Full Proofs for "Logarithm and Program Testing"

KUEN-BANG HOU (FAVONIA), University of Minnesota, USA
ZHUYANG WANG, University of Minnesota, USA

This is part of the supplementary material for the paper "Logarithm and Program Testing". It contains the full proofs of a variant of Lemma 3.2 and Theorem 4.1 from the main article.

1 A VARIANT OF LEMMA 3.2 FROM THE MAIN ARTICLE

This variant shows that the lack of strict positivity in the codomain type can also necessitate the testing of the empty type.

Lemma 1.1 (no single type, an alternative to Lemma 3.2 from the main article). *For any type $\cdot \vdash \tau$, there exist two functions f and g*

$$\cdot;\cdot \vdash_{\mathtt{poly}} f : \forall a.(a + \mathbb{1}) \to (a + a) + ((a \to \mathbb{0}) \to (\mathbb{1} + \mathbb{1}))$$
$$\cdot;\cdot \vdash_{\mathtt{poly}} g : \forall a.(a + \mathbb{1}) \to (a + a) + ((a \to \mathbb{0}) \to (\mathbb{1} + \mathbb{1}))$$

such that $f[\tau] \cong g[\tau]$ but $f \ncong g$. That is, τ cannot distinguish them.

Proof. Consider these three contextually distinct functions:

$$f \coloneqq \Lambda a.\lambda(x{:}a + \mathbb{1}).\mathtt{case}(x; y.\mathtt{inl}(\mathtt{inl}(y)); \_.\lambda(\_{:}a \to \mathbb{0}).\mathtt{inl}(\star))$$
$$g \coloneqq \Lambda a.\lambda(x{:}a + \mathbb{1}).\mathtt{case}(x; y.\mathtt{inl}(\mathtt{inl}(y)); \_.\lambda(\_{:}a \to \mathbb{0}).\mathtt{inr}(\star))$$
$$h \coloneqq \Lambda a.\lambda(x{:}a + \mathbb{1}).\mathtt{case}(x; y.\mathtt{inl}(\mathtt{inr}(y)); \_.\lambda(\_{:}a \to \mathbb{0}).\mathtt{inr}(\star))$$

On the one hand, if τ is non-empty, $f[\tau]$ and $g[\tau]$ are contextually equivalent, because there is no term of type $(\tau \to \mathbb{0})$. On the other hand, if τ is empty, $g[\tau]$ and $h[\tau]$ are contextually equivalent, because they always output $\lambda(\_{:}\tau \to \mathbb{0}).\mathtt{inr}(\star)$. In either case, there is a pair of indistinguishable functions for every τ. □

2 LOGICAL RELATIONS, FIXED POINTS, AND PREFIXED POINTS

Definition 2.1 (admissible relations). Let $\mathcal{R}$ be a relation between closed terms of closed types τ_1 and τ_2. We write $\mathcal{R} : \tau_1 \leftrightarrow \tau_2$ if it respects observational equivalence on both sides.

Definition 2.2 (functors). Let $F(a)$ be a type expression parametrized by a. We say F is a *positive functor* if a only appears in positive positions, a *negative functor* if a only appears in negative positions, and a *strictly positive functor* if a only appears in strictly positive positions.

Definition 2.3 (logical relation and prefixed points). We simultaneously define the prefixed points and logical relations by induction on the structure of the functor F and the type τ:

Authors' addresses: Kuen-Bang Hou (Favonia), Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota, 55455, USA, kbh@umn.edu; Zhuyang Wang, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota, 55455, USA, wang9163@umn.edu.

2475-1421/2022/1-ART64
https://doi.org/10.1145/3498726

Prefixed Points: For any kinding context Δ, any strictly positve functor F such that $\Delta, b \vdash F(b)$, two substitutions $\cdot \vdash \delta : \Delta$ and $\cdot \vdash \delta' : \Delta$, and a family of relations η indexed by Δ such that $\eta(a) : \delta(a) \leftrightarrow \delta'(a)$, a relation $\mathcal{R} : \delta(\mu b.F(b)) \leftrightarrow \delta'(\mu b.F(b))$ is a *prefixed point* of the functor F iff for any elements e_1 and e_2 such that $\cdot \vdash e_1 : \delta(F(\mu b.F(b)))$ and $\cdot \vdash e_2 : \delta'(F(\mu b.F(b)))$,

$$e_1 \sim_{F(b)} e_2 \ [\eta; b \mapsto \mathcal{R}] \implies \mathcal{R}(\mathtt{roll}_{b.\delta(F(b))}(e_1), \mathtt{roll}_{b.\delta'(F(b))}(e_2)).$$

Logical Relations For any kinding context Δ, any τ such that $\Delta \vdash \tau$, two substitutions $\cdot \vdash \delta : \Delta$ and $\cdot \vdash \delta' : \Delta$, and a family of relations η indexed by Δ such that $\eta(a) : \delta(a) \leftrightarrow \delta'(a)$, the logical relation $- \sim_\tau - \ [\eta]$ between elements e_1 and e_2 such that $\cdot \vdash e_1 : \delta(\tau)$ and $\cdot \vdash e_2 : \delta'(\tau)$ is defined as follows:

- $e_1 \sim_a e_2 \ [\eta]$ iff $\eta(a)(e_1, e_2)$.
- $e_1 \sim_{\mathbb{0}} e_2 \ [\eta]$ never holds.
- $e_1 \sim_{\tau_1+\tau_2} e_2 \ [\eta]$ iff
  - e_1 evalautes to $\mathtt{inl}(e_1')$ and e_2 evalautes to $\mathtt{inl}(e_2')$ and $e_1' \sim_{\tau_1} e_2' \ [\eta]$; or
  - e_1 evalautes to $\mathtt{inr}(e_1')$ and e_2 evalautes to $\mathtt{inr}(e_2')$ and $e_1' \sim_{\tau_2} e_2' \ [\eta]$.
- $e_1 \sim_{\mathbb{1}} e_2 \ [\eta]$ always holds.
- $e_1 \sim_{\tau_1 \times \tau_2} e_2 \ [\eta]$ iff $\mathtt{fst}(e_1) \sim_{\tau_1} \mathtt{fst}(e_2) \ [\eta]$ and $\mathtt{snd}(e_1) \sim_{\tau_2} \mathtt{snd}(e_2) \ [\eta]$.
- $e_1 \sim_{\tau_1 \to \tau_2} e_2 \ [\eta]$ iff for all e_1', e_2' such that $e_1' \sim_{\tau_1} e_2' \ [\eta]$, $e_1(e_1') \sim_{\tau_2} e_2(e_2') \ [\eta]$.
- $e_1 \sim_{\mu b.F(b)} e_2 \ [\eta]$ iff $\mathcal{R}(e_1, e_2)$ holds for any prefixed point $\mathcal{R}$ of the functor F.

Now that we have the logical relation defined, we can define the *fixed points* for a strictly positive functor as well:

Definition 2.4 (fixed points). For any kinding context Δ, any strictly positive functor F such that $\Delta, b \vdash F(b)$, two substitutions $\cdot \vdash \delta : \Delta$ and $\cdot \vdash \delta' : \Delta$, and a family of relations η indexed by Δ such that $\eta(a) : \delta(a) \leftrightarrow \delta'(a)$, a relation $\mathcal{R} : \delta(\mu b.F(b)) \leftrightarrow \delta'(\mu b.F(b))$ is a *fixed point* of the functor F iff for any elements e_1 and e_2 such that $\cdot \vdash e_1 : \delta(F(\mu b.F(b)))$ and $\cdot \vdash e_2 : \delta'(F(\mu b.F(b)))$,

$$e_1 \sim_{F(b)} e_2 \ [\eta; b \mapsto \mathcal{R}] \iff \mathcal{R}(\mathtt{roll}_{b.\delta(F(b))}(e_1), \mathtt{roll}_{b.\delta'(F(b))}(e_2)).$$

By the Knaster–Tarski theorem [Tarski 1955] and the positivity of a in the functor F, the least prefixed point and the least fixed point always exist and coincide. So the logical relation of type $\mu b.F(b)$ can also be defined as follows:

- $e_1 \sim_{\mu b.F(b)} e_2 \ [\eta]$ iff $\mathcal{R}(e_1, e_2)$ holds for any fixed point $\mathcal{R}$ of the functor F.

Lemma 2.5. *Logical relation is admissible.*

Lemma 2.6 (Compositionality). *For any types $\Delta, a \vdash \tau$ and $\Delta \vdash \tau_1$, two type substitutions $\cdot \vdash \delta : \Delta$ and $\cdot \vdash \delta' : \Delta$, a family of relations $\eta : \delta \leftrightarrow \delta'$,*

$$e \sim_{\tau[\tau_1/a]} e' \ [\eta] \text{ iff } e \sim_\tau e' \ [\eta, a \mapsto - \sim_{\tau_1} - \ [\eta]].$$

Lemma 2.7 (Parametricity). *If $\cdot; \cdot \vdash e : \tau$ then $e \sim_\tau e \ []$.*

Lemma 2.8. *Logical equivalence and observational equivalence coincide.*

Lemma 2.9. *If we have a type $a \vdash \tau$ and two elements $\cdot \vdash_{\mathtt{poly}} e : \forall a.\tau$ and $\cdot \vdash_{\mathtt{poly}} e' : \forall a.\tau$, then $e \cong e'$ iff $e[\tau_1] \cong e'[\tau_1]$ for any closed type τ_1.*

3 MAIN THEOREM STATEMENT

Our goal is to prove the following main theorem.

Theorem 3.1 (correctness with indexes, Theorem 4.1 from the main article). *Suppose we have*

- *An ambient kinding context* Δ*; and*
- *A type substitution* δ *such that* $\cdot \vdash \delta : \Delta$*, which applies to all open types in the theorem; and*
- *Two type expressions* $\alpha(a)$ *and* $H(a)$ *such that*

$$\Delta, a \vdash \alpha(a) \text{ and } a \in^{++} \log_a(\alpha(a))$$
$$\Delta, a \vdash H(a) \text{ and } a \in^{+} H(a)$$

- *Two functions* f *and* g *such that*

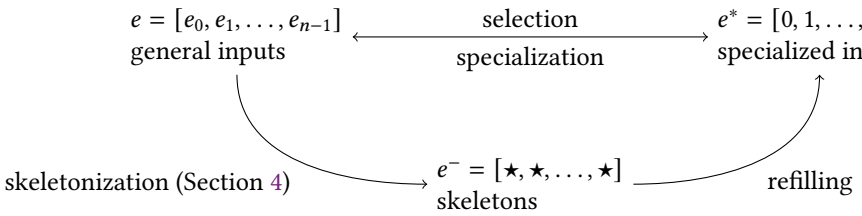
$$\cdot;\cdot \vdash_{\mathtt{poly}} f : \delta(\forall a.\alpha(a) \to H(a))$$
$$\cdot;\cdot \vdash_{\mathtt{poly}} g : \delta(\forall a.\alpha(a) \to H(a))$$

Then, $f \cong g$ *if and only if the following two conditions hold:*

(1) $f[\mathbb{0}] \cong g[\mathbb{0}]$
(2) $f[\delta(a^*)](\delta\{\alpha(a) \uparrow_a\}(e^-)) \cong g[\delta(a^*)](\delta\{\alpha(a) \uparrow_a\}(e^-))$
for every e^- *such that* $\cdot;\cdot \vdash e^- : \delta(\alpha^-(a^*))$

The "only if" direction is straightforward, and the interesting direction is from the two conditions to the contextual equivalence. It means the equivalence of any general argument is witnessed by at least one skeleton e^-. The proof involves two steps:

(1) **Focus on one function instead of two functions.**
We can compare two functions with a reduced domain because a polymorphic function is completely determined by its selected instances. If both functions are determined by their selected instances, and their selected instances agree, then they themselves agree as well. Thus, we can focus on why a polymorphic function is determined by its own instances.
(2) **Show that composition of skeletonization, refilling, and selection is identity.**
The insight is that every general input of type $\alpha(a)$ is related to a specialized input $\alpha(a^*)$ where a-elements and indexes are related by some left-unique relation $\mathcal{R}$. Given a general input e, the related specialized input may be computed by getting its skeleton e^- and then refilling it with indexes to obtain e^*. The core lemma is to establish an admissible relation between e and e^*. For example, consider the general input $[e_0, e_1, \ldots, e_{n-1}]$ of type $\mathtt{list}(a)$. Its skeleton e^- would be $[\star, \star, \ldots, \star]$ and the corresponding specialized input with the indexes would be $[0, 1, \ldots, n-1]$. The relation $\mathcal{R}$ would relate e with i if and only if $e_i \cong e$. The entire process may be summarized by the following diagram:



The following lemma (Lemma 3.2) states that the general behavior of a polymorphic function is indeed determined by some of its instances. For technical reasons, it only covers the cases where the a is instantiated with non-empty types, explaining why we had to consider the $\mathbb{0}$ thus the case-splitting in the main theorem. Note that the lemma uses the auxiliary functions defined in Sections 4 (skeletonization) and 5 (selection) to precisely write down e^- (the skeleton), e^* (the specialized input), and how the specialized input e^* is related to the general input e.

