

Fully Abstract Compilation via Universal Embedding

(Technical Appendix)

Max S. New
Northeastern University
maxnew@ccs.neu.edu

William J. Bowman
Northeastern University
wjb@williamjbowman.com

Amal Ahmed
Northeastern University
amal@ccs.neu.edu

Contents

1	Source Language	2
2	Target Language	4
3	Closure Conversion	7
4	Multi-Language	9
5	Multi-language Contexts and Contextual Equivalence	11
6	Multi-language Logical Relation	15
7	Multi-Language Logical Relation Corresponds to Contextual Equivalence	18
	7.1 Multi-Language Logical Relation: Fundamental Property	18
	7.2 Multi-Language Logical Relation Respects Contextual Equivalence	27
8	Back-Translation from Source to Target	29
9	Back Translation Correctness	35
10	Translation Correctness	45
	10.1 Semantics Preservation	45
	10.2 Full Abstraction	54

1 Source Language λ^S

<i>Types</i>	$\sigma ::= \alpha \mid \mathbf{1} \mid \sigma_1 + \sigma_2 \mid \sigma_1 \times \sigma_2 \mid \sigma_1 \rightarrow \sigma_2 \mid \mu\alpha. \sigma$
<i>Values</i>	$v ::= x \mid \langle \rangle \mid \text{inj}_1 v \mid \text{inj}_2 v \mid \langle v_1, v_2 \rangle \mid \lambda(x:\sigma).e \mid \text{fold}_{\mu\alpha.\sigma} v$
<i>Expressions</i>	$e ::= v \mid \text{case } v \text{ of } x_1.e_1 \mid x_2.e_2 \mid \pi_1 v \mid \pi_2 v v_1 v_2 \mid \text{unfold } v \mid \mid \text{let } x = e_1 \text{ in } e_2$
<i>Eval. Contexts</i>	$K ::= [\cdot] \mid \text{let } x = K \text{ in } e_2$

Figure 1: Source Language (STLC): Syntax

<i>Value Environment</i>	$\Gamma ::= \cdot \mid \Gamma, x : \sigma$
<i>Type Environment</i>	$\Delta ::= \cdot \mid \Delta, \alpha$

$\Delta \vdash \sigma$	$\frac{\alpha \in \Delta}{\Delta \vdash \alpha}$	$\frac{}{\Delta \vdash \mathbf{1}}$	$\frac{\Delta \vdash \sigma_1 \quad \Delta \vdash \sigma_2}{\Delta \vdash \sigma_1 + \sigma_2}$	$\frac{\Delta \vdash \sigma_1 \quad \Delta \vdash \sigma_2}{\Delta \vdash \sigma_1 \times \sigma_2}$	$\frac{\Delta \vdash \sigma_1 \quad \Delta \vdash \sigma_2}{\Delta \vdash \sigma_1 \rightarrow \sigma_2}$	$\frac{\Delta, \alpha \vdash \sigma}{\Delta \vdash \mu\alpha. \sigma}$
------------------------	--	-------------------------------------	---	--	---	--

$\Delta \vdash \Gamma$	$\frac{}{\Delta \vdash \cdot}$	$\frac{\Delta \vdash \Gamma \quad \Delta \vdash \sigma}{\Delta \vdash \Gamma, x : \sigma}$
------------------------	--------------------------------	--

$\Gamma \vdash e : \sigma$	$\frac{x : \sigma \in \Gamma \quad \cdot \vdash \Gamma}{\Gamma \vdash x : \sigma}$	$\frac{\cdot \vdash \Gamma}{\Gamma \vdash \langle \rangle : \mathbf{1}}$	$\frac{\Gamma \vdash e : \sigma_1 \quad \cdot \vdash \sigma_2}{\Gamma \vdash \text{inj}_1 e : \sigma_1 + \sigma_2}$	$\frac{\Gamma \vdash e : \sigma_2 \quad \cdot \vdash \sigma_1}{\Gamma \vdash \text{inj}_2 e : \sigma_1 + \sigma_2}$
	$\frac{\Gamma \vdash v : \sigma_1 + \sigma_2 \quad \Gamma, x_1 : \sigma_1 \vdash e_1 : \sigma \quad \Gamma, x_2 : \sigma_2 \vdash e_2 : \sigma}{\Gamma \vdash \text{case } v \text{ of } x_1.e_1 \mid x_2.e_2 : \sigma}$	$\frac{\Gamma \vdash v_1 : \sigma_1 \quad \Gamma \vdash v_2 : \sigma_2}{\Gamma \vdash \langle v_1, v_2 \rangle : \sigma_1 \times \sigma_2}$	$\frac{\Gamma \vdash v : \sigma_1 \times \sigma_2}{\Gamma \vdash \pi_1 : \sigma_1}$	
	$\frac{\Gamma \vdash v : \sigma_1 \times \sigma_2}{\Gamma \vdash \pi_2 : \sigma_2}$	$\frac{\Gamma, x : \sigma_1 \vdash e : \sigma_2}{\Gamma \vdash \lambda(x:\sigma_1).e : \sigma_1 \rightarrow \sigma_2}$	$\frac{\Gamma \vdash v_1 : \sigma_2 \rightarrow \sigma \quad \Gamma \vdash v_2 : \sigma_2}{\Gamma \vdash v_1 v_2 : \sigma}$	$\frac{\Gamma \vdash v : \sigma[\mu\alpha.\sigma/\alpha]}{\Gamma \vdash \text{fold}_{\mu\alpha.\sigma} v : \mu\alpha. \sigma}$
	$\frac{\Gamma \vdash v : \mu\alpha. \sigma}{\Gamma \vdash \text{unfold } v : \sigma[\mu\alpha.\sigma/\alpha]}$	$\frac{\Gamma \vdash e_1 : \sigma_1 \quad \Gamma, x : \sigma_1 \vdash e_2 : \sigma_2}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : \sigma_2}$		

Figure 2: Source Language (STLC): Static Semantics

$$\begin{array}{l}
\text{case } (\text{inj}_1 v) \text{ of } x_1. e_1 \mid x_2. e_2 \xrightarrow{S} e_1[v/x_1] \\
\text{case } (\text{inj}_2 v) \text{ of } x_1. e_1 \mid x_2. e_2 \xrightarrow{S} e_2[v/x_2] \\
\pi_1 \langle v_1, v_2 \rangle \xrightarrow{S} v_1 \\
\pi_2 \langle v_1, v_2 \rangle \xrightarrow{S} v_2 \\
(\lambda(x:\sigma). e) v \xrightarrow{S} e[v/x] \\
\text{unfold } (\text{fold}_{\mu\alpha.\sigma} v) \xrightarrow{S} v \\
\text{let } x = v \text{ in } e \xrightarrow{S} e[v/x] \\
\\
\frac{e \xrightarrow{S} e'}{K[e] \mapsto K[e']}
\end{array}$$

Figure 3: Source (λ^S): Operational Semantics

2 Target Language λ^T

<i>Value Types</i>	$\tau ::= \alpha \mid \tau_1 + \tau_2 \mid \langle \bar{\tau} \rangle \mid \forall[\alpha]. \tau \rightarrow \theta \mid \mu\alpha. \tau \mid \exists\alpha. \tau \mid \mathbf{0}$
<i>Computation Types</i>	$\theta ::= \mathbf{E} \tau_1 \tau_2$
<i>Values</i>	$v ::= x \mid \mathbf{inj}_1 v_1 \mid \mathbf{inj}_2 v_2 \mid \langle \bar{v} \rangle \mid \lambda[\alpha](x:\tau). e \mid \mathbf{fold}_{\mu\alpha. \tau} v \mid \mathbf{pack}(\tau, v) \mathbf{as} \exists\alpha. \tau'$
<i>Results</i>	$r ::= \mathbf{return} v \mid \mathbf{raise} v$
<i>Computations</i>	$e ::= r \mid v.i \mid \mathbf{unfold} v \mid \mathbf{handle} e \mathbf{with} (x. e_1) (y. e_2) \mid v_1 [\tau] v_2 \mid \mathbf{case} v \mathbf{of} x_1. e_1 \mid x_2. e_2 \mid \mathbf{unpack}(\alpha, x) = v \mathbf{in} e$
<i>Evaluation Contexts</i>	$\mathbf{K} ::= [\cdot] \mid \mathbf{handle} \mathbf{K} \mathbf{with} (x. e_1) (y. e_2)$

$$\boxed{e \xrightarrow{T} e'}$$

$$\begin{aligned}
& \mathbf{case}(\mathbf{inj}_1 v) \mathbf{of} x_1. e_1 \mid x_2. e_2 \xrightarrow{T} e_1[v/x_1] \\
& \mathbf{case}(\mathbf{inj}_2 v) \mathbf{of} x_1. e_1 \mid x_2. e_2 \xrightarrow{T} e_2[v/x_2] \\
& \langle v_1, \dots, v_n \rangle . i \xrightarrow{T} \mathbf{return} v_i \\
& (\lambda[\alpha](x:\tau). e) [\tau'] v \xrightarrow{T} e[\tau'/\alpha][v/x] \\
& \mathbf{unfold}(\mathbf{fold}_{\mu\alpha. \tau} v) \xrightarrow{T} \mathbf{return} v \\
& \mathbf{unpack}(\alpha, x) = (\mathbf{pack}(\tau, v) \mathbf{as} \exists\alpha. \tau) \mathbf{in} e \xrightarrow{T} e[\tau/\alpha][v/x] \\
& \mathbf{handle}(\mathbf{return} v) \mathbf{with} (x. e_1) (y. e_2) \xrightarrow{T} e_1[v/x] \\
& \mathbf{handle}(\mathbf{raise} v) \mathbf{with} (x. e_1) (y. e_2) \xrightarrow{T} e_2[v/y] \\
& \frac{e \xrightarrow{T} e'}{\mathbf{K}[e] \mapsto \mathbf{K}[e']}
\end{aligned}$$

Figure 4: Target Language (System F + exceptions): Syntax and Operational Semantics

Type Context $\Delta ::= \cdot \mid \Delta, \alpha$
Value Context $\Gamma ::= \cdot \mid \Gamma, x : \tau$

$\Delta \vdash \tau$

$$\frac{\alpha \in \Delta}{\Delta \vdash \alpha} \quad \frac{\Delta \vdash \tau_1 \quad \Delta \vdash \tau_2}{\Delta \vdash \tau_1 + \tau_2} \quad \frac{\Delta \vdash \tau_1 \cdots \Delta \vdash \tau_n}{\Delta \vdash \langle \tau_1, \dots, \tau_n \rangle} \quad \frac{\Delta, \alpha \vdash \tau_1 \quad \Delta, \alpha \vdash \tau_2}{\Delta \vdash \forall[\alpha]. \tau_1 \rightarrow \tau_2} \quad \frac{\Delta, \alpha \vdash \tau}{\Delta \vdash \mu\alpha. \tau} \quad \frac{}{\Delta \vdash 0}$$

$$\frac{\Delta, \alpha \vdash \tau}{\Delta \vdash \exists\alpha. \tau}$$

$\Delta \vdash \Gamma$

$$\frac{}{\Delta \vdash \cdot} \quad \frac{\Delta \vdash \Gamma \quad \Delta \vdash \tau}{\Delta \vdash \Gamma, x : \tau}$$

$\Delta; \Gamma \vdash v : \tau$

$$\frac{\Delta \vdash \Gamma \quad x : \tau \in \Gamma}{\Delta; \Gamma \vdash x : \tau} \quad \frac{\Delta; \Gamma \vdash v : \tau_1 \quad \Delta \vdash \tau_2}{\Delta; \Gamma \vdash \text{inj}_1 v : \tau_1 + \tau_2} \quad \frac{\Delta; \Gamma \vdash v : \tau_2 \quad \Delta \vdash \tau_1}{\Delta; \Gamma \vdash \text{inj}_2 v : \tau_1 + \tau_2} \quad \frac{\Delta; \Gamma \vdash v_i : \tau_i}{\Delta; \Gamma \vdash \langle \bar{v} \rangle : \langle \bar{\tau} \rangle}$$

$$\frac{\Delta \vdash \Gamma \quad \alpha; x : \tau \vdash e : \theta}{\Delta; \Gamma \vdash \lambda[\alpha](x : \tau). e : \forall[\alpha]. \tau \rightarrow \theta} \quad \frac{\Delta; \Gamma \vdash v : \tau[\mu\alpha. \tau/\alpha]}{\Delta; \Gamma \vdash \text{fold}_{\mu\alpha. \tau} v : \mu\alpha. \tau} \quad \frac{\Delta; \Gamma \vdash v : \tau[\tau'/\alpha] \quad \Delta \vdash \tau'}{\Delta; \Gamma \vdash \text{pack}(\tau', v) \text{ as } \exists\alpha. \tau : \exists\alpha. \tau}$$

$\Delta; \Gamma \vdash r : \theta$

$$\frac{\Delta; \Gamma \vdash v : \tau \quad \Delta \vdash \tau_{\text{exn}}}{\Delta; \Gamma \vdash \text{return } v : \mathbf{E} \tau_{\text{exn}} \tau} \quad \frac{\Delta; \Gamma \vdash v : \tau_{\text{exn}} \quad \Delta \vdash \tau}{\Delta; \Gamma \vdash \text{raise } v : \mathbf{E} \tau_{\text{exn}} \tau}$$

$\Delta; \Gamma \vdash e : \theta$

$$\frac{\Delta; \Gamma \vdash v : \tau_1 + \tau_2 \quad \Delta; \Gamma, x_1 : \tau_1 \vdash e_1 : \theta \quad \Delta; \Gamma, x_2 : \tau_2 \vdash e_2 : \theta}{\Delta; \Gamma \vdash \text{case } v \text{ of } x_1. e_1 \mid x_2. e_2 : \theta} \quad \frac{\Delta; \Gamma \vdash v : \langle \tau_1, \dots, \tau_n \rangle \quad \Delta \vdash \tau_{\text{exn}}}{\Delta; \Gamma \vdash v.i : \mathbf{E} \tau_{\text{exn}} \tau_i}$$

$$\frac{\Delta; \Gamma \vdash v_1 : \forall[\alpha]. \tau_2 \rightarrow \theta \quad \Delta \vdash \tau' \quad \Delta; \Gamma \vdash v_2 : \tau_2[\tau'/\alpha]}{\Delta; \Gamma \vdash v_1 [\tau'] v_2 : \theta[\tau'/\alpha]} \quad \frac{\Delta; \Gamma \vdash v : \mu\alpha. \tau \quad \Delta \vdash \tau_{\text{exn}}}{\Delta; \Gamma \vdash \text{unfold } v : \mathbf{E} \tau_{\text{exn}} (\tau[\mu\alpha. \tau/\alpha])}$$

$$\frac{\Delta; \Gamma \vdash v : \exists\alpha. \tau \quad \Delta, \alpha; \Gamma, x : \tau \vdash e : \theta}{\Delta; \Gamma \vdash \text{unpack}(\alpha, x) = v \text{ in } e : \theta}$$

$$\frac{\Delta; \Gamma \vdash e : \mathbf{E} \tau_{\text{exn}} \tau \quad \Delta; \Gamma, x : \tau \vdash e_1 : \theta \quad \Delta; \Gamma, y : \tau_{\text{exn}} \vdash e_2 : \theta}{\Delta; \Gamma \vdash \text{handle } e \text{ with } (x. e_1) (y. e_2) : \theta}$$

Figure 5: Target Language (System F): Static Semantics

let $x = e$ in e' $\stackrel{\text{def}}{=} \text{handle } e \text{ with } (x. e') (y. \text{raise } y)$
catch $y = e$ in e' $\stackrel{\text{def}}{=} \text{handle } e \text{ with } (x. \text{return } x) (y. e')$
 $1 \stackrel{\text{def}}{=} \langle \rangle$ (the empty tuple type)

Figure 6: Target Language (System F): Syntax Sugar

3 Closure Conversion

$$\begin{aligned}\alpha^+ &= \alpha \\ \mathbf{1}^+ &= \mathbf{1} \\ (\sigma_1 + \sigma_2)^+ &= \sigma_1^+ + \sigma_2^+ \\ (\sigma_1 \times \sigma_2)^+ &= \langle \sigma_1^+, \sigma_2^+ \rangle \\ (\sigma_1 \rightarrow \sigma_2)^+ &= \exists \alpha. \langle \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger, \alpha \rangle \\ (\mu \alpha. \sigma)^+ &= \mu \alpha. \sigma^+ \\ \sigma^\dagger &= \mathbf{E0} \sigma^+ \\ (\cdot)^+ &= \cdot \\ (\Gamma, \mathbf{x} : \sigma)^+ &= \Gamma^+, \mathbf{x} : \sigma^+\end{aligned}$$

Figure 7: Closure Conversion: Type Translation

$\boxed{\Gamma \vdash v : \sigma \rightsquigarrow_v v}$

$$\begin{array}{c}
\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma \rightsquigarrow_v x} \quad \frac{}{\Gamma \vdash \langle \rangle : 1 \rightsquigarrow_v \langle \rangle} \quad \frac{\Gamma \vdash v : \sigma_1 \rightsquigarrow_v v}{\Gamma \vdash \text{inj}_1 v : \sigma_1 + \sigma_2 \rightsquigarrow_v \text{inj}_1 v} \quad \frac{\Gamma \vdash v : \sigma_2 \rightsquigarrow_v v}{\Gamma \vdash \text{inj}_2 v : \sigma_1 + \sigma_2 \rightsquigarrow_v \text{inj}_2 v} \\
\\
\frac{\Gamma \vdash v_1 : \sigma_1 \rightsquigarrow_v v_1 \quad \Gamma \vdash v_2 : \sigma_2 \rightsquigarrow_v v_2}{\Gamma \vdash \langle v_1, v_2 \rangle : \sigma_1 \times \sigma_2 \rightsquigarrow_v \langle v_1, v_2 \rangle} \\
\\
\frac{\Gamma(y_i) = \sigma_i \quad \Gamma' = (y_1 : \sigma_1, \dots, y_n : \sigma_n) \quad \tau_{\text{env}} = \langle \sigma_1^+, \dots, \sigma_n^+ \rangle \quad \Gamma', x : \sigma \vdash e : \sigma' \rightsquigarrow_e e}{\Gamma \vdash \lambda(x : \sigma). e : \sigma \rightarrow \sigma' \rightsquigarrow_v \text{pack}(\tau_{\text{env}}, \langle \lambda(z : \langle \tau_{\text{env}}, \sigma^+ \rangle). \text{let } x_{\text{env}} = \text{return}_0 z.1 \text{ in} \\
\text{let } y_1 = \text{return}_0 x_{\text{env}}.1 \text{ in} \\
\vdots \\
\text{let } y_n = \text{return}_0 x_{\text{env}}.n \text{ in} \\
\text{let } x = \text{return}_0 z.2 \text{ in } e), \alpha)} \\
\\
\frac{\Gamma \vdash v : \sigma[\mu\alpha. \sigma/\alpha] \rightsquigarrow_v v}{\Gamma \vdash \text{fold}_{\mu\alpha. \sigma} v : \mu\alpha. \sigma \rightsquigarrow_v \text{fold}_{\mu\alpha. \sigma} v}
\end{array}$$

$\boxed{\Gamma \vdash e : \sigma \rightsquigarrow_e e}$

$$\begin{array}{c}
\frac{\Gamma \vdash v : \sigma \rightsquigarrow_v v}{\Gamma \vdash v : \sigma \rightsquigarrow_e \text{return}_0 v} \quad \frac{\Gamma \vdash v : \sigma_1 + \sigma_2 \rightsquigarrow_v v \quad \Gamma, x_1 : \sigma_1 \vdash e_1 : \sigma \rightsquigarrow_e e_1 \quad \Gamma, x_2 : \sigma_2 \vdash e_2 : \sigma \rightsquigarrow_e e_2}{\Gamma \vdash \text{case } v \text{ of } x_1. e_1 \mid x_2. e_2 : \sigma \rightsquigarrow_e \text{case } v \text{ of } x_1. e_1 \mid x_2. e_2} \\
\\
\frac{\Gamma \vdash v : \sigma_i \rightsquigarrow_v v}{\Gamma \vdash \pi_i v : \sigma_1 \times \sigma_2 \rightsquigarrow_e v.i} \quad \frac{\Gamma \vdash v_1 : \sigma_1 \rightarrow \sigma_2 \rightsquigarrow_v v_1 \quad \Gamma \vdash v_2 : \sigma_1 \rightsquigarrow_v v_2}{\Gamma \vdash v_1 v_2 : \sigma_2 \rightsquigarrow_e \text{unpack}(\alpha, z) = v_1 \text{ in} \\
\text{let } y_1 = \text{return } z.1 \text{ in} \\
\text{let } y_2 = \text{return } z.2 \text{ in} \\
y_1 \langle y_2, v_2 \rangle} \\
\\
\frac{\Gamma \vdash v : \mu\alpha. \sigma \rightsquigarrow_v v}{\Gamma \vdash \text{unfold } v : \sigma[\mu\alpha. \sigma/\alpha] \rightsquigarrow_e \text{return}_0 \text{unfold } v} \quad \frac{\Gamma \vdash e_1 : \sigma_1 \rightsquigarrow_e e_1 \quad \Gamma, x \vdash e_2 : \sigma_2 \rightsquigarrow_e e_2}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : \sigma_2 \rightsquigarrow_e \text{let } x = e_1 \text{ in } e_2}
\end{array}$$

Figure 8: Closure Conversion: Term Translation

4 Combined Language λ^{ST}

<i>Environments</i>	$\Gamma ::= \cdot \mid \Gamma, x : \sigma \mid \Gamma, \mathbf{x} : \boldsymbol{\tau}$
	$\Delta ::= \boldsymbol{\Delta}$
<i>Value Types</i>	$\tau ::= \sigma \mid \boldsymbol{\tau}$
<i>Computation Types</i>	$\theta ::= \sigma \mid \boldsymbol{\theta}$
<i>All Types</i>	$\varphi ::= \tau \mid \boldsymbol{\theta}$
<i>Variables</i>	$x ::= x \mid \mathbf{x}$
<i>Values</i>	$v ::= v \mid \mathbf{v}$
<i>Results</i>	$r ::= v \mid \mathbf{r}$
<i>Expressions</i>	$e ::= \dots \mid {}^{\sigma}ST \mathbf{e}$
	$\mathbf{e} ::= \dots \mid \mathcal{TS}^{\sigma} \mathbf{e}$
	$e ::= e \mid \mathbf{e}$
<i>Evaluation Contexts</i>	$K ::= \dots \mid {}^{\sigma}ST \mathbf{K}$
	$\mathbf{K} ::= \dots \mid \mathcal{TS}^{\sigma} \mathbf{K}$
	$K ::= K \mid \mathbf{K}$

Figure 9: Combined Language (λ^{ST}): Syntax

The syntax of the multi-language is defined by embedding the source and target syntax. Meta-variables defined by \dots indicate using the definitions from the corresponding source or target meta-variable. For instance, \mathbf{p} in the multi-language is exactly \mathbf{p} from the target language. However, \mathbf{e} in the multi-language is \mathbf{e} from the target language extended with a boundary term.

Typing in the multi-language, $\Delta; \Gamma \vdash e : \theta$, consists of the typing judgments from both the source and the target languages, with a few modifications. First, the judgments are modified to take the multi-language typing environments Δ and Γ instead of only the source or target typing environments. Next, the typing judgment for the source language is modified at the leaves of each derivation to check that $\Delta \vdash \Gamma$. Finally, two new rules are added to type-check boundary terms, given in Figure 11.

$$\boxed{e \xrightarrow{\text{ST}} e'}$$

$$\begin{array}{l}
{}^1\text{ST return } v \xrightarrow{\text{ST}} \langle \rangle \\
\sigma_1 + \sigma_2 \text{ST return inj}_i v \xrightarrow{\text{ST}} \text{let } x = {}^{\sigma_i} \text{ST return } v \text{ in inj}_i x \\
\sigma_1 \times \sigma_2 \text{ST return } v \xrightarrow{\text{ST}} \text{let } x_1 = {}^{\sigma_1} \text{ST } v.1 \text{ in let } x_2 = {}^{\sigma_2} \text{ST } v.2 \text{ in } \langle x_1, x_2 \rangle \\
\sigma_1 \rightarrow \sigma_2 \text{ST return } v \xrightarrow{\text{ST}} \lambda(x : \sigma_1). {}^{\sigma_2} \text{ST} \left(\begin{array}{l} \text{unpack } (\alpha, z) = v \text{ in let } x_f = z.1 \text{ in} \\ \text{let } x_{\text{env}} = z.2 \text{ in} \\ \text{let } x = \text{TS}^{\sigma_1} x \text{ in } x_f [\alpha] \langle x_{\text{env}}, x \rangle \end{array} \right) \\
\mu\alpha.\sigma \text{ST return } v \xrightarrow{\text{ST}} \text{let } x = {}^{\sigma[\mu\alpha.\sigma/\alpha]} \text{ST unfold } v \text{ in fold}_{\mu\alpha.\sigma} x \\
\text{TS}^1 v \xrightarrow{\text{ST}} \text{return } \langle \rangle \\
\text{TS}^{\sigma_1 + \sigma_2} \text{inj}_i v \xrightarrow{\text{ST}} \text{let } x = \text{TS}^{\sigma_i} v \text{ in return inj}_i x \\
\text{TS}^{\sigma_1 \times \sigma_2} v \xrightarrow{\text{ST}} \text{let } x_1 = \text{TS}^{\sigma_1} \pi_1 v \text{ in let } x_2 = \text{TS}^{\sigma_2} \pi_2 v \text{ in return } \langle x_1, x_2 \rangle \\
\text{TS}^{\sigma_1 \rightarrow \sigma_2} v \xrightarrow{\text{ST}} \text{return pack } (1, \langle \lambda(z : \langle 1, \sigma_1^+ \rangle). \\ \text{TS}^{\sigma_2} \left(\begin{array}{l} \text{let } x = {}^{\sigma_1} \text{ST } z.2 \text{ in} \\ v \ x \end{array} \right) \\ \rangle, \langle \rangle \rangle) \text{ as } (\sigma_1 \rightarrow \sigma_2)^+ \\
\text{TS}^{\mu\alpha.\sigma} v \xrightarrow{\text{ST}} \text{let } x = \text{TS}^{\sigma[\mu\alpha.\sigma/\alpha]} \text{unfold } v \text{ in return fold}_{(\mu\alpha.\sigma)^+} x \\
\frac{e \xrightarrow{S} e'}{K[e] \mapsto K[e']} \qquad \frac{e \xrightarrow{T} e'}{K[e] \mapsto K[e']} \qquad \frac{e \xrightarrow{\text{ST}} e'}{K[e] \mapsto K[e']}
\end{array}$$

Figure 10: Combined Language (λ^{ST}): Operational Semantics

$$\boxed{\Delta; \Gamma \vdash e : \theta}$$

$$\frac{\Delta; \Gamma \vdash e : \sigma^{\ddagger}}{\Delta; \Gamma \vdash {}^{\sigma} \text{ST } e : \sigma} \qquad \frac{\Delta; \Gamma \vdash e : \sigma}{\Delta; \Gamma \vdash \text{TS}^{\sigma} e : \sigma^{\ddagger}}$$

