$$

Then we can use induction hypotheses on the premises of 7 and derive:

$$(U\text{if})\frac{\Gamma \vdash \widetilde{t_1''} : U_1'' \qquad \Gamma \vdash \widetilde{t_2''} : U_2'' \qquad \Gamma \vdash \widetilde{t_3''} : U_3''}{\Gamma \vdash \mathsf{if}\ \widetilde{t_1''}\ \mathsf{then}\ \widetilde{t_2''}\ \mathsf{else}\ \widetilde{t_3''} : (U_2'' \sqcap U_3'')}$$

Where $U_1' \sqsubseteq U_1''$ and $U_2' \sqsubseteq U_2''$. Using the definition of type precision (Def. 2) we can infer that

$$(U_1' \sqcap U_2') \sqsubseteq (U_1'' \sqcap U_2'')$$

and the result holds.

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

$$(U::)\frac{\Gamma \vdash \widetilde{t} : U_1' \qquad U_1' \sim U_1}{\Gamma \vdash \widetilde{t} :: U_1' : U_1} \tag{9}$$

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

$$(\text{P}::)\frac{\widetilde{t} \sqsubseteq \widetilde{t'} \qquad U_1 \sqsubseteq U_2}{\widetilde{t} :: U_1 \sqsubseteq \widetilde{t'} :: U_2} \tag{10}$$

Using induction hypotheses on the premises of 9, $\Gamma \vdash \widetilde{t'} : U_2'$ where $U_1' \sqsubseteq U_2'$. We can use rule $(U::)$ and Lemma 4 to derive:

$$(U::)\frac{\Gamma \vdash \widetilde{t'} : U_2' \qquad U_2' \sim U_2}{\Gamma \vdash \widetilde{t'} :: U_2 : U_2}$$

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

4.2 Type Safety

In this section we present the proof of type safety for GTFL$^{\oplus}$.

Lemma 5 (Canonical forms). *Consider a value $v \in \text{TERM}_U$. Then either $v = u$, or $v = \varepsilon u :: U$ with $u \in \text{TERM}_{U'}$ and $\varepsilon \vdash U' \sim U$. Furthermore:*

1. *If $U = \mathsf{Bool}$ then either $v = b$ or $v = \varepsilon b :: \mathsf{Bool}$ with $b \in \text{TERM}_{\mathsf{Bool}}$.*
2. *If $U = \mathsf{Int}$ then either $v = n$ or $v = \varepsilon n :: \mathsf{Int}$ with $n \in \text{TERM}_{\mathsf{Int}}$.*
3. *If $U = U_1 \to U_2$ then either $v = (\lambda x^{U_1}.t^{U_2})$ with $t^{U_2} \in \text{TERM}_{U_2}$ or $v = \varepsilon(\lambda x^{U_1'}.t^{U_2'}) :: U_1 \to U_2$ with $t^{U_2'} \in \text{TERM}_{U_2'}$ and $\varepsilon \vdash U_1' \to U_2' \sim U_1 \to U_2$.*

Proof. By direct inspection of the formation rules of gradual intrinsic terms (Figure 5).

Lemma 6 (Substitution). *If $t^U \in \text{TERM}_U$ and $v \in \text{TERM}_{U_1}$, then $[v/x^{U_1}]t^U \in \text{TERM}_U$.*

Proof. By induction on the derivation of t^U.

Proposition 20 ($\to$ is well defined). *If $t^U \to r$, then $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$.*

Proof. By induction on the structure of a derivation of $t^U \to r$, considering the last rule used in the derivation.

Case (IU$\oplus$). Then $t^U = \varepsilon_1 n_1 + \varepsilon_2 n_2$. Then

$$(\mathrm{IU}+)\frac{n_1 \in \textsc{Term}_{\mathsf{Int}} \quad \varepsilon_1 \vdash \mathsf{Int} \sim \mathsf{Int} \quad n_2 \in \textsc{Term}_{\mathsf{Int}} \quad \varepsilon_2 \vdash \mathsf{Int} \sim \mathsf{Int}}{\varepsilon_1 n_1 + \varepsilon_2 n_2 \in \textsc{Term}_{\mathsf{Int}}}$$

Therefore

$$\varepsilon_1 n_1 + \varepsilon_2 n_2 \to n_3 \text{ where } n_3 = n_1 \,[\![+]\!]\, n_2$$

But $n_3 \in \textsc{Term}_{\mathsf{Int}}$ and the result holds.

Case (IU app). Then $t^U = \varepsilon_1(\lambda x^{U_{11}}.t_1^{U12})\ @^{U_1 \to U_2}\ (\varepsilon_2 u)$ and $U = U_2$. Then

$$(\mathrm{Iapp})\frac{\dfrac{\dfrac{\mathcal{D}_1}{t_1^{U12} \in \textsc{Term}_{U_{12}}}}{(\lambda x^{U_{11}}.t_1^{U12}) \in \textsc{Term}_{U_{11} \to U_{12}}} \quad \dfrac{\mathcal{D}_2}{u \in \textsc{Term}_{U_2'}} \quad \varepsilon_2 \vdash U_2' \sim U_1}{\varepsilon_1 \vdash U_{11} \to U_{12} \sim U_1 \to U_2}}{\varepsilon_1(\lambda x^{U_{11}}.t_1^{U12})\ @^{U_1 \to U_2}\ \varepsilon_2 u \in \textsc{Term}_{U_2}}$$

If $\varepsilon' = (\varepsilon_2 \circ^{=} \mathit{idom}(\varepsilon_1))$ is not defined, then $t^U \to \mathbf{error}$, and then the result hold immediately. Suppose that consistent transitivity does hold, then

$$\varepsilon_1(\lambda x^{U_{11}}.t_1^{U12})@^{U_1 \to U_2}\ \varepsilon_2 u \in \textsc{Term}_{U_2} \to \mathit{icod}(\varepsilon_1)([(\varepsilon' u :: U_{11})/x^{U_{11}}]t) :: U_2$$

As $\varepsilon_2 \vdash U_2' \sim U_1$ and by inversion lemma $\mathit{idom}(\varepsilon_1) \vdash U_1 \sim U_{11}$, then $\varepsilon' \vdash U_2' \sim U_{11}$. Therefore $\varepsilon' u :: U_{11} \in \textsc{Term}_{U_{11}}$, and by Lemma 6, $t'^{U_{12}} = [(\varepsilon' u :: U_{11})/x^{U_{11}}]t^{U_{12}} \in \textsc{Term}_{U_{12}}$.

Then

$$(\mathrm{IU}::)\frac{t'^{U_{12}} \in \textsc{Term}_{U_{12}} \qquad \mathit{icod}(\varepsilon_1) \vdash U_{12} \sim U_2}{\mathit{icod}(\varepsilon_1) t'^{U_{12}} :: U_2 \in \textsc{Term}_{U_2}}$$

and the result holds.

Case (IU if-true). Then $t^U = \mathsf{if}\ \varepsilon_1 b\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3}$, $U = U_2 \sqcap U_3$ and

$$(\mathrm{IU}\mathsf{if})\frac{\begin{array}{cc} b \in \textsc{Term}_{U_1} & \varepsilon_1 \vdash U_1 \sim \mathsf{Bool} \quad U = (U_2 \sqcap U_3) \\ t^{U_2} \in \textsc{Term}_{U_2} & \varepsilon_2 \vdash U_2 \sim U \\ t^{U_3} \in \textsc{Term}_{U_3} & \varepsilon_3 \vdash U_3 \sim U \end{array}}{\mathsf{if}\ \varepsilon_1 b\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \in \textsc{Term}_U}$$

Therefore

$$\mathsf{if}\ \varepsilon_1 b\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \to \varepsilon_2 t^{U_2} :: U_2 \sqcap U_3$$

But

$$(\mathrm{IU::})\frac{t^{U_2} \in \textsc{Term}_{U_2} \qquad \varepsilon \vdash U_2 \sim U_2 \sqcap U_3}{\varepsilon_2 t^{U_2} :: U_2 \sqcap U_3 \in \textsc{Term}_{U_2 \sqcap U_3}}$$

and the result holds.

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

Proposition 21 ($\longmapsto$ is well defined). *If $t^U \longmapsto r$, then $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$.*

Proof. By induction on the structure of a derivation of $t^U \longmapsto r$.

Case (R$\to$). $t^U \to r$. By well-definedness of $\to$ (Prop 20), $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$.

Case (Rf). $t^U = f[t_1^{U'}]$, $f[t^{U'}] \in \textsc{Term}_U$, $t_1^{U'} \longmapsto t_2^{U'}$, $t^{U'} \in \textsc{Term}_{U'}$, and $f : \textsc{Term}_{U'} \to \textsc{Term}_U$. By induction hypothesis, $t_2^{U'} \in \textsc{Term}_{U'}$, so $f[t_2^{U'}] \in \textsc{Term}_U$.

Case (Rferr, Rgerr). $r = \mathbf{error}$.

Case (Rg). $t^U = g[et]$, $g[t^{U'}] \in \textsc{Term}_U$, and $g : \textsc{EvTerm} \to \textsc{Term}_U$, and $et \to_c et'$. Then there exists U_e, U_x such that $et = \varepsilon_e t_e^{U_e}$ and $\varepsilon_e \vdash U_e \sim U_x$. Also, $t_e = \varepsilon_v v :: U_e$, with $v \in \textsc{Term}_{U_v}$ and $\varepsilon_v \vdash U_v \sim U_e$.
We know that $\varepsilon_c = \varepsilon_v \circ^{=} \varepsilon_e$ is defined, and $et = \varepsilon_e t_e \to_c \varepsilon_c v = et'$. By definition of $\circ^{=}$ we have $\varepsilon_c \vdash U_v \sim U_x$, so $g[et'] \in \textsc{Term}_U$.

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 22 (Type Safety). *If $t^U \in \textsc{Term}_U$ then either t^U is a value v; $t^U \longmapsto \mathbf{error}$; or $t^U \longmapsto t'^U$ for some term $t'^U \in \textsc{Term}_U$.*

Proof. By induction on the structure of t^U.