Lemma 3.2. *Suppose we have the following data:*

- *A kinding context* Δ.
- *A type substitution* δ *such that* $\cdot \vdash \delta : \Delta$.
- *The distinguished type variable* a.
- *A type expression* $\alpha(a)$ *such that* $\Delta, a \vdash \alpha(a)$ *and* $a \in^{++} \log_a(\alpha(a))$.
- *A type expression* $H(a)$ *such that* $\Delta, a \vdash H(a)$ *and* $a \in^{+} H(a)$.
- *A non-empty, closed type* τ *that is representing the general type to instantiate the type* a.
- *An element* $\overline{e}$ *witnessing the non-emptiness of* τ. *That is,* $\cdot;\cdot \vdash \overline{e} : \tau$.
- *An element* e *of type* $\cdot;\cdot \vdash e : \delta(\alpha(\tau))$ *as the general input.*

Let type a^* *be* $\mu a.\log_a(\alpha(a))$ *and the selection function* $\cdot;\cdot \vdash s : \delta(a^*) \to \tau$ *be*

$$s \coloneqq \lambda(x{:}\delta(a^*)).\mathtt{fold}^{\tau}_{a.\delta(\log_a(\alpha(a)))}(x; y.\delta\{\alpha(a)\ @_a\ [a \mapsto \tau]; \emptyset\}(\overline{e}; e; y))$$

Define e^- *(the skeleton) and* e^* *(the skeleton filled with indexes) as*

$$e^- \coloneqq \delta\{\alpha(a) \downarrow^{\emptyset}_a a^*; \tau; \emptyset\}(s; e)$$
$$e^* \coloneqq \delta\{\alpha(a) \uparrow^{\emptyset}_a [a \mapsto a^*]; \emptyset\}(\mathtt{roll}_{a.\delta(\log_a(\alpha(a)))}; e^-)$$

(These elements would satisfy the typing judgments $\cdot;\cdot \vdash e^- : \delta(\alpha^-(a^*))$ *and* $\cdot;\cdot \vdash e^* : \delta(\alpha(a^*))$.*)*
Given all these data, for any polymorphic function f *such that*

$$\cdot;\cdot \vdash_{\mathtt{poly}} f : \delta(\forall a.\alpha(a) \to H(a))$$

we can relate the behavior of f *on general inputs to that on specialized ones:*

$$f[\delta(a^*)](e^*) \sim_{\delta(H(a))} f[\tau](e)\ [a \mapsto s]$$

We demonstrate that the main theorem is implied by the above, justifying the first step of the development.

Proof of Theorem 3.1 from Lemma 3.2. The "only if" direction trivially follows the definition of contextual equivalence. For the more interesting "if" direction, by Lemma 2.9, it is sufficient to prove that for any closed type τ, $f[\tau] \cong g[\tau]$.

By Lemma 6.1, either τ is empty, or τ is not empty and we have a closed term $\overline{e}$ of type τ. If τ is empty, the equivalence is witnessed by the first condition $f[\mathbb{0}] \cong g[\mathbb{0}]$. If τ is non-empty, we only need to prove that for any element e such that $\cdot;\cdot \vdash e : \delta(\alpha(\tau))$,

$$f[\tau](e) \cong g[\tau](e)$$

By the skeleton e^- and the selection function s used in Lemma 3.2, we have

$$f[\delta(a^*)](\delta\{\alpha(a) \uparrow_a\}(e^-)) \sim_{\delta(H(a))} f[\tau](e)\ [a \mapsto s]$$
$$g[\delta(a^*)](\delta\{\alpha(a) \uparrow_a\}(e^-)) \sim_{\delta(H(a))} g[\tau](e)\ [a \mapsto s].$$

The second assumption states that the elements on the left-hand side are contextually equivalent. We wish to prove that those on the right-hand side are equivalent as well. Because $a \in^{+} \delta(H(a))$ and s is a function, by Lemma 8.1 we know that the relation

$$- \sim_{\delta(H(a))} - [a \mapsto s]$$

is also a function. Therefore, the equivalence on the left-hand side implies that on the right-hand side. That is,

$$f[\tau](e) \cong g[\tau](e)$$

□

$$\boxed{\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi\} : (a^* \to \tau) \to \xi(\alpha(\tau)) \to \xi(\alpha^-(a^*))}$$

Auxiliary skeletonization function.

$$\begin{aligned}
\{a \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq \star \\
\{b \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq e \qquad (a \neq b) \\
\{\mathbb{0} \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq e \\
\{\alpha_1(a) + \alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq \mathtt{case}(e; x.\mathtt{inl}(\{\alpha_1(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; x)); x.\mathtt{inr}(\{\alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; x))) \\
\{\mathbb{1} \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq e \\
\{\alpha_1(a) \times \alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq \langle \{\alpha_1(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; \mathtt{fst}(e)); \{\alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; \mathtt{snd}(e)) \rangle \\
\{\alpha_1(a) \to \alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq \lambda(x{:}\xi(\alpha_1(a^*))).\{\alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e(\langle a^+.\xi(\alpha_1(a)) \rangle(s; x))) \\
\{\mu b.\alpha_1(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e) &\coloneqq \mathtt{fold}^{\mu b.\xi(\alpha_1^-(a^*))}_{b.\xi(\alpha_1(\tau))}(e; x.\mathtt{roll}_{b.\xi(\alpha_1^-(a^*))}(\{\alpha_1(a) \downarrow_a^{\hat\Xi} a^*; \tau; \hat\xi\}(s; x))) \\
&\text{where } \hat\Xi \coloneqq \Xi, b \\
&\qquad \hat\xi \coloneqq \xi, b \mapsto \mu b.\xi(\alpha_1{}^-(a^*))
\end{aligned}$$

Fig. 1. The skeletonization function

The rest of the paper is organized as follows: Sections 4 and 5 define the skeletonization and selection functions used in Lemma 3.2. Section 6 elaborates on the procedure to construct an element of a non-empty type that is used in the proof of the main theorem using Lemma 3.2. Section 7 proves some basic lemmas about functoriality that we will take for granted in other sections. Section 8 shows a sufficient condition for a logical relation to be a function. Finally, Section 9 is devoted to the proof of the Lemma 3.2.

4 SKELETONIZATION

Skeletonization is the transformation from the general input e to the skeleton e^-. The function is formally written as $\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi\}$ where the symbols have the same meanings as in Lemma 3.2; see Figure 1 for its definition. For brevity, we write

$$\delta\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi\}(s; e)$$

for

$$\delta(\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi\})(s; e)$$

5 SELECTION FUNCTIONS

The selection function maps indexes in the specialized e^* back to the a-elements in the general input e. It is the key to relate the general input e and the specialized input e^* in Lemma 3.2. The function is formally written as $\{\tau \mathbin{@}_a^\Psi \sigma; \rho\}$ where the symbols have the same meanings as in Lemma 3.2; see Figure 2 for its definition. For brevity, we write

$$\delta\{\tau \mathbin{@}_a^\Psi \sigma; \rho\}(\overline{e}; e; t)$$

for

$$\delta(\{\tau \mathbin{@}_a^\Psi \sigma; \rho\})(\overline{e}; e; t)$$

$$\boxed{\ddot{\sigma}_a^{\psi;\rho}}$$

Substitutions parametrized by a to represent the input to the `select` function.

$$\ddot{\sigma}_a^{\psi;\rho}(b) \coloneqq \sigma(b) \qquad (b \notin \psi)$$
$$\ddot{\sigma}_a^{\psi;\rho}(b) \coloneqq \sigma(\rho(\psi(b)) \to a) \qquad (b \in \psi)$$

$$\boxed{\{\tau \mathbin{@}_a^{\psi} \sigma; \rho\} : \sigma(a) \to \ddot{\sigma}_a^{\psi;\rho}(\tau) \to \sigma(\rho(\log_a^{\psi}(\tau))) \to \sigma(a)}$$

Auxiliary selector.

$$\begin{aligned}
\{a \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq e \\
\{b \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq e(t) \qquad (a \neq b \text{ and } b \in \psi) \\
\{b \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \overline{e} \qquad (a \neq b \text{ and } b \notin \psi) \\
\{\mathbb{0} \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \overline{e} \\
\{\tau_1 + \tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \mathtt{case}(t; t.\mathtt{case}(e; x.\{\tau_1 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; x; t); _.\overline{e}); \\
&\qquad\qquad t.\mathtt{case}(e; _.\overline{e}; x.\{\tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; x; t))) \\
\{\mathbb{1} \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \overline{e} \\
\{\tau_1 \times \tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \mathtt{case}(t; t.\{\tau_1 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; \mathtt{fst}(e); t); \\
&\qquad\qquad t.\{\tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; \mathtt{snd}(e); t)) \\
\{\tau_1 \to \tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq \{\tau_2 \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e(\mathtt{fst}(t)); \mathtt{snd}(t)) \\
\{\mu b.\tau \mathbin{@}_a^{\psi} \sigma; \rho\}(\overline{e}; e; t) &\coloneqq (\mathtt{fold}_{b.\sigma(\tau)}^{\sigma(\hat{\rho}(b') \to a)}(e; x.\lambda t.\{\tau \mathbin{@}_a^{\hat{\psi}} \hat{\sigma}; \hat{\rho}\}(\overline{e}; x; \mathtt{unroll}(t))))(t) \\
&\text{where } \hat{\psi} \coloneqq \psi, b \mapsto b' \\
&\qquad \hat{\sigma} \coloneqq \sigma, b \mapsto \sigma(\mu b.\tau) \\
&\qquad \hat{\rho} \coloneqq \rho, b' \mapsto \rho(\log_a^{\psi}(\mu b.\tau))
\end{aligned}$$

Fig. 2. The selection function

6 DECIDABLE EMPTINESS

In the proof of Theorem 3.1 from Lemma 3.2, we need a concrete term $\overline{e}$ of a non-empty type. The following is the lemma that justifies such dichotomy even in the constructive setting. The formal emptiness and non-emptiness are written "$\tau\ \square$" ("τ is empty") and "$\tau\ \boxdot$" ("τ is non-empty"); see Figure 3. The formal definitions are extended to cover open types to better handle inductive types; emptiness of open types is defined in terms of their emptiness when all the type variables are instantiated with the empty type.