Figure 11: Combined Language (λ^{ST}): Static Semantics

5 λ^{ST} Contexts and Contextual Equivalence

$\mathbf{C}^\vee ::= [\cdot]^\vee \mid \text{inj}_i \mathbf{C}^\vee \mid \langle v, \mathbf{C}^\vee \rangle \mid \langle \mathbf{C}^\vee, v \rangle \mid \pi_i \mathbf{C}^\vee \mid \lambda(x:\sigma). \mathbf{C} \mid \text{fold}_{\mu\alpha.\sigma} \mathbf{C}^\vee$
 $\mathbf{C} ::= [\cdot] \mid [\cdot]^\vee \mid \text{case } \mathbf{C}^\vee \text{ of } x_1. e_1 \mid x_2. e_2 \mid \text{case } v \text{ of } x_1. \mathbf{C} \mid x_2. e_2$
 $\quad \mid \text{case } v \text{ of } x_1. e_1 \mid x_2. \mathbf{C} \mid \mathbf{C}^\vee v_2 \mid v_1 \mathbf{C}^\vee \mid \text{unfold } \mathbf{C}^\vee \mid \text{let } x = \mathbf{C} \text{ in } e_2 \mid \text{let } x = e_1 \text{ in } \mathbf{C} \mid {}^\sigma \mathcal{ST} \mathbf{C}$
 $\mathbf{C}^\vee ::= [\cdot]^\vee \mid \text{inj}_i \mathbf{C}^\vee \mid \langle v_1, \dots, \mathbf{C}^\vee, \dots, v_n \rangle \mid \lambda[\alpha](x:\tau). \mathbf{C} \mid \text{fold}_{\mu\alpha.\tau} \mathbf{C}^\vee \mid \text{pack}(\tau, \mathbf{C}^\vee) \text{ as } \exists\alpha. \tau$
 $\mathbf{C} ::= [\cdot] \mid \text{return } \mathbf{C}^\vee \mid \text{raise } \mathbf{C}^\vee \mid \text{case } \mathbf{C}^\vee \text{ of } x_1. e_1 \mid x_2. e_2 \mid \text{case } v \text{ of } x_1. \mathbf{C} \mid x_2. e_2 \mid \text{case } v \text{ of } x_1. e_1 \mid x_2. \mathbf{C}$
 $\quad \mid \mathbf{C}^\vee.i \mid \mathbf{C}^\vee [\tau] v_2 \mid v_1 [\tau] \mathbf{C}^\vee \mid \text{unfold } \mathbf{C}^\vee \mid \text{unpack}(\alpha, x) = \mathbf{C}^\vee \text{ in } e \mid \text{unpack}(\alpha, x) = v \text{ in } \mathbf{C}$
 $\quad \mid \text{handle } \mathbf{C} \text{ with } (x. e_1) (y. e_2) \text{ handle } e \text{ with } (x. \mathbf{C}) (y. e_2) \mid \text{handle } e \text{ with } (x. e_1) (y. \mathbf{C}) \mid \mathcal{TS}^\sigma \mathbf{C}$
 $\mathbf{C}^\xi ::= \mathbf{C}^\vee \mid \mathbf{C}$
 $\mathbf{C}^\xi ::= \mathbf{C}^\vee \mid \mathbf{C}$
 $\mathcal{C} ::= \mathbf{C}^\xi \mid \mathbf{C}^\xi$

$\boxed{\vdash \mathcal{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \varphi')}$

$$\frac{\Delta \vdash \Gamma}{\vdash [\cdot] : (\Delta; \Gamma \vdash \sigma) \Rightarrow (\Delta; \Gamma \vdash \sigma)} \quad \frac{\Delta \vdash \Gamma}{\vdash [\cdot]^\vee : (\Delta; \Gamma \vdash \sigma) \Rightarrow (\Delta; \Gamma \vdash \sigma)} \quad \frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_i)}{\vdash \text{inj}_i \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 + \sigma_2)}$$

$$\frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 + \sigma_2) \quad \Gamma', x_1 : \sigma_1 \vdash e_1 : \sigma \quad \Gamma', x_2 : \sigma_2 \vdash e_2 : \sigma}{\vdash \text{case } \mathbf{C}^\vee \text{ of } x_1. e_1 \mid x_2. e_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma)}$$

$$\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', x_1 : \sigma_1 \vdash \sigma) \quad \Gamma' \vdash v : \sigma_1 + \sigma_2 \quad \Gamma', x_2 : \sigma_2 \vdash e_2 : \sigma}{\vdash \text{case } v \text{ of } x_1. \mathbf{C} \mid x_2. e_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma)}$$

$$\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', x_2 : \sigma_2 \vdash \sigma) \quad \Gamma' \vdash v : \sigma_1 + \sigma_2 \quad \Gamma', x_1 : \sigma_1 \vdash e_1 : \sigma}{\vdash \text{case } v \text{ of } x_1. e_1 \mid x_2. \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma)}$$

$$\frac{\Delta' \vdash \Gamma' : v\sigma_1 \quad \vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_2)}{\vdash \langle v, \mathbf{C}^\vee \rangle : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \times \sigma_2)}$$

$$\frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1) \quad \Delta' \vdash \Gamma' : v\sigma_2 \vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \times \sigma_2)}{\vdash \langle \mathbf{C}^\vee, v \rangle : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \times \sigma_2)} \quad \frac{\Delta' \vdash \Gamma' : v\sigma_2 \vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \times \sigma_2)}{\vdash \pi_i \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_i)}$$

$$\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', x : \sigma_1 \vdash \sigma_2)}{\vdash \lambda(x:\sigma_1). \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \rightarrow \sigma_2)} \quad \frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1 \rightarrow \sigma_2) \quad \Delta'; \Gamma' \vdash v_2 : \sigma_1}{\vdash \mathbf{C}^\vee v_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_2)}$$

$$\frac{\Delta'; \Gamma' \vdash v_1 : \sigma_1 \rightarrow \sigma_2 \quad \vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1)}{\vdash v_1 \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_2)} \quad \frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma'[\mu\alpha.\sigma'/\alpha])}{\vdash \text{fold}_{\mu\alpha.\sigma'} \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mu\alpha.\sigma')}$$

$$\frac{\vdash \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mu\alpha.\sigma')}{\vdash \text{unfold } \mathbf{C}^\vee : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma'[\mu\alpha.\sigma'/\alpha])} \quad \frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_1) \quad \Delta'; \Gamma', x : \sigma_1 \vdash e_2 : \sigma_2}{\vdash \text{let } x = \mathbf{C} \text{ in } e_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_2)}$$

$$\frac{\Delta'; \Gamma' \vdash e_1 : \sigma_1 \quad \vdash \mathbf{C} : (\Delta; \Gamma, x : \sigma_1 \vdash \varphi) \Rightarrow (\Delta'; \Gamma', x : \sigma_1 \vdash \sigma_2)}{\vdash \text{let } x = e_1 \text{ in } \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma_2)} \quad \frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma^\dagger)}{\vdash {}^\sigma \mathcal{ST} \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma)}$$

Figure 12: λ^{ST} Contexts and Context Typing

$$\begin{array}{c}
\frac{\Delta \vdash \Gamma}{\vdash [\cdot]^{\mathbf{V}} : (\Delta; \Gamma \vdash \boldsymbol{\tau}) \Rightarrow (\Delta; \Gamma \vdash \boldsymbol{\tau})} \qquad \frac{\Delta \vdash \Gamma}{\vdash [\cdot] : (\Delta; \Gamma \vdash \boldsymbol{\theta}) \Rightarrow (\Delta; \Gamma \vdash \boldsymbol{\theta})} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau})}{\vdash \mathbf{return}_{\boldsymbol{\tau}_{\text{exn}}} \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \boldsymbol{\tau} \boldsymbol{\tau}_{\text{exn}})} \qquad \frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}_{\text{exn}})}{\vdash \mathbf{raise}_{\boldsymbol{\tau}} \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \boldsymbol{\tau} \boldsymbol{\tau}_{\text{exn}})} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}_i)}{\vdash \mathbf{inj}_i \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2)} \\
\frac{\Delta'; \Gamma' \vdash \mathbf{v}_i : \boldsymbol{\tau}_i \quad \vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau})}{\vdash \langle \mathbf{v}_1, \dots, \mathbf{C}^{\mathbf{V}}, \dots, \mathbf{v}_n \rangle : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \langle \boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}, \dots, \boldsymbol{\tau}_n \rangle)} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \langle \boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_i, \dots, \boldsymbol{\tau}_n \rangle) \quad \Delta' \vdash \boldsymbol{\tau}_{\text{exn}}}{\vdash \mathbf{C}^{\mathbf{V}}.i : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \boldsymbol{\tau}_i \boldsymbol{\tau}_{\text{exn}})} \\
\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\boldsymbol{\alpha}; \mathbf{x} : \boldsymbol{\tau} \vdash \boldsymbol{\theta})}{\vdash \boldsymbol{\lambda}[\boldsymbol{\alpha}](\mathbf{x} : \boldsymbol{\tau}). \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \forall[\boldsymbol{\alpha}]. \boldsymbol{\tau} \rightarrow \boldsymbol{\theta})} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \forall[\boldsymbol{\alpha}]. \boldsymbol{\tau} \rightarrow \boldsymbol{\theta}) \quad \Delta'; \Gamma' \vdash \mathbf{v}_2 : \boldsymbol{\tau}[\boldsymbol{\tau}'/\boldsymbol{\alpha}] \quad \Delta' \vdash \boldsymbol{\tau}'}{\vdash \mathbf{C}^{\mathbf{V}} [\boldsymbol{\tau}'] \mathbf{v}_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\theta}[\boldsymbol{\tau}'/\boldsymbol{\alpha}])} \\
\frac{\Delta'; \Gamma' \vdash \mathbf{v}_1 : \forall[\boldsymbol{\alpha}]. \boldsymbol{\tau} \rightarrow \boldsymbol{\theta} \quad \vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}[\boldsymbol{\tau}'/\boldsymbol{\alpha}]) \quad \Delta' \vdash \boldsymbol{\tau}'}{\vdash \mathbf{v}_1 [\boldsymbol{\tau}'] \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\theta}[\boldsymbol{\tau}'/\boldsymbol{\alpha}])} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\alpha}])}{\vdash \mathbf{fold}_{\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}} \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau})} \qquad \frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}) \quad \Delta' \vdash \boldsymbol{\tau}_{\text{exn}}}{\vdash \mathbf{unfold} \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \boldsymbol{\tau}_{\text{exn}} \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\alpha}])} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\tau}_1[\boldsymbol{\tau}_2/\boldsymbol{\alpha}])}{\vdash \mathbf{pack}(\boldsymbol{\tau}_2, \mathbf{C}^{\mathbf{V}}) \mathbf{as} \exists \boldsymbol{\alpha}. \boldsymbol{\tau}_1 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \exists \boldsymbol{\alpha}. \boldsymbol{\tau}_1)} \\
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \exists \boldsymbol{\alpha}. \boldsymbol{\tau}) \quad \Delta', \boldsymbol{\alpha}; \Gamma', \mathbf{x} : \boldsymbol{\tau} \vdash \mathbf{e} : \boldsymbol{\theta}}{\vdash \mathbf{unpack}(\boldsymbol{\alpha}, \mathbf{x}) = \mathbf{C}^{\mathbf{V}} \mathbf{in} \mathbf{e} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\theta})} \\
\frac{\Delta'; \Gamma' \vdash \mathbf{v} : \exists \boldsymbol{\alpha}. \boldsymbol{\tau} \quad \vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta', \boldsymbol{\alpha}; \Gamma', \mathbf{x} : \boldsymbol{\tau} \vdash \boldsymbol{\theta})}{\vdash \mathbf{unpack}(\boldsymbol{\alpha}, \mathbf{x}) = \mathbf{v} \mathbf{in} \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \boldsymbol{\theta})}
\end{array}$$

Figure 13: λ^{ST} Context Typing (continued)

$$\begin{array}{c}
\frac{\vdash \mathbf{C}^{\mathbf{V}} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \tau_1 + \tau_2) \quad \Delta'; \Gamma', \mathbf{x}_1 : \tau_1 \vdash \mathbf{e}_1 : \theta \quad \Delta'; \Gamma', \mathbf{x}_2 : \tau_2 \vdash \mathbf{e}_2 : \theta}{\vdash \text{case } \mathbf{C}^{\mathbf{V}} \text{ of } \mathbf{x}_1. \mathbf{e}_1 \mid \mathbf{x}_2. \mathbf{e}_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \theta)} \\
\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', \mathbf{x}_1 : \tau_1 \vdash \theta) \quad \Delta'; \Gamma' \vdash \mathbf{v} : \tau_1 + \tau_2 \quad \Delta'; \Gamma', \mathbf{x}_2 : \tau_2 \vdash \mathbf{e}_2 : \theta}{\vdash \text{case } \mathbf{v} \text{ of } \mathbf{x}_1. \mathbf{C} \mid \mathbf{x}_2. \mathbf{e}_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \theta)} \\
\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', \mathbf{x}_2 : \tau_2 \vdash \theta) \quad \Delta'; \Gamma' \vdash \mathbf{v} : \tau_1 + \tau_2 \quad \Delta'; \Gamma', \mathbf{x}_1 : \tau_1 \vdash \mathbf{e}_1 : \theta}{\vdash \text{case } \mathbf{v} \text{ of } \mathbf{x}_1. \mathbf{e}_1 \mid \mathbf{x}_2. \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \theta)} \\
\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \tau'_{\text{exn}} \tau') \quad \Delta'; \Gamma', x : \tau' \vdash \mathbf{e}_1 : \mathbf{E} \tau_{\text{exn}} \tau \quad \Delta'; \Gamma', y : \tau'_{\text{exn}} \vdash \mathbf{e}_2 : \mathbf{E} \tau_{\text{exn}} \tau}{\vdash \text{handle } \mathbf{C} \text{ with } (\mathbf{x}. \mathbf{e}_1) (\mathbf{y}. \mathbf{e}_2) : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \tau_{\text{exn}} \tau)} \\
\frac{\Delta'; \Gamma' \vdash \mathbf{e} : \mathbf{E} \tau'_{\text{exn}} \tau' \quad \vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', \mathbf{x} : \tau' \vdash \mathbf{E} \tau_{\text{exn}} \tau) \quad \Delta'; \Gamma', \mathbf{y} : \tau'_{\text{exn}} \vdash \mathbf{e}_2 : \mathbf{E} \tau_{\text{exn}} \tau}{\vdash \text{handle } \mathbf{e} \text{ with } (\mathbf{x}. \mathbf{C}) (\mathbf{y}. \mathbf{e}_2) : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \tau_{\text{exn}} \tau)} \\
\frac{\Delta'; \Gamma' \vdash \mathbf{e} : \mathbf{E} \tau'_{\text{exn}} \tau' \quad \Delta'; \Gamma', \mathbf{x} : \tau' \vdash \mathbf{e}_1 : \mathbf{E} \tau_{\text{exn}} \tau \quad \vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma', \mathbf{y} : \tau'_{\text{exn}} \vdash \mathbf{E} \tau_{\text{exn}} \tau)}{\vdash \text{handle } \mathbf{e} \text{ with } (\mathbf{x}. \mathbf{e}_1) (\mathbf{y}. \mathbf{C}) : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \mathbf{E} \tau_{\text{exn}} \tau)} \\
\frac{\vdash \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \sigma')}{\vdash \mathcal{TS}^{\sigma'} \mathbf{C} : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash (\sigma')^+)}
\end{array}$$

Figure 14: λ^{ST} Context Typing (continued, continued)

$$\begin{array}{l}
\Delta \vDash \delta \stackrel{\text{def}}{=} \forall \alpha \in \Delta. \Delta \vdash \delta(\alpha) \\
\delta, \Gamma \vDash \gamma \stackrel{\text{def}}{=} \forall x : \tau \in \Gamma. \cdot \vdash \gamma(x) : \delta(\tau) \\
\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{\text{ciu}} e_2 : \theta \stackrel{\text{def}}{=} \Delta; \Gamma \vdash e_1 : \theta \wedge \Delta; \Gamma \vdash e_2 : \theta \wedge \\
\forall \delta, \gamma, K. (\Delta \vDash \delta \wedge \delta, \Gamma \vDash \gamma \wedge \vdash K : (\cdot \vdash \theta) \Rightarrow (\cdot \vdash \mathbf{1})) \implies \\
(K[\delta(\gamma(e_1))] \Downarrow K[\delta(\gamma(e_2))])
\end{array}$$

Figure 15: CIU Equivalence

$$\begin{array}{l}
\Gamma \vdash e_1 \approx_5^{\text{ctx}} e_2 : \sigma \stackrel{\text{def}}{=} \Gamma \vdash e_1 : \sigma \wedge \Gamma \vdash e_2 : \sigma \wedge \\
\forall \mathbf{C}. \text{source } \mathbf{C} \wedge \vdash \mathbf{C} : (\cdot; \Gamma \vdash \sigma) \Rightarrow (\cdot; \cdot \vdash \mathbf{1}) \\
\implies (\mathbf{C}[e_1] \Downarrow \mathbf{C}[e_2]) \\
\Delta; \Gamma \vdash e_1 \approx_{\mathbf{T}}^{\text{ctx}} e_2 : \theta \stackrel{\text{def}}{=} \Delta; \Gamma \vdash e_1 : \theta \wedge \Delta; \Gamma \vdash e_2 : \theta \wedge \\
\forall \mathbf{C}. \text{target } \mathbf{C} \wedge \vdash \mathbf{C} : (\Delta; \Gamma \vdash \theta) \Rightarrow (\cdot; \cdot \vdash \mathbf{E} \mathbf{0} \mathbf{1}) \\
\implies (\mathbf{C}[e_1] \Downarrow \mathbf{C}[e_2]) \\
\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{\text{ctx}} e_2 : \theta \stackrel{\text{def}}{=} \Delta; \Gamma \vdash e_1 : \theta \wedge \Delta; \Gamma \vdash e_2 : \theta \wedge \\
\forall \mathbf{C}. \vdash \mathbf{C} : (\Delta; \Gamma \vdash \theta) \Rightarrow (\cdot; \cdot \vdash \mathbf{1}) \\
\implies (\mathbf{C}[e_1] \Downarrow \mathbf{C}[e_2])
\end{array}$$

Figure 16: Source, Target and Multi-language Contextual Equivalence

6 λ^{ST} Logical Relation

$$\begin{aligned}
\text{running}(k, e) &\stackrel{\text{def}}{=} \exists e'. e \mapsto^k e' \\
\text{Atom}[\varphi_1, \varphi_2] &\stackrel{\text{def}}{=} \{ (k, e_1, e_2) \mid k \in \mathbb{N} \wedge \cdot; \cdot \vdash e_1 : \varphi_1 \wedge \cdot; \cdot \vdash e_2 : \varphi_2 \} \\
\text{Atom}[\varphi]\rho &\stackrel{\text{def}}{=} \text{Atom}[\rho_1(\varphi), \rho_2(\varphi)] \\
\text{Atom}^{\text{val}}[\tau_1, \tau_2] &\stackrel{\text{def}}{=} \{ (k, v_1, v_2) \mid (k, v_1, v_2) \in \text{Atom}[\tau_1, \tau_2] \} \\
\text{Atom}^{\text{val}}[\tau]\rho &\stackrel{\text{def}}{=} \text{Atom}^{\text{val}}[\rho_1(\tau), \rho_2(\tau)] \\
\text{Atom}^{\text{res}}[\theta_1, \theta_2] &\stackrel{\text{def}}{=} \{ (k, r_1, r_2) \mid (k, r_1, r_2) \in \text{Atom}[\theta_1, \theta_2] \} \\
\text{Atom}^{\text{res}}[\theta]\rho &\stackrel{\text{def}}{=} \text{Atom}^{\text{res}}[\rho_1(\theta), \rho_2(\theta)] \\
\text{Atom}^{\mathcal{K}}[\theta_1, \theta_2] &\stackrel{\text{def}}{=} \{ (k, K_1, K_2) \mid k \in \mathbb{N} \wedge \exists \theta'_1, \theta'_2. \vdash K_1 : (\cdot; \cdot \vdash \theta_1) \Rightarrow (\cdot; \cdot \vdash \theta'_1) \wedge \vdash K_2 : (\cdot; \cdot \vdash \theta_2) \Rightarrow (\cdot; \cdot \vdash \theta'_2) \} \\
\text{Atom}^{\mathcal{K}}[\theta]\rho &\stackrel{\text{def}}{=} \text{Atom}^{\mathcal{K}}[\rho_1(\theta), \rho_2(\theta)] \\
\text{Rel}[\boldsymbol{\tau}_1, \boldsymbol{\tau}_2] &\stackrel{\text{def}}{=} \{ R \in \mathcal{P}(\text{Atom}^{\text{val}}[\boldsymbol{\tau}_1, \boldsymbol{\tau}_2]) \mid \forall (k, \mathbf{v}_1, \mathbf{v}_2) \in R. \forall j < k. (j, \mathbf{v}_1, \mathbf{v}_2) \in R \}
\end{aligned}$$

Figure 17: Logical Relation Auxiliary Definitions

$$\begin{aligned}
\mathcal{V}[\tau]\rho &\subset \text{Atom}^{\text{val}}[\tau]\rho \\
\mathcal{V}[\mathbf{1}]\rho &\stackrel{\text{def}}{=} \{(k, \langle \rangle, \langle \rangle)\} \\
\mathcal{V}[\sigma_1 + \sigma_2]\rho &\stackrel{\text{def}}{=} \{(k, \text{inj}_i v_1, \text{inj}_i v_2) \mid i \in \{1, 2\} \wedge (k, v_1, v_2) \in \mathcal{V}[\sigma_i]\rho\} \\
\mathcal{V}[\sigma \times \sigma']\rho &\stackrel{\text{def}}{=} \{(k, \langle v_1, v'_1 \rangle, \langle v_2, v'_2 \rangle) \mid (k, v_1, v_2) \in \mathcal{V}[\sigma]\rho \wedge (k, v'_1, v'_2) \in \mathcal{V}[\sigma']\rho\} \\
\mathcal{V}[\sigma \rightarrow \sigma']\rho &\stackrel{\text{def}}{=} \{(k, \lambda(x:\sigma).e_1, \lambda(x:\sigma).e_2) \mid \forall j \leq k. \forall v_1, v_2. (j, v_1, v_2) \in \mathcal{V}[\sigma]\rho \implies (j, e_1[v_1/x], e_2[v_2/x]) \in \mathcal{E}[\sigma']\rho\} \\
\mathcal{V}[\mu\alpha.\sigma]\rho &\stackrel{\text{def}}{=} \{(k, \text{fold}_{\mu\alpha.\sigma} v_1, \text{fold}_{\mu\alpha.\sigma} v_2) \mid \forall j < k. (j, v_1, v_2) \in \mathcal{V}[\sigma[\mu\alpha.\sigma/\alpha]]\rho\} \\
\mathcal{V}[\alpha]\rho &\stackrel{\text{def}}{=} \rho_R(\alpha) \\
\mathcal{V}[\tau_1 + \tau_2]\rho &\stackrel{\text{def}}{=} \{(k, \text{inj}_i v_1, \text{inj}_i v_2) \mid i \in \{1, 2\} \wedge (k, v_1, v_2) \in \mathcal{V}[\tau_i]\rho\} \\
\mathcal{V}[\langle \tau_1, \dots, \tau_n \rangle]\rho &\stackrel{\text{def}}{=} \{(k, \langle v_1, \dots, v_n \rangle, \langle v'_1, \dots, v'_n \rangle) \mid \forall i \in \{1 \dots n\}. (k, v_i, v'_i) \in \mathcal{V}[\tau_i]\rho\} \\
\mathcal{V}[\forall[\alpha].\tau \rightarrow \theta]\rho &\stackrel{\text{def}}{=} \{(k, \lambda[\alpha](x:\rho_1(\tau)).e_1, \lambda[\alpha](x:\rho_2(\tau)).e_2) \mid \\
&\quad \forall \tau_1, \tau_2, R \in \text{Rel}[\tau_1, \tau_2]. \\
&\quad \forall j \leq k. \forall (j, v_1, v_2) \in \mathcal{V}[\tau]\rho[\alpha \mapsto (\tau_1, \tau_2, R)]. \\
&\quad (j, e_1[\tau_1/\alpha][v_1/x], e_2[\tau_2/\alpha][v_2/x]) \in \mathcal{E}[\theta]\rho[\alpha \mapsto (\tau_1, \tau_2, R)]\} \\
\mathcal{V}[\mu\alpha.\tau]\rho &\stackrel{\text{def}}{=} \{(k, \text{fold}_{\rho_1(\mu\alpha.\tau)} v_1, \text{fold}_{\rho_2(\mu\alpha.\tau)} v_2) \mid \forall j < k. (j, v_1, v_2) \in \mathcal{V}[\tau[\mu\alpha.\tau/\alpha]]\rho\} \\
\mathcal{V}[\mathbf{0}]\rho &\stackrel{\text{def}}{=} \emptyset \\
\mathcal{V}[\exists\alpha.\tau]\rho &\stackrel{\text{def}}{=} \{(k, \text{pack}(\tau_1, v_1) \text{ as } \rho_1(\exists\alpha.\tau), \text{pack}(\tau_2, v_2) \text{ as } \rho_2(\exists\alpha.\tau)) \mid \\
&\quad \exists R \in \text{Rel}[\tau_1, \tau_2]. (k, v_1, v_2) \in \mathcal{V}[\tau]\rho[\alpha \mapsto (\tau_1, \tau_2, R)]\} \\
\mathcal{R}[\theta]\rho &\subset \text{Atom}^{\text{res}}[\theta]\rho \\
\mathcal{R}[\sigma]\rho &\stackrel{\text{def}}{=} \mathcal{V}[\sigma]\rho \\
\mathcal{R}[\mathbf{E}\tau_{\text{exn}}\tau]\rho &\stackrel{\text{def}}{=} \{(k, \text{return } v_1, \text{return } v_2) \mid (k, v_1, v_2) \in \mathcal{V}[\tau]\rho\} \\
&\quad \cup \\
&\quad \{(k, \text{raise } v_1, \text{raise } v_2) \mid (k, v_1, v_2) \in \mathcal{V}[\tau_{\text{exn}}]\rho\} \\
\mathcal{E}[\theta]\rho &\subset \text{Atom}[\theta]\rho \\
\mathcal{E}[\theta]\rho &\stackrel{\text{def}}{=} \{(k, e_1, e_2) \mid \forall K_1, K_2. (k, K_1, K_2) \in \mathcal{K}[\theta]\rho \implies (k, K_1[e_1], K_2[e_2]) \in \mathcal{O}\} \\
\mathcal{K}[\theta]\rho &\subset \text{Atom}^{\mathcal{K}}[\theta]\rho \\
\mathcal{K}[\theta]\rho &\stackrel{\text{def}}{=} \{(k, K_1, K_2) \mid \forall j \leq k, r_1, r_2. (j, r_1, r_2) \in \mathcal{R}[\theta]\rho \implies (j, K_1[r_1], K_2[r_2]) \in \mathcal{O}\} \\
\mathcal{O} &\stackrel{\text{def}}{=} \{(k, e_1, e_2) \mid (e_1 \Downarrow \wedge e_2 \Downarrow) \vee (\text{running}(k, e_1) \wedge \text{running}(k, e_2))\} \\
\mathcal{D}[\cdot] &\stackrel{\text{def}}{=} \{\emptyset\} \\
\mathcal{D}[\Delta, \alpha] &\stackrel{\text{def}}{=} \{\rho[\alpha \mapsto (\tau_1, \tau_2, R)] \mid \rho \in \mathcal{D}[\Delta] \wedge R \in \text{Rel}[\tau_1, \tau_2]\} \\
\mathcal{G}[\cdot]\rho &\stackrel{\text{def}}{=} \{(k, \emptyset) \mid k \in \mathbb{N}\} \\
\mathcal{G}[\Gamma, x:\tau]\rho &\stackrel{\text{def}}{=} \{(k, \gamma[x \mapsto (v_1, v_2)]) \mid (k, \gamma) \in \mathcal{G}[\Gamma]\rho \wedge (k, v_1, v_2) \in \mathcal{V}[\tau]\rho\}
\end{aligned}$$

Figure 18: Combined Language (λ^{ST}): Logical Relations for Closed Terms

$$\begin{aligned}
\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{\text{log}} e_2 : \theta &\stackrel{\text{def}}{=} \Delta; \Gamma \vdash e_1 : \theta \wedge \Delta; \Gamma \vdash e_2 : \theta \wedge \\
&\quad \forall k \geq 0. \forall \rho, \gamma. \rho \in \mathcal{D} \llbracket \Delta \rrbracket \wedge (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho \implies \\
&\quad (k, \rho_1(\gamma_1(e_1)), \rho_2(\gamma_2(e_2))) \in \mathcal{E} \llbracket \theta \rrbracket \rho \\
\Delta; \Gamma \vdash v_1 \approx_{\mathcal{V}}^{\text{log}} v_2 : \tau &\stackrel{\text{def}}{=} \Delta; \Gamma \vdash v_1 : \tau \wedge \Delta; \Gamma \vdash v_2 : \tau \wedge \\
&\quad \forall k \geq 0. \forall \rho, \gamma. \rho \in \mathcal{D} \llbracket \Delta \rrbracket \wedge (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho \implies \\
&\quad (k, \rho_1(\gamma_1(v_1)), \rho_2(\gamma_2(v_2))) \in \mathcal{V} \llbracket \tau \rrbracket \rho \\
\vdash C_1 \approx_{\mathcal{I} \Rightarrow \mathcal{J}}^{\text{log}} C_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \varphi') &\stackrel{\text{def}}{=} \vdash C_1 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \varphi') \wedge \\
&\quad \vdash C_2 : (\Delta; \Gamma \vdash \varphi) \Rightarrow (\Delta'; \Gamma' \vdash \varphi') \\
&\quad \wedge \forall e_1, e_2. \Delta; \Gamma \vdash e_1 \approx_{\mathcal{I}}^{\text{log}} e_2 : \varphi \implies \\
&\quad \Delta'; \Gamma' \vdash C_1[e_1] \approx_{\mathcal{J}}^{\text{log}} C_2[e_2] : \varphi'
\end{aligned}$$

Figure 19: Combined Language (λ^{ST}): Logical Relations for Open Terms

$$\begin{aligned}
\mathcal{V}^+ \llbracket \sigma \rrbracket &\stackrel{\text{def}}{=} \{(k, \mathbf{v}_1, \mathbf{v}_2) \in \text{Atom}[\sigma, \sigma^+] \mid \\
&\quad \exists \mathbf{v}_2. {}^\sigma \text{ST } \mathbf{v}_2 \mapsto^* \mathbf{v}_2 \wedge (k, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma \rrbracket \emptyset\} \\
\mathcal{E}^\div \llbracket \sigma \rrbracket &\stackrel{\text{def}}{=} \{(k, \mathbf{e}, \mathbf{e}) \in \text{Atom}[\sigma, \sigma^\div] \mid (k, \mathbf{e}, {}^\sigma \text{ST } \mathbf{e}) \in \mathcal{E} \llbracket \sigma \rrbracket \emptyset\} \\
\mathcal{G}^+ \llbracket \cdot \rrbracket &\stackrel{\text{def}}{=} \{(k, \emptyset, \emptyset) \mid k \in \mathbb{N}\} \\
\mathcal{G}^+ \llbracket \Gamma, x : \sigma \rrbracket &\stackrel{\text{def}}{=} \{(k, \gamma[x \mapsto \mathbf{v}], \gamma[x \mapsto \mathbf{v}]) \mid \\
&\quad (k, \gamma, \gamma) \in \mathcal{G}^+ \llbracket \Gamma \rrbracket \wedge (k, \mathbf{v}, \mathbf{v}) \in \mathcal{V}^+ \llbracket \sigma \rrbracket\} \\
\Gamma \vdash \mathbf{v} \approx_{\approx_+} \mathbf{v} : \sigma &\stackrel{\text{def}}{=} \mathbf{v} \in \lambda^{\text{S}} \wedge \mathbf{v} \in \lambda^{\text{T}} \wedge \Gamma \vdash \mathbf{v} : \sigma \wedge \cdot; \Gamma^+ \vdash \mathbf{v} : \sigma^+ \wedge \\
&\quad \forall (k, \gamma, \gamma) \in \mathcal{G}^+ \llbracket \Gamma \rrbracket. (k, \gamma(\mathbf{v}), \gamma(\mathbf{v})) \in \mathcal{V}^+ \llbracket \sigma \rrbracket \\
\Gamma \vdash \mathbf{e} \approx_{\approx_\div} \mathbf{e} : \sigma &\stackrel{\text{def}}{=} \mathbf{e} \in \lambda^{\text{S}} \wedge \mathbf{e} \in \lambda^{\text{T}} \wedge \Gamma \vdash \mathbf{e} : \sigma \wedge \cdot; \Gamma^+ \vdash \mathbf{e} : \sigma^\div \wedge \\
&\quad \forall (k, \gamma, \gamma) \in \mathcal{G}^+ \llbracket \Gamma \rrbracket. (k, \gamma(\mathbf{e}), \gamma(\mathbf{e})) \in \mathcal{E}^\div \llbracket \sigma \rrbracket
\end{aligned}$$

Figure 20: Cross Language (λ^{ST}) Logical Relations for Closure Conversion Semantics Preservation

7 λ^{ST} Logical Relation Corresponds to Contextual Equivalence

7.1 λ^{ST} Logical Relation: Fundamental Property

Unless otherwise specified, all of the following lemmas additionally assume $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$ and $\Delta \vdash \Gamma$.