Case (Iu,In, Ib, Ix, Iλ). t^U is a value.

Case (I::). $t^U = \varepsilon_1 t^{U_1} :: U_2$, and

$$(\mathrm{I::})\frac{t^{U_1} \in \textsc{Term}_{U_1} \qquad \varepsilon_1 \vdash U_1 \sim U_2}{\varepsilon_1 t^{U_1} :: U_2 \in \textsc{Term}_{U_2}}$$

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. $t^{U_1} \longmapsto r_1$ for some $r_1 \in \textsc{Term}_{U_1} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rf), or (R$f$err).

Case (IUif). $t^U = \mathsf{if}\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3}$ and

$$(\mathrm{I}U\mathrm{if})\frac{\begin{array}{ccc} t^{U_1} \in \textsc{Term}_{U_1} & \varepsilon_1 \vdash U_1 \sim \mathsf{Bool} & U = (U_2 \sqcap U_3) \\ t^{U_2} \in \textsc{Term}_{U_2} & & \varepsilon_2 \vdash U_2 \sim U \\ t^{U_3} \in \textsc{Term}_{U_3} & & \varepsilon_3 \vdash U_3 \sim U \end{array}}{\mathsf{if}\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \in \textsc{Term}_U}$$

By induction hypothesis on t^{U_1}, one of the following holds:

1. t^{U_1} is a value u, then by (R$\rightarrow$), $t^U \longmapsto r$ and $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21.
2. t^{U_1} is an ascribed value v, then, $\varepsilon_1 t^{U_1} \rightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rg), or (Rgerr).
3. $t^{U_1} \longmapsto r_1$ for some $r_1 \in \textsc{Term}_{U_1} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rf), or (R$f$err).

Case (IUapp). $t^U = (\varepsilon_1 t^{U_1})\ @^{U_{11}\rightarrow U_{12}}\ (\varepsilon_2 t^{U_2})$

$$(\mathrm{I}U\mathrm{app})\frac{\begin{array}{cc} t^{U_1} \in \textsc{Term}_{U_1} & \varepsilon_1 \vdash U_1 \sim U_{11} \rightarrow U_{12} \\ t^{U_2} \in \textsc{Term}_{U_2} & \varepsilon_2 \vdash U_2 \sim U_{11} \end{array}}{(\varepsilon_1 t^{U_1})\ @^{U_{11}\rightarrow U_{12}}\ (\varepsilon_2 t^{U_2}) \in \textsc{Term}_{U_{12}}}$$

By induction hypothesis on t^{U_1}, one of the following holds:

1. t^{U_1} is a value $(\lambda x^{U'_{11}}.t^{U'_{12}})$ (by canonical forms Lemma 5), posing $U_1 = U'_{11} \rightarrow U'_{12}$.
   Then by induction hypothesis on t^{U_2}, one of the following holds:
   (a) t^{U_2} is a value u, then by (R$\rightarrow$), $t^U \longmapsto r$ and $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21.
   (b) t^{U_2} is an ascribed value v, then, $\varepsilon_2 t^{U_2} \rightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rg), or (Rgerr).
   (c) $t^{U_2} \longmapsto r_2$ for some $r_2 \in \textsc{Config}_{U_2} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rf), or (R$f$err).
2. t^{U_1} is an ascribed value v, then, $\varepsilon_1 t^{U_1} \rightarrow_c et'$ for some $et' \in \textsc{EvTerm} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rg), or (Rgerr).
3. $t^{U_1} \longmapsto r_1$ for some $r_1 \in \textsc{Config}_{U_1} \cup \{\,\mathbf{error}\,\}$. Hence $t^U \longmapsto r$ for some $r \in \textsc{Config}_U \cup \{\,\mathbf{error}\,\}$ by Prop 21 and either (Rf), or (R$f$err).

Case (IU+). Similar case to (IUapp)

4.3 Dynamic Gradual Guarantee

In this section we present the proof the Dynamic Gradual Guarantee for GTFL$^{\oplus}$.

$$\frac{}{\Omega \cup \{\, x^{U_1} \sqsubseteq x^{U_2} \,\} \vdash x^{U_1} \sqsubseteq x^{U_2}} \qquad \frac{}{\Omega \vdash b \sqsubseteq b} \qquad \frac{}{\Omega \vdash n \sqsubseteq n}$$

$$\frac{U_{11} \sqsubseteq U_{12} \qquad \Omega \cup \{\, x^{U_{11}} \sqsubseteq x^{U_{12}} \,\} \vdash t^{U_{12}} \sqsubseteq t^{U_{22}}}{\Omega \vdash (\lambda x^{U_{11}}.t^{U_{12}}) \sqsubseteq (\lambda x^{U_{21}}.t^{U_{22}})}$$

$$\frac{\begin{array}{c}\Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \\ U_{12} \sqsubseteq U_{22} \qquad \varepsilon_1 \sqsubseteq \varepsilon_2\end{array}}{(\varepsilon_1 t^{U_{11}} :: U_{12}) \sqsubseteq (\varepsilon_2 t^{U_{21}} :: U_{22})} \qquad \frac{\begin{array}{c}\Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} \qquad \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \\ \varepsilon_{11} \sqsubseteq \varepsilon_{21} \qquad\qquad \varepsilon_{12} \sqsubseteq \varepsilon_{22} \\ U_1 \sqsubseteq U_2\end{array}}{\begin{array}{l}\Omega \vdash \varepsilon_{11} t^{U_{11}} \mathbin{@^{U_1}} \varepsilon_{12} t^{U_{12}} \sqsubseteq \\ \varepsilon_{21} t^{U_{21}} \mathbin{@^{U_2}} \varepsilon_{22} t^{U_{22}}\end{array}}$$

$$\frac{\begin{array}{cc}\Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} & \varepsilon_{11} \sqsubseteq \varepsilon_{21} \\ \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{23}} & \varepsilon_{12} \sqsubseteq \varepsilon_{22} \\ \Omega \vdash t^{U_{13}} \sqsubseteq t^{U_{23}} & \varepsilon_{13} \sqsubseteq \varepsilon_{23}\end{array}}{\begin{array}{l}\Omega \vdash \mathsf{if}\ \varepsilon_{11} t^{U_{11}}\ \mathsf{then}\ \varepsilon_{12} t^{U_{12}}\ \mathsf{else}\ \varepsilon_{13} t^{U_{13}} \sqsubseteq \\ \mathsf{if}\ \varepsilon_{21} t^{U_{21}}\ \mathsf{then}\ \varepsilon_{22} t^{U_{22}}\ \mathsf{else}\ \varepsilon_{23} t^{U_{23}}\end{array}}$$

$$\frac{\begin{array}{cc}\Omega \vdash t^{U_{11}} \sqsubseteq t^{U_{21}} & \Omega \vdash t^{U_{12}} \sqsubseteq t^{U_{22}} \\ \varepsilon_{11} \sqsubseteq \varepsilon_{21} & \varepsilon_{12} \sqsubseteq \varepsilon_{22}\end{array}}{\Omega \vdash (\varepsilon_{11} t^{U_{11}} + \varepsilon_{12} t^{U_{12}}) \sqsubseteq (\varepsilon_{21} t^{U_{21}} + \varepsilon_{22} t^{U_{22}})}$$

Fig. 10. Intrinsic term precision

Definition 18 (Intrinsic term precision). *Let* $\Omega \in \mathcal{P}(\textsc{Var}_* \times \textsc{Var}_*)$ *be defined as* $\Omega ::= \{\, \overline{x^{U_{i1}} \sqsubseteq x^{U_{i2}}} \,\}$ *We define an ordering relation* $(\cdot \vdash \cdot \sqsubseteq \cdot) \in (\mathcal{P}(\textsc{Var}_* \times \textsc{Var}_*) \times \textsc{Term}_* \times \textsc{Term}_*$ *shown in Figure 10.*

Definition 19 (Well Formedness of Ω). *We say that* Ω *is well formed iff* $\forall \{\, x^{G_{i1}} \sqsubseteq x^{G_{i2}} \,\} \in \Omega. G_{i1} \sqsubseteq G_{i2}$

Before proving the gradual guarantee, we first establish some auxiliary properties of precision. For the following propositions, we assume Well Formedness of Ω (Definition 19).

Proposition 23. *If* $\Omega \vdash t^{U_1} \sqsubseteq t^{U_2}$ *for some* $\Omega \in \mathcal{P}(\textsc{Var}_* \times \textsc{Var}_*)$*, 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 24. *Let* $g_1, g_2 \in \textsc{EvFrame}$ *such that* $g[\varepsilon_{11} t_1^{U_1}] \in \textsc{Term}_{U_1'}$, $g[\varepsilon_{21} t_1^{U_2}] \in \textsc{Term}_{U_2'}$*, 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 $(\Box\ @^U\ et)$. Then for $i \in \{1,2\}$ g_i must have the form $\Box\ @^{U_i''}\ \varepsilon_i' t^{U_i'}$ 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 $\sqsubseteq_{APP}$ $\varepsilon_{12} t_2^{U_1}\ @^{U_1''}\ \varepsilon_1' t^{U_1'} \sqsubseteq \varepsilon_{22} t_2^{U_2}\ @^{U_2''}$ $\varepsilon_2' t^{U_2'}$, and the result holds.

Case $(\Box + et, ev + \Box, ev\ @^U\ \Box, \Box :: U, \mathsf{if}\ \Box\ \mathsf{then}\ et\ \mathsf{else}\ et)$. Straightforward using similar argument to the previous case.

Proposition 25. *Let* $g_1, g_2 \in$ EvFrame *such that* $g_1[\varepsilon_1 t^{U_1}] \in \text{Term}_{U_1'}$, $g_2[\varepsilon_2 t^{U_2}] \in \text{Term}_{U_2'}$, *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 $(\Box\ @^U\ et)$. Then there must exist some U_i'', ε_i' and $t^{U_i'}$ such that $g[\varepsilon_1 t^{U_1}] = \varepsilon_1 t^{U_1}\ @^{U_1''}\ \varepsilon_1' t^{U_1'}$ and $g[\varepsilon_2 t^{U_2}] = \varepsilon_2 t^{U_2}\ @^{U_2''}\ \varepsilon_2' t^{U_2'}$. 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 $(\Box + et, ev + \Box, ev\ @^U\ \Box, \Box :: U, \mathsf{if}\ \Box\ \mathsf{then}\ et\ \mathsf{else}\ et)$. Straightforward using similar argument to the previous case.