$\boxed{\tau\ \square}$ and $\boxed{\tau\ \boxdot}$

"τ is empty" and "τ is non-empty"

$$\frac{}{a\ \square} \qquad \frac{}{\mathbb{0}\ \square} \qquad \frac{}{\mathbb{1}\ \boxdot} \qquad \frac{\tau_1\ \square}{\tau_1 \times \tau_2\ \square} \qquad \frac{\tau_2\ \square}{\tau_1 \times \tau_2\ \square} \qquad \frac{\tau_1\ \boxdot \qquad \tau_2\ \boxdot}{\tau_1 \times \tau_2\ \boxdot} \qquad \frac{\tau_1\ \boxdot}{\tau_1 + \tau_2\ \boxdot} \qquad \frac{\tau_2\ \boxdot}{\tau_1 + \tau_2\ \boxdot}$$

$$\frac{\tau_1\ \square \qquad \tau_2\ \square}{\tau_1 + \tau_2\ \square} \qquad \frac{\tau_1\ \square}{\tau_1 \to \tau_2\ \boxdot} \qquad \frac{\tau_2\ \boxdot}{\tau_1 \to \tau_2\ \boxdot} \qquad \frac{\tau_1\ \boxdot \qquad \tau_2\ \square}{\tau_1 \to \tau_2\ \square} \qquad \frac{\tau\ \square}{\mu a.\tau\ \square} \qquad \frac{\tau\ \boxdot}{\mu a.\tau\ \boxdot}$$

Fig. 3. Emptiness of types

Lemma 6.1. *For any kinding context Δ, any type τ such that $\Delta \vdash \tau$, and any type substitution δ such that $\cdot \vdash \delta : \Delta$ and $\delta(a) = \mathbb{0}$ for any type variable $a \in \Delta$, exactly one of the following two statements holds:*

- *$\tau\ \boxdot$ and there is an element e of type $\cdot;\cdot \vdash e : \delta(\tau)$ can be constructed.*
- *$\tau\ \square$ and there is an element e of type $\cdot;\cdot \vdash e : \delta(\tau) \to \mathbb{0}$ can be constructed.*

Proof. If both statements hold, then we have a closed term of the empty type $\mathbb{0}$. Thus, at most one of the statements is true. We prove that at least one of the statements holds by induction on τ.

- $\tau = a$.
  $a\ \square$ and we can pick $e = \mathtt{id}_{\mathbb{0}}$.
- $\tau = \mathbb{0}$.
  $\mathbb{0}\ \square$ and we can pick $e = \mathtt{id}_{\mathbb{0}}$.
- $\tau = \mathbb{1}$.
  $\mathbb{1}\ \boxdot$ and we can pick $e = \star$.
- $\tau = \tau_1 + \tau_2$.
  If $\tau_1\ \boxdot$ and there is an element e_1 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1)$, then $\tau_1 + \tau_2\ \boxdot$, and we can pick $e = \mathtt{inl}(e_1)$. If $\tau_2\ \boxdot$ and there is an element e_2 such that $\cdot;\cdot \vdash e_2 : \delta(\tau_2)$, then $\tau_1 + \tau_2\ \boxdot$, and we can pick $e = \mathtt{inr}(e_2)$. Otherwise, $\tau_1\ \square$ and $\tau_2\ \square$ and there are elements e_1 and e_2 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1) \to \mathbb{0}$ and $\cdot;\cdot \vdash e_2 : \delta(\tau_2) \to \mathbb{0}$. In this case, $\tau_1 + \tau_2\ \square$, and we can pick
  $$e = \lambda(z{:}\delta(\tau_1 + \tau_2)).\mathsf{case}(z; x.e_1(x); y.e_2(y)).$$
- $\tau = \tau_1 \times \tau_2$.
  If $\tau_1\ \square$ and there is an element e_1 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1) \to \mathbb{0}$, then $\tau_1 \times \tau_2\ \square$ and we can pick $e = \lambda(x{:}\delta(\tau_1 \times \tau_2)).e_1(\mathtt{fst}(x))$. If $\tau_2\ \square$ and there is an element e_2 such that $\cdot;\cdot \vdash e_2 : \delta(\tau_2) \to \mathbb{0}$, then $\tau_1 \times \tau_2\ \square$ and we can pick $e = \lambda(x{:}\delta(\tau_1 \times \tau_2)).e_2(\mathsf{snd}(x))$. Otherwise, $\tau_1\ \boxdot$ and $\tau_2\ \boxdot$ and there are elements e_1 and e_2 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1)$ and $\cdot;\cdot \vdash e_2 : \delta(\tau_2)$. In this case, $\tau_1 \times \tau_2\ \boxdot$ and we can pick $e = \langle e_1; e_2 \rangle$.
- $\tau = \tau_1 \to \tau_2$.
  If $\tau_1\ \square$ and there is an element e_1 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1) \to \mathbb{0}$, then $\tau_1 \to \tau_2\ \boxdot$ and we can pick $e = \lambda(x{:}\delta(\tau_1)).\mathtt{abort}(e_1(x))$. If $\tau_2\ \boxdot$ and there is an element e_2 such that $\cdot;\cdot \vdash e_2 : \delta(\tau_2)$, then $\tau_1 \to \tau_2\ \boxdot$ and we can pick $e = \lambda(x{:}\delta(\tau_1)).e_2$. Otherwise, $\tau_1\ \boxdot$ and $\tau_2\ \square$ and there are two elements e_1 and e_2 such that $\cdot;\cdot \vdash e_1 : \delta(\tau_1)$ and $\cdot;\cdot \vdash e_2 : \delta(\tau_2) \to \mathbb{0}$. In this case, $\tau_1 \to \tau_2\ \square$ and we can pick $e = \lambda(f{:}\delta(\tau_1 \to \tau_2)).e_2(f(e_1))$.
- $\tau = \mu a.\tau_1$.

Let the substitution δ' be $\delta, a \mapsto \mathbb{0}$. If $\tau_1 \ \square$ and there is an element e_1 such that $\cdot;\cdot \vdash e_1 : \delta'(\tau_1) \to \mathbb{0}$, then $\mu a.\tau_1 \ \square$ and we can pick

$$e = \lambda(x{:}\delta(\mu a.\tau_1)).\mathtt{fold}^{\mathbb{0}}_{a.\delta(\tau_1)}(x; y.e_1(y)).$$

If $\tau_1 \ \boxdot$ and there is an element e_1 sich that $\cdot;\cdot \vdash e_1 : \delta'(\tau_1)$, then $\mu a.\tau_1 \ \boxdot$ and we can pick

$$e = \mathtt{roll}_{a.\delta(\tau_1)}(\langle a^+.\delta(\tau_1)\rangle(\mathtt{abort}; e_1)).$$

□

7 POLARITIES AND FUNCTORIALITY

Functoriality plays an important role in our proofs. It appears in handling a-elements in the result type $H(a)$, mapping indexes back to their corresponding a-elements in the general input e, and also inductive types. First of all, functoriality satisfies the following basic properties: *compositionality* of functions applied to the same type variable and *commutativity* of functions applied to distinct type variables.

Lemma 7.1 (composition). *Given these data:*

- *A kinding context* Δ.
- *A type substitution* δ *such that* $\cdot \vdash \delta : \Delta$.
- *The distinguished type variable* $a \notin \Delta$.
- *Four type expressions* $\alpha(a)$, τ_1, τ_2, *and* τ_3*:*

(1) $\Delta, a \vdash \alpha(a)$
(2) $\Delta \vdash \tau_1$
(3) $\Delta \vdash \tau_2$
(4) $\Delta \vdash \tau_3$

- *An element* e *such that* $\Delta;\cdot \vdash e : \alpha(\tau_1)$.
- *Two functions* f *and* g *such that* $\Delta;\cdot \vdash f : \tau_1 \to \tau_2$ *and* $\Delta;\cdot \vdash g : \tau_2 \to \tau_3$.

We have the following:

$$\delta(\langle a^+.\alpha(a)\rangle(g; \langle a^+.\alpha(a)\rangle(f; e))) \sim_{\delta(\alpha(\tau_3))} \delta(\langle a^+.\alpha(a)\rangle(g \circ f; e))\ []$$

Proof. By simultaneous induction on the type expression $\alpha(a)$ for both positive and negative polarities. (The lemma only states the results for the positive polarity, but its proof needs both polarities due to domains of function types being contravariant.) □

Lemma 7.2 (commutativity). *Given these data:*

- *A kinding context* Δ.
- *A type substitution* δ *such that* $\cdot \vdash \delta : \Delta$,
- *Two different distinguished type variables* a *and* b *where* $a \neq b$, $a \notin \Delta$, $b \notin \Delta$.
- *Five type expressions* $\alpha(a, b)$, τ_1, τ_2, τ_3, *and* τ_4 *such that*

(1) $\Delta, a, b \vdash \alpha(a, b)$.
(2) $\Delta \vdash \tau_1$.
(3) $\Delta \vdash \tau_2$.
(4) $\Delta \vdash \tau_3$.
(5) $\Delta \vdash \tau_4$.

- *An element* e *such that* $\Delta;\cdot \vdash e : \alpha(\tau_1, \tau_3)$.
- *Two functions* f_a *and* f_b *such that* $\Delta;\cdot \vdash f_a : \tau_1 \to \tau_2$ *and* $\Delta;\cdot \vdash f_b : \tau_3 \to \tau_4$.

We have the following:

$$\delta(\langle b^+.\alpha(a,b)\rangle(f_b; \langle a^+.\alpha(a,b)\rangle(f_a; e))) \sim_{\delta(\alpha(\tau_2,\tau_4))} \delta(\langle a^+.\alpha(a,b)\rangle(f_a; \langle b^+.\alpha(a,b)\rangle(f_b; e)))\ []$$

$$\boxed{\langle \Xi^p.\tau\rangle(\phi; e)}$$

Multi-variable functoriality with functions $\phi(b) : \sigma_1(b) \to \sigma_2(b)$
and type τ with respect to type variables Ξ of polarity p.

$$\begin{aligned}
\langle \Xi^+.b\rangle(\phi; e) &\coloneqq \phi(b)(e) \qquad (b \in \Xi)\\
\langle \Xi^p.b\rangle(\phi; e) &\coloneqq e \qquad (b \notin \Xi)\\
\langle \Xi^p.\mathbb{0}\rangle(\phi; e) &\coloneqq e\\
\langle \Xi^p.\tau_{\mathsf{left}} + \tau_{\mathsf{right}}\rangle(\phi; e) &\coloneqq \mathtt{case}(e; x.\mathtt{inl}(\langle \Xi^p.\tau_{\mathsf{left}}\rangle(\phi; x)); y.\mathtt{inr}(\langle \Xi^p.\tau_{\mathsf{right}}\rangle(\phi; y)))\\
\langle \Xi^p.\mathbb{1}\rangle(\phi; e) &\coloneqq e\\
\langle \Xi^p.\tau_{\mathsf{fst}} \times \tau_{\mathsf{snd}}\rangle(\phi; e) &\coloneqq \langle \langle \Xi^p.\tau_{\mathsf{fst}}\rangle(\phi; \mathtt{fst}(e)); \langle \Xi^p.\tau_{\mathsf{snd}}\rangle(\phi; \mathtt{snd}(e))\rangle\\
\langle \Xi^+.\tau_{\mathsf{dom}} \to \tau_{\mathsf{cod}}\rangle(\phi; e) &\coloneqq \lambda(x{:}\sigma_2(\tau_{\mathsf{dom}})).\langle \Xi^+.\tau_{\mathsf{cod}}\rangle(\phi; e(\langle \Xi^-.\tau_{\mathsf{dom}}\rangle(\phi; x)))\\
\langle \Xi^-.\tau_{\mathsf{dom}} \to \tau_{\mathsf{cod}}\rangle(\phi; e) &\coloneqq \lambda(x{:}\sigma_1(\tau_{\mathsf{dom}})).\langle \Xi^-.\tau_{\mathsf{cod}}\rangle(\phi; e(\langle \Xi^+.\tau_{\mathsf{dom}}\rangle(\phi; x)))\\
\langle \Xi^+.\mu b.\tau\rangle(\phi; e) &\coloneqq \mathtt{fold}^{\mu b.\sigma_2(\tau)}_{b.\sigma_1(\tau)}(e; x.\mathtt{roll}_{b.\sigma_2(\tau)}(\langle \Xi^+.\tau\rangle(\phi; x)))\\
\langle \Xi^-.\mu b.\tau\rangle(\phi; e) &\coloneqq \mathtt{fold}^{\mu b.\sigma_1(\tau)}_{b.\sigma_2(\tau)}(e; x.\mathtt{roll}_{b.\sigma_1(\tau)}(\langle \Xi^-.\tau\rangle(\phi; x)))
\end{aligned}$$

Fig. 4. Functoriality with respect to multiple variables

Proof. By simultaneous induction on the type expression $\alpha(a, b)$ for both positive and negative polarities. (The lemma only states the results for the positive polarity, but its proof needs both polarities due to domains of function types being contravariant.) □

7.1 Generalization with Multiple Variables

Because inductive types can be nested, we also have to generalize the original definition of functoriality to deal with multiple variables. The generalized functoriality is defined in Figure 4. To reduce cluttering, Lemmas 7.1 and 7.2 and their natural extensions to multi-variable functoriality (where functions on the same variable compose and those on different variables fuse) are often used implicitly in the rest of this paper.