Lemma 7.1 (Unique Decomposition)

If $\cdot; \cdot \vdash K[e] : \theta$ and $e \mapsto e'$, then $K[e] \mapsto K[e']$.

Proof

Omitted, standard. □

Lemma 7.2 (Compositionality of Typing)

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$, $(k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$ and $\Delta; \Gamma \vdash e : \sigma$, then $\cdot; \cdot \vdash \rho_1(\gamma_1(e)) : \rho_1(\theta)$ and $\cdot; \cdot \vdash \rho_2(\gamma_2(e)) : \rho_2(\theta)$

Proof

Omitted, standard. □

Lemma 7.3 (Admissibility of Value Relation)

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$ and $\Delta \vdash \tau$, then $\mathcal{V} \llbracket \tau \rrbracket \rho \in \text{Rel}[\rho_1(\tau), \rho_2(\tau)]$

Proof

Omitted. □

Lemma 7.4 (Weakening of Logical Relations)

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$ ($\Delta \vdash \tau$), ($\Delta \vdash \theta$), $R \in \text{Rel}[\tau_1, \tau_2]$ and $\rho' = \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$ then

1. $\mathcal{V} \llbracket \tau \rrbracket \rho' = \mathcal{V} \llbracket \tau \rrbracket \rho$
2. $\mathcal{E} \llbracket \theta \rrbracket \rho' = \mathcal{E} \llbracket \theta \rrbracket \rho$
3. $\mathcal{R} \llbracket \theta \rrbracket \rho' = \mathcal{R} \llbracket \theta \rrbracket \rho$
4. $\mathcal{K} \llbracket \theta \rrbracket \rho' = \mathcal{K} \llbracket \theta \rrbracket \rho$

Proof

By mutual induction on $k, \Delta \vdash \tau, \Delta \vdash \theta$. □

Lemma 7.5 (Compositionality of Logical Relations)

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$, ($\Delta, \alpha \vdash \tau$), ($\Delta \vdash \tau$) and $\Delta, \alpha \vdash \theta$ then

1. $(j, e_1, e_2) \in \mathcal{E} \llbracket \theta \rrbracket \rho'$ if and only if $(j, e_1, e_2) \in \mathcal{E} \llbracket \theta[\tau/\alpha] \rrbracket \rho$
2. $(j, e_1, e_2) \in \mathcal{V} \llbracket \tau \rrbracket \rho'$ if and only if $(j, e_1, e_2) \in \mathcal{V} \llbracket \tau[\tau/\alpha] \rrbracket \rho$

where $\rho' = \rho[\alpha \mapsto (\rho_1(\tau), \rho_2(\tau), \mathcal{V} \llbracket \tau \rrbracket \rho)]$

Proof

By mutual induction on $k, \Delta, \alpha \vdash \tau, \Delta, \alpha \vdash \theta$. □

Lemma 7.6 (Monotonicity of Value Relation)

If $j, k \in \mathbb{N}$, $j \leq k$, and $(k, v_1, v_2) \in \mathcal{V} \llbracket \tau \rrbracket \rho$ then $(j, v_1, v_2) \in \mathcal{V} \llbracket \tau \rrbracket \rho$.

Proof

By induction on τ .

- Case 1, immediate.
- Case $\sigma_1 + \sigma_2$ by inductive hypothesis.
- Case $\sigma_1 \times \sigma_2$ by inductive hypothesis.
- Case $\sigma \rightarrow \sigma'$, by transitivity of \leq .
- Case $\mu\alpha.\sigma$, by transitivity of $<$.
- Case 0, vacuously true.
- Case $\langle \bar{\tau} \rangle$, by inductive hypothesis.
- Case $\tau_1 + \tau_2$ by inductive hypothesis.
- Case α , by definition of Rel and $\rho_R(\alpha) \in \text{Rel}[\tau_1, \tau_2]$ for some τ_1, τ_2 since $\rho \in \mathcal{D}[\Delta]$.
- Case $\forall[\alpha].\tau \rightarrow \mathbf{E}\tau_{\text{exn}}\tau'$, by transitivity of \leq .
- Case $\mu\alpha.\tau$, by transitivity of $<$.
- Case $\exists\alpha.\tau$, by inductive hypothesis.

□

Lemma 7.7 (Monotonicity of G Relation)

If $j, k \in \mathbb{N}, j \leq k$, and $(k, \gamma) \in \mathcal{G}[\Gamma]\rho$ then $(j, \gamma) \in \mathcal{G}[\Gamma]\rho$.

Proof

By induction on structure of γ , and Lemma 7.6.

□

Lemma 7.8 (Result Relation Embeds in Expression Relation)

$\mathcal{R}[\theta]\rho \subset \mathcal{E}[\theta]\rho$.

Proof

Immediate by definition of $\mathcal{K}[\theta]\rho$.

□

Lemma 7.9 (Monadic Bind)

If $(k, e_1, e_2) \in \mathcal{E}[\theta]\rho$

and $(\forall j \leq k, r_1, r_2, (j, r_1, r_2) \in \mathcal{R}[\theta]\rho \implies (j, K_1[r_1], K_2[r_2]) \in \mathcal{E}[\theta]\rho)$,

then $(k, K_1[e_1], K_2[e_2]) \in \mathcal{E}[\theta]\rho$.

Proof

Suppose $(k, K'_1, K'_2) \in \mathcal{K}[\theta]\rho$, we need to show that $(k, K'_1[K_1[e_1]], K'_2[K_2[e_2]]) \in \mathcal{O}$.

Since $(k, e_1, e_2) \in \mathcal{E}[\theta]\rho$, it is sufficient to show that $(k, K'_1[K_1], K'_2[K_2]) \in \mathcal{K}[\theta]\rho$.

Suppose $j \leq k, (j, r_1, r_2) \in \mathcal{R}[\theta]\rho$, we seek to prove that $(j, K'_1[K_1[r_1]], K'_2[K_2[r_2]]) \in \mathcal{O}$.

By hypothesis, $(j, K_1[r_1], K_2[r_2]) \in \mathcal{E}[\theta]\rho$, so $(j, K'_1[K_1[r_1]], K'_2[K_2[r_2]]) \in \mathcal{O}$ by definition of $\mathcal{E}[\theta]\rho$.

□

Lemma 7.10 (Observation Relation closed under Anti-Reduction)

If $e_1 \mapsto^{k_1} e'_1, e_2 \mapsto^{k_2} e'_2$ and $(k', e'_1, e'_2) \in \mathcal{O}$

then for any $0 \leq k \leq k' + \min(k_1, k_2)$, $(k, e_1, e_2) \in \mathcal{O}$.

Proof

If $e'_1 \Downarrow \wedge e'_2 \Downarrow$, then $e_1 \Downarrow \wedge e_2 \Downarrow$.

Otherwise we know that there exist e''_1, e''_2 such that $e'_1 \mapsto^{k'+1} e''_1$ and $e'_2 \mapsto^{k'+1} e''_2$.

Thus $e_1 \mapsto^{k'+k_1+1} e''_1$ and $e_2 \mapsto^{k'+k_2+1} e''_2$, and since $k \leq k' + k_1, k + 1 \leq k' + k_1 + 1$ and similarly $k + 1 \leq k' + k_2 + 1$ there must exist e'''_1, e'''_2 such that $e_1 \mapsto^{k+1} e'''_1$ and $e_2 \mapsto^{k+1} e'''_2$, so $(k, e_1, e_2) \in \mathcal{O}$.

□

Lemma 7.11 (Expression Relation closed under Anti-Reduction)

If $(k, e_1, e_2) \in \text{Atom}[\theta]\rho$, $e_1 \mapsto^{k_1} e'_1$, $e_2 \mapsto^{k_2} e'_2$, $(k', e'_1, e'_2) \in \mathcal{E}[\theta]\rho$ and $k \leq k' + \min(k_1, k_2)$ then $(k, e_1, e_2) \in \mathcal{E}[\theta]\rho$.

Proof

By definition of \mathcal{E} , Lemma 7.1 and Lemma 7.10. □

Lemma 7.12 (Compatibility Source Var)

If $x : \sigma \in \Gamma$ and $\Delta \vdash \Gamma$ then $\Delta; \Gamma \vdash x \approx_{\mathcal{V}}^{\text{log}} x : \sigma$.

Proof

$\Delta; \Gamma \vdash x : \sigma$ by definition of the type system.

Suppose $\rho \in \mathcal{D}[\Delta]$, $(k, \gamma) \in \mathcal{G}[\Gamma]\rho$. Then, by definition of \mathcal{D}, \mathcal{G} , $(k, \rho_1(\gamma_1(x)), \rho_2(\gamma_2(x))) \in \mathcal{V}[\theta]\rho$. □

Lemma 7.13 (Compatibility Source Unit)

$\Delta; \Gamma \vdash \langle \rangle \approx_{\mathcal{V}}^{\text{log}} \langle \rangle : \mathbf{1}$

Proof

Immediate by definition of $\mathcal{V}[\mathbf{1}]\rho$. □

Lemma 7.14 (Compatibility Source Sum)

If $\Delta; \Gamma \vdash v_1 \approx_{\mathcal{V}}^{\text{log}} v_2 : \sigma_i$ then $\Delta; \Gamma \vdash \text{inj}_i v_1 \approx_{\mathcal{V}}^{\text{log}} \text{inj}_i v_2 : \sigma_1 + \sigma_2$

Proof

Standard. □

Lemma 7.15 (Compatibility Source Case)

If $\Delta; \Gamma \vdash v_1 \approx_{\mathcal{V}}^{\text{log}} v_2 : \sigma_1 + \sigma_2$, $\Delta; \Gamma, x : \sigma_1 \vdash e_1 \approx_{\mathcal{E}}^{\text{log}} e'_1 : \sigma$, and $\Delta; \Gamma, y : \sigma_2 \vdash e_2 \approx_{\mathcal{E}}^{\text{log}} e'_2 : \sigma$ then $\Delta; \Gamma \vdash \text{case } v_1 \text{ of } x. e_1 \mid y. e_2 \approx_{\mathcal{E}}^{\text{log}} \text{case } v_2 \text{ of } x. e'_1 \mid y. e'_2 : \sigma$.

Proof

Standard. □

Lemma 7.16 (Compatibility Source Pair)

If $\Delta; \Gamma \vdash v_1 \approx_{\mathcal{V}}^{\text{log}} v'_1 : \sigma_1$ and $\Delta; \Gamma \vdash v_2 \approx_{\mathcal{V}}^{\text{log}} v'_2 : \sigma_2$ then $\Delta; \Gamma \vdash \langle v_1, v_2 \rangle \approx_{\mathcal{V}}^{\text{log}} \langle v'_1, v'_2 \rangle : \sigma_1 \times \sigma_2$

Proof

Standard. □

Lemma 7.17 (Compatibility Source Projection)

If $\Delta; \Gamma \vdash v \approx_{\mathcal{V}}^{\text{log}} v' : \sigma_1 \times \sigma_2$ then $\Delta; \Gamma \vdash \pi_i v \approx_{\mathcal{E}}^{\text{log}} \pi_i v' : \sigma_i$

Proof

Standard. □

Lemma 7.18 (Compatibility Source Abs)

If $\Delta; \Gamma, x : \sigma \vdash e_1 \approx_{\mathcal{E}}^{\text{log}} e_2 : \sigma'$
then $\Delta; \Gamma \vdash \lambda(x : \sigma). e_1 \approx_{\mathcal{V}}^{\text{log}} \lambda(x : \sigma). e_2 : \sigma \rightarrow \sigma'$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$. Then $(k, \rho_1(\gamma_1(\lambda(x : \sigma). e_1)), \rho_2(\gamma_2(\lambda(x : \sigma). e_2))) \in \text{Atom}[\sigma \rightarrow \sigma', \sigma \rightarrow \sigma']$ by Lemma 7.2.

Suppose $j \leq k, (j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma \rrbracket \rho$. We need to show that $(j, \rho_1(\gamma_1(\mathbf{e}_1))[\mathbf{v}_1/x], \rho_2(\gamma_2(\mathbf{e}_2))[\mathbf{v}_2/x]) \in \mathcal{E} \llbracket \sigma' \rrbracket \rho$.

Let $\gamma' = \gamma[x \mapsto (\mathbf{v}_1, \mathbf{v}_2)]$, then by hypothesis, it is sufficient to show that $(j, \gamma') \in \mathcal{G} \llbracket \Gamma, x : \sigma \rrbracket \rho$. This holds by assumption that $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma \rrbracket \rho$ and Lemma 7.7. \square

Lemma 7.19 (Compatibility Source App)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \sigma \rightarrow \sigma'$ and $\Delta; \Gamma \vdash \mathbf{v}'_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}'_2 : \sigma$
then $\Delta; \Gamma \vdash \mathbf{v}_1 \mathbf{v}'_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{v}_2 \mathbf{v}'_2 : \sigma'$.

Proof

Direct from definition of value relation at function type and Lemma 7.11. \square

Lemma 7.20 (Compatibility Source Fold)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \sigma[\mu\alpha. \sigma/\alpha]$
then $\Delta; \Gamma \vdash \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_2 : \mu\alpha. \sigma$.

Proof

Direct from definition of value relation and Lemma 7.6. \square

Lemma 7.21 (Compatibility Source Unfold)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \mu\alpha. \sigma$
then $\Delta; \Gamma \vdash \text{unfold} \mathbf{v}_1 \approx_{\mathcal{E}}^{\text{log}} \text{unfold} \mathbf{v}_2 : \sigma[\mu\alpha. \sigma/\sigma]$.

Proof

Direct from definition of value relation and hypothesis. \square

Lemma 7.22 (Compatibility Source Let)

If $\Delta; \Gamma \vdash \mathbf{e}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}_2 : \sigma_1$ and $\Delta; \Gamma, x : \sigma_1 \vdash \mathbf{e}'_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}'_2 : \sigma_2$
then $\Delta; \Gamma \vdash \text{let } x = \mathbf{e}_1 \text{ in } \mathbf{e}'_1 \approx_{\mathcal{E}}^{\text{log}} \text{let } x = \mathbf{e}_2 \text{ in } \mathbf{e}'_2 : \sigma_2$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$.

We seek to prove that $(k, \rho_1(\gamma_1(\text{let } x = \mathbf{e}_1 \text{ in } \mathbf{e}'_1)), \rho_2(\gamma_2(\text{let } x = \mathbf{e}_2 \text{ in } \mathbf{e}'_2))) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \rho$.

By Lemma 7.9, it is sufficient to show that for any $j \leq k, \mathbf{v}_1, \mathbf{v}_2$, if $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma_1 \rrbracket \rho$, then $(j, \text{let } x = \mathbf{v}_1 \text{ in } \rho_1(\gamma_1(\mathbf{e}'_1))\text{let } x = \mathbf{v}_2 \text{ in } \rho_2(\gamma_2(\mathbf{e}'_2))) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \rho$.

This holds by the fact that $(j, \gamma[x \mapsto (\mathbf{v}_1, \mathbf{v}_2)]) \in \mathcal{G} \llbracket \sigma_1 \rrbracket \rho$ as in the proof of Lemma 7.18. \square

Lemma 7.23 (Compatibility Target Var)

If $\mathbf{x} : \tau \in \Gamma$ and $\Delta \vdash \Gamma$ then $\Delta; \Gamma \vdash \mathbf{x} \approx_{\mathcal{V}}^{\text{log}} \mathbf{x} : \tau$.

Proof

Analogous to proof of Lemma 7.12 \square

Lemma 7.24 (Compatibility Target Sum)

If $\Delta; \Gamma \vdash \mathbf{v} \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}' : \tau_1$ then $\Delta; \Gamma \vdash \text{inj}_i \mathbf{v} \approx_{\mathcal{V}}^{\text{log}} \text{inj}_i \mathbf{v}' : \tau_1 + \tau_2$

Proof

Standard. \square

Lemma 7.25 (Compatibility Target Case)

If $\Delta; \Gamma \vdash \mathbf{v} \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}' : \tau_1 + \tau_2$, $\Delta; \Gamma, \mathbf{x} : \tau_1 \vdash \mathbf{e}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}'_1 : \theta$, $\Delta; \Gamma, \mathbf{y} : \tau_2 \vdash \mathbf{e}_2 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}'_2 : \theta$, then $\Delta; \Gamma \vdash \text{case } \mathbf{v} \text{ of } \mathbf{x}. \mathbf{e}_1 \mid \mathbf{y}. \mathbf{e}_2 \approx_{\mathcal{E}}^{\text{log}} \text{case } \mathbf{v}' \text{ of } \mathbf{x}. \mathbf{e}'_1 \mid \mathbf{y}. \mathbf{e}'_2 : \theta$

Proof

Standard. □

Lemma 7.26 (Compatibility Target Tuple)

If $n \geq 0$, $\forall i \in \{1 \dots n\}$. $\Delta; \Gamma \vdash \mathbf{v}_{1,i} \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_{2,i} : \tau_i$
then $\Delta; \Gamma \vdash \langle \mathbf{v}_{1,1}, \dots, \mathbf{v}_{1,n} \rangle \approx_{\mathcal{V}}^{\text{log}} \langle \mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,n} \rangle : \langle \tau_1, \dots, \tau_n \rangle$.

Proof

Direct from definition of $\mathcal{V}[\langle \tau_1, \dots, \tau_n \rangle] \rho$. □

Lemma 7.27 (Compatibility Target Projection)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \langle \tau_1, \dots, \tau_n \rangle$,
then for any $i \in \{1, \dots, n\}$, $\Delta; \Gamma \vdash \text{return}_{\tau_{\text{exn}}} \mathbf{v}_{1,i} \approx_{\mathcal{E}}^{\text{log}} \text{return}_{\tau_{\text{exn}}} \mathbf{v}_{2,i} : \mathbf{E} \tau_{\text{exn}} \tau_i$.

Proof

Suppose $k \geq 0$, $\rho \in \mathcal{D}[\Delta]$, $(k, \gamma) \in \mathcal{G}[\Gamma] \rho$.

We seek to prove that $(k, \text{return}(\rho_1(\gamma_1(\mathbf{v}_1))).\mathbf{i}, \text{return}(\rho_2(\gamma_2(\mathbf{v}_2))).\mathbf{i}) \in \mathcal{E}[\mathbf{E} \tau_{\text{exn}} \tau_i] \rho$.

By assumption, $(k, \rho_1(\gamma_1(\mathbf{v}_1)), \rho_2(\gamma_2(\mathbf{v}_2))) \in \mathcal{V}[\langle \tau_1, \dots, \tau_n \rangle] \rho$, so $\rho_1(\gamma_1(\mathbf{v}_1)) = \langle \mathbf{v}_{1,1}, \dots, \mathbf{v}_{1,n} \rangle$ and $\rho_2(\gamma_2(\mathbf{v}_2)) = \langle \mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,n} \rangle$, where importantly $(k, \mathbf{v}_{1,i}, \mathbf{v}_{2,i}) \in \mathcal{V}[\tau_i] \rho$.

Next, $\text{return}(\rho_1(\gamma_1(\mathbf{v}_1))) \mapsto \text{return} \mathbf{v}_{1,i}$ and $\text{return}(\rho_2(\gamma_2(\mathbf{v}_2))) \mapsto \text{return} \mathbf{v}_{2,i}$. So by Lemma 7.11, it is sufficient to show $(k-1, \text{return} \mathbf{v}_{1,i}, \text{return} \mathbf{v}_{2,i}) \in \mathcal{E}[\mathbf{E} \tau_{\text{exn}} \tau_i] \rho$, which follows from Lemma 7.8. □

Lemma 7.28 (Compatibility Target Abs)

If $\Delta, \alpha; \Gamma, \mathbf{x} : \tau \vdash \mathbf{e}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}_2 : \theta$
then $\Delta; \Gamma \vdash \lambda[\alpha](\mathbf{x} : \tau). \mathbf{e}_1 \approx_{\mathcal{V}}^{\text{log}} \lambda[\alpha](\mathbf{x} : \tau). \mathbf{e}_2 : \forall[\alpha]. \tau \rightarrow \theta$.

Proof

Suppose $k \geq 0$, $\rho \in \mathcal{D}[\Delta]$, $(k, \gamma) \in \mathcal{G}[\Gamma] \rho$.

We need to show that $(k, \lambda[\alpha](\mathbf{x} : \tau). \rho_1(\gamma_1(\mathbf{e}_1)), \lambda[\alpha](\mathbf{x} : \tau). \rho_2(\gamma_2(\mathbf{e}_2))) \in \mathcal{V}[\forall[\alpha]. \tau \rightarrow \theta] \rho$.

Suppose $\tau_1, \tau_2, R \in \text{Rel}[\tau_1, \tau_2]$, $j \leq k$, $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V}[\tau] \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$. We need to show that $(j, \rho_1(\gamma_1(\mathbf{e}_1))[\tau_1/\alpha][\mathbf{v}_1/\mathbf{x}], \rho_2(\gamma_2(\mathbf{e}_2))[\tau_2/\alpha][\mathbf{v}_2/\mathbf{x}]) \in \mathcal{E}[\theta] \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$

If we define $\rho' = \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$ and $\gamma' = \gamma[\mathbf{x} \mapsto (\mathbf{v}_1, \mathbf{v}_2)]$, then $\rho_1(\gamma_1(\mathbf{e}_1))[\tau_1/\alpha][\mathbf{v}_1/\mathbf{x}] = \rho'_1(\gamma'_1(\mathbf{e}_1))$ and $\rho_2(\gamma_2(\mathbf{e}_2))[\tau_2/\alpha][\mathbf{v}_2/\mathbf{x}] = \rho'_2(\gamma'_2(\mathbf{e}_2))$.

Furthermore, $\rho' \in \mathcal{D}[\Delta, \alpha]$ and $\gamma' \in \mathcal{G}[\Gamma, \mathbf{x} : \tau]$, which with our hypothesis gives us our goal $(j, \rho'_1(\gamma'_1(\mathbf{e}_1)), \rho'_2(\gamma'_2(\mathbf{e}_2))) \in \mathcal{E}[\theta] \rho'$. □

Lemma 7.29 (Compatibility Target App)

If $\Delta \vdash \tau'$ and $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \forall[\alpha]. \tau \rightarrow \theta$ and $\Delta; \Gamma \vdash \mathbf{v}'_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}'_2 : \tau[\tau'/\alpha]$
then $\Delta; \Gamma \vdash \mathbf{v}_1 [\tau'] \mathbf{v}'_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{v}_2 [\tau'] \mathbf{v}'_2 : \theta[\tau'/\alpha]$.

Proof

Suppose $k \geq 0$, $\rho \in \mathcal{D}[\Delta]$, $(k, \gamma) \in \mathcal{G}[\Gamma] \rho$.

We need to show that $(k, \rho_1(\gamma_1(\mathbf{v}_1 [\tau'] \mathbf{v}'_1)), \rho_2(\gamma_2(\mathbf{v}_2 [\tau'] \mathbf{v}'_2))) \in \mathcal{E}[\theta[\tau'/\alpha]] \rho$.

By definition of $\mathcal{V}[\forall[\alpha]. \tau \rightarrow \theta] \rho$, $\rho_1(\gamma_1(\mathbf{v}_1)) = \lambda[\alpha](\mathbf{x} : \tau_1). \mathbf{e}_1$ and $\rho_2(\gamma_2(\mathbf{v}_2)) = \lambda[\alpha](\mathbf{x} : \tau_2). \mathbf{e}_2$.

Then $\rho_1(\gamma_1(\mathbf{v}_1 [\boldsymbol{\tau}' \mathbf{v}'_1])) \mapsto \mathbf{e}_1[\rho_1(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_1(\gamma_1(\mathbf{v}'_1))/\mathbf{x}]$
and $\rho_2(\gamma_2(\mathbf{v}_2 [\boldsymbol{\tau}' \mathbf{v}'_2])) \mapsto \mathbf{e}_2[\rho_2(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_2(\gamma_2(\mathbf{v}'_2))/\mathbf{x}]$

Then by Lemma 7.11, it is sufficient to show that for $j < k$,
 $(k-1, \mathbf{e}_1[\rho_1(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_1(\gamma_1(\mathbf{v}'_1))/\mathbf{x}], \mathbf{e}_2[\rho_2(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_2(\gamma_2(\mathbf{v}'_2))/\mathbf{x}]) \in \mathcal{E} \llbracket \boldsymbol{\theta}[\boldsymbol{\tau}'/\boldsymbol{\alpha}] \rrbracket \rho$.