Proposition 26. *Let* $f_1, f_2 \in$ EvFrame *such that* $f_1[t_1^{U_1}] \in \text{Term}_{U_1'}$, $f_2[t_1^{U_2}] \in \text{Term}_{U_2'}$, *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 $g_1[\varepsilon_1 t_1^{U_1}] \in \text{Term}_{U_1'}$, $g_2[\varepsilon_2 t_1^{U_2}] \in \text{Term}_{U_2'}$ 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 25, $\varepsilon_1 \sqsubseteq \varepsilon_2$. Finally by Prop 24 we conclude that $g_1[\varepsilon_1 t_2^{U_1}] \sqsubseteq g_1[\varepsilon_1 t_2^{U_2}]$.

Proposition 27. *Let* $f_1, f_2 \in$ EvFrame *such that* $f_1[t^{U_1}] \in \text{Term}_{U_1'}$, $f_2[t^{U_2}] \in \text{Term}_{U_2'}$, *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 $g_1[\varepsilon_1 t_1^{U_1}] \in \text{Term}_{U_1'}$, $g_2[\varepsilon_2 t_1^{U_2}] \in \text{Term}_{U_2'}$ 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 25 we conclude that $t^{U_1} \sqsubseteq t^{U_2}$.

Proposition 28 (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.

Proposition 29 (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.

Proposition 30. *If $U_{11} \sqsubseteq U_{12}$ and $U_{21} \sqsubseteq U_{22}$ then $U_{11} \sqcap U_{21} \sqsubseteq U_{12} \sqcap U_{22}$.*

Proof. By induction on the type derivation of the types and meet.

Proposition 31 (Dynamic guarantee for $\longrightarrow$). *Suppose $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$. If $t_1^{U_1} \longrightarrow t_2^{U_1}$ then $t_1^{U_2} \longrightarrow t_2^{U_2}$, where $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$ 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 ($\longrightarrow +$). We know that $t_1^{U_1} = (\varepsilon_{11}(n_1) + \varepsilon_{12}(n_2))$ then by $(\sqsubseteq_+)$ $t_1^{U_2} = (\varepsilon_{21}(n_1) + \varepsilon_{22}(n_2))$ for some $\varepsilon_{21}, \varepsilon_{22}$ such that $\varepsilon_{11} \sqsubseteq \varepsilon_{21}$ and $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$.
If $t_1^{U_1} \longrightarrow n_3$ where $n_3 = (n_1 \, [\![\oplus]\!] \, n_2)$, then $t_1^{U_2} \longrightarrow n_3'$ where $n_3' = (n_1 \, [\![\oplus]\!] \, n_2)$. But $n_3 = n_3'$ and therefore $t_2^{U_1} \sqsubseteq t_2^{U_2}$ and the result holds.

Case ($\longrightarrow$app). We know that
$t_1^{U_1} = \varepsilon_{11}(\lambda x^{U_{11}}.t^{U_{12}}) \, @^{U_1 \longrightarrow U_2} \, \varepsilon_{12} u$ then by $(\sqsubseteq_{app})$ $t_1^{U_2}$ must have the form $t_1^{U_2} = \varepsilon_{21}(\lambda x^{U_{21}}.t^{U_{22}}) \, @^{U_3 \longrightarrow U_4} \, \varepsilon_{22} u_2$ for some $\varepsilon_{21}, x^{U_{21}}, t^{U_{22}}, U_3, U_4, \varepsilon_{22}$ and u_2.
Let us pose $\varepsilon_1 = \varepsilon_{12} \circ^= idom(\varepsilon_{11})$. Then
$t_1^{U_1} \longrightarrow icod(\varepsilon_{11}) t_1' :: U_2$ with $t_1' = [(\varepsilon_1 u_1 :: U_{11})/x^{U_{11}}]t^{U_{12}}$.
Also, let us pose $\varepsilon_2 = \varepsilon_{22} \circ^= idom(\varepsilon_{21})$. Then
$t_1^{U_2} \longrightarrow icod(\varepsilon_{21}) t_2' :: U_4$ 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 29 $\varepsilon_1 \sqsubseteq \varepsilon_2$. 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 28) $t_1' \sqsubseteq t_2'$, therefore $icod(\varepsilon_{11}) t_1' :: U_2 \sqsubseteq icod(\varepsilon_{21}) t_2' :: U_4$ by $(\sqsubseteq_{::})$. Then $t_2^{U_1} \sqsubseteq t_2^{U_2}$.

Case ($\longrightarrow$if-true). $t_1^{U_1} = \mathsf{if} \; \varepsilon_{11} \mathsf{true} \; \mathsf{then} \; \varepsilon_{12} t^{U_{12}} \; \mathsf{else} \; \varepsilon_{13} t^{U_{13}}$ then by $(\sqsubseteq_{if})$ $t_1^{U_2}$ has the form $t_1^{U_2} = \mathsf{if} \; \varepsilon_{21} \mathsf{true} \; \mathsf{then} \; \varepsilon_{22} t^{U_{22}} \; \mathsf{else} \; \varepsilon_{23} t^{U_{23}}$ for some $\varepsilon_{21}, \varepsilon_{22}, t^{U_2 2}, \varepsilon_{23}$, and $t^{U_{32}}$. Then $t_1^{U_1} \longrightarrow \varepsilon_{12} t^{U_{12}} :: (U_{12} \sqcap U_{13})$, and $t_1^{U_2} \longrightarrow \varepsilon_{22} t^{U_{22}} :: (U_{22} \sqcap U_{23})$. Using the fact that $t_1^{U_1} \sqsubseteq t_2^{U_2}$ we know that $\varepsilon_{12} \sqsubseteq \varepsilon_{22}$, $t^{U_{12}} \sqsubseteq t^{U_{22}}$ and by Prop 23, $U_{12} \sqsubseteq U_{22}$ and $U_{13} \sqsubseteq U_{23}$. Therefore by Prop 30 $(U_{12} \sqcap U_{13}) \sqsubseteq (U_{22} \sqcap U_{23})$. Then using $(\sqsubseteq_{::})$, $t_2^{U_1} \sqsubseteq t_2^{U_2}$.

Case ($\longrightarrow$if-false). Same as case $\longrightarrow$if-true, using the fact that $\varepsilon_{13} \sqsubseteq \varepsilon_{23}$ and $t^{U_{13}} \sqsubseteq t^{U_{23}}$.

Proposition 32 (Dynamic guarantee). *Suppose $t_1^{U_1} \sqsubseteq t_1^{U_2}$. If $t_1^{U_1} \longmapsto t_2^{U_1}$ then $t_1^{U_2} \longmapsto t_2^{U_2}$ where $t_2^{U_1} \sqsubseteq t_2^{U_2}$.*

Proof. We prove the following property instead: Suppose $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$. If $t_1^{U_1} \longmapsto t_2^{U_1}$ then $t_1^{U_2} \longmapsto t_2^{U_2}$ where $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$, 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⟶). $\Omega \vdash t_1^{U_1} \sqsubseteq t_1^{U_2}$, $t_1^{U_1} \longrightarrow t_2^{U_1}$. By dynamic guarantee of $\longrightarrow$ (Prop 31), $t_1^{U_2} \longrightarrow t_1^{U_2}$ where $\Omega' \vdash t_2^{U_1} \sqsubseteq t_2^{U_2}$, 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 23, $U_1' \sqsubseteq U_2'$. By Prop 27, we also know that $\Omega \vdash t_1^{U_1'} \sqsubseteq t_1^{U_2'}$. By induction hypothesis, $t_1^{U_1'} \longmapsto t_2^{U_1'}$, $t_1^{U_2'} \longmapsto t_2^{U_2'}$, $\Omega' \vdash t_2^{U_1'} \sqsubseteq t_2^{U_2'}$ for some $\Omega' \supseteq \Omega$. Then by Prop 26 then $\Omega' \vdash f_1[t_2^{U_1'}] \sqsubseteq f_2[t_2^{U_2'}]$ 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 25, $\varepsilon_{11} \sqsubseteq \varepsilon_{21}$, 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 29 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 24, it follows that $\Omega \vdash g_1[et_1'] \sqsubseteq g_1[et_2']$.

5 Compiling GTFL$^{\oplus}$ to Threesomes

This section presents the translational semantics of GTFL$^{\oplus}$. Section 5.1 presents the intermediate language. Section 5.2 presents the cast insertion rules, and Section 5.3 presents the full formalization of the correctness of the translational semantics.

5.1 Intermediate language: GTFL$^{\oplus}_{\Rightarrow}$

Figure 11 presents the syntax of GTFL$^{\oplus}_{\Rightarrow}$. Figure 12 presents the full type system of GTFL$^{\oplus}_{\Rightarrow}$.