7.2 Connections to Logical Relations

When a function is used as an admissible relation for some type variable a in logical relations at a type τ where $a \in^+ \tau$, functoriality coincides with those logical relations. That is, the logical relation itself becomes a function that matches the functorial action.

Lemma 7.3. *Given two closed types τ_{dom} and τ_{cod}, a function $\cdot; \cdot \vdash f : \tau_{\mathsf{dom}} \to \tau_{\mathsf{cod}}$, a distinguished type variable a, a type τ such that $a \vdash \tau$ where $a \in^+ \tau$, an element $\cdot; \cdot \vdash e : \tau[\tau_{\mathsf{dom}}/a]$, we have*

$$e \sim_\tau \langle a^+.\tau\rangle(f; e)\ [a \mapsto f]$$

Proof. This is the "forwarding" case clause of a special case of more general Lemma 7.4 below with $\Xi = \emptyset$ (that is, no bound variables introduced by inductive types). □

Lemma 7.4. *Given these data:*

- *Closed types τ_{dom} and τ_{cod}.*
- *A function $\cdot; \cdot \vdash f : \tau_{\mathsf{dom}} \to \tau_{\mathsf{cod}}$.*
- *A kinding context Ξ intended to be the set of type variables introduced by inductive types.*

- *A type substitution* $a \vdash \xi : \Xi$.
  For convenience, we define two type substitutions σ_{dom} *and* σ_{cod} *as follows:*

$$\sigma_{\mathsf{dom}} \coloneqq \xi[\tau_{\mathsf{dom}}/a], a \mapsto \tau_{\mathsf{dom}}$$
$$\sigma_{\mathsf{cod}} \coloneqq \xi[\tau_{\mathsf{cod}}/a], a \mapsto \tau_{\mathsf{cod}}$$

- *A type* τ *such that* $\Xi, a \vdash \tau$ *and* $\Xi \in^{++} \tau$.
- *A family of admissible relations* $\eta : \sigma_{\mathsf{dom}} \leftrightarrow \sigma_{\mathsf{cod}}$ *indexed by* Ξ.

The following two statements holds:

Positive, moving forward: *If* $a \in^{+} \tau$*, then for any family of "forward" functions* F *indexed by* Ξ *such that* $\cdot;\cdot \vdash F_b : \sigma_{\mathsf{dom}}(b) \to \sigma_{\mathsf{cod}}(b)$*, and any* $\cdot;\cdot \vdash e : \sigma_{\mathsf{dom}}(\tau)$ *such that* $e \sim_\tau e\ [\theta]$ *and where* $\theta : \sigma_{\mathsf{dom}} \leftrightarrow \sigma_{\mathsf{dom}}$ *is a family of admissible relations defined as*
(1) θ_a *is contextual equivalence at* τ_{dom} *for the distinguished type variable* a*; and*
(2) $\theta_b(e, e')$ *if and only if* $e \cong e'$ *and* $\eta_b(e, F_b(e))$ *for any type variable* $b \in \Xi$
we have the relation

$$e \sim_\tau \langle a^{+}.\tau\rangle(f; \langle \Xi^{+}.\tau\rangle(F; e))\ [a \mapsto f, \eta]$$

Negative, moving backward: *If* $a \in^{-} \tau$*, then for any family of "backward" functions* G *indexed by* Ξ *such that* $\cdot;\cdot \vdash G_b : \sigma_{\mathsf{cod}}(b) \to \sigma_{\mathsf{dom}}(b)$*, and any* $\cdot;\cdot \vdash e : \sigma_{\mathsf{cod}}(\tau)$ *such that* $e \sim_\tau e\ [\theta]$ *where* $\theta : \sigma_{\mathsf{cod}} \leftrightarrow \sigma_{\mathsf{cod}}$ *is a family of admissible relations defined as*
(1) θ_a *is contextual equivalence at* τ_{cod} *for the distinguished type variable* a*; and*
(2) $\theta_b(e, e')$ *if and only if* $e \cong e'$ *and* $\eta_b(G_b(e), e)$ *for any type variable* $b \in \Xi$
we have the relation

$$\langle a^{-}.\tau\rangle(f; \langle \Xi^{+}.\tau\rangle(G; e)) \sim_\tau e\ [a \mapsto f, \eta]$$

Proof. By induction on τ.

- $\tau = \mathbb{0}$ or $\tau = \mathbb{1}$.
  Trivial by parametricity.
- $\tau = a$ (the distinguished type variable).
  Since $a \in^{+} a$, we only need to prove that for any $\cdot;\cdot \vdash e : \tau_1$,

$$e \sim_a \langle a^{+}.a\rangle(f; e)\ [a \mapsto f, \eta]$$

  which is obvious from the definition of functoriality.
- $\tau = b \in \Xi$ (bound variables introduced by inductive types).
  If $a \in^{+} b$, we need to prove that for any $\cdot;\cdot \vdash e : \sigma_{\mathsf{dom}}(b)$ such that $e \sim_b e\ [\theta]$,

$$e \sim_b \langle \Xi^{+}.b\rangle(F; e)\ [a \mapsto f, \eta]$$

  which is equivalent to prove that $\eta_b(e, F_b(e))$. From $e \sim_b e\ [\theta]$ we know $\theta_b(e, e)$, which by definition is exactly $\eta_b(e, F_b(e))$. The case where $a \in^{-} b$ is similar.
- $\tau = \tau_1 \times \tau_2$ or $\tau = \tau_1 + \tau_2$.
  By definition and inductive hypotheses.
- $\tau = \tau_1 \to \tau_2$.
  We first consider the case where $a \in^{+} \tau$. Given $\cdot;\cdot \vdash e : \sigma_{\mathsf{dom}}(\tau)$ such that $e \sim_\tau e\ [\theta]$, by definition of the logical relation, in order to prove that

$$e \sim_\tau \langle a^{+}.\tau\rangle(f; \langle \Xi^{+}.\tau\rangle(F; e))\ [a \mapsto f, \eta]$$

  it suffices to prove that for any $e_1' \sim_{\tau_1} e_2'\ [a \mapsto f, \eta]$,

$$e(e_1') \sim_{\tau_2} \langle a^{+}.\tau\rangle(f; \langle \Xi^{+}.\tau\rangle(F; e))(e_2')\ [a \mapsto f, \eta]$$

which can be simplified to the following by evaluation:

$$e(e_1') \sim_{\tau_2} \langle a^+.\tau_2\rangle(f; \langle \Xi^+.\tau_2\rangle(F; e(\langle \Xi^-.\tau_1\rangle(F; \langle a^-.\tau_1\rangle(f; e_2'))))) \ [a \mapsto f, \eta]$$

Because $\Xi \in^{++} \tau_1 \to \tau_2$, which means $\Xi \notin \tau_1$ and $\langle \Xi^-.\tau_1\rangle(F; -)$ is a no-op, we have

$$\langle \Xi^-.\tau_1\rangle(F; \langle a^-.\tau_1\rangle(f; e_2')) = \langle a^-.\tau_1\rangle(f; e_2')$$

By the inductive hypothesis on τ_1, we have

$$\langle a^-.\tau_1\rangle(f; e_2') \sim_{\tau_1} e_2' \ [a \mapsto f]$$

But we also know $e_1' \sim_{\tau_1} e_2' \ [a \mapsto f]$. Therefore

$$\langle a^-.\tau_1\rangle(f; e_2') \cong e_1'$$

Hence it suffices to prove that

$$e(e_1') \sim_{\tau_2} \langle a^+.\tau_2\rangle(f; \langle \Xi^+.\tau_2\rangle(F; e(e_1'))) \ [F, a \mapsto f]$$

This follows the inductive hypothesis on τ_2 as long as we can show that $e(e_1') \sim_{\tau_2} e(e_1') \ [\theta]$. We know $e \sim_{\tau_1 \to \tau_2} e \ [\theta]$, and since $\Xi \notin \tau_1$, it is trivial that $e_1' \sim_{\tau_1} e_1' \ [\theta]$, and thus the condition $e(e_1') \sim_{\tau_2} e(e_1') \ [\theta]$ holds.

The case where $a \in^- \tau$ is similar.

- $\tau = \mu b.\tau_1$ where $\Xi, a, b \vdash \tau_1$ and $b \in^{++} \tau_1$.

  Positive/forward We start with the case where $a \in^+ \tau$. Let $\mathcal{R} : \sigma_{\mathsf{dom}}(\tau) \leftrightarrow \sigma_{\mathsf{dom}}(\tau)$ be an admissible relation defined as follows:

$$\mathcal{R}(e, e') \coloneqq e \cong e' \text{ and } e \sim_\tau \langle a^+.\tau\rangle(f; \langle \Xi^+.\tau\rangle(F; e)) \ [a \mapsto f, \eta]$$

  The goal is to prove $\mathcal{R}(e, e)$ under the assumption that $e \sim_\tau e \ [\theta]$. Let

$$\begin{aligned}
\sigma_{\mathsf{cod/dom}} &\coloneqq \xi[\tau_{\mathsf{cod}}/a], a \mapsto \tau_{\mathsf{dom}} \\
F_\tau^+ &\coloneqq \lambda(x{:}\sigma_{\mathsf{dom}}(\tau)).\langle \Xi^+.\tau\rangle(F; x) \\
f_\tau^+ &\coloneqq \lambda(x{:}\sigma_{\mathsf{cod/dom}}(\tau)).\langle a^+.\tau\rangle(f; x) \\
f_b &\coloneqq f_\tau^+ \circ F_\tau^+ \\
\Xi' &\coloneqq \Xi, b \\
\xi' &\coloneqq \xi, b \mapsto \tau \\
\sigma'_{\mathsf{dom}} &\coloneqq \xi'[\tau_{\mathsf{dom}}/a], a \mapsto \tau_{\mathsf{dom}} \\
F' &\coloneqq F, b \mapsto f_b \\
\eta' &\coloneqq \eta, b \mapsto - \sim_\tau - \ [a \mapsto f, \eta] \\
\theta' &\coloneqq \theta, b \mapsto \mathcal{R}
\end{aligned}$$

  Unfolding the definition of $e \sim_{\mu b.\tau_1} e \ [\theta]$, it suffices to show that $\mathcal{R}$ is a prefixed point respect to $\mu b.\tau_1$ and θ. That is, to prove $\mathcal{R}(e, e)$, it suffices to prove that for any $\cdot; \cdot \vdash e_1' : \sigma'_{\mathsf{dom}}(\tau_1)$ and $\cdot; \cdot \vdash e_2' : \sigma'_{\mathsf{dom}}(\tau_1)$ such that $e_1' \sim_{\tau_1} e_2' \ [\theta, b \mapsto \mathcal{R}]$, wa can show

$$\mathcal{R}(\mathtt{roll}_{b.\sigma_{\mathsf{dom}}(\tau_1)}(e_1'), \mathtt{roll}_{b.\sigma_{\mathsf{dom}}(\tau_1)}(e_2'))$$