Define $\rho' = \rho[\boldsymbol{\alpha} \mapsto (\rho_1(\boldsymbol{\tau}'), \rho_2(\boldsymbol{\tau}'), \mathcal{V} \llbracket \boldsymbol{\tau}' \rrbracket \rho)]$. By Lemma 7.3, $\mathcal{V} \llbracket \boldsymbol{\tau}' \rrbracket \rho \in \text{Rel}[\rho_1(\boldsymbol{\tau}')][\rho_2(\boldsymbol{\tau}')] \rho$, so $\rho' \in \mathcal{D} \llbracket \Delta, \boldsymbol{\alpha} \rrbracket$.

Then by Lemma 7.5 it is sufficient to show

$(k-1, \mathbf{e}_1[\rho_1(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_1(\gamma_1(\mathbf{v}'_1))/\mathbf{x}], \mathbf{e}_2[\rho_2(\boldsymbol{\tau}')/\boldsymbol{\alpha}][\rho_2(\gamma_2(\mathbf{v}'_2))/\mathbf{x}]) \in \mathcal{E} \llbracket \boldsymbol{\theta} \rrbracket \rho'$,

and so by definition of $\mathcal{V} \llbracket \forall[\boldsymbol{\alpha}]. \boldsymbol{\tau} \rightarrow \boldsymbol{\theta} \rrbracket \rho$, it is sufficient to show that $(k-1, \rho_1(\gamma_1(\mathbf{v}'_1)), \rho_2(\gamma_2(\mathbf{v}'_2))) \in \mathcal{V} \llbracket \boldsymbol{\tau} \rrbracket \rho'$, which follows from Lemma 7.5. \square

Lemma 7.30 (Compatibility Target Fold)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\alpha}]$

then $\Delta; \Gamma \vdash \mathbf{fold}_{\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}} \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{fold}_{\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}} \mathbf{v}_2 : \boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}$.

Proof

Analogous to proof of Lemma 7.20 \square

Lemma 7.31 (Compatibility Target Unfold)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}$

then $\Delta; \Gamma \vdash \mathbf{return}_{\boldsymbol{\tau}_{\text{exn}}} \mathbf{unfold} \mathbf{v}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{return}_{\boldsymbol{\tau}_{\text{exn}}} \mathbf{unfold} \mathbf{v}_2 : \mathbf{E} \boldsymbol{\tau}_{\text{exn}} \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\tau}]$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$.

We need to show that $(k, \mathbf{return}(\mathbf{unfold} \rho_1(\gamma_1(\mathbf{v}_1))), \mathbf{return}(\mathbf{unfold} \rho_2(\gamma_2(\mathbf{v}_2)))) \in \mathcal{E} \llbracket \mathbf{E} \boldsymbol{\tau}_{\text{exn}} \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\tau}] \rrbracket \rho$.

By hypothesis and definition of $\mathcal{V} \llbracket \boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau} \rrbracket \rho$, $\rho_1(\gamma_1(\mathbf{v}_1)) = \mathbf{fold}_{\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}} \mathbf{v}'_1$ and $\rho_2(\gamma_2(\mathbf{v}_2)) = \mathbf{fold}_{\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}} \mathbf{v}'_2$ where for all $j < k, (j, \mathbf{v}'_1, \mathbf{v}'_2) \in \mathcal{V} \llbracket \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\tau}] \rrbracket \rho$.

Therefore, $\mathbf{return}(\mathbf{unfold} \rho_1(\gamma_1(\mathbf{v}_1))) \mapsto \mathbf{return} \mathbf{v}'_1$ and

$\mathbf{return}(\mathbf{unfold} \rho_2(\gamma_2(\mathbf{v}_2))) \mapsto \mathbf{return} \mathbf{v}'_2$. Finally, for any $(k-1, \mathbf{return} \mathbf{v}'_1, \mathbf{return} \mathbf{v}'_2) \in \mathcal{E} \llbracket \mathbf{E} \boldsymbol{\tau}_{\text{exn}} \boldsymbol{\tau}[\boldsymbol{\mu}\boldsymbol{\alpha}. \boldsymbol{\tau}/\boldsymbol{\tau}] \rrbracket \rho$ by hypothesis and Lemma 7.8, so the result holds by Lemma 7.11. \square

Lemma 7.32 (Compatibility Target Pack)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \boldsymbol{\tau}[\boldsymbol{\tau}'/\boldsymbol{\alpha}]$

then $\Delta; \Gamma \vdash \mathbf{pack}(\boldsymbol{\tau}', \mathbf{v}_1) \text{ as } \exists \boldsymbol{\alpha}. \boldsymbol{\tau} \approx_{\mathcal{V}}^{\text{log}} \mathbf{pack}(\boldsymbol{\tau}', \mathbf{v}_2) \text{ as } \exists \boldsymbol{\alpha}. \boldsymbol{\tau} : \exists \boldsymbol{\alpha}. \boldsymbol{\tau}$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$. We need to show that

$(k, \mathbf{pack}(\rho_1(\boldsymbol{\tau}'), \rho_1(\gamma_1(\mathbf{v}_1))) \text{ as } \exists \boldsymbol{\alpha}. \rho_1(\boldsymbol{\tau}), \mathbf{pack}(\rho_2(\boldsymbol{\tau}'), \rho_2(\gamma_2(\mathbf{v}_2))) \text{ as } \exists \boldsymbol{\alpha}. \rho_2(\boldsymbol{\tau})) \in \mathcal{V} \llbracket \exists \boldsymbol{\alpha}. \boldsymbol{\tau} \rrbracket \rho$.

First, by Lemma 7.3, $\mathcal{V} \llbracket \boldsymbol{\tau}' \rrbracket \rho \in \text{Rel}[\rho_1(\boldsymbol{\tau}'), \rho_2(\boldsymbol{\tau}')] \rho$. Therefore it is sufficient to show that for any $j < k, (j, \rho_1(\gamma_1(\mathbf{v}_1)), \rho_2(\gamma_2(\mathbf{v}_2))) \in \mathcal{V} \llbracket \boldsymbol{\tau} \rrbracket \rho[\boldsymbol{\alpha} \mapsto (\rho_1(\boldsymbol{\tau}'), \rho_2(\boldsymbol{\tau}'), \mathcal{V} \llbracket \boldsymbol{\tau}' \rrbracket \rho)]$.

By Lemma 7.5, this is equivalent to showing $(j, \rho_1(\gamma_1(\mathbf{v}_1)), \rho_2(\gamma_2(\mathbf{v}_2))) \in \mathcal{V} \llbracket \boldsymbol{\tau}[\boldsymbol{\tau}'/\boldsymbol{\alpha}] \rrbracket \rho$, which holds by hypothesis and Lemma 7.6. \square

Lemma 7.33 (Compatibility Target Unpack)

If $\Delta; \Gamma \vdash \mathbf{v}_1 \approx_{\mathcal{V}}^{\text{log}} \mathbf{v}_2 : \exists \boldsymbol{\alpha}. \boldsymbol{\tau}$ and $\Delta, \boldsymbol{\alpha}; \Gamma, \mathbf{x} : \boldsymbol{\tau} \vdash \mathbf{e}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{e}_2 : \boldsymbol{\theta}$

then $\Delta; \Gamma \vdash \mathbf{unpack}(\boldsymbol{\alpha}, \mathbf{x}) = \mathbf{v}_1 \text{ in } \mathbf{e}_1 \approx_{\mathcal{E}}^{\text{log}} \mathbf{unpack}(\boldsymbol{\alpha}, \mathbf{x}) = \mathbf{v}_2 \text{ in } \mathbf{e}_2 : \boldsymbol{\theta}$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$.

We need to show that $(k, \mathbf{unpack}(\alpha, \mathbf{x}) = \rho_1(\mathbf{v}_1) \mathbf{in} \rho_1(\mathbf{e}_1), \mathbf{unpack}(\alpha, \mathbf{x}) = \rho_2(\mathbf{v}_2) \mathbf{in} \rho_2(\mathbf{e}_2)) \in \mathcal{E} \llbracket \theta \rrbracket \rho$.

By hypothesis and definition of $\mathcal{V} \llbracket \exists \alpha. \tau \rrbracket \rho$, $\rho_1(\mathbf{v}_1) = \mathbf{pack}(\mathbf{v}'_1, \tau_1) \mathbf{as} \exists \alpha. \rho_1(\tau)$ and $\rho_2(\mathbf{v}_2) = \mathbf{pack}(\mathbf{v}'_2, \tau_2) \mathbf{as} \exists \alpha. \rho_2(\tau)$, so $\mathbf{unpack}(\alpha, \mathbf{x}) = \rho_1(\mathbf{v}_1) \mathbf{in} \rho_1(\mathbf{e}_1) \mapsto \rho_1(\mathbf{e}_1)[\tau_1/\alpha][\mathbf{v}_1/\mathbf{x}]$ and $\mathbf{unpack}(\alpha, \mathbf{x}) = \rho_2(\mathbf{v}_2) \mathbf{in} \rho_2(\mathbf{e}_2) \mapsto \rho_2(\mathbf{e}_2)[\tau_2/\alpha][\mathbf{v}_2/\mathbf{x}]$.

Then the result holds by an analogous argument to that in the proof of Lemma 7.28. \square

Lemma 7.34 (Compatibility Target Handle)

If $\Delta; \Gamma \vdash \mathbf{e}_1 \approx_{\mathcal{E}}^{\log} \mathbf{e}_2 : \mathbf{E} \tau'_{\text{exn}} \tau'$

and $\Delta; \Gamma, \mathbf{x} : \tau' \vdash \mathbf{e}'_1 \approx_{\mathcal{E}}^{\log} \mathbf{e}'_2 : \mathbf{E} \tau_{\text{exn}} \tau$

and $\Delta; \Gamma, \mathbf{y} : \tau'_{\text{exn}} \vdash \mathbf{e}''_1 \approx_{\mathcal{E}}^{\log} \mathbf{e}''_2 : \mathbf{E} \tau_{\text{exn}} \tau$

then $\Delta; \Gamma \vdash \mathbf{handle} \mathbf{e}_1 \mathbf{with} (\mathbf{x}. \mathbf{e}'_1) (\mathbf{y}. \mathbf{e}''_1) \approx_{\mathcal{E}}^{\log} \mathbf{handle} \mathbf{e}_2 \mathbf{with} (\mathbf{x}. \mathbf{e}'_2) (\mathbf{y}. \mathbf{e}''_2) : \mathbf{E} \tau_{\text{exn}} \tau$.

Proof

Suppose $k \geq 0, \rho \in \mathcal{D} \llbracket \Delta \rrbracket, (k, \gamma) \in \mathcal{G} \llbracket \Gamma \rrbracket \rho$.

We need to show that

$(k, \mathbf{handle}(\rho_1(\gamma_1(\mathbf{e}_1))) \mathbf{with} (\mathbf{x}. \rho_1(\gamma_1(\mathbf{e}'_1))) (\mathbf{y}. \rho_1(\gamma_1(\mathbf{e}''_1))),$
 $\mathbf{handle}(\rho_2(\gamma_2(\mathbf{e}_2))) \mathbf{with} (\mathbf{x}. \rho_2(\gamma_2(\mathbf{e}'_2))) (\mathbf{y}. \rho_2(\gamma_2(\mathbf{e}''_2)))) \in \mathcal{E} \llbracket \mathbf{E} \tau_{\text{exn}} \tau \rrbracket \rho$

Applying Lemma 7.9, there are two cases

1. Suppose $j \leq k, (j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \tau' \rrbracket \rho$, then we need to show that

$(j, \mathbf{handle} \mathbf{return} \mathbf{v}_1 \mathbf{with} (\mathbf{x}. \rho_1(\gamma_1(\mathbf{e}'_1))) (\mathbf{y}. \rho_1(\gamma_1(\mathbf{e}''_1))),$
 $\mathbf{handle} \mathbf{return} \mathbf{v}_2 \mathbf{with} (\mathbf{x}. \rho_2(\gamma_2(\mathbf{e}'_2))) (\mathbf{y}. \rho_2(\gamma_2(\mathbf{e}''_2)))) \in \mathcal{E} \llbracket \mathbf{E} \tau_{\text{exn}} \tau \rrbracket \rho$.

Then $\mathbf{handle} \mathbf{return} \mathbf{v}_1 \mathbf{with} (\mathbf{x}. \rho_1(\gamma_1(\mathbf{e}'_1))) (\mathbf{y}. \rho_1(\gamma_1(\mathbf{e}''_1))) \mapsto \rho_1(\gamma_1(\mathbf{e}'_1))[\mathbf{v}_1/\mathbf{x}]$ and
 $\mathbf{handle} \mathbf{return} \mathbf{v}_2 \mathbf{with} (\mathbf{x}. \rho_2(\gamma_2(\mathbf{e}'_2))) (\mathbf{y}. \rho_2(\gamma_2(\mathbf{e}''_2))) \mapsto \rho_2(\gamma_2(\mathbf{e}'_2))[\mathbf{v}_2/\mathbf{x}]$.

Let $\gamma' = \gamma[\mathbf{x} \mapsto (\mathbf{v}_1, \mathbf{v}_2)]$. Then $\rho_1(\gamma_1(\mathbf{e}'_1))[\mathbf{v}_1/\mathbf{x}] = \rho_1(\gamma'_1(\mathbf{e}'_1))$ and $\rho_2(\gamma_2(\mathbf{e}'_2))[\mathbf{v}_2/\mathbf{x}] = \rho_2(\gamma'_2(\mathbf{e}'_2))$.
Furthermore, $\gamma' \in \mathcal{G} \llbracket \Gamma, \mathbf{x} : \tau' \rrbracket$, so by hypothesis $(j, \rho_1(\gamma'_1(\mathbf{e}'_1)), \rho_2(\gamma'_2(\mathbf{e}'_2))) \in \mathcal{E} \llbracket \mathbf{E} \tau_{\text{exn}} \tau \rrbracket \rho$. The result then holds by Lemma 7.11.

2. Suppose $j \leq k, (j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \tau'_{\text{exn}} \rrbracket \rho$, then we need to show that

$(j, \mathbf{handle} \mathbf{raise} \mathbf{v}_1 \mathbf{with} (\mathbf{x}. \rho_1(\gamma_1(\mathbf{e}'_1))) (\mathbf{y}. \rho_1(\gamma_1(\mathbf{e}''_1))),$
 $\mathbf{handle} \mathbf{raise} \mathbf{v}_2 \mathbf{with} (\mathbf{x}. \rho_2(\gamma_2(\mathbf{e}'_2))) (\mathbf{y}. \rho_2(\gamma_2(\mathbf{e}''_2)))) \in \mathcal{E} \llbracket \mathbf{E} \tau_{\text{exn}} \tau \rrbracket \rho$.

Analogous to the previous case. \square

Lemma 7.35 (Bridge Lemmas)

Let $\rho \in \mathcal{D} \llbracket \Delta \rrbracket, \Delta \vdash \sigma$.

1. If $(k, \mathbf{e}_1, \mathbf{e}_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$, then $(k, {}^\sigma \mathbf{ST} \mathbf{e}_1, {}^\sigma \mathbf{ST} \mathbf{e}_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$.
2. If $(k, \mathbf{r}_1, \mathbf{r}_2) \in \mathcal{R} \llbracket \sigma^\dagger \rrbracket \rho$ and ${}^\sigma \mathbf{ST} \mathbf{r}_1 \mapsto^n \mathbf{v}_1$ and ${}^\sigma \mathbf{ST} \mathbf{r}_2 \mapsto^m \mathbf{v}_2$, then $(k, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{R} \llbracket \sigma \rrbracket \rho$.
3. If $(k, \mathbf{e}_1, \mathbf{e}_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$, then $(k, \mathcal{TS}^\sigma \mathbf{e}_1, \mathcal{TS}^\sigma \mathbf{e}_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$.
4. If $(k, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{R} \llbracket \sigma \rrbracket \rho$ and $\mathcal{TS}^\sigma \mathbf{v}_1 \mapsto^n \mathbf{r}_1$ and ${}^\sigma \mathbf{ST} \mathbf{v}_1 \mapsto^m \mathbf{r}_2$, then $(k, \mathbf{r}_1, \mathbf{r}_2) \in \mathcal{R} \llbracket \sigma^\dagger \rrbracket \rho$.

Proof

Proved simultaneously by induction on σ, k .

1. By Lemma 7.9, it is sufficient to prove that for all $j \leq k, (j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket$,
 $(j, {}^\sigma \mathbf{ST} \mathbf{return} \mathbf{v}_1, {}^\sigma \mathbf{ST} \mathbf{return} \mathbf{v}_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$ and for all $j \leq k, (j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \mathbf{0} \rrbracket \rho$,
 $(j, {}^\sigma \mathbf{ST} \mathbf{raise} \mathbf{v}_1, {}^\sigma \mathbf{ST} \mathbf{raise} \mathbf{v}_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$. The latter is vacuously true since $\mathcal{V} \llbracket \mathbf{0} \rrbracket \rho = \emptyset$.
For the former case, note that $(j, \mathbf{return} \mathbf{v}_1, \mathbf{return} \mathbf{v}_2) \in \mathcal{R} \llbracket \sigma^\dagger \rrbracket \rho$ by definition of $\mathcal{R} \llbracket \sigma^\dagger \rrbracket \rho$ and the assumption that $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket$. The goal follows by case 2 of this lemma, and Lemma 7.8.

2. By case analysis of σ . We omit the uninteresting cases such as $\sigma_1 + \sigma_2$ and $\sigma_1 \times \sigma_2$

Case $\sigma = \sigma'' \rightarrow \sigma'$: then $\sigma^+ = \exists \alpha. \langle \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^+ \rangle, \alpha \rangle$.

By definition of \mathcal{V} , this means $\mathbf{v}_1 = \mathbf{pack}(\tau_1, \langle \mathbf{v}'_1, \mathbf{v}''_1 \rangle)$ as $(\sigma'' \rightarrow \sigma')^+$ and $\mathbf{v}_2 = \mathbf{pack}(\tau_2, \langle \mathbf{v}'_2, \mathbf{v}''_2 \rangle)$ as $(\sigma'' \rightarrow \sigma')^+$, where there is some relation $R \in \text{Rel}[\tau_1, \tau_2]$ such that $(k, \mathbf{v}'_1, \mathbf{v}'_2) \in \mathcal{V}[\langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^+] \rho'$ and $(k, \mathbf{v}''_1, \mathbf{v}''_2) \in \mathcal{V}[\alpha] \rho'$, where $\rho' = \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$. Next, $\sigma'' \rightarrow \sigma' \mathcal{ST} \text{ return } \mathbf{v}_1 \mapsto$

$$\lambda(x: \sigma''). \sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_1 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad) \\ \text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_1} \mathbf{x} \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle$$

and $\sigma'' \rightarrow \sigma' \mathcal{ST} \text{ return } \mathbf{v}_2 \mapsto$

$$\lambda(x: \sigma''). \sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_2 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad) \\ \text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_2} \mathbf{x} \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle$$

Suppose $j \leq k$ and $(j, \mathbf{v}'''_1, \mathbf{v}'''_2) \in \mathcal{V}[\sigma''] \rho$. We need to show that

$$(j, \sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_1 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad), \\ \text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_1} \mathbf{v}'''_1 \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle \\ \sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_2 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad)) \\ \text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_2} \mathbf{v}'''_2 \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle$$

$\in \mathcal{E}[\sigma'] \rho$.

By inductive hypothesis and Lemma 10.5, there exist $(j, \mathbf{v}'''_1, \mathbf{v}'''_2) \in \mathcal{V}[\sigma''^+] \rho'$ such that

$\mathcal{TS}^{\sigma_1} \mathbf{v}'''_1 \mapsto^{n'} \text{return } \mathbf{v}'''_1$ and similarly $\mathcal{TS}^{\sigma_2} \mathbf{v}'''_2 \mapsto^{m'} \text{return } \mathbf{v}'''_2$ for some n', m' .

Then $\sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_1 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad) \mapsto^{n'+5}$

$$\text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_1} \mathbf{v}'''_1 \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle$$

and similarly $\sigma' \mathcal{ST} \mathbf{v}'_1[\tau_1] \langle \mathbf{v}''_1, \mathbf{v}'''_1 \rangle$ and similarly $\sigma' \mathcal{ST}(\mathbf{unpack}(\alpha, z) = \mathbf{v}_2 \text{ in let } \mathbf{x}_f = \text{return } z.1 \text{ in } \quad) \mapsto^{m'+5}$

$$\text{let } \mathbf{x}_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } \mathbf{x} = \mathcal{TS}^{\sigma_2} \mathbf{v}'''_2 \text{ in } \mathbf{x}_f[\alpha] \langle \mathbf{x}_{\text{env}}, \mathbf{x} \rangle$$

$\sigma' \mathcal{ST} \mathbf{v}'_2[\tau_2] \langle \mathbf{v}''_2, \mathbf{v}'''_2 \rangle$.

The result then holds by inductive hypothesis, Lemma 7.11, and similar reasoning to Lemma 7.29 and Lemma 7.26.

Case $\sigma = \mu\alpha. \sigma'$: then $\sigma^+ = \mu\alpha. \sigma'^+$.

By definition of $\mathcal{V}[\mu\alpha. \sigma'^+] \rho$, $\mathbf{v}_1 = \mathbf{fold}_{\mu\alpha. \sigma'^+} \mathbf{v}'_1$ and $\mathbf{v}_2 = \mathbf{fold}_{\mu\alpha. \sigma'^+} \mathbf{v}'_2$ such that for every $j < k$, $(j, \mathbf{v}'_1, \mathbf{v}'_2) \in \mathcal{V}[\sigma'^+[\mu\alpha. \sigma'^+/\alpha]]$.

Next, $\mu\alpha. \sigma' \mathcal{ST} \text{ return } \mathbf{v}_1 \mapsto \text{let } \mathbf{x} = \sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \text{ unfold } \mathbf{v}_1 \text{ in fold}_{\mu\alpha. \sigma'} \mathbf{x}$ and $\mu\alpha. \sigma' \mathcal{ST} \text{ return } \mathbf{v}_2 \mapsto \text{let } \mathbf{x} = \sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \text{ unfold } \mathbf{v}_2 \text{ in fold}_{\mu\alpha. \sigma'} \mathbf{x}$.

Furthermore, by Lemma 10.5 and inductive hypothesis, $\sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \mathbf{v}'_1 \mapsto^n \mathbf{v}'_1$ and similarly $\sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \mathbf{v}'_2 \mapsto^m \mathbf{v}'_2$ and $(j, \mathbf{v}'_1, \mathbf{v}'_2) \in \mathcal{V}[\sigma'^+[\mu\alpha. \sigma'/\alpha]] \rho$ for every $j < k$.

Therefore $\text{let } \mathbf{x} = \sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \text{ unfold } \mathbf{v}_1 \text{ in fold}_{\mu\alpha. \sigma'} \mathbf{x} \mapsto^{n+2} \mathbf{fold}_{\mu\alpha. \sigma'} \mathbf{v}'_1$ and $\text{let } \mathbf{x} = \sigma'[\mu\alpha. \sigma'/\alpha] \mathcal{ST} \text{ return } \mathbf{0} \text{ unfold } \mathbf{v}_2 \text{ in fold}_{\mu\alpha. \sigma'} \mathbf{x} \mapsto^{m+2} \mathbf{fold}_{\mu\alpha. \sigma'} \mathbf{v}'_2$. So we need to show that $(k, \mathbf{fold}_{\mu\alpha. \sigma'} \mathbf{v}'_1, \mathbf{fold}_{\mu\alpha. \sigma'} \mathbf{v}'_2) \in \mathcal{V}[\mu\alpha. \sigma'] \rho$, which holds by definition of $\mathcal{V}[\mu\alpha. \sigma'] \rho$ and what we know about $\mathbf{v}'_1, \mathbf{v}'_2$.

3. By Lemma 7.9, it is sufficient to prove that for all $j \leq k$ if $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V}[\sigma] \rho$ then $(j, \mathcal{TS}^{\sigma} \mathbf{v}_1, \mathcal{TS}^{\sigma} \mathbf{v}_2) \in \mathcal{E}[\sigma^+] \rho$. The result then holds by the value case and Lemma 7.8.

4. By case analysis of σ . We omit the uninteresting cases such as $\sigma_1 + \sigma_2$ and $\sigma_1 \times \sigma_2$

Case $\sigma = \sigma'' \rightarrow \sigma'$: $\mathcal{TS}^\sigma v_1 \mapsto \mathbf{return_0 pack} (1, \langle \lambda(z : \langle 1, \sigma''^+ \rangle) \rangle)$, $\langle \rangle \rangle \mathbf{as} (\sigma_1 \rightarrow \sigma_2)^+$

$$\mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} z.2 \mathbf{in} \\ v'_1 x \end{array} \right)$$

and similarly $\mathcal{TS}^\sigma v_2 \mapsto \mathbf{return_0 pack} (1, \langle \lambda(z : \langle 1, \sigma''^+ \rangle) \rangle)$, $\langle \rangle \rangle \mathbf{as} (\sigma_2 \rightarrow \sigma_2)^+$

$$\mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} z.2 \mathbf{in} \\ v'_2 x \end{array} \right)$$

so we need to show that these **packs** are in $\mathcal{V} [\exists \alpha. \langle \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^+ \rangle, \alpha] \rho$.

For our relation we choose $\mathcal{V} [\mathbf{1}] \rho$ (justified by Lemma 7.3). By definition of \mathcal{V} , it is sufficient to prove that for any $j \leq k$ and $(j, \langle \rangle, v'_1), \langle \rangle, v'_2) \in \mathcal{V} [\langle \alpha, \sigma_1^+ \rangle] \rho'$,

$$\left(j, \mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} \langle \rangle, v'_1.2 \mathbf{in} \\ v'_1 x \end{array} \right), \mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} \langle \rangle, v'_2.2 \mathbf{in} \\ v'_2 x \end{array} \right) \right)$$

$\in \mathcal{E} [\sigma_2^+] \rho'$ where $\rho' = \rho[\alpha \mapsto (1, 1, \mathcal{V} [\mathbf{1}] \rho)]$.

By inductive hypothesis and Lemma 10.5, there exist v''_1, v''_2 such that $(j, v''_1, v''_2) \in \mathcal{V} [\sigma_1^+] \rho'$, $\sigma'' \mathcal{ST} \mathbf{return} v''_1 \mapsto^{n'} v'_1$ and $\sigma'' \mathcal{ST} \mathbf{return} v''_2 \mapsto^{m'} v'_2$.

Then $\mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} \langle \rangle, v'_1.2 \mathbf{in} \\ v'_1 x \end{array} \right) \mapsto^{n'+2} v''_1 v''_1$ and

$\mathcal{TS}^{\sigma'} \left(\begin{array}{c} \mathbf{let} x = \sigma'' \mathcal{ST} \mathbf{return_0} \langle \rangle, v'_2.2 \mathbf{in} \\ v'_2 x \end{array} \right) \mapsto^{m'+2} v''_2 v''_2$. So the result holds by similar reasoning to Lemma 7.19.

Case $\sigma = \mu \alpha. \sigma'$: the proof follows similarly to the corresponding case above. □

Lemma 7.36 (Compatibility Source Boundary)

If $\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{log} e_2 : \sigma^\ddagger$, then $\Delta; \Gamma \vdash \sigma \mathcal{ST} e_1 \approx_{\mathcal{E}}^{log} \sigma \mathcal{ST} e_2 : \sigma$.

Proof

Immediate by Lemma 7.35 □

Lemma 7.37 (Compatibility Target Boundary)

If $\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{log} e_2 : \sigma$, then $\Delta; \Gamma \vdash \mathcal{TS}^\sigma e_1 \approx_{\mathcal{E}}^{log} \mathcal{TS}^\sigma e_2 : \sigma^\ddagger$

Proof

Immediate by Lemma 7.35 □

Theorem 7.38 (Fundamental Properties)

The following are proved by mutual induction.

1. If $\Delta; \Gamma \vdash e : \theta$, then $\Delta; \Gamma \vdash e \approx_{\mathcal{E}}^{log} e : \theta$
2. If $\Delta; \Gamma \vdash v : \tau$, then $\Delta; \Gamma \vdash v \approx_{\mathcal{V}}^{log} v : \tau$

Proof

By induction on the typing derivation, then immediate by appropriate compatibility lemma. □

Lemma 7.39 (Context Fundamental Property)

There are four cases, depending on whether the context takes values or produces values.