Figure 13 presents the full dynamic semantics of GTFL$^{\oplus}_{\Rightarrow}$. Applications of a non-casted function are standard. The reduction rule for additions and conditionals use the *rval* metafunction, which strips away the surrounding cast, if any, to access the underlying value. The reduction of the application of a casted function is standard, splitting the function cast into a cast on the argument and a cast on the result. Two threesomes that coincide on their source/target types are combined by meeting their middle types. If the meet is undefined then the term steps to **error**. Otherwise both casts are merged to a new cast where the middle type is now the meet between the middle types. Note that new casts are introduced using the following cast metafunction, which avoids producing useless threesomes:

$$T \in \textsc{Type}, \quad x \in \textsc{Var}, \quad t \in \textsc{Term}, \quad \Gamma \in \textsc{Var} \xrightarrow{\text{fin}} \textsc{Type}$$

$$T ::= \mathsf{Bool} \mid \mathsf{Int} \mid T \to T \qquad \text{(types)}$$

$$u ::= n \mid b \mid (\lambda x : U.t) \qquad \text{(simple values)}$$

$$v ::= u \mid \langle U \overset{U}{\Leftarrow} U \rangle u \qquad \text{(values)}$$

$$t ::= v \mid x \mid t\ t \mid t + t \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t \mid t :: T \mid \langle U \overset{U}{\Leftarrow} U \rangle t \qquad \text{(terms)}$$

Fig. 11. Syntax of the intermediate language

$$(IT\text{x})\ \frac{x : U \in \Gamma}{\Gamma \vdash_i x : U} \qquad (\text{ITb})\ \frac{}{\Gamma \vdash_i b : \mathsf{Bool}} \qquad (\text{ITn})\ \frac{}{\Gamma \vdash_i n : \mathsf{Int}}$$

$$(\text{IT}\lambda)\ \frac{\Gamma, x : U_1 \vdash_i t : U_2}{\Gamma \vdash_i (\lambda x : U_1.t) : U_1 \to U_2} \qquad (\text{ITapp})\ \frac{\Gamma \vdash_i t_1 : U_1 \qquad \Gamma \vdash_i t_2 : \widetilde{dom}(U_1)}{\Gamma \vdash_i t_1\ t_2 : \widetilde{cod}(U_1)}$$

$$(\text{IT}\langle\rangle)\ \frac{\Gamma \vdash_i t : U_1 \qquad U_1 \sim U_2 \qquad U_1 \sim U_3 \qquad U_3 \sim U_2}{\Gamma \vdash_i \langle U_2 \overset{U_3}{\Leftarrow} U_1 \rangle t : U_2} \qquad (\text{IT}+)\ \frac{\Gamma \vdash_i t_1 : \mathsf{Int} \qquad \Gamma \vdash_i t_2 : \mathsf{Int}}{\Gamma \vdash_i t_1 + t_2 : \mathsf{Int}}$$

$$(\text{ITif})\ \frac{\Gamma \vdash_i t_1 : \mathsf{Bool} \qquad \Gamma \vdash_i t_2 : U_2 \qquad \Gamma \vdash_i t_3 : U_2}{\Gamma \vdash_i \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : U_2}$$

Fig. 12. GTFL$^{\oplus}_{\Rightarrow}$: Type system for the intermediate language

$$\langle\!\langle U_2 \overset{U_3}{\Leftarrow} U_1 \rangle\!\rangle t = \begin{cases} t & \text{if } U_1 = U_2 = U_3 \\ \langle U_2 \overset{U_3}{\Leftarrow} U_1 \rangle t & \text{otherwise} \end{cases}$$

5.2 Cast Insertion

We now briefly describe the cast insertion translation from a GTFL$^{\oplus}$ term $\widetilde{t}$ to a GTFL$^{\oplus}_{\Rightarrow}$ term t. The cast insertion rules are presented in Figure 14. Cast insertion rules use twosomes to ease readability; a twosome $\langle U_2 \Leftarrow U_1 \rangle t$ is equal to $\langle\!\langle U_2 \overset{U_1 \sqcap U_2}{\Longleftarrow} U_1 \rangle\!\rangle t$: the initial middle type is the meet of both ends [4]. Note that useless casts are not introduced by translation due to the use of the cast metafunction.

The key idea of the transformation is to insert casts in places where consistency is used to justify the typing derivation. For instance, if a term $\widetilde{t}$ of type $\mathsf{Int} \oplus \mathsf{Bool}$ is used where an Int is required, the translation inserts a cast $\langle \mathsf{Int} \Leftarrow \mathsf{Int} \oplus \mathsf{Bool} \rangle t$, where t is the recursive translation of $\widetilde{t}$. This cast plays

$$u ::= \mathsf{true} \mid \mathsf{false} \mid n \mid \lambda x.t$$
$$v ::= u \mid \langle U \overset{U}{\Leftarrow} U \rangle u \qquad \text{(values)}$$
$$f ::= \Box + t \mid v + \Box \mid \Box\ t \mid v\ \Box \mid \langle U \overset{U}{\Leftarrow} U \rangle \Box \mid \mathsf{if}\ \Box\ \mathsf{then}\ t\ \mathsf{else}\ t \qquad \text{(frames)}$$

$\boxed{t \longrightarrow t}$ **Notions of Reduction**

$$\frac{n_3 = rval(v_1)\ [\![+]\!]\ rval(v_2)}{v_1 + v_2 \longrightarrow n_3} \qquad (\lambda x.t)\ v \longrightarrow [v/x]t$$

$$\mathsf{if}\ v\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2 \longrightarrow \begin{cases} t_2 \text{ if } rval(v) = \mathsf{true} \\ t_3 \text{ if } rval(v) = \mathsf{false} \end{cases}$$

$$\langle U_{21} \to U_{22} \overset{U_3}{\Leftarrow} U_{11} \to U_{12} \rangle u\ v \longrightarrow \langle\!\langle U_{22} \overset{icod(U_3)}{\Longleftarrow} U_{12} \rangle\!\rangle (u\ \langle\!\langle U_{11} \overset{idom(U_3)}{\Longleftarrow} U_{21} \rangle\!\rangle v)$$

$$\langle U_3 \overset{U_{32}}{\Leftarrow} U_2 \rangle \langle U_2 \overset{U_{21}}{\Leftarrow} U_1 \rangle v \longrightarrow \begin{cases} \langle\!\langle U_3 \overset{U_{32} \sqcap U_{21}}{\Longleftarrow} U_1 \rangle\!\rangle v \\ \textbf{error} \text{ if } U_{32} \sqcap U_{21} \text{ is undefined} \end{cases}$$

$\boxed{t \longmapsto t}$ **Reduction**

$$\frac{t_1 \longrightarrow t_2}{t_1 \longmapsto t_2} \qquad \frac{t_1 \longmapsto t_2}{f[t_1] \longmapsto f[t_2]} \qquad \frac{}{f[\textbf{error}] \longmapsto \textbf{error}}$$

Fig. 13. $\mathrm{GTFL}^{\oplus}_{\Rightarrow}$: Dynamic Semantics of $\mathrm{GTFL}^{\oplus}_{\Rightarrow}$

the role of the implicit projection from the gradual union type. Dually, when a term of type Int is used where a gradual union is expected, the translation adds a cast that performs the implicit injection to the gradual union, *e.g.* $\langle \mathsf{Int} \oplus \mathsf{Bool} \Leftarrow \mathsf{Int} \rangle 10$. Note that a value with a cast that loses precision is like a tagged value in tagged union type systems; the difference again is that the "tag" is inserted implicitly.

Rule (C::) may insert a cast from the type of the body to the ascribed type. Rule (Capp) may insert two casts. By declaring that the resulting type of the application is $\widetilde{cod}(U_1)$ and that the argument is consistent with $\widetilde{dom}(U_1)$, we are implicitly assuming that U_1 is consistent with some function type, which justifies the cast on t_1. The second cast on t_2 comes from the consistent judgment $U_2 \sim \widetilde{dom}(U_1)$. Rule (C+) is similar.

$$\text{(Cx)}\ \frac{x : U \in \Gamma}{\Gamma \vdash x \Rightarrow x : U} \qquad \text{(Cb)}\ \frac{}{\Gamma \vdash b \Rightarrow b : \mathsf{Bool}} \qquad \text{(Cn)}\ \frac{}{\Gamma \vdash n \Rightarrow n : \mathsf{Int}}$$

$$\text{(C}\lambda\text{)}\ \frac{\Gamma, x : U_1 \vdash \widetilde{t} \Rightarrow t' : U_2}{\Gamma \vdash (\lambda x : U_1.\widetilde{t}) \Rightarrow (\lambda x : U_1.t') : U_1 \to U_2}$$

$$\text{(C::)}\ \frac{\Gamma \vdash \widetilde{t} \Rightarrow t' : U \qquad U \sim U_1}{\Gamma \vdash (\widetilde{t} :: U_1) \Rightarrow \langle U_1 \Leftarrow U \rangle t' : U_1}$$

$$\text{(Capp)}\ \frac{\Gamma \vdash \widetilde{t}_1 \Rightarrow t_1' : U_1 \qquad \Gamma \vdash \widetilde{t}_2 \Rightarrow t_2' : U_2 \qquad U_2 \sim \widetilde{dom}(U_1)}{\begin{array}{l}\Gamma \vdash \widetilde{t}_1\ \widetilde{t}_2 \Rightarrow \\ \quad \langle \widetilde{dom}(U_1) \to \widetilde{cod}(U_1) \Leftarrow U_1 \rangle t_1' \ \ \langle \widetilde{dom}(U_1) \Leftarrow U_2 \rangle t_2' : \widetilde{cod}(U_1)\end{array}}$$

$$\text{(C+)}\ \frac{\Gamma \vdash \widetilde{t}_1 \Rightarrow t_1' : U_1 \qquad \Gamma \vdash \widetilde{t}_2 \Rightarrow t_2' : U_2 \qquad U_1 \sim \mathsf{Int} \qquad U_2 \sim \mathsf{Int}}{\Gamma \vdash \widetilde{t}_1 + \widetilde{t}_2 \Rightarrow \langle \mathsf{Int} \Leftarrow U_1 \rangle t_1' + \langle \mathsf{Int} \Leftarrow U_2 \rangle t_2' : \mathsf{Int}}$$

$$\text{(Cif)}\ \frac{\Gamma \vdash \widetilde{t}_1 \Rightarrow t_1' : U_1 \qquad U_1 \sim \mathsf{Bool} \qquad \Gamma \vdash \widetilde{t}_2 \Rightarrow t_2' : U_2 \qquad \Gamma \vdash \widetilde{t}_3 \Rightarrow t_3' : U_3}{\begin{array}{l}\Gamma \vdash \mathsf{if}\ \widetilde{t}_1\ \mathsf{then}\ \widetilde{t}_2\ \mathsf{else}\ \widetilde{t}_3 \Rightarrow \\ \quad \mathsf{if}\ \langle \mathsf{Bool} \Leftarrow U_1 \rangle t_1'\ \mathsf{then}\ \langle U_2 \sqcap U_3 \Leftarrow U_2 \rangle t_2'\ \mathsf{else}\ \langle U_2 \sqcap U_3 \Leftarrow U_3 \rangle t_3' : U_2 \sqcap U_3\end{array}}$$