  Because $\mathcal{R}$ is an admissible subrelation of contextual equivalence, without loss of generality we may assume $e_1' = e_2' = e'$. By the definition of $\mathcal{R}$, the above goal is equivalent to

$$\mathtt{roll}_{b.\sigma_{\mathsf{dom}}(\tau_1)}(e') \sim_\tau f_\tau^+(F_\tau^+(\mathtt{roll}_{b.\sigma_{\mathsf{dom}}(\tau_1)}(e'))) \ [a \mapsto f, \eta]$$

We can evaluate its right-hand side as follows:

$$\begin{aligned}
& f_\tau^+(F_\tau^+(\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(e'))) \\
\cong\ & f_\tau^+(\langle \Xi^+.\tau\rangle(F; \mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(e'))) \\
\cong\ & f_\tau^+(\mathtt{fold}^{\mu b.\sigma_{\mathrm{cod/dom}}(\tau_1)}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(e'); x.\mathtt{roll}_{b.\sigma_{\mathrm{cod/dom}}(\tau_1)}(\langle \Xi^+.\tau_1\rangle(F; x)))) \\
\cong\ & f_\tau^+(\mathtt{roll}_{b.\sigma_{\mathrm{cod/dom}}(\tau_1)}(\langle \Xi^+.\tau_1\rangle(F; \langle b^+.\tau_1\rangle(F_\tau^+; e')))) \\
\cong\ & \langle a^+.\tau\rangle(f; \langle \Xi^+.\tau\rangle(F; \mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(e'))) \\
\cong\ & \mathtt{fold}^{\mu b.\sigma_{\mathrm{cod}}(\tau_1)}_{b.\sigma_{\mathrm{cod/dom}}(\tau_1)}(\mathtt{roll}_{b.\sigma_{\mathrm{cod/dom}}(\tau_1)}(\\
& \quad \langle \Xi^+.\tau_1\rangle(F; \langle b^+.\tau_1\rangle(F_\tau^+; e'))); x.\mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(\langle a^+.\tau_1\rangle(f; x))) \\
\cong\ & \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(\langle a^+.\tau_1\rangle(f; \langle b^+.\tau_1\rangle(f_\tau^+; \langle \Xi^+.\tau_1\rangle(F; \langle b^+.\tau_1\rangle(F_\tau^+; e'))))) \\
\cong\ & \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(\langle a^+.\tau_1\rangle(f; \langle \Xi'^+.\tau_1\rangle(F'; e')))
\end{aligned}$$

Therefore it is equivalent to prove that

$$\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(e') \sim_\tau \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(\langle a^+.\tau_1\rangle(f; \langle \Xi'^+.\tau_1\rangle(F'; e'))) \ [a \mapsto f, \eta]$$

By the definition of the logical relation, it suffices to prove that

$$e' \sim_{\tau_1} \langle a^+.\tau_1\rangle(f; \langle \Xi'^+.\tau_1\rangle(F'; e')) \ [a \mapsto f, \eta']$$

This follows the inductive hypothesis on τ_1 with $\theta = \theta'$ if we can verify that $\theta'_b(e, e')$ if and only if $e \cong e'$ and $\eta'_b(e, F'_b(e))$, which holds by definition because $\theta'_b = \mathcal{R}$.

Negative/backward Now consider the situation where $a \in^- \tau$. Let $\mathcal{R} : \sigma_{\mathrm{cod}}(\tau) \leftrightarrow \sigma_{\mathrm{cod}}(\tau)$ be an admissible relation defined as follows:

$$\mathcal{R}(e, e') \coloneqq e \cong e' \text{ and } \langle a^-.\tau\rangle(f; \langle \Xi^+.\tau\rangle(G; e)) \sim_\tau e \ [a \mapsto f, \eta]$$

The goal is to prove $\mathcal{R}(e, e)$ under the assumption that $e \sim_\tau e \ [\theta]$. Let

$$\begin{aligned}
\sigma_{\mathrm{dom/cod}} &\coloneqq [\xi[\tau_{\mathrm{dom}}/a], a \mapsto \tau_{\mathrm{cod}}] \\
G_\tau^+ &\coloneqq \lambda(x{:}\sigma_{\mathrm{cod}}(\tau)).\langle \Xi^+.\tau\rangle(G; x) \\
f_\tau^- &\coloneqq \lambda(x{:}\sigma_{\mathrm{dom/cod}}(\tau)).\langle a^-.\tau\rangle(f; x) \\
g_b &\coloneqq f_\tau^- \circ G_\tau^+ \\
\Xi' &\coloneqq \Xi, b \\
\xi' &\coloneqq \xi, b \mapsto \tau \\
\sigma'_{\mathrm{cod}} &\coloneqq \xi'[\tau_{\mathrm{cod}}/a], a \mapsto \tau_{\mathrm{cod}} \\
G' &\coloneqq G, b \mapsto g_b \\
\eta' &\coloneqq \eta, b \mapsto - \sim_\tau - \ [a \mapsto f, \eta] \\
\theta' &\coloneqq \theta, b \mapsto \mathcal{R}
\end{aligned}$$

Unfolding the definition of $e \sim_{\mu b.\tau_1} e \ [\theta]$, it suffices to show that $\mathcal{R}$ is a prefixed point respect to $\mu b.\tau_1$ and θ. That is, to prove $\mathcal{R}(e, e)$, it suffices to prove that for any $\cdot; \cdot \vdash e'_1 : \sigma'_{\mathrm{cod}}(\tau_1)$ and $\cdot; \cdot \vdash e'_2 : \sigma'_{\mathrm{cod}}(\tau_1)$ such that $e'_1 \sim_{\tau_1} e'_2 \ [\theta, b \mapsto \mathcal{R}]$, we can show

$$\mathcal{R}(\mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e_1), \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e_2))$$

Because $\mathcal{R}$ is an admissible subrelation of contextual equivalence, without loss of generality we may assume $e'_1 = e'_2 = e'$. By the definition of $\mathcal{R}$, it is equivalent to

$$f_\tau^-(G_\tau^+(\mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e'))) \sim_\tau \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e') \ [a \mapsto f, \eta]$$

We can evaluate its left-hand side as follows:

$$\begin{aligned}
&f_\tau^-(G_\tau^+(\mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e')))\\
\cong\ &f_\tau^-(\mathtt{roll}_{b.\sigma_{\mathrm{dom/cod}}(\tau_1)}(\langle \Xi^+.\tau_1\rangle(G;\langle b^+.\tau_1\rangle(G_\tau^+;e'))))\\
\cong\ &\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(\langle a^-.\tau_1\rangle(f;\langle b^+.\tau_1\rangle(f_\tau^-;\langle \Xi^+.\tau_1\rangle(G;\langle b^+.\tau_1\rangle(G_\tau^+;e')))))\\
\cong\ &\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(\langle a^-.\tau_1\rangle(f;\langle \Xi'^+.\tau_1\rangle(G';e')))
\end{aligned}$$

Therefore it is equivalent to prove that

$$\mathtt{roll}_{b.\sigma_{\mathrm{dom}}(\tau_1)}(\langle a^-.\tau_1\rangle(f;\langle \Xi'^+.\tau_1\rangle(G';e'))) \sim_\tau \mathtt{roll}_{b.\sigma_{\mathrm{cod}}(\tau_1)}(e')\ [a \mapsto f, \eta]$$

By definition of the logical relation, it suffices to prove that

$$\langle a^-.\tau_1\rangle(f;\langle \Xi'^+.\tau_1\rangle(G';e')) \sim_{\tau_1} x\ [a \mapsto f, \eta']$$

This follows the inductive hypothesis on τ_1 with $\theta = \theta'$ if we can verify that $\theta'_b(e,e)$ if and only if $e \cong e'$ and $\eta'_b(G'_b(e),e)$, which holds by definition because $\theta'_b = \mathcal{R}$.

□

8 FUNCTIONAL LOGICAL RELATIONS

Here are a few more lemmas used in the proofs of Theorem 3.1 and Lemma 3.2.

Lemma 8.1. *Given a type $\Delta \vdash \tau$ where $\Delta \in^+ \tau$, a family of relations $\eta : \delta_1 \leftrightarrow \delta_2$ where each relation η_a is a function, the relation $- \sim_\tau - \ [\eta]$ must be a function.*

Proof. By Lemma 8.2 with $\Xi = \emptyset$. □

Lemma 8.2. *Given a type $\Delta, \Xi \vdash \tau$ where $\Xi \in^{++} \tau$, a family of relations $\eta : \delta_1 \leftrightarrow \delta_2$ where each relation η_a is a function, a family of relations $\eta_1 : \sigma_1 \leftrightarrow \sigma_2$,*

- *If $\Delta \in^+ \tau$, then for any e such that $e \sim_{\delta_1(\tau)} e\ [\theta]$, there must be a unique term e' such that $e \sim_\tau e'\ [\eta, \eta_1]$. Here $\theta : \sigma_1 \leftrightarrow \sigma_1$ is a family of relations indexed by Ξ such that $\theta_b(e,e')$ if and only if $e \cong e'$ and there exists a unique e'' such that $\eta_1(e,e'')$,*
- *If $\Delta \in^- \tau$, then for any e such that $e \sim_{\delta_2(\tau)} e\ [\theta]$, there must be a unique term e' such that $e' \sim_\tau e\ [\eta, \eta_1]$. Here $\theta : \sigma_2 \leftrightarrow \sigma_2$ is a family of relations indexed by Ξ such that $\theta_b(e,e')$ if and only if $e \cong e'$ and there exists a unique e'' such that $\eta_1(e'',e)$,*

Proof. By induction on τ.

- $\tau = \mathbb{0}$ or $\tau = \mathbb{1}$.
  Trivial.
- $\tau = a \in \Delta$.
  We know $a \in^+ a$. Then by the definition of logical relations, it suffices to prove that there is a unique e' such that $\eta_a(e,e')$, which follows directly by the assumption that η_a is a function.
- $\tau = b \in \Xi$.
  By assumption we know $e \sim_{\delta_1(b)} e\ [\theta]$, which means $\theta_b(e,e)$. By definition of θ, we know there is a unique e' such that $\eta_1(e,e')$, which is equivalent to $e \sim_\tau e'\ [\eta, \eta_1]$.
- $\tau = \tau_1 \times \tau_2$.
  We first consider the case where $\Delta \in^+ \tau$. Because $e \sim_{\delta_1(\tau)} e\ [\theta]$, we know $\mathtt{fst}(e) \sim_{\delta_1(\tau_1)} \mathtt{fst}(e)\ [\theta]$. By inductive hypotheses on τ_1 we know there must be a unique term e_1 such that $\mathtt{fst}(e) \sim_{\tau_1} e_1\ [\eta, \eta_1]$. Similarly there exists a unique term e_2 such that $\mathtt{snd}(e) \sim_{\tau_2} e_2\ [\eta, \eta_1]$. Then by the definition of logical relations, we get

$$\langle \mathtt{fst}(e); \mathtt{snd}(e)\rangle \sim_{\tau_1 \times \tau_2} \langle e_1; e_2\rangle\ [\eta, \eta_1]$$

which means $e \sim_\tau \langle e_1; e_2\rangle\ [\eta, \eta_1]$ because $e \cong \langle \mathtt{fst}(e); \mathtt{snd}(e)\rangle$. Thus we proved there exists $e' = \langle \mathtt{fst}(e); \mathtt{snd}(e)\rangle$.
Now suppose there is another term e'' such that $e \sim_\tau e''\ [\eta, \eta_1]$, and we will show that $e'' \cong e'$. Then by the definition of logical relations, we have $\mathtt{fst}(e) \sim_{\tau_1} \mathtt{fst}(e'')\ [\eta, \eta_1]$. Because of the uniqueness of e_1, we must have $e_1 \cong \mathtt{fst}(e'')$. Similarly we have $e_2 \cong \mathtt{snd}(e'')$. Therefore $e' \cong \langle e_1; e_2\rangle$.
The case where $\Delta \in^- \tau$ is similar.