1. If $\vdash C : (\Delta; \Gamma \vdash \theta) \Rightarrow (\Delta'; \Gamma' \vdash \theta')$, then $\vdash C \approx_{\mathcal{E} \Rightarrow \mathcal{E}}^{log} C : (\Delta; \Gamma \vdash \theta) \Rightarrow (\Delta'; \Gamma' \vdash \theta')$.
2. If $\vdash C : (\Delta; \Gamma \vdash \theta) \Rightarrow (\Delta'; \Gamma' \vdash \tau')$, then $\vdash C \approx_{\mathcal{E} \Rightarrow \mathcal{V}}^{log} C : (\Delta; \Gamma \vdash \theta) \Rightarrow (\Delta'; \Gamma' \vdash \tau')$.
3. If $\vdash C : (\Delta; \Gamma \vdash \tau) \Rightarrow (\Delta'; \Gamma' \vdash \theta')$, then $\vdash C \approx_{\mathcal{V} \Rightarrow \mathcal{E}}^{log} C : (\Delta; \Gamma \vdash \tau) \Rightarrow (\Delta'; \Gamma' \vdash \theta')$.
4. If $\vdash C : (\Delta; \Gamma \vdash \tau) \Rightarrow (\Delta'; \Gamma' \vdash \tau')$, then $\vdash C \approx_{\mathcal{V} \Rightarrow \mathcal{V}}^{log} C : (\Delta; \Gamma \vdash \tau) \Rightarrow (\Delta'; \Gamma' \vdash \tau')$.

Proof

By induction on the context typing derivation, applying appropriate compatibility at each step. \square

7.2 Sound and Complete

Theorem 7.40 (Contextual Equivalence Implies CIU Equivalence)

If $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ctx} e_2 : \theta$, then $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ciu} e_2 : \theta$.

Proof

Since $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ctx} e_2 : \theta$, $\Delta; \Gamma \vdash e_1 : \theta$ and $\Delta; \Gamma \vdash e_2 : \theta$.

Suppose $\Delta \vDash \delta, \delta, \Gamma \vDash \gamma$ and $\vdash K : (\cdot; \cdot \vdash \theta) \Rightarrow (\cdot; \cdot \vdash \mathbf{1})$. We seek to prove that $K[\delta(\gamma(e_1))] \Downarrow K[\delta(\gamma(e_2))]$.

First, we split Γ into $\mathbf{\Gamma} = \{(x_1 : \sigma_1), \dots, (x_n : \sigma_n)\}$ and $\mathbf{\Gamma} = \{(x_1 : \tau_1), \dots, (x_m : \tau_m)\}$ and define $\{\alpha_1, \dots, \alpha_p\} = \Delta$.

For each x_i , define $C_i = \text{let } x_i = \gamma(x_i) \text{ in } [\cdot]$ and for each x_i , define $\mathbf{C}_i = \text{let } x_i = \text{return}_0 \gamma(x_i) \text{ in } [\cdot]$. Next, for each α_i , define $\mathbf{C}_{m+i} = (\lambda[\alpha_i](y : \mathbf{1}). [\cdot]) [\delta(\alpha_i)] \langle \cdot \rangle$. Finally, define

$$\mathbf{C} = {}^1\text{ST } \mathbf{C}_{m+1} [\dots \mathbf{C}_{m+p} [\mathbf{C}_1 [\dots \mathbf{C}_m [{}^1\text{TS} \mathbf{C}_1 [\dots \mathbf{C}_n [K] \dots]] \dots]] \dots]$$

Then $\vdash \mathbf{C} : (\Delta; \Gamma \vdash \theta) \Rightarrow (\cdot; \cdot \vdash \mathbf{1})$, so since e_1, e_2 are contextually equivalent, $\mathbf{C}[e_1] \Downarrow \mathbf{C}[e_2]$. Furthermore, $\mathbf{C}[e_1] \mapsto^{p+m+n} {}^1\text{ST } {}^1\text{TS} \mathbf{C}_1 [\delta(\gamma(e_1))]$, so $\mathbf{C}[e_1] \Downarrow {}^1\text{ST } {}^1\text{TS} \mathbf{C}_1 [\delta(\gamma(e_1))]$. Finally, by definition of the operational semantics, ${}^1\text{ST } {}^1\text{TS} \mathbf{C}_1 [\delta(\gamma(e_1))] \Downarrow K[\delta(\gamma(e_1))]$, so $\mathbf{C}[e_1] \Downarrow K[\delta(\gamma(e_1))]$. By analogous reasoning $\mathbf{C}[e_2] \Downarrow K[\delta(\gamma(e_2))]$.

Therefore, by transitivity of \Downarrow , $K[\delta(\gamma(e_1))] \Downarrow K[\delta(\gamma(e_2))]$. \square

Theorem 7.41 (CIU Equivalence Implies Logically Related)

If $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ciu} e_2 : \theta$, then $\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{log} e_2 : \theta$.

Proof

Since $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ciu} e_2 : \theta$, $\Delta; \Gamma \vdash e_1 : \theta$ and $\Delta; \Gamma \vdash e_2 : \theta$.

Suppose $(k, K_1, K_2) \in \mathcal{K} \llbracket \theta \rrbracket \rho$, we seek to prove that $(k, K_1[\rho_1(\gamma_1(e_1))], K_2[\rho_2(\gamma_2(e_2))]) \in \mathcal{O}$.

Using $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{ciu} e_2 : \theta$ twice and Theorem 7.38 twice, we get

1. $K_1[\rho_1(\gamma_1(e_1))] \Downarrow K_1[\rho_1(\gamma_1(e_2))]$
2. $K_2[\rho_2(\gamma_2(e_1))] \Downarrow K_2[\rho_2(\gamma_2(e_2))]$
3. $(k, K_1[\rho_1(\gamma_1(e_1))], K_2[\rho_2(\gamma_2(e_1))]) \in \mathcal{O}$
4. $(k, K_1[\rho_1(\gamma_1(e_2))], K_2[\rho_2(\gamma_2(e_2))]) \in \mathcal{O}$

By case analysis of 3:

Case $K_1[\rho_1(\gamma_1(e_1))] \Downarrow \wedge K_2[\rho_2(\gamma_2(e_1))] \Downarrow$: then by 2, $K_2[\rho_2(\gamma_2(e_2))] \Downarrow$.

Case running $(k, K_1[\rho_1(\gamma_1(e_1))]) \wedge$ running $(k, K_2[\rho_2(\gamma_2(e_2))])$: By case analysis of 4:

Case $K_1[\rho_1(\gamma_1(e_2))] \Downarrow \wedge K_2[\rho_2(\gamma_2(e_2))] \Downarrow$: then by 1, $K_1[\rho_1(\gamma_1(e_1))] \Downarrow$.

Case $\text{running}(k, K_2[\rho_2(\gamma_2(e_2))]) \wedge \text{running}(k, K_1[\rho_1(\gamma_1(e_2))])$: then we have precisely that $\text{running}(k, K_1[\rho_1(\gamma_1(e_1))]) \wedge \text{running}(k, K_2[\rho_2(\gamma_2(e_2))])$.

□

Theorem 7.42 (Logically Related Implies Contextual Equivalence)

If $\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{\text{log}} e_2 : \theta$, then $\Delta; \Gamma \vdash e_1 \approx_{\text{ST}}^{\text{ctx}} e_2 : \theta$.

Proof

Since $\Delta; \Gamma \vdash e_1 \approx_{\mathcal{E}}^{\text{log}} e_2 : \theta$, $\Delta; \Gamma \vdash e_1 : \theta$ and $\Delta; \Gamma \vdash e_2 : \theta$.

Suppose $\vdash C : (\Delta; \Gamma \vdash \theta) \Rightarrow (\cdot; \cdot \vdash \mathbf{1})$. Then by Lemma 7.39, $\cdot; \cdot \vdash C[e_1] \approx_{\mathcal{E}}^{\text{log}} C[e_2] : \mathbf{1}$.

We seek to prove that $C[e_1] \Downarrow C[e_2]$. Suppose $C[e_1] \Downarrow$. Then in particular there exists some $k \geq 0$ such that $\neg \text{running}(C[e_1], k)$. Furthermore, since $\cdot; \cdot \vdash C[e_1] \approx_{\mathcal{E}}^{\text{log}} C[e_2] : \mathbf{1}$, $(k, C[e_1], C[e_2]) \in \mathcal{O}$, so since $\neg \text{running}(C[e_1], k)$, $C[e_2] \Downarrow$. By symmetric reasoning, if $C[e_2] \Downarrow$, then $C[e_1] \Downarrow$. □

Theorem 7.43 (Logical Relation, Contextual Equivalence, CIU Equivalence Coincide)

$\Delta; \Gamma \vdash e \approx_{\mathcal{E}}^{\text{log}} e' : \theta$ if and only if $\Delta; \Gamma \vdash e \approx_{\text{ST}}^{\text{ctx}} e' : \theta$ if and only if $\Delta; \Gamma \vdash e \approx_{\text{ST}}^{\text{ciu}} e' : \theta$

Proof

By Lemma 7.40, Lemma 7.41, and Lemma 7.42. □

Theorem 7.44 (Logical Relation is Transitive)

If $\Delta; \Gamma \vdash e \approx_{\mathcal{E}}^{\text{log}} e' : \theta$ and $\Delta; \Gamma \vdash e' \approx_{\mathcal{E}}^{\text{log}} e'' : \theta$, then $\Delta; \Gamma \vdash e \approx_{\mathcal{E}}^{\text{log}} e'' : \theta$.

Proof

By Theorem 7.43 and transitivity of contextual equivalence. □

8 Back-Translation From λ^{ST} to λ^{S}

$$\begin{array}{ll}
 \delta & ::= \emptyset \mid \delta[\alpha \mapsto \sigma, \mathbf{x}] \\
 \emptyset_{\Gamma} & \stackrel{\text{def}}{=} . \\
 (\delta[\alpha \mapsto \sigma, \mathbf{x}])_{\Gamma} & \stackrel{\text{def}}{=} \delta_{\Gamma}, \mathbf{x} : \mathbf{1} \rightarrow ((\sigma \rightarrow \mathbf{U}) \times (\mathbf{U} \rightarrow \sigma)) \\
 \emptyset_{\sigma} & \stackrel{\text{def}}{=} \emptyset \\
 (\delta[\alpha \mapsto \sigma, \mathbf{x}])_{\sigma} & \stackrel{\text{def}}{=} \delta_{\sigma}[\alpha \mapsto \sigma] \\
 \emptyset_{\mathbf{x}} & \stackrel{\text{def}}{=} \emptyset \\
 (\delta[\alpha \mapsto \sigma, \mathbf{x}])_{\mathbf{x}} & \stackrel{\text{def}}{=} \delta_{\mathbf{x}}[\alpha \mapsto \mathbf{x}]
 \end{array}$$

Figure 21: Embedding-Projection Environment

$$\begin{array}{ll}
 \mathbf{U} & \stackrel{\text{def}}{=} \mu\alpha. \mathbf{1} + (\alpha + \alpha) + (\alpha \times \alpha) + (\alpha \rightarrow \mathbf{R}(\alpha)) + \alpha \\
 \mathbf{R}(\sigma) & \stackrel{\text{def}}{=} \sigma + \sigma \\
 \mathbf{R} & \stackrel{\text{def}}{=} \mathbf{R}(\mathbf{U})
 \end{array}$$

Figure 22: Universal Type and Result Type

if $\Delta; \Gamma \vdash v_f : (\sigma_1 \rightarrow \sigma_2) \rightarrow \sigma_1 \rightarrow \sigma_2$, then $\Delta; \Gamma \vdash \text{FIX}_{\sigma_1 \rightarrow \sigma_2}(v_f) : \sigma_1 \rightarrow \sigma_2$

if $\Delta; \Gamma \vdash v_f : (\sigma_1 \rightarrow \sigma_2) \rightarrow \sigma_1 \rightarrow \sigma_2$, then $\Delta; \Gamma \vdash \text{LOOP}(\sigma_1 \rightarrow \sigma_2, v_f) : (\mu\alpha. \alpha \rightarrow \sigma_1 \rightarrow \sigma_2) \rightarrow \sigma_1 \rightarrow \sigma_2$

$\text{FIX}_{\sigma_1 \rightarrow \sigma_2}(v_f)$	$\stackrel{\text{def}}{=} \lambda(z : \sigma_1). \text{let } x_{fix} = \text{LOOP}(\sigma_1 \rightarrow \sigma_2, v_f) (\text{fold}_{\mu\alpha. \alpha \rightarrow \sigma_1 \rightarrow \sigma_2} \text{LOOP}(\sigma_1 \rightarrow \sigma_2, v_f)) \text{ in } \text{let } x_f = v_f \ x_{fix} \text{ in } x_f \ z$
$\text{LOOP}(\sigma_1 \rightarrow \sigma_2, v_f)$	$\stackrel{\text{def}}{=} \lambda(x_{folded} : \mu\alpha. \alpha \rightarrow \sigma_1 \rightarrow \sigma_2). \text{let } x_{loop} = \text{unfold } x_{folded} \text{ in } \lambda(z : \sigma_1). \text{let } x_{fix} = x_{loop} (\text{fold}_{\mu\alpha. \alpha \rightarrow \sigma_1 \rightarrow \sigma_2} x_{loop}) \text{ in } \text{let } x_f = v_f \ x_{fix} \text{ in } x_f \ z$
UNIT	$\stackrel{\text{def}}{=} \text{fold}_U(\text{inj}_1 \langle \rangle)$
$\text{IN}(i, v)$	$\stackrel{\text{def}}{=} \text{fold}_U(\text{inj}_2(\text{inj}_1(\text{inj}_i v)))$
$\text{CONS}(v_1, v_2)$	$\stackrel{\text{def}}{=} \text{fold}_U(\text{inj}_2(\text{inj}_2(\text{inj}_1 \langle v_1, v_2 \rangle)))$
$\text{LAMBDA}(\lambda(x : U). e)$	$\stackrel{\text{def}}{=} \text{fold}_U(\text{inj}_2(\text{inj}_2(\text{inj}_2(\text{inj}_1(\lambda(x : U). e))))))$
$\text{FOLD}(v)$	$\stackrel{\text{def}}{=} \text{fold}_U(\text{inj}_2(\text{inj}_2(\text{inj}_2(\text{inj}_2(v))))))$
$\text{RETURN}(v)$	$\stackrel{\text{def}}{=} \text{inj}_1 v$
$\text{RAISE}(v)$	$\stackrel{\text{def}}{=} \text{inj}_2 v$
$\text{TOLHS}(v_u)$	$\stackrel{\text{def}}{=} \text{case } v_u \text{ of } x_1. x_1 \mid x_2. \bar{U}$
$\text{TORHS}(v_u)$	$\stackrel{\text{def}}{=} \text{case } v_u \text{ of } x_1. \bar{U} \mid x_2. x_2$
$\text{TOSUM}(v_u)$	$\stackrel{\text{def}}{=} \text{let } x_1 = \text{unfold } v_u \text{ in } \text{let } x_2 = \text{TORHS}(x_1) \text{ in } \text{TOLHS}(x_2)$
$\text{TOPAIR}(v_u)$	$\stackrel{\text{def}}{=} \text{let } x_1 = \text{unfold } v_u \text{ in } \text{let } x_2 = \text{TORHS}(x_1) \text{ in } \text{let } x_3 = \text{TORHS}(x_2) \text{ in } \text{TOLHS}(x_3)$
$\text{TOFUN}(v_u)$	$\stackrel{\text{def}}{=} \text{let } x_1 = \text{unfold } v_u \text{ in } \text{let } x_2 = \text{TORHS}(x_1) \text{ in } \text{let } x_3 = \text{TORHS}(x_2) \text{ in } \text{let } x_4 = \text{TORHS}(x_3) \text{ in } \text{TOLHS}(x_4)$
$\text{TOFOLD}(v_u)$	$\stackrel{\text{def}}{=} \text{let } x_1 = \text{unfold } v_u \text{ in } \text{let } x_2 = \text{TORHS}(x_1) \text{ in } \text{let } x_3 = \text{TORHS}(x_2) \text{ in } \text{let } x_4 = \text{TORHS}(x_3) \text{ in } \text{TORHS}(x_4)$
$\text{PRJ}(1, v_u)$	$\stackrel{\text{def}}{=} \text{let } x = \text{TOPAIR}(v_u) \text{ in } \pi_1 x$
$\text{PRJ}(i + 1, v_u)$	$\stackrel{\text{def}}{=} \text{let } x = \text{TOPAIR}(v_u) \text{ in } \text{let } y = \pi_2 x \text{ in } \text{PRJ}(i, y)$

Figure 23: Interpreter Metafunctions

$$\emptyset \vdash \text{PROJECT}(\sigma) : R \rightarrow \sigma$$
$$\delta_\Gamma \vdash \text{PROJECT}(\delta, \sigma) : U \rightarrow \delta_\sigma(\sigma)$$

$\text{PROJECT}(\sigma)$	$\stackrel{\text{def}}{=} \lambda(x_r : R). \text{let } x_u = \text{TOLHS}(x_r) \text{ in } \text{PROJECT}(\emptyset, \sigma) x_u$
$\text{PROJECT}(\delta, \alpha)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \text{let } x = \delta_x(\alpha) \langle \rangle \text{ in}$ $\text{let } x_f = \pi_2 x \text{ in } x' x$
$\text{PROJECT}(\delta, 1)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \langle \rangle$
$\text{PROJECT}(\delta, \sigma_1 + \sigma_2)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \text{let } x = \text{TOSUM}(x_u) \text{ in}$ $\text{case } x \text{ of}$ $x_1 . \text{let } x'_1 = \text{PROJECT}(\delta, \sigma_1) x_1 \text{ in } \text{inj}_1 x'_1$ $x_2 . \text{let } x'_2 = \text{PROJECT}(\delta, \sigma_2) x_2 \text{ in } \text{inj}_2 x'_2$
$\text{PROJECT}(\delta, \sigma_1 \times \sigma_2)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \text{let } x = \text{TOPAIR}(x_u) \text{ in}$ $\text{let } x_1 = \pi_1 x \text{ in}$ $\text{let } x'_1 = \text{PROJECT}(\delta, \sigma_1) x_1 \text{ in}$ $\text{let } y = \pi_2 x \text{ in}$ $\text{let } y' = \text{TOPAIR}(y) \text{ in}$ $\text{let } x_2 = \pi_1 y' \text{ in}$ $\text{let } x'_2 = \text{PROJECT}(\delta, \sigma_2) x_2 \text{ in}$ $\langle x'_1, x'_2 \rangle$
$\text{PROJECT}(\delta, \sigma_1 \rightarrow \sigma_2)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \text{let } x'_u = \text{TOPAIR}(x_u) \text{ in}$ $\text{let } x_f = \text{PRJ}(1, x'_u) \text{ in}$ $\text{let } x_{env} = \text{PRJ}(2, x'_u) \text{ in}$ $\lambda(y : \delta_\sigma(\sigma_1)). \text{let } y_u = \text{EMBED}(\delta, \sigma_1) y \text{ in}$ $\text{let } x = \text{CONS}(x_{env}, \text{CONS}(y_u, \text{UNIT})) \text{ in}$ $\text{let } x_r = x_f x \text{ in}$ $\text{let } x''_u = \text{TOLHS}(x_r) \text{ in}$ $\text{PROJECT}(\delta, \sigma_2) x''_u$
$\text{PROJECT}(\delta, \mu\alpha. \sigma)$	$\stackrel{\text{def}}{=} \lambda(x_u : U). \text{let } x = \text{EP}(\delta, \mu\alpha. \sigma) \langle \rangle \text{ in}$ $\text{let } x_f = \pi_2 x \text{ in } x_f x_u$

Figure 24: Projecting from the Universal Type

$$\emptyset \vdash \text{EMBED}(\sigma) : \sigma \rightarrow \mathbf{R}$$

$$\delta_\Gamma \vdash \text{EMBED}(\delta, \sigma) : \delta_\sigma(\sigma) \rightarrow \mathbf{U}$$

$$\begin{aligned} \text{EMBED}(\sigma) &\stackrel{\text{def}}{=} \lambda(x : \sigma). \text{let } x_u = \text{EMBED}(\emptyset, \sigma) \ x \text{ in } \text{RETURN}(x_u) \\ \text{EMBED}(\delta, \alpha) &\stackrel{\text{def}}{=} \lambda(x : \delta_\sigma(\alpha)). \text{let } x_{ep} = \delta_x(\alpha) \ \langle \rangle \text{ in} \\ &\quad \text{let } x_{embed} = \pi_1 x_{ep} \text{ in } x_{embed} \times \\ \text{EMBED}(\delta, 1) &\stackrel{\text{def}}{=} \lambda(x : \delta_\sigma(1)). \text{UNIT} \\ \text{EMBED}(\delta, \sigma_1 + \sigma_2) &\stackrel{\text{def}}{=} \lambda(x : \delta_\sigma(\sigma_1 + \sigma_2)). \text{case } x \text{ of} \\ &\quad x_1 . \text{let } x' = \text{EMBED}(\delta, \sigma_1) \ x_1 \text{ in} \\ &\quad \quad \text{IN}(1, x') \\ &\quad x_2 . \text{let } x' = \text{EMBED}(\delta, \sigma_2) \ x_2 \text{ in} \\ &\quad \quad \text{IN}(2, x') \\ \text{EMBED}(\delta, \sigma_1 \times \sigma_2) &\stackrel{\text{def}}{=} \lambda(x : \delta_\sigma(\sigma_1 \times \sigma_2)). \text{let } x_1 = \pi_1 x \text{ in} \\ &\quad \text{let } x_2 = \pi_2 x \text{ in} \\ &\quad \text{let } x'_1 = \text{EMBED}(\delta, \sigma_1) \ x_1 \text{ in} \\ &\quad \text{let } x'_2 = \text{EMBED}(\delta, \sigma_2) \ x_2 \text{ in} \\ &\quad \text{CONS}(x'_1, \text{CONS}(x'_2, \text{UNIT})) \\ \text{EMBED}(\delta, \sigma_1 \rightarrow \sigma_2) &\stackrel{\text{def}}{=} \lambda(x_f : \delta_\sigma(\sigma_1 \rightarrow \sigma_2)). \text{let } x'_f = \lambda(x_u : \mathbf{U}). \text{let } x'_u = \text{PRJ}(2, x_u) \text{ in} \quad \text{in} \\ &\quad \text{let } x = \text{PROJECT}(\delta, \sigma_1) \ x'_u \text{ in} \\ &\quad \text{let } y = x_f \ x \text{ in} \\ &\quad \text{let } x''_u = \text{EMBED}(\delta, \sigma_2) \ y \text{ in} \\ &\quad \text{RETURN}(x''_u) \\ &\quad \text{CONS}(x'_f, \text{CONS}(\text{UNIT}, \text{UNIT})) \\ \text{EMBED}(\delta, \mu\alpha. \sigma) &\stackrel{\text{def}}{=} \lambda(x : \delta_\sigma(\mu\alpha. \sigma)). \text{let } x_{ep} = \text{EP}(\delta, \mu\alpha. \sigma) \ \langle \rangle \text{ in} \\ &\quad \text{let } x_{embed} = \pi_1 x_{ep} \text{ in } x_{embed} \times \end{aligned}$$

Figure 25: Embedding into the Universal Type

$$\delta_\Gamma \vdash \text{EP}(\delta, \mu\alpha. \sigma) : \mathbf{1} \rightarrow ((\delta_\sigma(\mu\alpha. \sigma) \rightarrow \mathbf{U}) \times (\mathbf{U} \rightarrow \delta_\sigma(\mu\alpha. \sigma)))$$

$$\begin{aligned} \text{EP}(\delta, \mu\alpha. \sigma) &\stackrel{\text{def}}{=} \text{FIX}_{\mathbf{1} \rightarrow ((\delta_\sigma(\mu\alpha. \sigma) \rightarrow \mathbf{U}) \times (\mathbf{U} \rightarrow \delta_\sigma(\mu\alpha. \sigma)))} \\ &\quad \lambda(x_{\mu\alpha. \sigma} : \mathbf{1} \rightarrow ((\delta_\sigma(\mu\alpha. \sigma) \rightarrow \mathbf{U}) \times (\mathbf{U} \rightarrow \delta_\sigma(\mu\alpha. \sigma))))). \\ &\quad \lambda(x_{unit} : \mathbf{1}). \\ &\quad \text{let } x_{embed} = \\ &\quad \quad \lambda(x : \delta_\sigma(\mu\alpha. \sigma)). \\ &\quad \quad \text{let } y = \text{unfold } x \text{ in} \\ &\quad \quad \text{let } y_u = \text{EMBED}(\delta[\alpha \mapsto \mu\alpha. \sigma, x_{\mu\alpha. \sigma}], \sigma) \ y \text{ in} \\ &\quad \quad \text{FOLD}(y_u) \\ &\quad \text{in let } x_{project} = \\ &\quad \quad \lambda(x_u : \mathbf{U}). \\ &\quad \quad \text{let } y_u = \text{TOFOLD}(x_u) \text{ in} \\ &\quad \quad \text{let } y = \text{PROJECT}(\delta[\alpha \mapsto \mu\alpha. \sigma, x_{\mu\alpha. \sigma}], \sigma) \ y_u \text{ in} \\ &\quad \quad \text{fold}_{\mu\alpha. \sigma} \ y \\ &\quad \text{in } \langle x_{embed}, x_{project} \rangle \end{aligned}$$

Figure 26: Embedding-Projection Pair for Recursive Types

$$\boxed{\rightarrow : \Gamma \rightarrow \Gamma}$$

$$\begin{aligned} (\cdot)^{\rightarrow} &= \cdot \\ (\Gamma, \mathbf{x} : \sigma)^{\rightarrow} &= \Gamma^{\rightarrow}, \mathbf{x} : \sigma \\ (\Gamma, \mathbf{y} : \tau)^{\rightarrow} &= \Gamma^{\rightarrow}, \mathbf{y} : \mathbf{U} \end{aligned}$$

$$\boxed{\Delta; \Gamma \vdash \mathbf{e} : \sigma \rightarrow \mathbf{e}'}$$

where $\mathbf{e}' \in \lambda^{\mathbf{S}}$ and $\Delta; \Gamma \vdash \mathbf{e} : \sigma$ and $\Gamma^{\rightarrow} \vdash \mathbf{e}' : \sigma$

$$\begin{array}{c} \frac{}{\Delta; \Gamma \vdash \mathbf{x} : \sigma \rightarrow \mathbf{x}} \qquad \frac{\Delta; \Gamma \vdash \mathbf{v} : \sigma_i \rightarrow \mathbf{v}' \quad i \in \{1, 2\}}{\Delta; \Gamma \vdash \text{inj}_i \mathbf{v} : \sigma_1 + \sigma_2 \rightarrow \text{inj}_i \mathbf{v}'} \\ \frac{\Delta; \Gamma \vdash \mathbf{v} : \sigma_1 + \sigma_2 \rightarrow \mathbf{v}' \quad \Delta; \Gamma, \mathbf{x}_1 : \sigma_1 \vdash \mathbf{e}_1 : \sigma \rightarrow \mathbf{e}'_1 \quad \Delta; \Gamma, \mathbf{x}_2 : \sigma_2 \vdash \mathbf{e}_2 : \sigma \rightarrow \mathbf{e}'_2}{\Delta; \Gamma \vdash \text{case } \mathbf{v} \text{ of } \mathbf{x}_1. \mathbf{e}_1 \mid \mathbf{x}_2. \mathbf{e}_2 : \sigma \rightarrow \text{case } \mathbf{v}' \text{ of } \mathbf{x}_1. \mathbf{e}'_1 \mid \mathbf{x}_2. \mathbf{e}'_2} \\ \frac{\Delta; \Gamma \vdash \mathbf{v}_1 : \sigma_1 \rightarrow \mathbf{v}'_1 \quad \Delta; \Gamma \vdash \mathbf{v}_2 : \sigma_2 \rightarrow \mathbf{v}'_2}{\Delta; \Gamma \vdash \langle \mathbf{v}_1, \mathbf{v}_2 \rangle : \sigma_1 \times \sigma_2 \rightarrow \langle \mathbf{v}'_1, \mathbf{v}'_2 \rangle} \qquad \frac{\Delta; \Gamma \vdash \mathbf{v} : \sigma_1 \times \sigma_2 \rightarrow \mathbf{v}' \quad i \in \{1, 2\}}{\Delta; \Gamma \vdash \pi_i \mathbf{v} : \sigma_i \rightarrow \pi_i \mathbf{v}'} \\ \frac{\Delta; \Gamma, \mathbf{x} : \sigma_1 \vdash \mathbf{e} : \sigma_2 \rightarrow \mathbf{e}'}{\Delta; \Gamma \vdash \lambda(\mathbf{x} : \sigma_1). \mathbf{e} : \sigma_1 \rightarrow \sigma_2 \rightarrow \lambda(\mathbf{x} : \sigma_1). \mathbf{e}'} \qquad \frac{\Delta; \Gamma \vdash \mathbf{v}_1 : \sigma_2 \rightarrow \sigma \rightarrow \mathbf{v}'_1 \quad \Delta; \Gamma \vdash \mathbf{v}_2 : \sigma_2 \rightarrow \mathbf{v}'_2}{\Delta; \Gamma \vdash \mathbf{v}_1 \mathbf{v}_2 : \sigma \rightarrow \mathbf{v}'_1 \mathbf{v}'_2} \\ \frac{\Delta; \Gamma \vdash \mathbf{v} : \sigma[\mu\alpha. \sigma/\alpha] \rightarrow \mathbf{v}'}{\Delta; \Gamma \vdash \text{fold}_{\mu\alpha. \sigma} \mathbf{v} : \mu\alpha. \sigma \rightarrow \text{fold}_{\mu\alpha. \sigma} \mathbf{v}'} \qquad \frac{\Delta; \Gamma \vdash \mathbf{v} : \mu\alpha. \sigma \rightarrow \mathbf{v}'}{\Delta; \Gamma \vdash \text{unfold } \mathbf{v} : \sigma[\mu\alpha. \sigma/\alpha] \rightarrow \text{unfold } \mathbf{v}'} \\ \frac{\Delta; \Gamma \vdash \mathbf{e}_1 : \sigma_1 \rightarrow \mathbf{e}'_1 \quad \Delta; \Gamma, \mathbf{x} : \sigma_1 \vdash \mathbf{e}_2 : \sigma_2 \rightarrow \mathbf{e}'_2}{\Delta; \Gamma \vdash \text{let } \mathbf{x} = \mathbf{e}_1 \text{ in } \mathbf{e}_2 : \sigma_2 \rightarrow \text{let } \mathbf{x} = \mathbf{e}'_1 \text{ in } \mathbf{e}'_2} \qquad \frac{\Delta; \Gamma \vdash^{\dagger} \mathbf{e} : \sigma^{\dagger} \rightarrow \mathbf{e}_u}{\Delta; \Gamma \vdash^{\sigma} \mathcal{ST} \mathbf{e} : \sigma \rightarrow \text{let } \mathbf{x} = \mathbf{e}_u \text{ in } \text{PROJECT}(\sigma) \mathbf{x}}$$