Fig. 14. Cast insertion rules

$$
\begin{aligned}
(b_1, b_2) \in \mathcal{U}_k[\![\mathsf{Bool}]\!] &\iff b_1 \in \textsc{Term}_{\mathsf{Bool}} \wedge b_2 : \mathsf{Bool} \wedge b_1 = b_2 \\
(n_1, n_2) \in \mathcal{U}_k[\![\mathsf{Int}]\!] &\iff n_1 \in \textsc{Term}_{\mathsf{Int}} \wedge n_2 : \mathsf{Int} \wedge n_1 = n_2 \\
(\check{u}_1, u_2) \in \mathcal{U}_k[\![U_1 \to U_2]\!] &\iff \check{u}_1 \in \textsc{Term}_{U_1 \to U_2} \wedge u_2 : U_1 \to U_2 \wedge \\
&\quad \forall U' = U_1'' \to U_2'', \varepsilon_1 \vdash U_1 \to U_2 \sim U_1'' \to U_2'', \text{ and} \\
&\quad \varepsilon_2 \vdash U_1' \sim U_1'', \text{ we have: } \forall j \leq k, (\check{v}_1, v_2) \in \mathcal{V}_j[\![U_1']\!], \\
&\quad (\varepsilon_1 \check{u}_1 \mathbin{@}^{U'} \varepsilon_2 \check{v}_1, \langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2 \rangle u_2 \ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1' \rangle v_2) \in \mathcal{T}_j[\![U_2'']\!] \\
(\varepsilon \check{u}_1 :: U, u_2) \in \mathcal{V}_k[\![U]\!] &\iff \varepsilon \check{u}_1 :: U \in \textsc{Term}_U \wedge \varepsilon = U \wedge (\check{u}_1, u_2) \in \mathcal{U}_k[\![U]\!] \\
(\varepsilon \check{u}_1 :: U, \langle U \overset{\varepsilon}{\Leftarrow} U' \rangle u_2) \in \mathcal{V}_k[\![U]\!] &\iff \varepsilon \check{u}_1 :: U \in \textsc{Term}_U \wedge (\check{u}_1, u_2) \in \mathcal{U}_{k-1}[\![U']\!] \\
(\check{u}_1, u_2) \in \mathcal{V}_k[\![U]\!] &\iff (\check{u}_1, u_2) \in \mathcal{U}_k[\![U]\!] \\
(\check{t}_1, t_2) \in \mathcal{T}_k[\![U]\!] &\iff \check{t}_1 \in \textsc{Term}_U \wedge \vdash t_2 : U \wedge \forall j < k \\
&\quad (\check{t}_1 \longrightarrow^j \check{v}_1 \Rightarrow (t_2 \longrightarrow^* v_2 \wedge (\check{v}_1, v_2) \in \mathcal{V}_{k-j}[\![U]\!])) \wedge \\
&\quad (t_2 \longrightarrow^j v_2 \Rightarrow (\check{t}_1 \longrightarrow^* \check{v}_1 \wedge (\check{v}_1, v_2) \in \mathcal{V}_{k-j}[\![U]\!])) \wedge \\
&\quad (\check{t}_1 \longrightarrow^j \mathbf{error} \Rightarrow t_2 \longrightarrow^* \mathbf{error}) \wedge \\
&\quad (\check{t}_2 \longrightarrow^j \mathbf{error} \Rightarrow t_1 \longrightarrow^* \mathbf{error})
\end{aligned}
$$

Fig. 15. Logical relations between intrinsic terms and cast calculus terms.

5.3 Correctness of the Translational Semantics

This section present all the definitions and properties used to prove the correctness of translational semantics.

One of the novelty of this work is to establish that the translational semantics of the gradual language enjoys both type safety and the gradual guarantees, without relying on the usual proof techniques. The typical approach is to prove type safety of a gradual language by first establishing type safety of the target cast calculus and second proving that the cast insertion translation preserves typing. Proving the gradual guarantees is a separate effort [6].

Here, we instead directly establish that the translation semantics is equivalent to the reference semantics derived with AGT. Because the reference semantics describes a type-safe gradual language that satisfies the gradual guarantees, so does the translational semantics. We establish the equivalence between the semantics using step-indexed: logical relations. We use step-indexed logical relations so the relation is well-founded: the definition without indexes may contain some vicious cycles in presence of gradual unions. Equivalence between two terms is acknowledged when either both evaluate to the same value, or both lead to an **error**.

Figure 15 presents the logical relations between simple values, values and computations, which are defined mutually recursively. The logical relations are defined for pairs composed of an intrinsic term $\check{t}$, which denotes the typing derivation for a GTFL$^{\oplus}$ term $\widetilde{t}$, and a GTFL$^{\oplus}_{\Rightarrow}$ term t. For simplicity, we write $t : U$ for $\cdot \vdash_{\widetilde{\imath}} t : U$.

A pair of simple values $(\check{u}_1, u_2)$ are related for k steps at type U, notation $(\check{u}_1, u_2) \in \mathcal{U}_k[\![U]\!]$, if they both have the same type U and, if U is either Bool

or Int, then the values are also equal. If the simple values are functions, then they are related if their application to related arguments , for $j \leq k$ steps, yields *related computations*, as explained below. Note that the relation between simple values need not consider the case of gradual types, as no literal values have gradual types.

A pair of values $(\check{v}_1, v_2)$ are related for k steps at type U, notation $(\check{v}_1, v_2) \in \mathcal{V}_k[\![U]\!]$, if both have the same type and their underlying simple values are related. One important point to notice is that we may only relate an ascribed value $\varepsilon\check{u}_1 :: U$ to a simple value u_2 if we do not learn anything new from the ascription, *i.e.* both the type of $\check{u}_1$ and the evidence ε are U. This corresponds to the case where the reference semantics carries useless evidence—recall that the cast insertion translation does not insert useless casts. Additionally, an ascribed value $\varepsilon\check{u}_1 :: U$ is related to a casted value if the evidence and ascription correspond to the threesome. More precisely, the evidence ε must be exactly the middle type of the threesome, and the source and target types of the threesome must correspond to the type of $\check{u}_1$ and the ascribed type U, respectively. Finally, in order to reason about the underlying simple values, the ascription and cast must be eliminated by combining them with an evidence and a cast respectively. Because of this extra step, the underlying simple values must be related for $k-1$ steps instead.

A pair of terms $(\check{t}_1, t_2)$ are related computations for k steps at type U, notation $(\check{t}_1, t_2) \in \mathcal{T}_k[\![U]\!]$, if both terms have the same type U, then either both terms reduce to related values at type U, or both terms reduce to an error. Formally, for any $j < k$, if the evaluation of the intrinsic term $\check{t}_1$ terminates in a value v_1 at least in j steps, then the compiled term t_2 also reduces to a value v_2, and the resulting values are related values for $k-j$ steps at type U. Analogously, if the evaluation of the compiled term t_2 reduces to a value v_2 at least in j steps, then the intrinsic term $\check{t}_1$ also reduces to a related value v_1. Finally, if either term reduces to an error in at least j steps, then the other also reduces to an error. Note that this last condition is only required because we do not assume type safety of $\mathrm{GTFL}^{\oplus}_{\Rightarrow}$.

Armed with these logical relations we can state the notion of semantic equivalence between a $\mathrm{GTFL}^{\oplus}$ intrinsic term and a $\mathrm{GTFL}^{\oplus}_{\Rightarrow}$ term.

Definition 20 (Semantic equivalence). *Let $\check{t} \in \textsc{Term}_U$, $\Gamma = FV(\check{t})$ and a $GTFL^{\oplus}_{\Rightarrow}$ term t such that $\Gamma \vdash_i t : U$. We say that $\check{t}$ and t are semantically equivalent, notation $\check{t} \approx t : U$, if and only if for any $k \geq 0, (\sigma_1, \sigma_2) \in \mathcal{G}_k[\![\Gamma]\!]$, we have $(\sigma_1(\check{t}), \sigma_2(t)) \in \mathcal{T}_k[\![U]\!]$.*

The definition of semantic equivalence appeals to the notion of related substitutions. Two substitutions σ_1 and σ_2 are related for k steps at type environment Γ, notation $(\sigma_1, \sigma_2) \in \mathcal{G}_k[\![\Gamma]\!]$, if they map each variable in Γ to related values (full definition in 5.3).

Note that we write $\check{t}$ instead of t^U when it is clear from the context that it is an intrinsic term. Also note that $t : U \equiv \cdot \vdash_i t : U$.

Definition 21. *Let σ be a substitution and Γ a type substitution. We say that substitution σ satisfy environment Γ, written $\sigma \models \Gamma$, if and only if $dom(\sigma) = dom(\Gamma)$.*

Definition 22 (Related substitutions). *Let σ_1 be a substitution function from intrinsic variables to intrinsic values, and let σ_2 be a substitution function from variables to values from the intermediate language. Then we define related substitution as follows:*

$$(\sigma_1, \sigma_2) \in \mathcal{G}_k[\![\Gamma]\!] \iff \sigma_i \models \Gamma \wedge \forall x \in \Gamma.(\sigma_1(x^{\Gamma(x)}), \sigma_2(x)) \in \mathcal{V}_k[\![\Gamma(x)]\!]$$

Lemma 7 (Reduction preserves relations). *Consider $\Gamma \vdash \widetilde{t} : U$, $t^U \in \text{TERM}_U$ and $\Gamma \vdash \widetilde{t} \Rightarrow t : U$. Consider $k, j > 0$, if $t^U \longrightarrow^j t'^U$ and $t \longrightarrow^j t'$, then we have $(t^U, t) \in \mathcal{T}_k[\![U]\!]$ if and only if $(t'^U, t') \in \mathcal{T}_{k-j}[\![U]\!]$*

Proof. The $\Rightarrow$ direction relies on the determinism of the reduction relation and the definition of related computations. The $\Leftarrow$ direction follows direct from the definition of $(t^U, t) \in \mathcal{T}_k[\![U]\!]$ and transitivity of $\longrightarrow$.