- $\tau = \tau_1 + \tau_2$.
We first consider the case where $\Delta \in^+ \tau$. When $e \cong \mathtt{inl}(e_1)$, then by inductive hypothesis on τ_1, we know there is a unique e_1' such that $e_1 \sim_{\tau_1} e_1'\ [\eta, \eta_1]$. Thus we have $e' = \mathtt{inl}(e_1')$ such that $e \sim_\tau e'\ [\eta, \eta_1]$. The uniqueness of e' comes from the uniqueness of e_1'. Similarly when $e \cong \mathtt{inr}(e_1)$, there also exists a unique e' such that $e \sim_\tau e'\ [\eta, \eta_1]$.
- $\tau = \tau_1 \to \tau_2$.
If $\Delta \in^+ \tau$, we first rename e into f to indicate it is a function. Because $\Delta \in^+ \tau$ and $\Xi \in^{++} \tau$, we know $\Delta \in^- \tau_1$ and $\Xi \notin \tau$. So we know for any term e_1 of type $\delta_2(\tau_1)$ it always holds that $e_1 \sim_{\delta_1(\tau_1)} e_1\ [\theta]$. And then by inductive hypothesis on τ_1, there is a unique e_1' such that $e_1' \sim_{\tau_1} e_1\ [\eta, \eta_1]$. This means there is a function f_1 such that $e_1' = f_1(e_1)$.
By inductive hypothesis on τ_2 we know there is a unique e_2 such that $f(e_1') \sim_{\tau_2} e_2\ [\eta, \eta']$, which means there is a function f_2 such that $e_2 = f_2(f(e_1')) = f_2(f(f_1(e_1)))$. Hence we have $f \sim_\tau f_2 \circ f \circ f_1\ [\eta]$, *i.e.,* for any f there exists $f' = f_2 \circ f \circ f_1$ such that $f \sim_\tau f'\ [\eta]$.
Now to prove f' is unique, suppose there is another term f'' such that $f \sim_\tau f''\ [\eta]$ and we will show that $f'' \cong f'$. It suffices to prove that $f''(e_1) \cong f_2(f(f_1(e_1)))$ for any e_1 of type $\delta_2(\tau_1)$. By definition of f_1 we have $f_1(e_1) \sim_{\tau_1} e_1\ [\eta]$. Thus $f(f_1(e_1)) \sim_{\tau_2} f''(e_1)\ [\eta]$. Then by definition of f_2 we know $f''(e_1) \cong f'(e_1)$.
The case where $\Delta \in^- \tau$ is similar.
- $\tau = \mu b.\tau_1$.
If $\Delta \in^+ \tau$, first define a relation $\mathcal{R}$ as $\mathcal{R}(e, e')$ iff $e \cong e'$ and there exists a unique term e'' satisfying $e \sim_\tau e''\ [\eta, \eta_1]$. Because $e \sim_{\delta_1(\tau)} e\ [\theta]$, by the definition of logical relations, to prove $\mathcal{R}(e, e)$, it suffices to prove that $\mathcal{R}$ is a prefixed point: for any $e_1 \sim_{\delta_1(\tau_1)} e_1\ [\theta, b \mapsto \mathcal{R}]$, we always have $\mathcal{R}(\mathtt{roll}_{b.\delta_1(\tau_1)}(e_1), \mathtt{roll}_{b.\delta_1(\tau_1)}(e_1))$. By the definition of $\mathcal{R}$, it is equivalent to prove that there is a unique term e'' satisfying $\mathtt{roll}_{b.\delta_1(\tau_1)}(e_1) \sim_\tau e''\ [\eta, \eta_1]$. Now let

$$\theta' \coloneqq \theta, b \mapsto \mathcal{R}$$
$$\eta_1' \coloneqq \eta, b \mapsto - \sim_\tau -\ [\eta, \eta_1]$$

Because $e_1 \sim_{\delta_1(\tau_1)} e_1\ [\theta']$, by the inductive hypothesis on τ_1, we know the relation $- \sim_{\tau_1} -\ [\eta, \eta_1']$ is a function, and let's call it f. Then we have an $e'' = \mathtt{roll}_{b.\delta_2(\tau_1)}(f(e_1))$. And by the definition of logical relations, it holds that $\mathtt{roll}_{b.\delta_1(\tau_1)}(e_1) \sim_\tau e''\ [\eta, \eta_1]$.
Now we show that e'' is unique. Suppose there is another $\hat{e}''$ satisfying $\mathtt{roll}_{b.\delta_1(\tau_1)}(e_1) \sim_\tau \hat{e}''\ [\eta, \eta_1]$, there must be a term $\hat{e}_1''$ such that $\hat{e}'' \cong \mathtt{roll}_{b.\delta_2(\tau_1)}(\hat{e}_1'')$. By the definition of logical relations, we have $e_1 \sim_{\tau_1} \hat{e}_1''\ [\eta, \eta_1']$. Then by the definition of f and the fact that f is a function, we know $\hat{e}_1'' \cong f(e_1)$. Therefore $\hat{e}'' \cong e''$.
The case where $\Delta \in^- \tau$ is similar.

□

9 PROOF OF LEMMA 3.2

This section is devoted to proving Lemma 3.2, the pinnacle of our technical development. The lemma is necessarily vastly generalized to Lemma 9.1 to account for bound type variables introduced by inductive types. We should first describe the reduction:

Proof of Lemma 3.2 from Lemma 9.1. By parametricity on polymorphic types[1] we know

$$f \sim_{\delta(\forall a.\alpha(a)\to H(a))} f\ []$$

By choosing s as the admissible relation for the type variable a, it suffices to prove that

$$e^* \sim_{\delta(\alpha(a))} e\ [a \mapsto s]$$

Then we can apply Lemma 9.1 with

$$\begin{aligned} \Xi &\coloneqq \emptyset \\ \Xi' &\coloneqq \emptyset \\ a^* &\coloneqq \mu a.\log_a(\alpha(a)) \\ \iota &\coloneqq \mathtt{roll}_{a.\delta(\log_a(\alpha(a)))} \\ s &\coloneqq s \end{aligned}$$

where all the conditions are satisifed; in particular,

(1) $e \sim_{\delta(\alpha(\tau))} e\ []$ by parametricity; and
(2) ι (as $\mathtt{roll}_{a.-}$) is injective; and
(3) for any t such that $\cdot;\cdot \vdash t : \delta(\log_a(\alpha(a))[\delta(a^*)/a])$,

$$s(\iota(t)) \cong \delta\{\alpha(a)\ @_a\ [a \mapsto \tau]; \emptyset\}(\overline{e}; e; \langle a^+.\delta(\log_a(\alpha(a)))\rangle(s;t))$$

by the definitions of s and ι and the β-rule of recursive types.

The definition of s is reproduced here for the convenience of readers:

$$\lambda(x{:}\delta(a^*)).\mathtt{fold}^{\tau}_{a.\delta(\log_a(\alpha(a)))}(x; y.\delta\{\alpha(a)\ @_a\ [a \mapsto \tau]; \emptyset\}(\overline{e}; e; y))$$

□

Lemma 9.1. *Suppose we have the following data:*

- *Distinct type variables:*
  - *An ambient kinding context Δ.*
  - *A type substitution δ such that $\cdot \vdash \delta : \Delta$, applied to all open types in the theorem.*
  - *The distinguished type variable a.*
  - *A kinding context Ξ intended to be the set of type variables introduced by inductive types.*
  - *Another kinding context Ξ' intended to be the logarithm representatives of Ξ.*
  - *A one-to-one function ψ from Ξ to Ξ'. ψ is intended to send a type variable in Ξ to its logarithm representative in Ξ'.*
- *Some types and type substitutions:*

(1) *$\alpha(a)$ such that $\Delta, a, \Xi \vdash \alpha(a)$ and $a \in^+ \log_a^{\psi}(\alpha(a))$*
(2) *$\Delta \vdash a^*$, intended to be the type of indexes in Lemma 3.2.*
(3) *$\cdot \vdash \tau$, intended to be the type of general a-elements we wish to relate to.*
(4) *$\Delta, a \vdash \xi : \Xi$, intended to be the type of the parts associated with variables introduced by inductive types.*

[1] This parametricity is different from the one given in Section 2 which only applies to rank-0 types. Instead, the parametricity here is the same as the logical relation for the standard System F without the prenex restriction. We use a different version here to handle the universal quantifiers, and for rank-0 types it collides with the parametricity given in Section 2.

(5) $\Delta \vdash \xi^- : \Xi$ *intended to be the type of the skeletons associated with variables introduced by inductive types.*
(6) $\Delta, a, \Xi \vdash \rho : \Xi'$ *intended to be the type of logarithm of the recursive parts.*
For convenience, σ *is defined to be* $[\xi[a^*/a], a \mapsto a^*]$ *and* ς *to be* $[\xi[\tau/a], a \mapsto \tau]$.

- *An element* $\cdot;\cdot \vdash \overline{e} : \tau$ *showing that* τ *is not empty.*
- *An element* $\cdot;\cdot \vdash e : \delta(\varsigma(\alpha(a)))$ *intended to be the general input.*
- *A family of functions* $\mathsf{Skeletonize}_{b:\Xi}$ *where* $\cdot;\cdot \vdash \mathsf{Skeletonize}_b : \delta(\varsigma(b) \to \xi^-(b))$
- *A family of functions* $\mathsf{Refill}_{b:\Xi}$ *where* $\cdot;\cdot \vdash \mathsf{Refill}_b : \delta(\xi^-(b) \to \sigma_a^{\psi;\rho}(b))$
- *A family of functions* $\mathsf{Select}_{b:\Xi}$ *where* $\cdot;\cdot \vdash \mathsf{Select}_b : \delta(\varsigma(b) \to \ddot{\varsigma}_a^{\psi;\rho}(b))$
- *A function* $\cdot;\cdot \vdash \iota : \delta(\sigma(\rho(\log_a^\psi(\alpha(a))))) \to \delta(a^*)$ *where* ι *is injective.*
- *A global selection function* $\cdot;\cdot \vdash s : \delta(a^*) \to \tau$ *such that for any* $\cdot;\cdot \vdash t : \delta(\sigma(\rho(\log_a^\psi(\alpha(a)))))$,
$$s(\iota(t)) \cong \delta\{\alpha(a) \mathbin{@}_a^\psi \varsigma; \rho\}(\overline{e}; \langle \Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Select}; e); \langle a^+.\delta(\xi(\rho(\log_a^\psi(\alpha(a))))))\rangle(s; t))$$
- *A family of relations* $\eta : \delta(\xi[a^*/a]) \leftrightarrow \delta(\xi[\tau/a])$
- $e \sim_{\delta(\alpha(\tau))} e\ [\theta]$ *where* $\theta : \delta(\xi[\tau/a]) \leftrightarrow \delta(\xi[\tau/a])$ *are defined as follows:* $\theta_b(e, e')$ *if and only if* $e \cong e'$ *and* $\eta_b(\mathsf{Refill}_b(\mathsf{Skeletonize}_b(e))(\iota), e)$ *holds for any* $b \in \Xi$, *for any* $\cdot;\cdot \vdash e : \delta(\varsigma(b))$, *and for any injective* $\cdot;\cdot \vdash \iota : \delta(\sigma(\rho(\psi(b)))) \to \delta(a^*)$ *satisfying the condition that for any* $\cdot;\cdot \vdash t : \delta(\sigma(\rho(\psi(b))))$, *we have* $s(\iota(t)) \cong \mathsf{Select}_b(e)(\langle a^+.\delta(\xi(\rho(\psi(b))))\rangle(s; t))$.[2]

Define $\cdot;\cdot \vdash e^- : \delta(\xi^-(\alpha^-(a^*)))$ *and* $\cdot;\cdot \vdash e^* : \delta(\sigma(\alpha(a)))$ *to be*

$$e^- \coloneqq \delta\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi^-\}(s; \langle \Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Skeletonize}; e))$$
$$e^* \coloneqq \delta\{\alpha(a) \uparrow_a^\psi \sigma; \rho\}(\iota; \langle \Xi^+.\delta(\alpha^-(a^*))\rangle(\mathsf{Refill}; e^-))$$

We have

$$e^* \sim_{\delta(\alpha(a))} e\ [\eta, a \mapsto s]$$

Proof. Prove by induction on the type structure of $\alpha(a)$.