Figure 27: Relating $\lambda^{\mathbf{ST}}$ terms to $\lambda^{\mathbf{S}}$ terms (“Back-Translation”)

$\Delta; \Gamma \vdash^+ \mathbf{v} : \tau \rightarrow \mathbf{v}$ where $\mathbf{v} \in \lambda^S$ and $\Delta; \Gamma \vdash \mathbf{v} : \tau$ and $\Gamma \rightarrow \vdash \mathbf{v} : \mathbf{U}$

$$\begin{array}{c}
\frac{}{\Delta; \Gamma \vdash^+ \mathbf{y} : \sigma^+ \rightarrow \mathbf{y}} \quad \frac{}{\Delta; \Gamma \vdash^+ \langle \rangle : \langle \rangle \rightarrow \text{UNIT}} \quad \frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau_i \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^+ \text{inj}_i \mathbf{v} : \tau_1 + \tau_2 \rightarrow \text{IN}(i, \mathbf{v}_u)} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v}_1 : \tau \rightarrow \mathbf{v} \quad \Delta; \Gamma \vdash^+ \langle \mathbf{v}_1, \dots, \mathbf{v}_n \rangle : \langle \tau_1, \dots, \tau_n \rangle \rightarrow \mathbf{v}'}{\Delta; \Gamma \vdash^+ \langle \mathbf{v}, \mathbf{v}_1, \dots, \mathbf{v}_n \rangle : \langle \tau, \tau_1, \dots, \tau_n \rangle \rightarrow \text{CONS}(\mathbf{v}, \mathbf{v}')} \\
\frac{\alpha; \mathbf{x} : \tau \vdash^{\dot{+}} \mathbf{e} : \theta \rightarrow \mathbf{e}_u}{\Delta; \Gamma \vdash^+ \lambda[\alpha](\mathbf{x} : \tau). \mathbf{e} : \forall[\alpha]. \tau \rightarrow \theta \rightarrow \text{LAMBDA}(\lambda(\mathbf{x} : \mathbf{U}). \mathbf{e}_u)} \quad \frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau[\mu\alpha. \tau/\alpha] \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^+ \text{fold}_{\mu\alpha. \tau} \mathbf{v} : \mu\alpha. \tau \rightarrow \text{FOLD}(\mathbf{v}_u)} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau[\tau'/\alpha] \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^+ \text{pack}(\tau', \mathbf{v}) \text{ as } \exists\alpha. \tau : \exists\alpha. \tau \rightarrow \mathbf{v}_u}
\end{array}$$

$\Delta; \Gamma \vdash^{\dot{+}} \mathbf{r} : \theta \rightarrow \mathbf{v}_u$ where $\mathbf{e} \in \lambda^S$ and $\Delta; \Gamma \vdash \mathbf{r} : \theta$ and $\Gamma \rightarrow \vdash \mathbf{v}_u : \mathbf{R}$

$$\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^{\dot{+}} \text{return } \mathbf{v} : \mathbf{E} \tau_{\text{exn}} \tau \rightarrow \text{RETURN}(\mathbf{v}_u)} \quad \frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau_{\text{exn}} \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^{\dot{+}} \text{raise } \mathbf{v} : \mathbf{E} \tau_{\text{exn}} \tau \rightarrow \text{RAISE}(\mathbf{v}_u)}$$

$\Delta; \Gamma \vdash^{\dot{+}} \mathbf{e} : \theta \rightarrow \mathbf{e}$ where $\mathbf{e} \in \lambda^S$ and $\Delta; \Gamma \vdash \mathbf{e} : \theta$ and $\Gamma \rightarrow \vdash \mathbf{e} : \mathbf{R}$

$$\begin{array}{c}
\frac{\Delta; \Gamma \vdash \mathbf{e} : \sigma \rightarrow \mathbf{e}'}{\Delta; \Gamma \vdash^{\dot{+}} \mathcal{TS}^\sigma \mathbf{e} : \sigma^{\dot{+}} \rightarrow \text{let } \mathbf{x} = \mathbf{e}' \text{ in } \text{EMBED}(\sigma) \mathbf{x}} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \tau_1 + \tau_2 \rightarrow \mathbf{v}_u \quad \Delta; \Gamma, \mathbf{x}_1 : \tau_1 \vdash^{\dot{+}} \mathbf{e}_1 : \theta \rightarrow \mathbf{e}_1 \quad \Delta; \Gamma, \mathbf{x}_1 : \tau_2 \vdash^{\dot{+}} \mathbf{e}_2 : \theta \rightarrow \mathbf{e}_2}{\Delta; \Gamma \vdash^{\dot{+}} \text{case } \mathbf{v} \text{ of } \mathbf{x}_1. \mathbf{e}_1 \mid \mathbf{x}_2. \mathbf{e}_2 : \theta \rightarrow \text{let } \mathbf{x} = \text{TOSUM}(\mathbf{v}_u) \text{ in case } \mathbf{x} \text{ of } \mathbf{x}_1. \mathbf{e}_1 \mid \mathbf{x}_2. \mathbf{e}_2} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \langle \tau_1, \dots, \tau_n \rangle \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^{\dot{+}} \mathbf{v}.i : \langle \tau_1, \dots, \tau_n \rangle \rightarrow \text{let } \mathbf{x} = \text{PRJ}(i, \mathbf{v}_u) \text{ in } \text{RETURN}(\mathbf{x})} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v}_1 : \forall[\alpha]. \tau \rightarrow \theta \rightarrow \mathbf{v}_1 \quad \Delta; \Gamma \vdash^+ \mathbf{v}_2 : \tau[\tau'/\alpha] \rightarrow \mathbf{v}_2}{\Delta; \Gamma \vdash^{\dot{+}} \mathbf{v}_1 [\tau'] \mathbf{v}_2 : \theta[\tau'/\alpha] \rightarrow \text{let } \mathbf{x} = \text{TOFUN}(\mathbf{v}_1) \text{ in } \mathbf{x} \mathbf{v}_2} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \mathbf{E} \tau_{\text{exn}} \mu\alpha. \tau \rightarrow \mathbf{v}_u}{\Delta; \Gamma \vdash^{\dot{+}} \text{unfold } \mathbf{v} : \tau[\mu\alpha. \tau/\alpha] \rightarrow \text{let } \mathbf{x} = \text{TOFOLD}(\mathbf{v}_u) \text{ in } \text{RETURN}(\mathbf{x})} \\
\frac{\Delta; \Gamma \vdash^+ \mathbf{v} : \exists\alpha. \tau \rightarrow \mathbf{v}_u \quad \Delta, \alpha; \Gamma, \mathbf{x} : \tau \vdash^{\dot{+}} \mathbf{e} : \theta \rightarrow \mathbf{e}_u}{\Delta; \Gamma \vdash^{\dot{+}} \text{unpack}(\alpha, \mathbf{x}) = \mathbf{v} \text{ in } \mathbf{e} : \theta \rightarrow \text{let } \mathbf{x} = \mathbf{v}_u \text{ in } \mathbf{e}_u} \\
\frac{\Delta; \Gamma \vdash^{\dot{+}} \mathbf{e} : \mathbf{E} \tau_{\text{exn}} \tau \rightarrow \mathbf{e} \quad \Delta; \Gamma, \mathbf{x}_1 : \tau \vdash^{\dot{+}} \mathbf{e}_1 : \theta \rightarrow \mathbf{e}_1 \quad \Delta; \Gamma, \mathbf{x}_2 : \tau_{\text{exn}} \vdash^{\dot{+}} \mathbf{e}_2 : \theta \rightarrow \mathbf{e}_2}{\Delta; \Gamma \vdash^{\dot{+}} \text{handle } \mathbf{e} \text{ with } (\mathbf{x}_1. \mathbf{e}_1) (\mathbf{x}_2. \mathbf{e}_2) : \theta \rightarrow \text{let } \mathbf{x}_r = \mathbf{e} \text{ in case } \mathbf{x}_r \text{ of} \\
\mathbf{x}_1. \mathbf{e}_1 \\
\mathbf{x}_2. \mathbf{e}_2}
\end{array}$$

Figure 28: Relating λ^{ST} terms to λ^S terms

9 Back Translation Correctness

$$\begin{aligned}
\text{Atom}^V[\tau] &\stackrel{\text{def}}{=} \{(k, \mathbf{v}, \mathbf{v}) \mid k \in \mathbb{N} \wedge \cdot \vdash \mathbf{v} : \mathbf{U} \wedge \cdot \vdash \mathbf{v} : \tau\} \\
\text{Atom}^R[\theta] &\stackrel{\text{def}}{=} \{(k, \mathbf{v}, \mathbf{r}) \mid k \in \mathbb{N} \wedge \cdot \vdash \mathbf{v} : \mathbf{R} \wedge \cdot \vdash \mathbf{r} : \theta\} \\
\text{Atom}^E[\theta] &\stackrel{\text{def}}{=} \{(k, \mathbf{e}, \mathbf{e}) \mid k \in \mathbb{N} \wedge \cdot \vdash \mathbf{e} : \mathbf{R} \wedge \cdot \vdash \mathbf{e} : \theta\} \\
\text{Atom}^K[\theta] &\stackrel{\text{def}}{=} \{(k, K_1, K_2) \mid k \in \mathbb{N} \wedge \exists \theta. \vdash K_1 : (\cdot \vdash \mathbf{R}) \Rightarrow (\cdot \vdash \theta) \wedge \vdash K_2 : (\cdot \vdash \theta) \Rightarrow (\cdot \vdash \theta)\} \\
\text{Rel}^U[\tau] &\stackrel{\text{def}}{=} \{R \in \mathcal{P}(\text{Atom}^V[\tau]) \mid \forall j \leq k, \mathbf{v}, \mathbf{v}. (k, \mathbf{v}, \mathbf{v}) \in R \implies (j, \mathbf{v}, \mathbf{v}) \in R\}
\end{aligned}$$

Figure 29: Universal Type Logical Relation Auxiliary Definitions

$$\begin{aligned}
\mathcal{V}^U[\tau]\rho^U &\subset \text{Atom}^V[\rho^U(\tau)] \\
\mathcal{V}^U[\alpha]\rho^U &\stackrel{\text{def}}{=} \rho^U_R(\alpha) \\
\mathcal{V}^U[\langle \rangle]\rho^U &\stackrel{\text{def}}{=} \{(k, \text{UNIT}, \langle \rangle)\} \\
\mathcal{V}^U[\tau_1 + \tau_2]\rho^U &\stackrel{\text{def}}{=} \{(k, \text{IN}(i, \mathbf{v}_u), \text{inj}_i \mathbf{v}) \mid \\
&\quad (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau_i]\rho^U\} \\
\mathcal{V}^U[\langle \tau_1, \tau_2, \dots, \tau_n \rangle]\rho^U &\stackrel{\text{def}}{=} \{(k, \text{CONS}(\mathbf{v}_u, \mathbf{v}'_u), \langle \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \rangle) \mid \\
&\quad (k, \mathbf{v}_u, \mathbf{v}_1) \in \mathcal{V}^U[\tau_1]\rho^U \wedge (k, \mathbf{v}'_u, \langle \mathbf{v}_2, \dots, \mathbf{v}_n \rangle) \in \mathcal{V}^U[\langle \tau_2, \dots, \tau_n \rangle]\rho^U\} \\
\mathcal{V}^U[\forall[\alpha]. \tau \rightarrow \theta]\rho^U &\stackrel{\text{def}}{=} \{(k, \text{LAMBDA}(\lambda(x_u : \mathbf{U}). e_u), \lambda[\alpha](\mathbf{x} : \tau). \mathbf{e}) \mid \\
&\quad \forall \tau', R \in \text{Rel}^U[\rho^U(\tau')], j \leq k, (j, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U. \\
&\quad (j, e_u[\mathbf{v}_u/x_u], \mathbf{e}[\tau'/\alpha][\mathbf{v}/\mathbf{x}]) \in \mathcal{E}^U[\theta]\rho^U \\
&\quad \text{where. } \rho^U = \rho^U[\alpha \mapsto \tau', R]\} \\
\mathcal{V}^U[\mu\alpha. \tau]\rho^U &\stackrel{\text{def}}{=} \{(0, \mathbf{v}_u, \mathbf{v})\} \\
&\quad \cup \\
&\quad \{(k + 1, \text{FOLD}(\mathbf{v}_u), \text{fold}_{\rho^U(\mu\alpha. \tau)} \mathbf{v}) \mid (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau[\mu\alpha. \tau/\alpha]]\rho^U\} \\
\mathcal{V}^U[0]\rho^U &\stackrel{\text{def}}{=} \emptyset \\
\mathcal{V}^U[\exists\alpha. \tau]\rho^U &\stackrel{\text{def}}{=} \{(k, \mathbf{v}_u, \text{pack}(\tau', \mathbf{v}) \text{ as } \rho^U(\exists\alpha. \tau)) \mid \\
&\quad \exists R \in \text{Rel}^U[\rho^U(\tau')]. (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U[\alpha \mapsto \tau', R]\} \\
\mathcal{R}^U[\theta]\rho^U &\subset \text{Atom}^R[\rho^U(\theta)] \\
\mathcal{R}^U[\mathbf{E} \tau_{\text{exn}} \tau]\rho^U &\stackrel{\text{def}}{=} \{(k, \text{RETURN}(\mathbf{v}_u), \text{return } \mathbf{v}) \mid (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U\} \\
&\quad \cup \\
&\quad \{(k, \text{RAISE}(\mathbf{v}_u), \text{raise } \mathbf{v}) \mid (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau_{\text{exn}}]\rho^U\} \\
\mathcal{E}^U[\theta]\rho^U &\subset \text{Atom}^E[\rho^U(\theta)] \\
\mathcal{E}^U[\theta]\rho^U &\stackrel{\text{def}}{=} \{(k, \mathbf{e}_u, \mathbf{e}) \mid \\
&\quad \forall j \leq k, K_1, K_2. (j, K_1, K_2) \in \mathcal{K}[\theta]\rho \implies (j, K_1[\mathbf{e}_u], K_2[\mathbf{e}]) \in \mathcal{O}\} \\
\mathcal{K}^U[\theta]\rho^U &\subset \text{Atom}^K[\rho^U(\theta)] \\
\mathcal{K}^U[\theta]\rho^U &\stackrel{\text{def}}{=} \{(k, K_1, K_2) \mid \\
&\quad \forall j \leq k, \mathbf{v}_u, \mathbf{r}. (j, \mathbf{v}_u, \mathbf{r}) \in \mathcal{R}[\theta]\rho \implies (j, K_1[\mathbf{v}_u], K_2[\mathbf{r}]) \in \mathcal{O}\} \\
\mathcal{D}^U[\cdot] &\stackrel{\text{def}}{=} \{\emptyset\} \\
\mathcal{D}^U[\Delta, \alpha] &\stackrel{\text{def}}{=} \{\rho^U[\alpha \mapsto \tau, R] \mid \rho^U \in \mathcal{D}[\Delta] \wedge R \in \text{Rel}^U[\tau]\} \\
\mathcal{G}^U[\cdot]\rho^U &\stackrel{\text{def}}{=} \{(k, \emptyset) \mid k \in \mathbb{N}\} \\
\mathcal{G}^U[\Gamma, \mathbf{x} : \sigma]\rho^U &\stackrel{\text{def}}{=} \{(k, \gamma^U[\mathbf{x} \mapsto \mathbf{v}_1, \mathbf{v}_2]) \mid (k, \gamma^U) \in \mathcal{G}^U[\Gamma]\rho^U \wedge (k, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V}[\sigma]\emptyset\} \\
\mathcal{G}^U[\Gamma, \mathbf{x} : \tau]\rho^U &\stackrel{\text{def}}{=} \{(k, \gamma^U[\mathbf{x} \mapsto \mathbf{v}_u, \mathbf{v}]) \mid (k, \gamma^U) \in \mathcal{G}^U[\Gamma]\rho^U \wedge (k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U\}
\end{aligned}$$

Figure 30: Universal Type Logical Relation

$$\begin{aligned}
\Delta; \Gamma \vdash v' \approx_{\mathcal{V}^U}^{\text{log}} v : \sigma &\stackrel{\text{def}}{=} v' \in \lambda^S \wedge \forall \rho^U \in \mathcal{D}^U[\Delta], (k, \gamma^U) \in \mathcal{G}^U[\Gamma] \rho^U. \\
&\quad (k, \gamma^U(v'), \rho^U(\gamma^U(v))) \in \mathcal{V}[\sigma] \emptyset \\
\Delta; \Gamma \vdash e' \approx_{\mathcal{E}^U}^{\text{log}} e : \sigma &\stackrel{\text{def}}{=} e' \in \lambda^S \wedge \forall \rho^U \in \mathcal{D}^U[\Delta], (k, \gamma^U) \in \mathcal{G}^U[\Gamma] \rho^U. \\
&\quad (k, \gamma^U(e'), \rho^U(\gamma^U(e))) \in \mathcal{E}[\sigma] \emptyset \\
\Delta; \Gamma \vdash v_u \approx_{\mathcal{V}^U}^{\text{log}} v : \tau &\stackrel{\text{def}}{=} v_u \in \lambda^S \wedge \forall \rho^U \in \mathcal{D}^U[\Delta], (k, \gamma^U) \in \mathcal{G}^U[\Gamma] \rho^U. \\
&\quad (k, \gamma^U(v_u), \rho^U(\gamma^U(v))) \in \mathcal{V}^U[\tau] \rho^U \\
\Delta; \Gamma \vdash v_u \approx_{\mathcal{R}^U}^{\text{log}} r : \theta &\stackrel{\text{def}}{=} v_u \in \lambda^S \wedge \forall \rho^U \in \mathcal{D}^U[\Delta], (k, \gamma^U) \in \mathcal{G}^U[\Gamma] \rho^U. \\
&\quad (k, \gamma^U(v_u), \rho^U(\gamma^U(r))) \in \mathcal{R}^U[\theta] \rho^U \\
\Delta; \Gamma \vdash e_u \approx_{\mathcal{E}^U}^{\text{log}} e : \theta &\stackrel{\text{def}}{=} e_u \in \lambda^S \wedge \forall \rho^U \in \mathcal{D}^U[\Delta], (k, \gamma^U) \in \mathcal{G}^U[\Gamma] \rho^U. \\
&\quad (k, \gamma^U(e_u), \rho^U(\gamma^U(e))) \in \mathcal{E}^U[\theta] \rho^U
\end{aligned}$$

Figure 31: Universal Type Logical Relation for Open Terms

Lemma 9.1 (Universal Type Logical Relation Weakening)

If $\rho^U \in \mathcal{D}[\Delta]$, $\Delta \vdash \tau$ and $\Delta \vdash \theta$, $\Delta \vdash \tau'$, $\Delta \vdash \Gamma$ and $R \in \text{Rel}^U[\tau']$, then

1. $\mathcal{V}^U[\tau]\rho^U = \mathcal{V}^U[\tau]\rho^{U'}$
2. $\mathcal{R}^U[\theta]\rho^U = \mathcal{R}^U[\theta]\rho^{U'}$
3. $\mathcal{E}^U[\theta]\rho^U = \mathcal{E}^U[\theta]\rho^{U'}$
4. $\mathcal{K}^U[\theta]\rho^U = \mathcal{K}^U[\theta]\rho^{U'}$
5. $\mathcal{G}^U[\Gamma]\rho^U = \mathcal{G}^U[\Gamma]\rho^{U'}$

where $\rho^{U'} = \rho^U[\alpha \mapsto \tau', R]$.

Proof

The first 4 are proven by mutual induction on types. Then the $\mathcal{G}^U[\]$ case follows by induction on Γ . \square

Lemma 9.2 (Universal Type Logical Relation Compositionality)

If $\rho^U \in \mathcal{D}[\Delta]$, $\Delta \vdash \tau'$, and $R \in \text{Rel}^U[\tau']$, then if $\Delta, \alpha \vdash \tau$ and $\Delta, \alpha \vdash \theta$,

1. $\mathcal{V}^U[\tau]\rho^{U'} = \mathcal{V}^U[\tau[\alpha/\tau']]\rho^U$
2. $\mathcal{R}^U[\tau]\rho^{U'} = \mathcal{R}^U[\tau[\alpha/\tau']]\rho^U$
3. $\mathcal{E}^U[\tau]\rho^{U'} = \mathcal{E}^U[\tau[\alpha/\tau']]\rho^U$
4. $\mathcal{K}^U[\tau]\rho^{U'} = \mathcal{K}^U[\tau[\alpha/\tau']]\rho^U$

where $\rho^{U'} = \rho^U[\alpha \mapsto \tau', R]$.

Proof

By induction k , τ and θ , using Lemma 9.1 where appropriate. \square

Lemma 9.3 (Monotonicity)

If $j < k$ then

1. If $(k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U$, then $(j, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\tau]\rho^U$.
2. If $(k, \mathbf{v}_u, \mathbf{r}) \in \mathcal{R}^U[\theta]\rho^U$, then $(j, \mathbf{v}_u, \mathbf{r}) \in \mathcal{R}^U[\theta]\rho^U$.
3. If $(k, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\theta]\rho^U$, then $(j, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\theta]\rho^U$.
4. If $(k, K_1, K_2) \in \mathcal{K}^U[\theta]\rho^U$, then $(j, K_1, K_2) \in \mathcal{K}^U[\theta]\rho^U$.
5. If $(k, \gamma^U) \in \mathcal{G}^U[\Gamma]\rho^U$, then $(j, \gamma^U) \in \mathcal{G}^U[\Gamma]\rho^U$.

Lemma 9.4 (Universal Type Value Relation is Admissible)

$\mathcal{V}^U[\tau]\rho^U \in \text{Rel}^U[\rho^U(\tau)]$

Proof

Immediate corollary of Lemma 9.3. \square

Lemma 9.5 (Universal Type Logical Relation Monadic Bind)

There are a few different versions, depending on how the two logical relations are interacting, however the proofs are essentially the same.

1. If $(k, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\theta]\rho^U$ and for any $j \leq k$, $(j, \mathbf{v}, \mathbf{r}) \in \mathcal{R}^U[\theta]\rho^U$ and $(j, K_1[\mathbf{v}], K_2[\mathbf{r}]) \in \mathcal{E}[\theta]\rho$ then $(k, K_1[\mathbf{e}_u], K_2[\mathbf{e}]) \in \mathcal{E}[\theta]\rho$.
2. If $(k, e_1, e_2) \in \mathcal{E}[\theta]\rho$ and for any $j \leq k$, $(j, r_1, r_2) \in \mathcal{R}[\theta]\rho$, $(j, \mathbf{K}[r_1], \mathbf{K}[r_2]) \in \mathcal{E}^U[\theta]\rho^U$, then $(\mathbf{K}[e_1], \mathbf{K}[e_2]) \in \mathcal{E}^U[\theta]\rho^U$.
3. If $(k, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\theta]\rho^U$ and for any $j \leq k$, $(j, \mathbf{v}, \mathbf{r}) \in \mathcal{R}^U[\theta]\rho^U$ and $(j, \mathbf{K}[\mathbf{v}], \mathbf{K}[\mathbf{r}]) \in \mathcal{E}^U[\theta']\rho^U$ then $(k, \mathbf{K}[\mathbf{e}_u], \mathbf{K}[\mathbf{e}]) \in \mathcal{E}^U[\theta']\rho^U$.

Proof

We present a proof of the first case, the others are essentially the same. Let $(k, K'_1, K'_2) \in \mathcal{K}[\theta]\rho$. We want to show that $(k, K'_1[K_1[\mathbf{e}_u]], K'_2[K_2[\mathbf{e}]]) \in \mathcal{O}$. By definition of $\mathcal{E}^U[\]$, it is sufficient to show that $(k, K'_1[K_1], K'_2[K_2]) \in \mathcal{K}^U[\theta]\rho^U$.

So, let $j \leq k$, $(j, \mathbf{v}, \mathbf{r}) \in \mathcal{R}^U[\theta]\rho^U$, we need to show that $(j, K'_1[K_1[\mathbf{v}]], K'_2[K_2[\mathbf{r}]]) \in \mathcal{O}$. By Lemma 7.6, $(j, K'_1, K'_2) \in \mathcal{K}[\theta]\rho$, so the result follows from the assumption and that $(j, K_1[\mathbf{v}], K_2[\mathbf{r}]) \in \mathcal{E}[\theta]\rho$. \square

Lemma 9.6 (Universal Type Logical Relation Anti-reduction)

If $\mathbf{e}_u \mapsto^{k_u} \mathbf{e}'_u$ and $\mathbf{e} \mapsto^{k_t} \mathbf{e}'$ and $k \leq \min(k_u, k_t) + k'$ then if $(k', \mathbf{e}'_u, \mathbf{e}') \in \mathcal{E}^U[\theta]\rho^U$, then $(k, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\theta]\rho^U$.