Lemma 8. *If $(\widetilde{t}_1, t_2) \in \mathcal{T}_k[\![U]\!]$ then if $\varepsilon \vdash U \sim U'$, then $(\varepsilon\widetilde{t}_1 :: U', \langle U' \overset{\varepsilon}{\Leftarrow} U\rangle t_2) \in \mathcal{T}_{k+1}[\![U']\!]$*

Proof. If either term reduce to an **error** then the lemma trivially holds. If either one of the term reduce to a value, then it holds by definition of related values and Lemma 7

Lemma 9. *Consider $k > 0$. If $(\widetilde{t}_1, t_2) \in \mathcal{T}_k[\![U]\!]$ then $(\widetilde{t}_1, t_2) \in \mathcal{T}_{k-1}[\![U]\!]$*

Proof. Trivial by definition of related computations, as $(\widetilde{t}_1, t_2) \in \mathcal{T}_k[\![U]\!]$ is a stronger property than $(\widetilde{t}_1, t_2) \in \mathcal{T}_{k-1}[\![U]\!]$.

Finally, semantic equivalence between the reference and the translational semantics says that given a well-typed term $\widetilde{t}$ from the gradual source language, its corresponding intrinsic term $\check{t}$ is semantically equivalent to the cast calculus term t obtained after the cast insertion translation.

Proposition 33 (Equivalence of reference and translational semantics). *If $\Gamma \vdash \widetilde{t} : U$, represented as the intrinsic term $\check{t} \in \text{TERM}_U$, and $\Gamma \vdash \widetilde{t} \Rightarrow t : U$, then $\check{t} \approx t : U$.*

We open the proposition to prove this instead:

If $\Gamma \vdash \widetilde{t} : U$, $t^U \in \text{TERM}_U$, $\Gamma \vdash \widetilde{t} \Rightarrow t : U$, then $\forall k \geq 0, (\sigma_1, \sigma_2) \in \mathcal{G}_k[\![\Gamma]\!]$, $(\sigma_1(t^U), \sigma_2(t)) \in \mathcal{T}_k[\![\Gamma]\!]$.

Proof. By induction on the type derivation of $\widetilde{t}$.

Case (U b). Then $\widetilde{t} = b$ and therefore:

$$(U\text{b})\frac{}{\Gamma \vdash b : \mathsf{Bool}}$$

where $U = \mathsf{Bool}$. Then the corresponding intrinsic term is:

$$(\text{I}U\text{b})\frac{}{b \in \textsc{Term}_{\mathsf{Bool}}}$$

and the type derivation of the compiled term is:

$$(\text{IT b})\frac{}{\Gamma \vdash_i \ b : \mathsf{Bool}}$$

But $\sigma_1(b) = b, \sigma_2(b) = b$, and $b = b$ and the result holds immediately.

Case (U n). Then $\widetilde{t} = n$ and therefore:

$$(U\text{n})\frac{}{\Gamma \vdash n : \mathsf{Int}}$$

where $U = \mathsf{Int}$. Then the corresponding intrinsic term is:

$$(\text{I}U\text{n})\frac{}{n \in \textsc{Term}_{\mathsf{Int}}}$$

and the type derivation of the compiled term is:

$$(\text{IT n})\frac{}{\Gamma \vdash_i \ n : \mathsf{Int}}$$

But $\sigma_1(n) = n, \sigma_2(n) = n$, and $n = n$ and the result holds immediately.

Case (U x). Then $\widetilde{t} = x$ and therefore:

$$(U\text{x})\frac{x : U \in \Gamma}{\Gamma \vdash x : U}$$

Then the corresponding intrinsic term is:

$$(\text{I}U\text{x})\frac{}{x^U \in \textsc{Term}_U}$$

and the type derivation of the compiled term is:

$$(\text{IT x})\frac{x : U \in \Gamma}{\Gamma \vdash_i \ x : U}$$

As $(\sigma_1, \sigma_2) \in \mathcal{G}_k[\![\Gamma]\!]$ and $x \in dom(\Gamma)$, then $(x^U, x) \in \mathcal{V}_k[\![U]\!]$ and the result holds immediately.

Case (U λ). Then $\widetilde{t} = (\lambda x : U_1.\widetilde{t_2})$ and therefore:

$$(U\lambda)\frac{\Gamma, x : U_1 \vdash \widetilde{t_2} : U_2}{\Gamma \vdash (\lambda x : U_1.\widetilde{t_2}) : U_1 \to U_2}$$

where $U = U_1 \to U_2$. Then the corresponding intrinsic term is:

$$(\text{I}U\lambda)\frac{t_2^{U_2} \in \textsc{Term}_{U_2}}{(\lambda x^{U_1}.t_2^{U_2}) \in \textsc{Term}_{U_1 \to U_2}}$$

and the type derivation of the compiled term is:

$$(\mathrm{IT}\lambda)\frac{\Gamma, x : U_1 \vdash_{\mathsf{i}} \widetilde{t}_2' : U_2}{\Gamma \vdash_{\mathsf{i}} (\lambda x : U_1.\widetilde{t}_2) : U_1 \to U_2}$$

where $\Gamma, x : U_1 \vdash \widetilde{t}_2 \Rightarrow t_2' : U_2$. Consider $j \leq k, U' = U_1'' \to U_2'', \varepsilon_1 \vdash U_1 \to U_2 \sim U_1'' \to U_2'', \varepsilon_2 \vdash U_1' \sim U_1'', \check{v}_1'$, and v_2', such that $(\check{v}_1', v_2') \in \mathcal{V}_j[\![U_1']\!]$. We have to prove that:

$$(\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))@^{U'}\varepsilon_2\check{v}_1', \langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle v_2') \in \mathcal{T}_j[\![U_2'']\!]$$

Then we proceed depending on the structure of $(\check{v}_1', v_2')$, but ultimately we converge to analogous cases where the argument are just related simple values.

1. If $(\check{v}_1', v_2') = (\check{u}_1', u_2') \in \mathcal{V}_k[\![U_1']\!]$, then $(\check{u}_1', u_2') \in \mathcal{U}_k[\![U_1']\!]$. Therefore

$$\begin{aligned}&\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle u_2'\\ \longrightarrow&\langle U_2'' \overset{icod(\varepsilon_1)}{\Longleftarrow} U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1 \overset{idom(\varepsilon_1)}{\Longleftarrow} U_1''\rangle\langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle u_2')\end{aligned}$$

   (a) If $\varepsilon_2 \circ^{=} idom(\varepsilon_1)$ is not defined, then $\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2\check{u}_1' \longrightarrow$ **error**. But by definition of consistent transitivity $\varepsilon_2 \sqcap idom(\varepsilon_1) = \emptyset$ and therefore $\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle u_2' \longrightarrow^2$ **error** and the result holds.

   (b) If $\varepsilon' = \varepsilon_2 \circ^{=} idom(\varepsilon_1)$ is defined, then

$$\begin{aligned}&\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2\check{u}_1'\\ \longrightarrow&icod(\varepsilon_1)([\varepsilon'\check{u}_1' :: U_1/x^{U_1}]\sigma_1(t_2^{U_2})) :: U_2''\\ =&icod(\varepsilon_1)(\sigma_1[x^{U_1} \mapsto \varepsilon'\check{u}_1' :: U_1](t_2^{U_2})) :: U_2''\end{aligned}$$

   and, let us suppose than $\neg(U_1' = U_1'' = \varepsilon')$ (the other case is similar modulo one step of evaluation)

$$\begin{aligned}&\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle u_2'\\ \longrightarrow^2&\langle U_2'' \overset{icod(\varepsilon_1)}{\Longleftarrow} U_2\rangle[\langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2'/x]\sigma_2(t_2)\\ =&\langle U_2'' \overset{icod(\varepsilon_1)}{\Longleftarrow} U_2\rangle\sigma_2[x \mapsto \langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2'](t_2)\end{aligned}$$

   As $(\check{u}_1', u_2') \in \mathcal{U}_j[\![U_1']\!]$ then by definition of related values, $(\varepsilon'\check{u}_1' :: U_1, \langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2') \in \mathcal{V}_{j+1}[\![U]\!]$, then by Lemma 9 $(\varepsilon'\check{u}_1' :: U_1, \langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2') \in \mathcal{V}_j[\![U]\!]$. Therefore by definition of related substitutions, $(\sigma_1[x^{U_1} \mapsto \varepsilon'\check{u}_1' :: U_1], \sigma_2[x \mapsto \langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2']) \in \mathcal{G}_j[\![\Gamma, x : U_1]\!]$. Then by induction hypothesis on pair $(t_2^{T_2}, \Gamma, x : U_1 \vdash \widetilde{t}_2 : U_2)$,

$$((\sigma_1[x^{U_1} \mapsto \varepsilon'\check{u}_1' :: U_1](t_2^{U_2})), \sigma_2[x \mapsto \langle U_1 \overset{\varepsilon'}{\Leftarrow} U_1'\rangle u_2'](t_2)) \in \mathcal{T}_j[\![U_2]\!]$$

   and the result holds by Lemma 9, and backward preservation of the relations (Lemma 7).