- $\alpha(a) = a$.
By evaluation we get $e^- = \star$ and $e^* = \iota(\star)$. Then by the definition of the logical relation, it suffices to prove that $s(\iota(\star)) \cong e$.
Let t be $\star$ in the assumption of s, we get
$$s(\iota(\star)) \cong \delta\{a \mathbin{@}_a^\psi \varsigma; \rho\}(\overline{e}; \langle \Xi^+.a\rangle(\mathsf{Select}; e); \langle a^+.\delta(\xi(\rho(\log_a^\psi(a)))))\rangle(s; \star))$$
But the right-hand side is equal to e by definition. Thus we proved $s(\iota(\star)) \cong e$.
- $\alpha(a) = b \in \Xi$.
By definition we know
$$\begin{aligned} e^- &= \delta\{b \downarrow_a^\Xi a^*; \tau; \xi\}(s; \langle \Xi^+.b\rangle(\mathsf{Skeletonize}; e)) \\ &= \mathsf{Skeletonize}_b(e) \end{aligned}$$
and then
$$\begin{aligned} e^* &= \delta\{b \uparrow_a^\psi \sigma; \rho\}(\iota; \langle \Xi^+.b\rangle(\mathsf{Refill}; e^-)) \\ &= \langle \Xi^+.b\rangle(\mathsf{Refill}; e^-)(\iota) \\ &= \mathsf{Refill}_b(e^-)(\iota) \\ &= \mathsf{Refill}_b(\mathsf{Skeletonize}_b(e))(\iota) \end{aligned}$$

[2]This convoluted condition is essentially saying that, when $\alpha(a) = b \in \Xi$, the theorem is automatically true. It is in a sense self-recursive and is needed for inductive types. It is perhaps easier to check how this condition is used in the case where $\alpha(a) = b \in \Xi$ instead of attempting to understand it as a standalone statement.

We want to prove

$$e^* \sim_b e\ [\eta, a \mapsto s]$$

By the definition of the logical relation, it suffices to prove that

$$\eta_b(\mathsf{Refill}_b(\mathsf{Skeletonize}_b(e))(\iota), e)$$

By the assumption that $e \sim_b e\ [\theta]$ we get $\theta_b(e, e)$, which leads to our goal above if we can show that for any $\cdot;\cdot \vdash t : \delta(\sigma(\rho(\psi(b))))$,

$$s(\iota(t)) \cong \mathsf{Select}_b(e)(\langle a^+.\delta(\xi(\rho(\psi(b))))\rangle(s;t))$$

But from the assumption about s we know

$$s(\iota(t)) \cong \delta\{b\ @_a^\psi\ \varsigma;\rho\}(\overline{e};\langle\Xi^+.b\rangle(\mathsf{Select};e);\langle a^+.\delta(\xi(\rho(\psi(b))))\rangle(s;t))$$

We can see they are equivalent by simplifying the right-hand side.

- $\alpha(a) = b \in \Delta$.
  By definition $e^* = e$, so the result follows by parametricity.
- $\alpha(a) = \mathbb{1}$ or $\alpha(a) = \mathbb{0}$.
  The logical relation holds trivially by definition.
- $\alpha(a) = \alpha_1(a) \times \alpha_2(a)$.
  By definition we get $e^- = \langle e_1^-; e_2^-\rangle$ and $e^* = \langle e_1^*; e_2^*\rangle$ where

$$\begin{aligned}
e_1^- &\coloneqq \delta\{\alpha_1(a) \downarrow_a^\Xi a^*;\tau;\xi^-\}(s;\langle\Xi^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Skeletonize};\mathtt{fst}(e)))\\
e_2^- &\coloneqq \delta\{\alpha_2(a) \downarrow_a^\Xi a^*;\tau;\xi^-\}(s;\langle\Xi^+.\delta(\alpha_2(\tau))\rangle(\mathsf{Skeletonize};\mathtt{snd}(e)))\\
e_1^* &\coloneqq \delta\{\alpha_1(a) \uparrow_a^\psi \sigma;\rho\}(\iota\circ\mathtt{inl};\langle\Xi^+.\delta(\alpha_1^-(a^*))\rangle(\mathsf{Refill};e_1^-))\\
e_2^* &\coloneqq \delta\{\alpha_2(a) \uparrow_a^\psi \sigma;\rho\}(\iota\circ\mathtt{inr};\langle\Xi^+.\delta(\alpha_2^-(a^*))\rangle(\mathsf{Refill};e_2^-))
\end{aligned}$$

  We want to prove

$$e^* \sim_{\delta(\alpha_1(a)\times\alpha_2(a))} e\ [\eta, a \mapsto s]$$

  By the definition of the logical relation, it suffices to prove that

$$\begin{aligned}
e_1^* &\sim_{\delta(\alpha_1(a))} \mathtt{fst}(e)\ [\eta, a \mapsto s]\\
e_2^* &\sim_{\delta(\alpha_2(a))} \mathtt{snd}(e)\ [\eta, a \mapsto s]
\end{aligned}$$

  Now we use the inductive hypothesis on $\alpha_1(a)$ (then on $\alpha_2(a)$, similarly) by instantiating $\alpha(a)$ by $\alpha_1(a)$, e by $\mathtt{fst}(e)$, and ι by $\iota\circ\mathtt{inl}$.
  The condition $\mathtt{fst}(e) \sim_{\delta(\alpha_1(\tau))} \mathtt{fst}(e)\ [\theta]$ can be proved by $e \sim_{\delta(\alpha_1(\tau)\times\alpha_2(\tau))} e\ [\theta]$ and the definition of the logical relation. The function $\iota\circ\mathtt{inl}$ is injective because ι and $\mathtt{inl}$ are both injective. Now we only need to prove that for any $\cdot;\cdot \vdash t_1 : \delta(\sigma(\rho(\log_a^\psi(\alpha_1(a)))))$,

$$\begin{aligned}
s((\iota\circ\mathtt{inl})(t_1)) \cong \delta\{\alpha_1(a)\ @_a^\psi\ \varsigma;\rho\}(\overline{e};&\langle\Xi^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Select};\mathtt{fst}(e));\\
&\langle a^+.\delta(\xi(\rho(\log_a^\psi(\alpha_1(a))))))\rangle(s;t_1))
\end{aligned}$$

  By assumption we know that for any $\cdot;\cdot \vdash t : \delta(\sigma(\rho(\log_a^\psi(\alpha(a))))))$,

$$\begin{aligned}
s(\iota(t)) \cong \delta\{\alpha(a)\ @_a^\psi\ \varsigma;\rho\}(\overline{e};&\langle\Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Select};e);\\
&\langle a^+.\delta(\xi(\rho(\log_a^\psi(\alpha(a))))))\rangle(s;t))
\end{aligned}$$

  Let t be $\mathtt{inl}(t_1)$ and then it can be proved by definition.

- $\alpha(a) = \alpha_1(a) + \alpha_2(a)$.
  We consider two cases where $e \cong \mathtt{inl}(e_1)$ and $e \cong \mathtt{inr}(e_2)$.
  If $e \cong \mathtt{inl}(e_1)$, then by definition we have $e^- = \mathtt{inl}(e_1^-)$ and $e^* = \mathtt{inl}(e_1^*)$.

$$e_1^- \coloneqq \delta\{\alpha_1(a) \downarrow_a^\Xi a^*; \tau; \xi^-\}(s; \langle \Xi^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Skeletonize}; e_1))$$
$$e_1^* \coloneqq \delta\{\alpha_1(a) \uparrow_a^\psi \sigma; \rho\}(\iota \circ \mathtt{inl}; \langle \Xi^+.\delta(\alpha_1^-(a^*))\rangle(\mathsf{Refill}; e_1^-))$$

  So it is equivalent to prove

$$\mathtt{inl}(e_1^*) \sim_{\delta(\alpha_1(a)+\alpha_2(a))} \mathtt{inl}(e_1)\ [\eta, a \mapsto s]$$

  By the definition of the logical relation, it suffices to prove that

$$e_1^* \sim_{\delta(\alpha_1(a))} e_1\ [\eta, a \mapsto s]$$

  Now we can use the inductive hypothesis on $\alpha_1(a)$ by instantiating $\alpha(a)$ by $\alpha_1(a)$, e by e_1, and ι by $\iota \circ \mathtt{inl}$. The condition $e_1 \sim_{\delta(\alpha_1(\tau))} e_1\ [\theta]$ can be proved by $e \sim_{\delta(\alpha_1(\tau)+\alpha_2(\tau))} e\ [\theta]$ and the definition of the logical relation. The function $\iota \circ \mathtt{inl}$ is injective because ι and $\mathtt{inl}$ are both injective. Now we only need to prove that for any $\cdot; \cdot \vdash t_1 : \delta(\sigma(\rho(\log_a^\psi(\alpha_1(a)))))$,

$$s((\iota \circ \mathtt{inl})(t_1)) \cong \delta\{\alpha_1(a) @_a^\psi \varsigma; \rho\}(\overline{e}; \langle \Xi^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Select}; e_1); \langle a^+.\delta(\xi(\rho(\log_a^\psi(\alpha_1(a))))))\rangle(s; t_1))$$

  By assumption we know that for any $\cdot; \cdot \vdash t : \delta(\sigma(\rho(\log_a^\psi(\alpha(a)))))$,

$$s(\iota(t)) \cong \delta\{\alpha(a) @_a^\psi \varsigma; \rho\}(\overline{e}; \langle \Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Select}; e); \langle a^+.\delta(\xi(\rho(\log_a^\psi(\alpha(a))))))\rangle(s; t))$$

  Let t be $\mathtt{inl}(t_1)$ and then it can be proved by definition.
- $\alpha(a) = \alpha_1(a) \to \alpha_2(a)$.
  By the definition of the logical relation, it is equivalent to prove that for any $\cdot \vdash e_1 : \delta(\sigma(\alpha_1(a)))$ and $\cdot \vdash e_1' : \delta(\varsigma(\alpha_1(a)))$, if $e_1 \sim_{\delta(\alpha_1(a))} e_1'\ [\eta, a \mapsto s]$, then

$$e^*(e_1) \sim_{\delta(\alpha_2(a))} e(e_1')\ [\eta, a \mapsto s]$$

  Because $\Xi \in^{++} \alpha(a)$, we know $\Xi \notin \alpha_1(a)$. Thus $e_1 \sim_{\delta(\alpha_1(a))} e_1'\ [\eta, a \mapsto s]$ is equivalent to $e_1 \sim_{\delta(\alpha_1(a))} e_1'\ [a \mapsto s]$. By Lemma 7.3 we know

$$e_1 \sim_{\delta(\alpha_1(a))} \langle a^+.\delta(\alpha_1(a))\rangle(s; e_1)\ [a \mapsto s]$$