Proof

Direct from definition of \mathcal{O} . \square

Lemma 9.7 (Universal Type Derived Computation Rules)

For appropriately typed expressions,

1. $\text{TOSUM}(\text{IN}(i, v_u)) \mapsto^* \text{inj}_i v_u$
2. $\text{TOPAIR}(\text{CONS}(v_u, v'_u)) \mapsto^* \langle v_u, v'_u \rangle$
3. $\text{TOFUN}(\text{LAMBDA}(\lambda(x_u : U). e_u)) \mapsto^* \lambda(x_u : U). e_u$
4. $\text{TOFOLD}(\text{FOLD}(v_u)) \mapsto^{\geq 1} v_u$

Proof

Trivial. \square

Lemma 9.8 (Correctness of Fix)

If $\cdot; \cdot \vdash v_f : (\sigma_1 \rightarrow \sigma_2) \rightarrow \sigma_1 \rightarrow \sigma_2$ and $\cdot; \cdot \vdash v_{arg} : \sigma_1$, then

$$\text{FIX}_{\sigma_1 \rightarrow \sigma_2}(v_f) v_{arg} \mapsto^* \text{let } x_f = v_f \text{ FIX}_{\sigma_1 \rightarrow \sigma_2}(v_f) \text{ in } x_f v_{arg}$$

Proof

Straightforward calculation. \square

Lemma 9.9 (Embed/Project Unroll)

$$\text{EP}(\emptyset, \mu\alpha. \sigma) \langle \rangle \mapsto^* \langle \lambda(x : \delta_\sigma(\mu\alpha. \sigma)). \lambda(x_u : U). \text{let } y = \text{unfold } x \text{ in let } y_u = \text{EMBED}(\cdot, \sigma[\alpha/\mu\alpha. \sigma]) y \text{ in FOLD}(y_u), \lambda(x_u : U). \text{let } y_u = \text{TOFOLD}(x_u) \text{ in let } y = \text{PROJECT}(\cdot, \sigma[\alpha/\mu\alpha. \sigma]) y_u \text{ in fold}_{\mu\alpha. \sigma} y \rangle.$$

Proof

The result is a simple consequence of Lemma 9.8 and the following lemma:

1. $\text{EMBED}(\delta[\alpha \mapsto \mu\alpha. \sigma, x_{\mu\alpha. \sigma}], \sigma')[\text{EP}(\delta, \mu\alpha. \sigma)/x_{\mu\alpha. \sigma}] = \text{EMBED}(\delta, \sigma'[\mu\alpha. \sigma/\alpha])$

2. $\text{PROJECT}(\delta[\alpha \mapsto \mu\alpha. \sigma, x_{\mu\alpha.\sigma}], \sigma')[\text{EP}(\delta, \mu\alpha. \sigma)/x_{\mu\alpha.\sigma}] = \text{PROJECT}(\delta, \sigma'[\mu\alpha. \sigma/\alpha])$

which holds by a straightforward induction on σ' . \square

Theorem 9.10 (Interpret = Interoperate)

1. If $(k, \mathbf{e}_u, \mathbf{e}) \in \mathcal{E}^U[\![\sigma^\dagger]\!] \emptyset$, then $(k, \text{let } x = \mathbf{e}_u \text{ in } \text{PROJECT}(\sigma) \ x, {}^\sigma \text{ST } \mathbf{e}) \in \mathcal{E}[\![\sigma]\!] \emptyset$.
2. If $(k, \mathbf{e}, \mathbf{e}') \in \mathcal{E}[\![\sigma]\!] \emptyset$, then $(k, \text{let } x = \mathbf{e} \text{ in } \text{EMBED}(\sigma) \ x, \text{TS } {}^\sigma \mathbf{e}') \in \mathcal{E}^U[\![\sigma^\dagger]\!] \emptyset$.
3. If $(k, \mathbf{v}, \mathbf{r}) \in \mathcal{R}^U[\![\sigma^\dagger]\!] \emptyset$, then either
 $\text{PROJECT}(\sigma) \ \mathbf{v} \mapsto^k$ and ${}^\sigma \text{ST } \mathbf{r} \mapsto^k$, or
 $\text{PROJECT}(\sigma) \ \mathbf{v} \mapsto^* \mathbf{v}'_1$, ${}^\sigma \text{ST } \mathbf{r} \mapsto^* \mathbf{v}'_2$ and $(k, \mathbf{v}'_1, \mathbf{v}'_2) \in \mathcal{R}[\![\sigma]\!] \emptyset$.
4. If $(k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\![\sigma^+]\!] \emptyset$, then either
 $\text{PROJECT}(\cdot, \sigma) \ \mathbf{v}_u \mapsto^k$ and ${}^\sigma \text{ST } \text{return } \mathbf{v} \mapsto^k$, or
 $\text{PROJECT}(\cdot, \sigma) \ \mathbf{v}_u \mapsto^* \mathbf{v}$, ${}^\sigma \text{ST } \text{return } \mathbf{v} \mapsto^* \mathbf{v}'$ and $(k, \mathbf{v}, \mathbf{v}') \in \mathcal{V}[\![\sigma]\!] \emptyset$.
5. If $(k, \mathbf{v}, \mathbf{v}') \in \mathcal{V}[\![\sigma]\!] \emptyset$, then either
 $\text{EMBED}(\cdot, \sigma) \ \mathbf{v} \mapsto^k$ and $\text{TS } {}^\sigma \mathbf{v}' \mapsto^k$ or $\text{EMBED}(\cdot, \sigma) \ \mathbf{v} \mapsto^* \text{RETURN}(\mathbf{v}_u)$, $\text{TS } {}^\sigma \mathbf{v}' \mapsto^* \text{return } \mathbf{v}$
and $(k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\![\sigma^+]\!] \emptyset$.

Proof

The first 2 cases follow from the latter cases. The third case follows from the later ones and the interpretation of $\mathbf{0}$.

For the last 2 cases, we proceed by nested induction on k, σ .

Case $(k, \mathbf{v}_u, \mathbf{v}) \in \mathcal{V}^U[\![\sigma^+]\!] \emptyset$:

Case 1: trivial.

Case $\sigma_1 + \sigma_2$: $\mathbf{v}_u = \text{IN}(i, \mathbf{v}'_u)$ and $\mathbf{v} = \text{inj}_i \ \mathbf{v}'$. By Lemma 9.7,

$\text{PROJECT}(\cdot, \sigma_1 + \sigma_2) \ \text{IN}(i, \mathbf{v}'_u) \mapsto^* \text{let } x = \text{PROJECT}(\cdot, \sigma_i) \ \mathbf{v}'_u \text{ in } \text{inj}_i \ x$. Next, ${}^{\sigma_1 + \sigma_2} \text{ST } \text{inj}_i \ \mathbf{v}' \mapsto^* \text{let } x = {}^{\sigma_i} \text{ST } \text{return } \mathbf{v}' \text{ in } \text{inj}_i \ x$, so the result follows by inductive hypothesis and Lemma 9.6.

Case $\sigma_1 \times \sigma_2$: By Lemma 9.7 and inductive hypothesis.

Case $\sigma_1 \rightarrow \sigma_2$: $\mathbf{v}_u = \text{CONS}(\text{LAMBDA}(\lambda(x_u : U). \mathbf{e}_u), \text{CONS}(\mathbf{v}_{env}, \text{UNIT}))$ and $\mathbf{v} = \text{pack}(\tau, \langle \lambda(x : \langle \tau', \sigma_1^+ \rangle). \mathbf{e}, \mathbf{v}_{env} \rangle)$ and there exists $R \in \text{Atom}^V[\tau]$ such that $(k, \mathbf{v}_{env}, \mathbf{v}_{env}) \in R$ and $(k, \text{LAMBDA}(\lambda(x_u : U). \mathbf{e}_u), \lambda(x : \langle \tau', \sigma_1^+ \rangle). \mathbf{e}) \in \mathcal{V}^U[\![\langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger]\!] \rho^{U'}$ where $\rho^{U'} = \rho^U[\emptyset \mapsto \alpha, \tau]R$.

First,

$$\begin{aligned} \text{PROJECT}(\cdot, \sigma_1 \rightarrow \sigma_2) \ \mathbf{v}_u \mapsto^* & \lambda(y : \delta_\sigma(\sigma_1)). \text{let } y_u = \text{EMBED}(\cdot, \sigma_1) \ y \text{ in} \\ & \text{let } x = \text{CONS}(\mathbf{v}_{env}, \text{CONS}(y_u, \text{UNIT})) \ \text{in} \\ & \text{let } x_r = \text{LAMBDA}(\lambda(x_u : U). \mathbf{e}_u) \ x \ \text{in} \\ & \text{let } x'_u = \text{TOLHS}(x_r) \ \text{in} \\ & \text{PROJECT}(\cdot, \sigma_2) \ x'_u \end{aligned}$$

and

$${}^{\sigma_1 \rightarrow \sigma_2} \text{ST } \text{return } \mathbf{v} \mapsto^* \lambda(x : \sigma_1). {}^{\sigma_2} \text{ST} \left(\begin{array}{l} \text{unpack } (\alpha, z) = \mathbf{v} \text{ in let } x_f = z.1 \text{ in} \\ \text{let } x_{env} = z.2 \text{ in} \\ \text{let } x = \text{TS } {}^{\sigma_1} x \ \text{in } x_f \ [\alpha] \ \langle x_{env}, x \rangle \end{array} \right)$$

Let $j \leq k$ and $(j, \mathbf{v}_{larg}, \mathbf{v}_{rarg}) \in \mathcal{V} \llbracket \sigma_1 \rrbracket \emptyset$. By Lemma 7.11, it is sufficient to show that $(j,$

$\text{let } y_u = \text{EMBED}(\cdot, \sigma_1) \mathbf{v}_{larg} \text{ in}$
 $\text{let } x = \text{CONS}(\mathbf{v}_{env}, \text{CONS}(y_u, \text{UNIT})) \text{ in}$
 $\text{let } x_r = \text{LAMBDA}(\lambda(x_u : U). e_u) x \text{ in}$
 $\text{let } x''_u = \text{TOLHS}(x_r) \text{ in}$
 $\text{PROJECT}(\cdot, \sigma_2) x''_u$
 $\sigma_2 \mathcal{ST} \text{ let } x = \mathcal{TS}^{\sigma_1} \mathbf{v}_{rarg} \text{ in } (\lambda(x : \langle \tau', \sigma_1^+ \rangle). e) \langle \mathbf{v}_{env}, x \rangle) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \emptyset.$

By inductive hypothesis either both $\text{EMBED}(\cdot, \sigma_1) \mathbf{v}_{larg} \mapsto^j$ and $\mathcal{TS}^{\sigma_1} \mathbf{v}_{rarg} \mapsto^j$ and we're done, or $\text{EMBED}(\cdot, \sigma_1) \mathbf{v}_{larg} \mapsto^* \mathbf{v}_{uarg}$ and $\mathcal{TS}^{\sigma_1} \mathbf{v}_{rarg} \mapsto^* \text{return } \mathbf{v}_{targ}$ where $(j, \mathbf{v}_{uarg}, \mathbf{v}_{targ}) \in \mathcal{V}^U \llbracket \sigma_1^+ \rrbracket \emptyset$. So it is sufficient to show

$(j, \text{let } x_r = \text{LAMBDA}(\lambda(x_u : U). e_u) \text{CONS}(\mathbf{v}_{env}, \text{CONS}(\mathbf{v}_{uarg}, \text{UNIT})) \text{ in},$
 $\text{let } x''_u = \text{TOLHS}(x_r) \text{ in}$
 $\text{PROJECT}(\cdot, \sigma_2) x''_u$
 $\sigma_2 \mathcal{ST} (\lambda(x : \langle \tau', \sigma_1^+ \rangle). e) \langle \mathbf{v}_{env}, \mathbf{v}_{targ} \rangle) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \emptyset$

Next, $(j, \text{CONS}(\mathbf{v}_{env}, \text{CONS}(\mathbf{v}_{uarg}, \text{UNIT})), \langle \mathbf{v}_{env}, \mathbf{v}_{targ} \rangle) \in \mathcal{V}^U \llbracket \langle \alpha, \sigma_1^+ \rangle \rrbracket \rho^U$ by assumption and Lemma 9.3. Then by Lemma 9.5, it is sufficient to show for any $l \leq j$, $(l, \mathbf{v}_r, \mathbf{r}) \in \mathcal{R} \llbracket \sigma_2^\div \rrbracket \emptyset$,

$$(j, \text{let } x''_u = \text{TOLHS}(\mathbf{v}_r) \text{ in}, \sigma_2 \mathcal{ST} \mathbf{r}) \in \mathcal{E} \llbracket \sigma \rrbracket \emptyset.$$

$$\text{PROJECT}(\cdot, \sigma_2) x''_u$$

Which follows by inductive hypothesis and the definition of $\mathcal{V}^U \llbracket \mathbf{0} \rrbracket \emptyset$.

Case $\mu\alpha. \sigma$: Either $k = 0$ and we're done or there is some k' such that $k = k' + 1$. In the latter case, we have $\mathbf{v}_u = \text{FOLD}(\mathbf{v}'_u)$, $\mathbf{v} = \text{fold}_{\mu\alpha. \sigma} \mathbf{v}'$ where $(k', \mathbf{v}'_u, \mathbf{v}') \in \mathcal{V}^U \llbracket \sigma^+ [\mu\alpha. \sigma^+ / \alpha] \rrbracket \emptyset$. By Lemma 9.9 and further calculation,

$$\text{PROJECT}(\cdot, \mu\alpha. \sigma) \text{FOLD}(\mathbf{v}'_u) \mapsto^{\geq 1} \text{let } y = \text{PROJECT}(\cdot, \sigma[\alpha / \mu\alpha. \sigma]) \mathbf{v}'_u \text{ in}$$

$$\text{fold}_{\mu\alpha. \sigma} y$$

and

$$\mu\alpha. \sigma \mathcal{ST} \text{return } \text{fold}_{\mu\alpha. \sigma} \mathbf{v}' \mapsto^* \text{let } x = \sigma[\mu\alpha. \sigma / \alpha] \mathcal{ST} \mathbf{v}' \text{ in } \text{fold}_{\mu\alpha. \sigma} x$$

Then by inductive hypothesis either both $\text{PROJECT}(\cdot, \sigma[\alpha / \mu\alpha. \sigma]) \mathbf{v}'_u \mapsto^{k'}$ and $\sigma[\mu\alpha. \sigma / \alpha] \mathcal{ST} \mathbf{v}' \mapsto^{k'}$, or $\text{PROJECT}(\cdot, \sigma[\alpha / \mu\alpha. \sigma]) \mathbf{v}'_u \mapsto^* \mathbf{v}_l$ and $\sigma[\mu\alpha. \sigma / \alpha] \mathcal{ST} \mathbf{v}' \mapsto^* \mathbf{v}_r$ and $(k', \mathbf{v}_l, \mathbf{v}_r) \in \mathcal{V} \llbracket \sigma[\mu\alpha. \sigma / \alpha] \rrbracket \emptyset$. Then we have

$$\text{let } y = \text{PROJECT}(\cdot, \sigma[\alpha / \mu\alpha. \sigma]) \mathbf{v}'_u \text{ in} \mapsto^* \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_l$$

$$\text{fold}_{\mu\alpha. \sigma} y$$

and

$$\text{let } x = \sigma[\mu\alpha. \sigma / \alpha] \mathcal{ST} \mathbf{v}' \text{ in } \text{fold}_{\mu\alpha. \sigma} x \mapsto^* \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_r$$

and we have $(k' + 1, \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_l, \text{fold}_{\mu\alpha. \sigma} \mathbf{v}_r) \in \mathcal{V} \llbracket \mu\alpha. \sigma \rrbracket \emptyset$.

Case $(k, \mathbf{v}, \mathbf{v}') \in \mathcal{V} \llbracket \sigma \rrbracket \emptyset$:

Case 1: trivial.

Case $\sigma_1 + \sigma_2$: Then $\mathbf{v}_1 = \text{inj}_i \mathbf{v}_{i,1}$, $\mathbf{v}_2 = \text{inj}_i \mathbf{v}_{i,2}$ where $(k, \mathbf{v}_{i,1}, \mathbf{v}_{i,2}) \in \mathcal{V} \llbracket \sigma_i \rrbracket \emptyset$. Next by Lemma 9.7,

$$\text{EMBED}(\cdot, \sigma_1 + \sigma_2) (\text{inj}_i \mathbf{v}_{i,1}) \mapsto^* \text{let } x' = \text{EMBED}(\cdot, \sigma_i) \mathbf{v}_{i,1} \text{ in}$$

$$\text{IN}(i, x')$$

and

$$\mathcal{TS}^{\sigma_1 + \sigma_2} (\text{inj}_i \mathbf{v}_{i,2}) \mapsto^* \text{let } x = \mathcal{TS}^{\sigma_i} \mathbf{v}_{i,2} \text{ in } \text{return } \text{inj}_i x$$

so the result holds by inductive hypothesis and Lemma 9.6.

Case $\sigma_1 \times \sigma_2$: By straightforward computation and inductive hypothesis.

Case $\sigma \rightarrow \sigma'$: Then $v_1 = \lambda(x_1 : \sigma). e_1$ and $v_2 = \lambda(x_2 : \sigma). e_1$. Next,

$$\text{EMBED}(\cdot, \sigma \rightarrow \sigma') (\lambda(x_1 : \sigma). e_1) \mapsto^* \text{CONS}(\lambda(x_u : U). \text{let } x'_u = \text{PRJ}(2, x_u) \text{ in } \text{CONS}(\text{UNIT}, \text{UNIT})) \\ \text{let } x = \text{PROJECT}(\cdot, \sigma) x'_u \text{ in} \\ \text{let } y = \lambda(x_1 : \sigma). e_1 x \text{ in} \\ \text{let } x''_u = \text{EMBED}(\cdot, \sigma') y \text{ in} \\ \text{RETURN}(x''_u)$$

and

$$\text{return pack } (1, \langle \lambda(z : \langle 1, \sigma^+ \rangle). \langle \rangle \rangle) \text{ as } (\sigma \rightarrow \sigma')^+ \\ \mathcal{TS}^{\sigma'} \left(\text{let } x = {}^\sigma \text{ST } \mathbf{z.2} \text{ in} \right. \\ \left. \lambda(x_2 : \sigma). e_2 x \right)$$

For τ , we select $\langle \rangle$ and for R we select $\text{Atom}^V[\langle \rangle]$, which is obviously in $\text{Rel}^U[\langle \rangle]$ and $(k, \text{UNIT}, \langle \rangle) \in \mathcal{V}^U[\alpha](\emptyset[\alpha \mapsto \langle \rangle, \text{Atom}^V[\langle \rangle]]) = \text{Atom}^V[\langle \rangle]$ as needed. Let $j \leq k$ and $(j, \text{CONS}(v'_u, \text{CONS}(v_u, \text{UNIT})), \langle v', v \rangle) \in \mathcal{V}^U[\langle \alpha, \sigma^+ \rangle](\emptyset[\alpha \mapsto \langle \rangle, \text{Atom}^V[\langle \rangle]])$. Then by Lemma 9.7 and Lemma 9.6, it is sufficient to show that

$$(j, \text{let } x = \text{PROJECT}(\cdot, \sigma) v_u \text{ in, } \mathcal{TS}^{\sigma'} \left(\text{let } x = {}^\sigma \text{ST } \text{return } \mathbf{v} \text{ in} \right. \\ \left. \text{let } y = \lambda(x_1 : \sigma). e_1 x \text{ in} \right. \\ \left. \text{let } x''_u = \text{EMBED}(\cdot, \sigma') y \text{ in} \right. \\ \left. \text{RETURN}(x''_u) \right) \in \mathcal{E}^U[\sigma'^+] \emptyset$$

By inductive hypothesis either both $\text{PROJECT}(\cdot, \sigma) v_u \mapsto^k$ and ${}^\sigma \text{ST } \text{return } \mathbf{v} \mapsto^k$, or $\text{PROJECT}(\cdot, \sigma) v_u \mapsto^* v_l$ and ${}^\sigma \text{ST } \text{return } \mathbf{v} \mapsto^* v_r$ and $(j, v_l, v_r) \in \mathcal{V}[\sigma] \emptyset$.

Then $(j, e_1[x_1/v_l], e_2[x_2/v_r]) \in \mathcal{E}[\sigma] \emptyset$, so by Lemma 9.5, Lemma 9.6 and computation, it is sufficient to show that for any $j' \leq j$, $(j', v_{l,2}, v_{r,2}) \in \mathcal{V}[\sigma] \emptyset$,

$$(j', \text{let } x''_u = \text{EMBED}(\cdot, \sigma') v_{l,2} \text{ in, } \mathcal{TS}^{\sigma'} v_{r,2}) \in \mathcal{E}^U[\sigma'^+] \emptyset \\ \text{RETURN}(x''_u)$$

which follows by inductive hypothesis.

Case $\mu\alpha. \sigma$: If $k = 0$, we're done. Otherwise $k = k' + 1$, $v = \text{fold}_{\mu\alpha. \sigma} v_l$ and $v' = \text{fold}_{\mu\alpha. \sigma} v_r$. By Lemma 9.9 and further calculation,

$$\text{EMBED}(\cdot, \mu\alpha. \sigma) \text{fold}_{\mu\alpha. \sigma} v_l \mapsto^{\geq 1} \text{let } y_u = \text{EMBED}(\cdot, \sigma[\mu\alpha. \sigma/\alpha]) v_l \text{ in} \\ \text{FOLD}(y_u)$$

and

$$\mathcal{TS}^{\mu\alpha. \sigma} \text{fold}_{\mu\alpha. \sigma} v_r \mapsto^{\geq 1} \text{let } \mathbf{x} = \mathcal{TS}^{\sigma[\mu\alpha. \sigma/\alpha]} v_r \text{ in return fold}_{(\mu\alpha. \sigma)^+} \mathbf{x}$$

By inductive hypothesis either both $\text{EMBED}(\cdot, \sigma[\mu\alpha. \sigma/\alpha]) v_l \mapsto^k$ and $\mathcal{TS}^{\sigma[\mu\alpha. \sigma/\alpha]} v_r \mapsto^k$, or $\text{EMBED}(\cdot, \sigma[\mu\alpha. \sigma/\alpha]) v_l \mapsto^* v_u$ and $\mathcal{TS}^{\sigma[\mu\alpha. \sigma/\alpha]} v_r \mapsto^* \text{return } \mathbf{v}$ and $(k', v_u, \mathbf{v}) \in \mathcal{V}[\sigma[\mu\alpha. \sigma/\alpha]] \emptyset$. Thus, we have

$$\text{let } y_u = \text{EMBED}(\cdot, \sigma[\mu\alpha. \sigma/\alpha]) v_l \text{ in } \mapsto^* \text{FOLD}(v_u) \\ \text{FOLD}(y_u)$$

and

$$\text{let } \mathbf{x} = \mathcal{TS}^{\sigma[\mu\alpha. \sigma/\alpha]} v_r \text{ in return fold}_{(\mu\alpha. \sigma)^+} \mathbf{x} \mapsto^* \text{return fold}_{(\mu\alpha. \sigma)^+} \mathbf{v}$$

and we have $(k' + 1, \text{FOLD}(v_u), \text{fold}_{(\mu\alpha. \sigma)^+} \mathbf{v}) \in \mathcal{V}^U[(\mu\alpha. \sigma)^+] \emptyset$.

□

Theorem 9.11 (Interpreter Fundamental Property)

1. If $\Delta; \Gamma \vdash v : \sigma$ and $\Delta; \Gamma \vdash v : \sigma \rightarrow v'$, then $\Delta; \Gamma \vdash v' \approx_{\mathcal{V}^U}^{log} v : \sigma$.
2. If $\Delta; \Gamma \vdash e : \sigma$ and $\Delta; \Gamma \vdash e : \sigma \rightarrow e'$, then $\Delta; \Gamma \vdash e' \approx_{\mathcal{E}^U}^{log} e : \sigma$.
3. If $\Delta; \Gamma \vdash v : \tau$ and $\Delta; \Gamma \vdash^+ v : \tau \rightarrow v_u$, then $\Delta; \Gamma \vdash v_u \approx_{\mathcal{V}^U}^{log} v : \tau$.
4. If $\Delta; \Gamma \vdash r : \theta$ and $\Delta; \Gamma \vdash^{\dot{+}} r : \theta \rightarrow v_u$, then $\Delta; \Gamma \vdash v_u \approx_{\mathcal{R}^U}^{log} r : \theta$.
5. If $\Delta; \Gamma \vdash e : \theta$ and $\Delta; \Gamma \vdash^{\dot{+}} e : \theta \rightarrow e_u$, then $\Delta; \Gamma \vdash e_u \approx_{\mathcal{E}^U}^{log} e : \theta$.

Proof

By induction over the implicit k and mutual induction over typing/translation derivations.

For each case let $\rho^U \in \mathcal{D}^U[\Delta]$ and $(k, \gamma^U) \in \mathcal{G}^U[\Gamma]\rho^U$.

Case $\Delta; \Gamma \vdash v : \sigma$ and $\Delta; \Gamma \vdash v : \sigma \rightarrow v'$. We need to show that $\Delta; \Gamma \vdash v' \approx_{\mathcal{V}^U}^{log} v : \sigma$. Every case follows by the same reasoning as in the proof of Theorem 7.38.

Case $\Delta; \Gamma \vdash e : \sigma$ and $\Delta; \Gamma \vdash e : \sigma \rightarrow e'$. We need to show that $\Delta; \Gamma \vdash e' \approx_{\mathcal{E}^U}^{log} e : \sigma$. Almost every case follows as in Theorem 7.38.

Case $e = {}^\sigma\mathcal{ST} e$ then $e' = \text{let } x = e_u \text{ in } \text{PROJECT}(\sigma) x$. We need to show that

$$(k, \text{let } x = \gamma^U(e_u) \text{ in } \text{PROJECT}(\sigma) x, {}^\sigma\mathcal{ST} \rho^U(\gamma^U(e))) \in \mathcal{E}[\sigma] \emptyset.$$

By inductive hypothesis, $(k, \gamma^U(e_u), \rho^U(\gamma^U(e))) \in \mathcal{E}^U[\sigma^{\dot{+}}]\rho^U$. Then by Lemma 9.5, it is sufficient to show that for any $j \leq k$, $(j, v_u, r) \in \mathcal{R}^U[\sigma^{\dot{+}}]\rho^U$,

$$(j, \text{let } x = v_u \text{ in } \text{PROJECT}(\sigma) x, {}^\sigma\mathcal{ST} r) \in \mathcal{E}[\sigma] \emptyset.$$

The result then holds by Lemma 9.10.

Case $\Delta; \Gamma \vdash v : \tau$ and $\Delta; \Gamma \vdash^+ v : \tau \rightarrow v_u$. We need to show that $\Delta; \Gamma \vdash v_u \approx_{\mathcal{V}^U}^{log} v : \tau$. Let ρ^U, k, γ^U as appropriate. Most cases follow immediately by definition.

Case $\Delta; \Gamma \vdash^+ \lambda[\alpha](x : \tau). e : \forall[\alpha]. \tau \rightarrow \theta \rightarrow \text{LAMBDA}(\lambda(x : U). e_u)$, where $\alpha; x : \tau \vdash^{\dot{+}} e : \theta \rightarrow e_u$. Given $\cdot \vdash \tau', R \in \text{Rel}^U[\rho^U(\tau')]$, $j \leq k$, $(j, v_u, v) \in \mathcal{V}^U[\tau]\rho^U$ where $\rho^U = \rho^U[\alpha \mapsto \tau', R]$, we need to show that $(j, \gamma^U(e_u)[x_u/v_u], \rho^U(\gamma^U(e))[\alpha/\tau'][x/v]) \in \mathcal{E}^U[\theta]\rho'$. Since e, e_u only have α, x and x free in them, this is equivalent to showing that $(j, e_u[x_u/v_u], e[\alpha/\tau'][x/v]) \in \mathcal{E}^U[\theta]\rho'$. By repeated use of Lemma 9.1, this is equivalent to showing $(j, e_u[x_u/v_u], e[\alpha/\tau'][x/v]) \in \mathcal{E}^U[\theta](\emptyset[\alpha \mapsto \tau', R])$, which holds by inductive hypothesis.

Case $\Delta; \Gamma \vdash^+ \text{pack}(\tau', v) \text{ as } \exists \alpha. \tau : \exists \alpha. \tau \rightarrow v_u$, where $\Delta; \Gamma \vdash^+ v : \tau[\tau'/\alpha] \rightarrow v_u$. Choose $R = \mathcal{V}^U[\tau']\rho^U$, which is a valid choice by Lemma 9.4. Then the result holds by inductive hypothesis and Lemma 9.2.

Case $\Delta; \Gamma \vdash^+ \text{fold}_{\mu\alpha.\tau} v : \mu\alpha. \tau \rightarrow \text{FOLD}(v_u)$, where $\Delta; \Gamma \vdash^+ v : \tau[\mu\alpha. \tau/\alpha] \rightarrow v_u$. If $k = 0$, we're done. Otherwise the result holds by inductive hypothesis.

Case $\Delta; \Gamma \vdash r : \theta$ and $\Delta; \Gamma \vdash^{\dot{+}} r : \theta \rightarrow v_u$. We need to show that $\Delta; \Gamma \vdash v_u \approx_{\mathcal{R}^U}^{log} r : \theta$. Both cases follow immediately by definition.

Case $\Delta; \Gamma \vdash e : \theta$ and $\Delta; \Gamma \vdash^{\dot{+}} e : \theta \rightarrow e_u$. We need to show that $\Delta; \Gamma \vdash e_u \approx_{\mathcal{E}^U}^{log} e : \theta$. Most cases follow immediately by definition, Lemma 9.7 and Lemma 9.6.