2. If $(\check{v}_1', v_2') = (\varepsilon\check{u}_1' :: U_1', u_2') \in \mathcal{V}_j[\![U_1']\!]$, then $\varepsilon = U_1'$ and $(\check{u}_1', u_2') \in \mathcal{U}_j[\![U_1']\!]$. Then $\varepsilon_2' = \varepsilon \circ^{=} \varepsilon_2$ is defined because $U_1' \sim U_1''$, and therefore: $\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))@^{U'}$ $\varepsilon_2\check{v}_1' \longrightarrow \varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2'\check{u}_1'$. Then we proceed analogous to (1) and the result holds.
3. If $(\check{v}_1', v_2') = (\varepsilon\check{u}_1' :: U_1', \langle U_1' \overset{\varepsilon}{\Leftarrow} U'\rangle u_2') \in \mathcal{V}_j[\![U_1']\!]$, then $(\check{u}_1', u_2') \in \mathcal{U}_j[\![U']\!]$, for some U' such that $\varepsilon \vdash U' \sim U_1'$.
   (a) If $\varepsilon \circ^{=} \varepsilon_2$ is not defined then $\varepsilon \sqcap \varepsilon_2 = \emptyset$,
   $\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2\varepsilon\check{u}_1' :: U_1' \longrightarrow$ **error** and
   $\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle\langle U_1' \overset{\varepsilon}{\Leftarrow} U'\rangle u_2' \longrightarrow$ **error**, and the result holds.
   (b) If $\varepsilon_2' = \varepsilon \circ^{=} \varepsilon_2$ is defined then

$$\varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2\varepsilon\check{u}_1' :: U_1' \longrightarrow \varepsilon_1(\lambda x^{U_1}.\sigma_1(t_2^{U_2}))\ @^{U'}\ \varepsilon_2'\check{u}_1'$$

   and

$$\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2}{\Leftarrow} U_1'\rangle\langle U_1' \overset{\varepsilon}{\Leftarrow} U'\rangle u_2' \longrightarrow$$
$$\langle U' \overset{\varepsilon_1}{\Leftarrow} U_1 \to U_2\rangle(\lambda x : U_1.\sigma_2(t_2))\ \langle U_1'' \overset{\varepsilon_2'}{\Leftarrow} U'\rangle u_2'$$

   and we proceed analogous to (1) and the result holds.

Case (U ::). Then $\widetilde{t} = t_1 :: U$ and therefore:

$$(U ::)\frac{\Gamma \vdash t_1 : U' \qquad U' \sim U}{\Gamma \vdash t_1 :: U : U}$$

Then the corresponding non ascribed intrinsic term is:

$$(\mathrm{I}U ::)\frac{t_1^{U'} \in \textsc{Term}_{U'} \qquad \varepsilon \vdash U' \sim U}{\varepsilon t_1^{U'} :: U \in \textsc{Term}_U}$$

Let us assume $U' \neq U$ (the other case is analogous). The type derivation of the compiled term is:

$$(\mathrm{IT}{<>})\frac{\Gamma \vdash_i\ t_1' : U'}{\Gamma \vdash_i\ \langle U \overset{\varepsilon}{\Leftarrow} U'\rangle t_1'}$$

Then we have to prove that

$$(\varepsilon\sigma_1(t_1^{U'}) :: U, \langle U \overset{\varepsilon}{\Leftarrow} U'\rangle\sigma_2(t_1')) \in \mathcal{T}_k[\![U]\!]$$

By induction hypotheses on $\widetilde{t}_1$ $(\sigma_1(t_1^{U'}), \sigma_2(t_1')) \in \mathcal{T}_k[\![U']\!]$. If either term reduces to an error in less than k steps, then the result holds immediately. The interesting case if they reduce to related values in less than k steps. Then suppose $\sigma_1(t^{U_1}) \longrightarrow^j \check{v}_1$, $\sigma_2(t_1') \longrightarrow^* v_1'$, where $j < k$ $(\check{v}_1, v_1') \in \mathcal{V}_{k-j}[\![U_1]\!]$.

Let us assume $\check{v}_1 = \check{u}_1$ then $v_1' = u_1'$ (if they are ascribed values, then the argument is similar modulo one extra step of evaluation, where a runtime error may be produced).

$$\varepsilon\sigma_1(t_1^{U'}) :: U \longrightarrow^j \varepsilon\check{u}_1 :: U$$

and

$$\langle U \overset{\varepsilon}{\Leftarrow} U' \rangle \sigma_2(t_1') \longrightarrow^j \langle U \overset{\varepsilon}{\Leftarrow} U' \rangle u_1'$$

We need to prove that

$$(\varepsilon \check{u}_1 :: U, \langle U \overset{\varepsilon}{\Leftarrow} U' \rangle u_1') \in \mathcal{V}_{k-j}[\![U]\!]$$

but $(\check{v}_1, v_1') \in \mathcal{V}_{k-j}[\![U_1]\!]$ and by Lemma 9, $(\check{v}_1, v_1') \in \mathcal{V}_{k-j-1}[\![U_1]\!]$, and the result holds by backward preservation lemma.

Case (U app). Then $\widetilde{t} = \widetilde{t}_1\ \widetilde{t}_2$ and therefore:

$$(U\text{app})\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \qquad \Gamma \vdash \widetilde{t}_2 : U_2 \qquad U_2 \sim \widetilde{dom}(U_1)}{\Gamma \vdash \widetilde{t}_1\ \widetilde{t}_2 : \widetilde{cod}(U_1)}$$

and $U = \widetilde{cod}(U_1)$. Then the corresponding intrinsic term is:

$$(\text{I}U\text{app})\frac{t^{U_1} \in \text{Term}_{U_1} \qquad \varepsilon_1 \vdash U_1 \sim U_{11} \to U_{12} \qquad t^{U_2} \in \text{Term}_{U_2} \qquad \varepsilon_2 \vdash U_2 \sim U_{11}}{(\varepsilon_1 t^{U_1})\ @^{U_{11} \to U_{12}}\ (\varepsilon_2 t^{U_2}) \in \text{Term}_{U_{12}}}$$

where $U_{11} = \widetilde{dom}(U_1)$ and $U_{12} = \widetilde{cod}(U_1)$.

We proceed assuming that the compilation always inserts casts (the other cases are similar because then the evidences are equal to the types in the judgment. Therefore the next combinations of those evidences are redundant and never fails). Then suppose $U_1 \neq U_{11} \to U_{12}$ and $U_2 \neq U_{11}$. As $\varepsilon_1 = U_1 \sqcap U_{11} \to U_{12}$, and $\varepsilon_2 = U_2 \sqcap U_{11}$, the type derivation of the compiled term is:

$$(\text{IT app})\frac{\Gamma \vdash_{\mathfrak{i}} t_1' : U_1 \qquad \Gamma \vdash_{\mathfrak{i}} t_2' : U_{11}}{\Gamma \vdash_{\mathfrak{i}} \langle U_{11} \to U_{12} \overset{\varepsilon_1}{\Leftarrow} U_1 \rangle t_1'\ \langle U_{11} \overset{\varepsilon_2}{\Leftarrow} U_2 \rangle t_2' : U_{12})}$$

Then we have to prove that

$$((\varepsilon_1 \sigma_1(t^{U_1}))@^{U_{11} \to U_{12}}(\varepsilon_2 \sigma_1(t^{U_2})), \langle U_{11} \to U_{12} \overset{\varepsilon_1}{\Leftarrow} U_1 \rangle \sigma_2(t_1')\ \langle U_{11} \overset{\varepsilon_2}{\Leftarrow} U_2 \rangle \sigma_2(t_2')) \in \mathcal{T}_k[\![U_{12}]\!]$$

By induction hypotheses on $\widetilde{t}_1$ and $\widetilde{t}_2$ $(\sigma_1(t^{U_1}), \sigma_2(t_1')) \in \mathcal{T}_k[\![U_1]\!]$ and $(\sigma_1(t^{U_2}), \sigma_2(t_2')) \in \mathcal{T}_k[\![U_2]\!]$. If either term reduces to an error in less than k steps, then the result holds immediately. The interesting case if they reduce to related values in less than k steps. Then suppose $\sigma_1(t^{U_1}) \longrightarrow^j \check{v}_1$, $\sigma_1(t^{U_2}) \longrightarrow^j \check{v}_2$, $\sigma_2(t_1') \longrightarrow^* v_1'$, $\sigma_2(t_2') \longrightarrow^* v_2'$, where $j < k$ $(\check{v}_1, v_1') \in \mathcal{V}_{k-j}[\![U_1]\!]$ and $(\check{v}_2, v_2') \in \mathcal{V}_{k-j}[\![U_2]\!]$. If $\check{v}_1 = \check{u}_1$ then $v_1' = u_1'$, by canonical forms the simple values must be lambdas and the proof follows from Case($U\lambda$). If $\check{v}_1 = \varepsilon_1' \check{u}_1 :: U_1$ and $\check{u}_1 \in \text{Term}_{U_1'}$, then suppose $v_1' = \langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1' \rangle u_1'$ (the other case is similar but the evidence combination never fails). Also $(\check{u}_1, u_1') \in \mathcal{U}_{k-j-1}[\![U_1']\!]$. Suppose $\varepsilon_1' \circ^{=} \varepsilon_1$ is not defined (which is equivalent to $\varepsilon_1' \sqcap \varepsilon_1 = \emptyset$), therefore

$$(\varepsilon_1 \varepsilon_1' \check{u}_1 :: U_1)\ @^{U_{11} \to U_{12}}\ (\varepsilon_2 \check{v}_2) \longrightarrow \textbf{error} \iff$$

$$\langle U_{11} \to U_{12} \overset{\varepsilon_1}{\Leftarrow} U_1 \rangle \langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1' \rangle u_1'\ \langle U_{11} \overset{\varepsilon_2}{\Leftarrow} U_2 \rangle v_2' \longrightarrow \textbf{error}$$

and the result follows. Suppose $\varepsilon_1'' = \varepsilon_1' \circ^{=} \varepsilon_1$ is defined. Then

$$\begin{array}{rl} (\varepsilon_1\varepsilon_1'\breve{u}_1 :: U_1)\ @^{U_{11}\to U_{12}}\ (\varepsilon_2\breve{v}_2) \longrightarrow & \\ (\varepsilon_1''\breve{u}_1)\ @^{U_{11}\to U_{12}}\ (\varepsilon_2\breve{v}_2) & \Rightarrow \\ \langle U_{11}\to U_{12} \overset{\varepsilon_1}{\Leftarrow} U_1\rangle\langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1'\rangle u_1'\ \langle U_{11} \overset{\varepsilon_2}{\Leftarrow} U_2\rangle v_2' \longrightarrow & \\ \langle U_{11}\to U_{12} \overset{\varepsilon_1''}{\Leftarrow} U_1'\rangle u_1'\ \langle U_{11} \overset{\varepsilon_2}{\Leftarrow} U_2\rangle v_2' & \end{array}$$

which is exactly the definition of related functions and then the result holds by backward preservation of the relations (Lemma 7) and Lemma 9.