  And then by Lemma 8.1 we can prove that $\langle a^+.\delta(\alpha_1(a))\rangle(s; e_1) \cong e_1'$.
  By definition we have

$$\begin{aligned}
e^- &= \delta\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi^-\}(s; \langle \Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Skeletonize}; e)) \\
&= \delta\{\alpha(a) \downarrow_a^\Xi a^*; \tau; \xi^-\}(s; \lambda(x{:}\delta(\alpha_1(\tau))).\langle \Xi^+.\delta(\alpha_2(\tau))\rangle(\mathsf{Skeletonize}; e(x))) \\
&= \lambda(x{:}\delta(\alpha_1(a^*))).\delta\{\alpha_2(a) \downarrow_a^\Xi a^*; \tau; \xi^-\}(s; \langle \Xi^+.\delta(\alpha_2(\tau))\rangle(\mathsf{Skeletonize}; e(\langle a^+.\delta(\alpha_1(a))\rangle(s; x)))) \\
e^* &= \delta\{\alpha(a) \uparrow_a^\psi \sigma; \rho\}(\iota; \langle \Xi^+.\delta(\alpha^-(a^*))\rangle(\mathsf{Refill}; e^-)) \\
&= \delta\{\alpha(a) \uparrow_a^\psi \sigma; \rho\}(\iota; \lambda(x{:}\delta(\alpha_1(a^*))).\langle \Xi^+.\delta(\alpha_2^-(a^*))\rangle(\mathsf{Refill}; e^-(x))) \\
&= \lambda(x{:}\delta(\alpha_1(a^*))).\delta\{\alpha_2(a) \uparrow_a^\psi \sigma; \rho\}(\iota \circ \lambda(y{:}\delta(\sigma(\rho(\log_a^\psi(\alpha_2(a)))))).\langle x; y\rangle; \\
&\qquad \langle \Xi^+.\delta(\alpha_2^-(a^*))\rangle(\mathsf{Refill}; e^-(x)))
\end{aligned}$$

Since $\langle a^+.\delta(\alpha_1(a))\rangle(s;e_1) \cong e_1'$, we now have $e^*(e_1) = e_2^*$ where

$$\begin{aligned}
e_2^* &\coloneqq \delta\{\alpha_2(a) \uparrow_a^{\psi} \sigma;\rho\}(\iota';\langle\Xi^+.\delta(\alpha_2^-(a^*))\rangle(\mathsf{Refill};e_2^-))\\
e_2^- &\coloneqq \delta\{\alpha_2(a) \downarrow_a^{\Xi} a^*;\tau;\xi^-\}(s;\langle\Xi^+.\delta(\alpha_2(\tau))\rangle(\mathsf{Skeletonize};e(e_1')))\\
\iota' &\coloneqq \iota \circ \lambda(y{:}\delta(\sigma(\rho(\log_a^{\psi}(\alpha_2(a)))))).\langle e_1;y\rangle
\end{aligned}$$

Now we can use the inductive hypothesis on $\alpha_2(a)$ by instantiating $\alpha(a)$ by $\alpha_2(a)$, e by $e(e_1')$, and ι by ι'. The condition $e(e_1') \sim_{\delta(\alpha_2(\tau))} e(e_1')\ [\theta]$ can be proved by the definition of the logical relation, because $e \sim_{\delta(\alpha_1(\tau)\to\alpha_2(\tau))} e\ [\theta]$ and $e_1' \sim_{\delta(\alpha_1(\tau))} e_1'\ []$. The function ι' is injective because ι and $\lambda y.\langle e_1;y\rangle$ are both injective. Now we only need to prove that for any $\cdot;\cdot \vdash t_2 : \delta(\sigma(\rho(\log_a^{\psi}(\alpha_2(a)))))$,

$$\begin{aligned}
s(\iota'(t_2)) \cong \delta\{\alpha_2(a) @_a^{\psi} \varsigma;\rho\}(\overline{e};&\langle\Xi^+.\delta(\alpha_2(\tau))\rangle(\mathsf{Select};e(e_1'));\\
&\langle a^+.\delta(\xi(\rho(\log_a^{\psi}(\alpha_2(a))))))\rangle(s;t_2))
\end{aligned}$$

By assumption we know that for any $\cdot;\cdot \vdash t : \delta(\sigma(\rho(\log_a^{\psi}(\alpha(a)))))$,

$$\begin{aligned}
s(\iota(t)) \cong \delta\{\alpha(a) @_a^{\psi} \varsigma;\rho\}(\overline{e};&\langle\Xi^+.\delta(\alpha(\tau))\rangle(\mathsf{Select};e);\\
&\langle a^+.\delta(\xi(\rho(\log_a^{\psi}(\alpha(a))))))\rangle(s;t))
\end{aligned}$$

Let t be $\langle e_1;t_2\rangle$ and then it can be proved by definition.

- Case $\alpha(a) = \mu b.\alpha_1(a)$: Let the new types and type substitutions be:

$$\begin{aligned}
\hat{\Xi} &\coloneqq \Xi, b\\
\hat{\Xi}' &\coloneqq \Xi', b'\\
\hat{\psi} &\coloneqq \psi, b \mapsto b'\\
\hat{\xi} &\coloneqq \xi, b \mapsto \xi(\mu b.\alpha_1(a))\\
\hat{\xi}^- &\coloneqq \xi^-, b \mapsto \xi^-(\mu b.\alpha_1^-(\tau))\\
\hat{\rho} &\coloneqq \rho, b' \mapsto \rho(\log_a^{\psi}(\mu b.\alpha_1(a)))
\end{aligned}$$

And define some functions and families of functions:

$$\begin{aligned}
g_1 &\coloneqq \lambda(x{:}\mu b.\delta(\varsigma(\alpha_1(\tau)))).\langle\Xi^+.\mu b.\delta(\alpha_1(\tau))\rangle(\mathsf{Skeletonize};x)\\
g_2 &\coloneqq \lambda(x{:}\mu b.\delta(\xi^-(\alpha_1(\tau)))).\delta\{\mu b.\alpha_1(a) \downarrow_a^{\Xi} a^*;\tau;\xi^-\}(s;x)\\
g_3 &\coloneqq \lambda(x{:}\mu b.\delta(\xi^-(\alpha_1^-(a^*)))).\langle\Xi^+.\mu b.\delta(\alpha_1^-(a^*))\rangle(\mathsf{Refill};x)\\
g_4 &\coloneqq \lambda x.\lambda\iota.\delta\{\mu b.\alpha_1(a) \uparrow_a^{\psi} \sigma;\rho\}(\iota;x)\\
\mathsf{sel} &\coloneqq \lambda e.\lambda t.\delta\{\mu b.\alpha_1(a) @_a^{\psi} \varsigma;\rho\}(\overline{e};\langle\Xi^+.\mu b.\delta(\alpha(\tau))\rangle(\mathsf{Select};e);t)\\
\mathsf{Skeletonize}' &\coloneqq \mathsf{Skeletonize}, b \mapsto g_2 \circ g_1\\
\mathsf{Refill}' &\coloneqq \mathsf{Refill}, b \mapsto g_4 \circ g_3\\
\mathsf{Select}' &\coloneqq \mathsf{Select}, b \mapsto \mathsf{sel}
\end{aligned}$$

The goal is to prove

$$\mathsf{Refill}_b'(\mathsf{Skeletonize}_b'(e))(\iota) \sim_{\mu b.\delta(\alpha_1(a))} e\ [\eta, a \mapsto s]$$

Let $\mathcal{R}$ be a relation such that $\mathcal{R}(x,x)$ iff

$$\mathsf{Refill}_b'(\mathsf{Skeletonize}_b'(x))(\iota) \sim_{\mu b.\delta(\alpha_1(a))} x\ [\eta, a \mapsto s]$$

holds for any $\cdot;\cdot \vdash \iota : \delta(\hat{\rho}(b')) \to \delta(a^*)$, where ι satisfies that for any $\cdot;\cdot \vdash t : \delta(\hat{\sigma}(\hat{\rho}(b')))$,

$$s(\iota(t)) \cong \mathsf{Select}'_b(x)(\langle a^+.\delta(\hat{\xi}(\hat{\rho}(b')))\rangle(s;t))$$

By assumption we know that $e \sim_{\mu b.\delta(\alpha_1(a))} e\ [\theta]$. Then to prove $\mathcal{R}(e,e)$, by the definition of the logical relation of the inductive type, it suffices to prove that for any e_1 such that

$$e_1 \sim_{\delta(\alpha_1(a))} e_1\ [\theta, b \mapsto \mathcal{R}]$$

it always holds that $\mathcal{R}(\mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1), \mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1))$, *i.e.,* for any $\cdot;\cdot \vdash \iota : \delta(\hat{\rho}(b')) \to \delta(a^*)$ satisfying that for any $\cdot;\cdot \vdash t : \delta(\hat{\sigma}(\hat{\rho}(b')))$,

$$s(\iota(t)) \cong \mathsf{Select}'_b(\mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1))(\langle a^+.\delta(\hat{\xi}(\hat{\rho}(b')))\rangle(s;t))$$

we need to prove

$$\mathsf{Refill}'_b(\mathsf{Skeletonize}'_b(\mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1)))(\iota) \sim_{\mu b.\delta(\alpha_1(a))} \mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1)\ [\eta, a \mapsto s]$$

By evaluating the left hand side, it is equivalent to prove

$$\mathtt{roll}_{b.\delta(\sigma(\alpha_1(a)))}(e_1^*) \sim_{\mu b.\delta(\alpha_1(a))} \mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1)\ [\eta, a \mapsto s]$$

where

$$\begin{aligned} e_1^* &\coloneqq \delta\{\alpha_1(a) \uparrow_a^{\hat{\psi}} \hat{\sigma}; \hat{\rho}\}(\iota'; \langle \hat{\Xi}^+.\delta(\alpha_1^-(a))\rangle(\mathsf{Refill}'; e_1^-)) \\ e_1^- &\coloneqq \delta\{\alpha_1(a) \downarrow_a^{\hat{\Xi}} a^*; \tau; \hat{\xi}^-\}(s; \langle \hat{\Xi}^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Skeletonize}'; e_1)) \\ \iota' &\coloneqq \iota \circ \mathtt{roll}_{b'.\delta(\hat{\sigma}(\rho(\log_a^{\hat{\psi}}(\alpha_1(a)))))} \end{aligned}$$

Then by definition of the logical relation, it is equivalent to

$$e_1^* \sim_{\delta(\alpha_1(a))} e_1\ [\eta, a \mapsto s, b \mapsto - \sim_{\mu b.\delta(\alpha_1(a))} -\ [\eta, a \mapsto s]]$$

Let

$$\begin{aligned} \hat{\eta} &\coloneqq \eta, b \mapsto - \sim_{\mu b.\delta(\alpha_1(a))} -\ [\eta, a \mapsto s] \\ \hat{\theta} &\coloneqq \theta, b \mapsto \mathcal{R} \end{aligned}$$

Then we can use the inductive hypothesis on $\alpha_1(a)$. Let a, τ, a^*, $\overline{e}$, s be the same. Let Ξ, Ξ', ψ, ξ, ξ^-, ρ, Refill, Skeletonize, Select, θ, η, ι be the corresponding hatted versions. Let e be e_1. To instantiate the inductive hypothesis, we need to show that $\hat{\theta}_b$ (which is $\mathcal{R}$) matches its definition in the assumption. This can be verified by the definition of $\mathcal{R}$. We also need to show that for any $\cdot;\cdot \vdash t_1 : \delta(\hat{\sigma}(\hat{\rho}(\log_a^{\hat{\psi}}(\alpha_1(a)))))$,

$$s(\iota'(t_1)) \cong \delta\{\alpha_1(a)\ @_a^{\hat{\psi}}\ \hat{\varsigma}; \hat{\rho}\}(\overline{e}; \langle \hat{\Xi}^+.\delta(\alpha_1(\tau))\rangle(\mathsf{Select}'; e_1); \langle a^+.\delta(\hat{\xi}(\hat{\rho}(\log_a^{\hat{\psi}}(\alpha_1(a))))))\rangle(s; t_1))$$

But we have the assumption that for any $\cdot;\cdot \vdash t : \delta(\hat{\sigma}(\hat{\rho}(b')))$,

$$s(\iota(t)) \cong \mathsf{Select}'_b(\mathtt{roll}_{b.\delta(\varsigma(\alpha_1(a)))}(e_1))(\langle a^+.\delta(\hat{\xi}(\hat{\rho}(b')))\rangle(s;t))$$

Let t be $\mathtt{roll}_{b'.\delta(\hat{\sigma}(\rho(\log_a^{\hat{\psi}}(\alpha_1(a)))))}(t_1)$. Then the equivalence can be proved by unfolding the definitions.

□

REFERENCES

Alfred Tarski. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* 5, 2 (1955), 285 – 309. https://doi.org/10.2140/pjm.1955.5.285