Case (U if). Then $\widetilde{t} = \widetilde{t}_1\ \widetilde{t}_2$ and therefore:

$$(U\text{if})\frac{\Gamma \vdash \widetilde{t}_1 : U_1 \quad U_1 \sim \mathsf{Bool} \quad \Gamma \vdash \widetilde{t}_2 : U_2 \quad \Gamma \vdash \widetilde{t}_3 : U_3}{\Gamma \vdash \mathsf{if}\ \widetilde{t}_1\ \mathsf{then}\ \widetilde{t}_2\ \mathsf{else}\ \widetilde{t}_3 : U_2 \sqcap U_3}$$

and $U = U_2 \sqcap U_3$. Then the corresponding intrinsic term is:

$$(\text{IU}\text{if})\frac{\begin{array}{ccccc} t^{U_1} \in \textsc{Term}_{U_1} & & \varepsilon_1 \vdash U_1 \sim \mathsf{Bool} & & U = (U_2 \sqcap U_3) \\ t^{U_2} \in \textsc{Term}_{U_2} & \varepsilon_2 \vdash U_2 \sim U & t^{U_3} \in \textsc{Term}_{U_3} & \varepsilon_3 \vdash U_3 \sim U & \end{array}}{\mathsf{if}\ \varepsilon_1 t^{U_1}\ \mathsf{then}\ \varepsilon_2 t^{U_2}\ \mathsf{else}\ \varepsilon_3 t^{U_3} \in \textsc{Term}_U}$$

We proceed assuming that the compilation always inserts casts (the other cases are similar because then the evidences are equal to the types in the judgment. Therefore the next combinations of those evidences are redundant and never fails). Then suppose $U_1 \neq U_{11} \to U_{12}$ and $U_2 \neq U_{11}$. As $\varepsilon_1 = U_1 \sqcap U_{11} \to U_{12}$, and $\varepsilon_2 = U_2 \sqcap U_{11}$, the type derivation of the compiled term is:

$$(\text{C}\text{if})\frac{\Gamma \vdash_{\mathfrak{i}} \widetilde{t}_1' : U_1 \quad U_1 \sim \mathsf{Bool} \quad \Gamma \vdash_{\mathfrak{i}} \widetilde{t}_2' : U_2 \quad \Gamma \vdash_{\mathfrak{i}} \widetilde{t}_3' : U_3}{\mathsf{if}\ \langle \mathsf{Bool} \overset{\varepsilon_1}{\Leftarrow} U_1\rangle t_1'\ \mathsf{then}\ \langle U_2 \sqcap U_3 \overset{\varepsilon_2}{\Leftarrow} U_2\rangle t_2'\ \mathsf{else}\ \langle U_2 \sqcap U_3 \overset{\varepsilon_3}{\Leftarrow} U_3\rangle t_3' : U_2 \sqcap U_3}$$

where $\Gamma \vdash \widetilde{t}_1 \Rightarrow \widetilde{t}_1' : U_1, \Gamma \vdash \widetilde{t}_2 \Rightarrow \widetilde{t}_2' : U_2$, and $\Gamma \vdash \widetilde{t}_3 \Rightarrow \widetilde{t}_3' : U_3$.

But by definition of substitution, $\sigma_1(t^U) = \mathsf{if}\ \varepsilon_1\sigma_1(t^{U_1})\ \mathsf{then}\ \ \mathsf{else}\ \varepsilon_2\sigma_1(t^{U_2})\varepsilon_3\sigma_1(t^{U_3})$ and $\sigma_2(t') = \mathsf{if}\ \langle \mathsf{Bool} \overset{\varepsilon_1}{\Leftarrow} U_1\rangle\sigma_2(t_1')\ \mathsf{then}\ \langle U_2 \sqcap U_3 \overset{\varepsilon_2}{\Leftarrow} U_2\rangle\sigma_2(t_2')\ \mathsf{else}\ \langle U_2 \sqcap U_3 \overset{\varepsilon_3}{\Leftarrow} U_3\rangle\sigma_2(t_3')$.

By induction hypotheses on $\widetilde{t}_1$, $\widetilde{t}_2$ and $\widetilde{t}_3$, $(\sigma_1(t^{U_1}), \sigma_2(t_1')) \in \mathcal{T}_k[\![U_1]\!]$ and $(\sigma_1(t^{U_2}), \sigma_2(t_2')) \in \mathcal{T}_k[\![U_2]\!]$ and $(\sigma_1(t^{U_3}), \sigma_2(t_3')) \in \mathcal{T}_k[\![U_3]\!]$ If either $\sigma_1(t^{U_1})$ or $\sigma_2(t_1')$ term reduces to an error then the result holds immediately. The interesting case is when they reduce to related values. Then suppose $\sigma_1(t^{U_1}) \longrightarrow^j \breve{v}_1$, $\sigma_2(t_1') \longrightarrow^j v_1'$, where $(\breve{v}_1, v_1') \in \mathcal{V}_{k-j}[\![U_1]\!]$ and $(\breve{v}_2, v_2') \in \mathcal{V}_{k-j}[\![U_2]\!]$.

1. If $\breve{v}_1 = \breve{u}_1$ then $v_1' = u_1'$, by canonical forms u_i must be booleans b^{Bool} and b. Suppose that $b = \mathsf{true}$ (the other case is analogous). Then $t^U \longrightarrow^j \varepsilon_2\sigma_1(t^{U_2}) :: U_2 \sqcap U_3$ and $t' \longrightarrow^j \langle U_2 \sqcap U_3 \overset{\varepsilon_2}{\Leftarrow} U_2\rangle\sigma_2(t_2')$. Then as $(\sigma_1(t^{U_2}), \sigma_2(t_2')) \in \mathcal{T}_k[\![U_2]\!]$ and Lemma 8, Lemma 9 and backward preservation of the relation the result holds.

2. If $\check{v}_1 = \varepsilon_1' \check{u}_1 :: U_1$ and $\check{u}_1 \in \text{TERM}_{U_1'}$, then suppose $v_1' = \langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1' \rangle u_1'$ (the other case is similar but the evidence combination never fails). Also $(\check{u}_1, u_1') \in \mathcal{U}_{k-j}[\![U_1']\!]$. Suppose $\varepsilon_1' \circ^{=} \varepsilon_1$ is not defined (which is equivalent to $\varepsilon_1' \sqcap \varepsilon_1 = \emptyset$), therefore

$$\mathsf{if}\ \varepsilon_1 \varepsilon_1' \check{u}_1 :: U_1\ \mathsf{then}\ \varepsilon_2 \sigma_1(t^{U_2})\ \mathsf{else}\ \varepsilon_3 \sigma_1(t^{U_3}) \longrightarrow \mathbf{error} \Longleftrightarrow$$

$$\mathsf{if}\ \langle \mathsf{Bool} \overset{\xi_1}{\Leftarrow} U_1 \rangle \langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1' \rangle u_1'\ \mathsf{then}\ \ldots\ \mathsf{else}\ \ldots \longrightarrow \mathbf{error}$$

and the result follows. Suppose $\varepsilon_1'' = \varepsilon_1' \circ^{=} \varepsilon_1$ is defined. Then

$$\begin{array}{l} \mathsf{if}\ \varepsilon_1 \varepsilon_1' \check{u}_1 :: U_1\ \mathsf{then}\ \varepsilon_2 \sigma_1(t^{U_2})\ \mathsf{else}\ \varepsilon_3 \sigma_1(t^{U_3}) \longrightarrow \\ \qquad \mathsf{if}\ \varepsilon_1'' \check{u}_1\ \mathsf{then}\ \varepsilon_2 \sigma_1(t^{U_2})\ \mathsf{else}\ \varepsilon_3 \sigma_1(t^{U_3}) \qquad\qquad \Rightarrow \\ \mathsf{if}\ \langle \mathsf{Bool} \overset{\xi_1}{\Leftarrow} U_1 \rangle \langle U_1 \overset{\varepsilon_1'}{\Leftarrow} U_1' \rangle u_1'\ \mathsf{then}\ \ldots\ \mathsf{else}\ \ldots \longrightarrow \\ \qquad \mathsf{if}\ \langle \mathsf{Bool} \overset{\varepsilon_1''}{\Leftarrow} U_1' \rangle u_1'\ \mathsf{then}\ \ldots\ \mathsf{else}\ \ldots \end{array}$$

Then as $(\check{u}_1, u_1') \in \mathcal{U}_{k-j}[\![U_1']\!]$ we proceed analogous to (1) and the result holds.

Case (U +). Similar to the (Uapp) and (Uif) case.

References

[1] A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5(2):56–68, 06 1940.

[2] R. Garcia, A. M. Clark, and É. Tanter. Abstracting gradual typing. In *43rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2016)*, St Petersburg, FL, USA, Jan. 2016. ACM Press.

[3] W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pages 479–490. Academic Press, New York, 1980. Reprint of 1969 article.

[4] J. Siek and P. Wadler. Threesomes, with and without blame. In *37th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2010)*, pages 365–376. ACM Press, Jan. 2010.

[5] J. G. Siek and W. Taha. Gradual typing for functional languages. In *Scheme and Functional Programming Workshop*, pages 81–92, Sept. 2006.

[6] J. G. Siek, M. M. Vitousek, M. Cimini, and J. T. Boyland. Refined criteria for gradual typing. In *1st Summit on Advances in Programming Languages (SNAPL 2015)*, pages 274–293, 2015.