Case $\Delta; \Gamma \vdash^\dagger \mathcal{TS}^\sigma e : \sigma^\dagger \Rightarrow \text{let } x = e_u \text{ in } \text{EMBED}(\sigma) x$ where $\Delta; \Gamma \vdash e : \sigma \Rightarrow e_u$.

By inductive hypothesis, we know $(k, \gamma^U(e_u), \rho^U(\gamma^U(e))) \in \mathcal{E}^U[\sigma^\dagger] \rho^U$ and we need to show that

$$(k, \text{let } x = \gamma^U(e_u) \text{ in } \text{EMBED}(\sigma) x, \mathcal{TS}^\sigma \rho^U(\gamma^U(e))) \in \mathcal{E}[\sigma] \emptyset.$$

Then the result holds by Lemma 9.5 and Lemma 9.10.

Case $\Delta; \Gamma \vdash^\dagger v_1 [\tau'] v_2 : \theta[\tau'/\alpha] \Rightarrow \text{let } x = \text{TOFUN}(v_1) \text{ in } x v_2$ where

$\Delta; \Gamma \vdash^\dagger v_1 : \forall[\alpha]. \tau \rightarrow \theta \Rightarrow v_1$, and $\Delta; \Gamma \vdash^\dagger v_2 : \tau[\tau'/\alpha] \Rightarrow v_2$.

By inductive hypothesis, $(k, \gamma^U(v_1), \rho^U(\gamma^U(v_1))) \in \mathcal{V}^U[\forall[\alpha]. \tau \rightarrow \theta] \rho^U$ so in particular $\gamma^U(v_1) = \text{LAMBDA}(\lambda(x_u : U). e)$ and $\rho^U(\gamma^U(v_1)) = \lambda[\alpha](x : \tau). e$. Next by Lemma 9.7, Lemma 9.6 it is sufficient to show $(k, e[x_u/\gamma^U(v_2)], e[\alpha/\tau'][x/\rho^U(\gamma^U(v_2))]) \in \mathcal{V}^U[\theta[\alpha/\tau']] \rho^U$. By picking $\rho^{U'} = \rho^U[\alpha \mapsto \tau', \mathcal{V}^U[\tau'] \rho^U]$, the result follows by inductive hypothesis, Lemma 9.4 and Lemma 9.2.

Case $\Delta; \Gamma \vdash^\dagger \text{unpack}(\alpha, x) = v \text{ in } e : \theta \Rightarrow \text{let } x = v_u \text{ in } e_u$ where $\Delta; \Gamma \vdash^\dagger v : \exists\alpha. \tau \Rightarrow v_u$ and $\Delta, \alpha; \Gamma, x : \tau \vdash^\dagger e$

By inductive hypothesis, $\rho^U(\gamma^U(v)) = \text{pack}(\tau', v')$ as $\exists\alpha. \tau$ and there exists $R \in \text{Rel}^U[\rho^U(\tau')]$ such that $(k, v'_u, v') \in \mathcal{V}^U[\tau] \rho^{U'}$ where $\rho^{U'} = \rho^U[\alpha \mapsto \tau', R]$. Then by Lemma 9.6 and Lemma 9.7 it is sufficient to show $(k, \gamma^U(e_u)[v'_u/x], \rho^U(\gamma^U(e))[\tau'/\alpha][v'/x]) \in \mathcal{E}^U[\tau] \rho^U$. Since α is not free in τ , by Lemma 9.1, $\mathcal{E}^U[\tau] \rho^U = \mathcal{E}^U[\tau] \rho^{U'}$. Then the result follows by inductive hypothesis since $\rho^{U'} \in \mathcal{D}^U[\Delta, \alpha]$ and $\gamma^U[x \mapsto v'_u, v'] \in \mathcal{G}^U[\Gamma] \rho^{U'}$ by Lemma 9.1.

Case $\Delta; \Gamma \vdash^\dagger \text{unfold } v : \tau[\mu\alpha. \tau/\alpha] \Rightarrow \text{let } x = \text{TOFOLD}(v_u) \text{ in } \text{RETURN}(x)$ where

$\Delta; \Gamma \vdash^\dagger v : \mathbf{E} \tau_{\text{exn}} \mu\alpha. \tau \Rightarrow v_u$. If $k = 0$, we're done. Otherwise $k = k' + 1$, by inductive hypothesis $\gamma^U(v_u) = \text{FOLD}(v'_u)$, and $\rho^U(\gamma^U(v)) = \text{fold}_{\mu\alpha. \tau} v'$ and $(k', v_u, v') \in \mathcal{V}^U[\tau[\mu\alpha. \tau/\alpha]]$. Next, by Lemma 9.7, $\text{let } x = \text{TOFOLD}(\text{FOLD}(v'_u)) \text{ in } \text{RETURN}(x) \mapsto^{\geq 1} \text{RETURN}(v'_u)$ and $\text{unfold fold}_{\mu\alpha. \tau} v' \mapsto \text{return } v'$. Then the result holds by Lemma 9.6 and definition of $\mathcal{K}^U[\cdot]$.

Case $\Delta; \Gamma \vdash^\dagger \text{handle } e \text{ with } (x_1. e_1) (x_2. e_2) : \theta \Rightarrow \text{let } x_r = e \text{ in case } x_r \text{ of,}$

$$x_1. e_1$$

$$x_2. e_2$$

where $\Delta; \Gamma \vdash^\dagger e : \mathbf{E} \tau_{\text{exn}} \tau \Rightarrow e$, $\Delta; \Gamma, x_1 : \tau \vdash^\dagger e_1 : \theta \Rightarrow e_1$, and

$\Delta; \Gamma, x_1 : \tau_{\text{exn}} \vdash^\dagger e_2 : \theta \Rightarrow e_2$.

By inductive hypothesis and Lemma 9.5, it is sufficient to suppose $j \leq k$, $(j, v_r, r) \in \mathcal{R}^U[\theta] \rho^U$ and prove $(j, \text{let } x_r = v_r \text{ in case } x_r \text{ of } \text{, handle } r \text{ with } (x_1. \rho^U(\gamma^U(e_1))) (x_2. \rho^U(\gamma^U(e_2)))) \in \mathcal{E}^U[\theta] \rho^U$.

$$x_1. \gamma^U(e_1)$$

$$x_2. \gamma^U(e_2)$$

There are two cases, we consider the case where $v_r = \text{RETURN}(v_u)$ and $r = \text{return } v$, the other case is symmetric.

By computation and Lemma 9.6, it is sufficient to show $(j, \gamma^U(e_1)[x_1/v_u], \rho^U(\gamma^U(e_1))[x/v]) \in \mathcal{E}^U[\theta] \rho^U$.

By inductive hypothesis it is sufficient to show that $(j, \gamma^U[x_1 \mapsto v_u, v]) \in \mathcal{G}^U[\Gamma, x_1 : \tau_1] \rho^U$, which holds by assumptions about v_u, v and Lemma 9.3.

□

Lemma 9.12 (Universal Type Equivalence and Logical Equivalence Coincide in Source Contexts)

$$.; \Gamma \vdash e' \approx_{\mathcal{E}}^{\text{log}} e : \sigma \text{ iff } ; \Gamma \vdash e' \approx_{\mathcal{E}^U}^{\text{log}} e : \sigma.$$

Proof

Follows directly from $\mathcal{G}[\Gamma] \emptyset$ iff $\mathcal{G}^U[\Gamma] \emptyset$ which is direct from the definition. □

Theorem 9.13 (Back Translation Preserves Equivalence)

$$\text{If } ; \Gamma \vdash e' \approx_{\mathcal{E}}^{\text{log}} e : \sigma \text{ and } ; \Gamma \vdash e' : \sigma \Rightarrow e'', \text{ then } ; \Gamma \vdash e'' \approx_{\mathcal{E}}^{\text{log}} e : \sigma.$$

Proof

Direct corollary of Lemma 9.12 and Theorem 9.11. □

Lemma 9.14 (Back Translation is Identity on Source Terms)

1. If $e \in \lambda^S$ and $;\Gamma \vdash e : \sigma \rightarrow e'$ then $e = e'$.
2. If $v \in \lambda^S$ and $;\Gamma \vdash v : \sigma \rightarrow e'$ then $v = v'$.

Proof

Trivial by induction. □

Lemma 9.15 (Context Back-Translation)

If $\Delta; \Gamma \vdash e_1 : \sigma \rightarrow e'_1$ and $\Delta; \Gamma \vdash e_2 : \sigma \rightarrow e'_2$, then if $\Delta'; \Gamma' \vdash C[e_1] : \sigma' \rightarrow e'$, and $\Delta'; \Gamma' \vdash C[e_1] : \sigma' \rightarrow e''$, then there exists C such that $e' = C[e'_1]$ and $e'' = C[e'_2]$.

Proof

By induction on contexts. The construction can be realized by lifting the back-translation to contexts, adding a new rule:

$$\overline{\Delta; \Gamma \vdash [\] : \sigma \rightarrow [\]}$$

□

10 Translation Correctness

10.1 Semantics Preservation

Theorem 10.1 (Type Preservation)

1. If $\Gamma \vdash v : \sigma$ and $\Gamma \vdash v : \sigma \rightsquigarrow_v v$, then $;\Gamma^+ \vdash v : \sigma^+$.
2. If $\Gamma \vdash e : \sigma$ and $\Gamma \vdash e : \sigma \rightsquigarrow_e e$, then $;\Gamma^+ \vdash e : \sigma^\dagger$.

Proof

Proved simultaneously by mutual induction on the structure of v and e . We consider only the abstraction introduction case, all others follow trivially by induction.

If $\Gamma \vdash \lambda(x : \sigma).e : \sigma \rightarrow \sigma'$, then

$$v = \mathbf{pack}(\tau_{\mathbf{env}}, \langle \lambda(z : \langle \tau_{\mathbf{env}}, \sigma^+ \rangle). \mathbf{let} \ x_{\mathbf{env}} = \mathbf{return}_0 \ z.1 \ \mathbf{in} \ \mathbf{let} \ y_1 = \mathbf{return}_0 \ x_{\mathbf{env}}.1 \ \mathbf{in} \ \dots \ \mathbf{let} \ y_n = \mathbf{return}_0 \ x_{\mathbf{env}}.n \ \mathbf{in} \ \mathbf{let} \ x = \mathbf{return}_0 \ z.2 \ \mathbf{in} \ e \rangle, \langle y_1, \dots, y_n \rangle) \ \mathbf{as} \ \exists \alpha. \langle (\langle \alpha, \sigma^+ \rangle \rightarrow \sigma'^{\dagger}), \alpha \rangle$$

Where $\text{fv}(\lambda(x : \sigma').e) = (y_1, \dots, y_n)$, $\Gamma(y_i) = \sigma_i$, $\Gamma' = (y_1 : \sigma_1, \dots, y_n : \sigma_n)$, $\tau_{\mathbf{env}} = \langle \sigma_1^+, \dots, \sigma_n^+ \rangle$, and $\Gamma', x : \sigma \vdash e : \sigma' \rightsquigarrow_e e$.

We need to show that $;\Gamma^+ \vdash v : \exists \alpha. \langle (\langle \alpha, \sigma^+ \rangle \rightarrow \sigma'^{\dagger}), \alpha \rangle$.

Applying the typing rules, this reduces to showing that $;\mathbf{x}_{\mathbf{env}} : \tau_{\mathbf{env}}, \Gamma^+, x : \sigma^+ \vdash e : \sigma'^{\dagger}$.

By weakening it is sufficient to show that $;\Gamma^+, x : \sigma^+ \vdash e : \sigma'^{\dagger}$ since $\mathbf{x}_{\mathbf{env}} \notin \text{fve}$.

By inductive hypothesis and the fact that $\Gamma', x : \sigma \vdash e : \sigma' \rightsquigarrow_e e$, it is sufficient to show that $\Gamma', x : \sigma \vdash e : \sigma'$. Which holds by the fact that $\Gamma, x : \sigma \vdash e : \sigma'$ and that Γ' is a subset of Γ containing all of the free variables in e besides x . \square

Lemma 10.2 (Translation Weakening)

If $\Gamma \vdash e : \sigma$ and $\Gamma \vdash e : \sigma \rightsquigarrow_e e$, then for any $\Gamma' \subset \Gamma$ such that $\Gamma \vdash e : \sigma$, $\Gamma' \vdash e : \sigma \rightsquigarrow_e e$.

Proof

By induction on e . \square

Lemma 10.3 (Context Translation)

If $\vdash C : (\Gamma \vdash \sigma) \Rightarrow (\Gamma' \vdash \sigma')$, $\Gamma \vdash e : \sigma$ and $\Gamma \vdash e : \sigma \rightsquigarrow_e e$, then there exists C such that $\Gamma' \vdash C[e] : \sigma \rightsquigarrow_e C[e]$.
Furthermore if $\Gamma \vdash e' : \sigma$ and $\Gamma \vdash e' : \sigma \rightsquigarrow_e e'$, then $\Gamma' \vdash C[e] : \sigma \rightsquigarrow_e C[e']$.

Proof

Both follow by induction on C , using Lemma 10.2 in the abstraction case. \square

Lemma 10.4 (Boundary Terminates (Source to Target))

If $\cdot \vdash$

If $\Delta; \cdot \vdash v : \sigma$, then there exist n, \mathbf{v} such that $\mathcal{TS}^\sigma v \mapsto^n \mathbf{return}_0 \ \mathbf{v}$.

Proof

By induction on the typing derivation. We omit the cases for unit, sums, and pairs.

Case $\Delta; \cdot \vdash v : \sigma_1 \rightarrow \sigma_2$: Then $\mathcal{TS}^{\sigma_1 \rightarrow \sigma_2} v \mapsto$
 $\lambda(z : \langle \mathbf{1}, \sigma_1^+ \rangle).$

$$\mathcal{TS}^{\sigma_2} \left(\begin{array}{l} \text{let } a = {}^{\sigma_1} ST \text{ return}_0 z.2 \text{ in} \\ v \ a \end{array} \right)$$

Case $\Delta; \cdot \vdash \text{fold}_{\mu\alpha.\sigma'} v' : \mu\alpha.\sigma'$:

First, $\mathcal{TS}^{\mu\alpha.\sigma'} \text{fold}_{\mu\alpha.\sigma'} v' \mapsto^2 \text{let } v = \mathcal{TS}^{\sigma[\mu\alpha.\sigma/\alpha]} v' \text{ in return}_0 \text{fold}_{\mu\alpha.\tau} v$. By inductive hypothesis there exist n, v' such that $\mathcal{TS}^{\sigma[\mu\alpha.\sigma/\alpha]} v' \mapsto^n \text{return}_0 v'$. Then by definition of the operational semantics, $\mathcal{TS}^{\mu\alpha.\sigma'} \text{fold}_{\mu\alpha.\sigma'} v' \mapsto^{n+3} \text{return}_0 \text{fold}_{\mu\alpha.\tau} v'$.

□

Lemma 10.5 (Boundary Terminates (Target to Source))

If $\Delta; \cdot \vdash v : \sigma^+$, then there exist n, v such that ${}^\sigma ST \text{return } v \mapsto^n v$.

Proof

By induction on the typing derivation. We omit the cases for sums and tuples.

Case $\Delta; \cdot \vdash v : \sigma_1 \rightarrow \sigma_2^+$: Then ${}^{\sigma_1 \rightarrow \sigma_2} ST \text{return } v \mapsto$

$$\lambda(x : \sigma_1). {}^{\sigma_2} ST \left(\begin{array}{l} \text{unpack } (\alpha, z) = v \text{ in let } x_f = \text{return}_0 z.1 \text{ in} \\ \quad \text{let } x_{\text{env}} = \text{return}_0 z.2 \text{ in} \\ \quad \text{let } x = \mathcal{TS}^{\sigma_1} x \text{ in } x_f [\alpha] \langle x_{\text{env}}, x \rangle \end{array} \right)$$

Case $\Delta; \cdot \vdash \text{fold}_{\mu\alpha.\sigma'} v' : \mu\alpha.\sigma'^+$: Directly analogous to the case in Lemma 10.4.

□

Lemma 10.6 (Boundary Cancellation (Source round-trip))

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$ and $\cdot \vdash \sigma$, then

1. If $(k, e_1, e_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$ then $(k, e_1, {}^\sigma ST \mathcal{TS}^\sigma e_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$
2. If $(k, v_1, v_2) \in \mathcal{V} \llbracket \sigma \rrbracket \rho$ and ${}^\sigma ST \mathcal{TS}^\sigma v_2 \mapsto^n v'_2$ then $(k, v_1, v'_2) \in \mathcal{V} \llbracket \sigma \rrbracket \rho$

Proof

Proved simultaneously by induction on k and σ . We omit the cases for unit, sums, and pairs.

1. By Lemma 7.9, it is sufficient to prove that for every $j \leq k$, if $(j, v_1, v_2) \in \mathcal{V} \llbracket \sigma \rrbracket \rho$ then $(j, v_1, {}^\sigma ST \mathcal{TS}^\sigma v_2) \in \mathcal{E} \llbracket \sigma \rrbracket \rho$. Then by Lemma 10.5, Lemma 10.4, ${}^\sigma ST \mathcal{TS}^\sigma v_2 \mapsto^n v'_2$ for some n, v'_2 , so the result holds by inductive hypothesis, Lemma 7.11 and Lemma 7.8.
2. Values

Case $\sigma = \sigma_1 \rightarrow \sigma_2$:

By definition of $\mathcal{V} \llbracket \sigma_1 \rightarrow \sigma_2 \rrbracket \rho$, $v_1 = \lambda(x : \sigma_1). e_1$ and $v_2 = \lambda(x : \sigma_1). e_2$ where for every $j \leq k$, $(j, v'_1, v'_2) \in \mathcal{V} \llbracket \sigma_1 \rrbracket \rho$, $(j, e_1[v'_1/x], e_2[v'_2/x]) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \rho$.

Then, as in Lemma 10.4 and Lemma 10.5,

${}^\sigma ST \mathcal{TS}^\sigma v_2 \mapsto$

${}^\sigma ST \text{return}_0 \text{pack } (\mathbf{1}, \langle \lambda(z : \langle \mathbf{1}, \sigma_1^+ \rangle).$

$$\left. \begin{array}{l} \text{let } a = {}^{\sigma_1} ST \text{return}_0 z.2 \text{ in} \\ v_2 \ a \end{array} \right), \langle \rangle \rangle \text{ as } (\sigma_1 \rightarrow \sigma_2)^+$$

By Lemma 10.4, $\mathcal{TS}^{\sigma_1} v_2''' \mapsto^n v_2''''$ for some n, v_2'''' . Then,

$$\begin{aligned} & \sigma_2 ST \text{let } x = \mathcal{TS}^{\sigma_1} v_2' \text{ in } \left(\begin{array}{c} \lambda(z : \langle 1, \sigma_1^+ \rangle). \\ \mathcal{TS}^{\sigma_2} \left(\begin{array}{c} \text{let } a = \sigma_1 ST \text{return}_0 z.2 \text{ in} \\ v_2 a \end{array} \right) \end{array} \right) [\alpha] \langle \langle \rangle, x \rangle \\ & \mapsto^4 \sigma_2 ST \mathcal{TS}^{\sigma_2} \left(\begin{array}{c} \text{let } a = \sigma_1 ST \text{return}_0 v_2'''' \text{ in} \\ v_2 a \end{array} \right) \text{ By Lemma 10.5, } \sigma_1 ST \text{return}_0 v_2'''' \mapsto^m \\ & v_2'''' \text{ for some } m, v_2'''''. \text{ Note that this means } \sigma_1 ST \mathcal{TS}^{\sigma_1} v_2'' \mapsto^{m+n} v_2''''', \text{ so by inductive hypothesis, } (j, v_1'', v_2''''') \in \mathcal{V} \llbracket \sigma_1 \rrbracket \rho. \text{ Finally,} \end{aligned}$$

$$\sigma_2 ST \mathcal{TS}^{\sigma_2} \left(\begin{array}{c} \text{let } a = \sigma_1 ST \text{return}_0 v_2'''' \text{ in} \\ v_2 a \end{array} \right) \mapsto^{m+2} \sigma_2 ST \mathcal{TS}^{\sigma_2} e_2[v_2''''/x], \text{ so by Lemma 7.11,}$$

it is sufficient to prove $(j, e_1[v_1''/x], \sigma_2 ST \mathcal{TS}^{\sigma_2} e_2[v_2''''/x])$ which holds by inductive hypothesis and the fact that $(j, e_1[v_1''/x], e_2[v_2''/x]) \in \mathcal{E} \llbracket \sigma_2 \rrbracket \rho$.

Case $\sigma = \mu\alpha. \sigma'$: By definition of $\mathcal{V} \llbracket \mu\alpha. \sigma' \rrbracket \rho$, $v_1 = \text{fold}_{\mu\alpha. \sigma'} v_1'$ and $v_2 = \text{fold}_{\mu\alpha. \sigma'} v_2'$. where $(k-1, v_1', v_2') \in \mathcal{V} \llbracket \sigma'[\mu\alpha. \sigma'/\alpha] \rrbracket \rho$.

Next as in the proof of Lemma 10.4, $\mu\alpha. \sigma' ST \mathcal{TS}^{\mu\alpha. \sigma'} 3 \mapsto^{n+3} \mu\alpha. \sigma' ST \text{return}_0 \text{fold}_{\mu\alpha. \sigma'} v_2''$ where $\mathcal{TS}^{\sigma'[\mu\alpha. \sigma'/\alpha]} v_2'' \mapsto^n \text{return } v_2''$. Then as in the proof of Lemma 10.5, $\mu\alpha. \sigma' ST \text{return}_0 \text{fold}_{\mu\alpha. \sigma'} v_2'' \mapsto^{m+3} \text{fold}_{\mu\alpha. \sigma'} v_2''$ where $\sigma'[\mu\alpha. \sigma'/\alpha] ST \text{return } v_2'' \mapsto^m v_2''''$. Then $\sigma'[\mu\alpha. \sigma'/\alpha] ST \mathcal{TS}^{\sigma'[\mu\alpha. \sigma'/\alpha]} v_2'' \mapsto^{m+n} v_2''''$, so by inductive hypothesis $(k-1, v_1', v_2') \in \mathcal{V} \llbracket \sigma'[\mu\alpha. \sigma'/\alpha] \rrbracket \rho$.

□

Lemma 10.7 (Boundary Cancellation (Target round-trip))

If $\rho \in \mathcal{D} \llbracket \Delta \rrbracket$ and $\cdot \vdash \sigma$, then

1. If $(k, e_1, e_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$ then $(k, e_1, \mathcal{TS}^{\sigma} \sigma ST e_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$
2. If $(k, v_1, v_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket \rho$ and $\mathcal{TS}^{\sigma} \sigma ST \text{return } v_2 \mapsto^n \text{return } v_2'$ then $(k, v_1, v_2') \in \mathcal{V} \llbracket \sigma^+ \rrbracket \rho$

Proof

1. Applying Lemma 7.9 there are two cases.

Case Suppose $j \leq k$ and $(j, v_1, v_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket \rho$. We need to show that $(j, \text{return } v_1, \mathcal{TS}^{\sigma} \sigma ST \text{return } v_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$.

By Lemma 10.5 and Lemma 10.4, there exist n, v_2' such that $\mathcal{TS}^{\sigma} \sigma ST \text{return } v_2 \mapsto^n \text{return } v_2'$. By Lemma 7.11, it is sufficient to show that $(j, \text{return } v_1, \text{return } v_2') \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$, which holds by Lemma 7.8 and part 2.

Case Suppose $j \leq k$ and $(j, v_1, v_2) \in \mathcal{V} \llbracket \mathbf{0} \rrbracket \rho$. We need to show that $(j, \text{raise } v_1, \mathcal{TS}^{\sigma} \sigma ST \text{raise } v_2) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \rho$. This holds vacuously since $\mathcal{V} \llbracket \mathbf{0} \rrbracket \rho = \emptyset$.

2. Values We omit the cases for unit, sums, and pairs.

Case $\sigma = \sigma_1 \rightarrow \sigma_2, (\sigma_1 \rightarrow \sigma_2)^+ = \exists \alpha. \langle \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger \rangle, \alpha \rangle$:

By definition of $\mathcal{V} \llbracket (\sigma_1 \rightarrow \sigma_2)^+ \rrbracket \rho$, $v_1 = \text{pack}(\tau_1, \langle v_1'', v_1''' \rangle) \text{ as } (\sigma_1 \rightarrow \sigma_2)^+$ and $v_2 = \text{pack}(\tau_2, \langle v_2'', v_2''' \rangle) \text{ as } (\sigma_1 \rightarrow \sigma_2)^+$ such that there exists $R \in \text{Rel}[\tau_1, \tau_2]$ such that $(k, v_1'', v_2''') \in \mathcal{V} \llbracket \alpha \rrbracket \rho'$ and $(k, v_1', v_2''') \in \mathcal{V} \llbracket \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger \rrbracket \rho'$ where $\rho' = \rho[\alpha \mapsto (\tau_1, \tau_2, R)]$.

Furthermore, $\mathbf{v}_1'' = \lambda(\mathbf{x} : \langle \tau_1, \sigma_1^+ \rangle) \cdot \mathbf{e}_1$ and $\mathbf{v}_2'' = \lambda(\mathbf{x} : \langle \tau_2, \sigma_1^+ \rangle) \cdot \mathbf{e}_2$ such that for any $j \leq k$, $(j, \mathbf{v}_1''', \mathbf{v}_2''') \in \mathcal{V} \llbracket \langle \alpha, \sigma_1^+ \rangle \rrbracket \rho'$, $(j, \mathbf{e}_1[\mathbf{v}_1'''/\mathbf{x}], \mathbf{e}_2[\mathbf{v}_2'''/\mathbf{x}]) \in \mathcal{E} \llbracket \sigma_2^\dagger \rrbracket \rho'$.

Next, as in the proofs of Lemma 10.5 and Lemma 10.4,

$\sigma_1 \rightarrow \sigma_2$ ST **return** $\mathbf{v}_2 \mapsto$
 $\lambda(\mathbf{x} : \sigma_1) \cdot \sigma_2 ST$ (**unpack** $(\alpha, \mathbf{z}) = \mathbf{v}_2$ **in** **let** $\mathbf{x}_f =$ **return** $\mathbf{z}.1$ **in** \quad) which we denote \mathbf{v}'_2
 $\quad \quad \quad \mathbf{let} \mathbf{x}_{env} =$ **return** $\mathbf{z}.2$ **in**
 $\quad \quad \quad \mathbf{let} \mathbf{x} = TS^{\sigma_1} \mathbf{x}$ **in** $\mathbf{x}_f [\alpha] \langle \mathbf{x}_{env}, \mathbf{x} \rangle$

and $TS^{\sigma_1 \rightarrow \sigma_2} \mathbf{v}'_2 \mapsto$
return **pack** $(1, \langle \lambda(\mathbf{z} : \langle 1, \sigma''^+ \rangle) \cdot \langle \rangle \rangle)$ **as** $(\sigma_1 \rightarrow \sigma_2)^+$ and we define

$$TS^{\sigma'} \left(\begin{array}{l} \mathbf{let} \mathbf{x} = \sigma'' ST \mathbf{return}_0 \mathbf{z}.2 \mathbf{in} \\ \mathbf{v}'_2 \mathbf{x} \end{array} \right)$$

the value in the **return** here to be \mathbf{v}'_2 . Then $TS^{\sigma_1 \rightarrow \sigma_2} \sigma_1 \rightarrow \sigma_2 ST$ **return** $\mathbf{v}_2 \mapsto^2$ **return** \mathbf{v}'_2 , so we need to show that $(k, \mathbf{v}_1, \mathbf{v}'_2) \in \mathcal{V} \llbracket \exists \alpha. \langle \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger \rangle, \alpha \rrbracket \rho$.

We define R' to be the relation $\{(j, \mathbf{v}_1''', \langle \rangle) \mid j \leq k\}$. Then $R' \in \text{Rel}[\tau_1, \mathbf{1}]$. Define $\rho'' = \rho[\alpha \mapsto (\tau_1, \mathbf{1}, R')]$. Then $(k, \mathbf{v}_1''', \langle \rangle) \in \mathcal{V} \llbracket \alpha \rrbracket \rho''$.

Next, we need to show that for every $(j, \mathbf{v}^2_1, \mathbf{v}^2_2) \in \mathcal{V} \llbracket \langle \alpha, \sigma_1^+ \rangle \rrbracket \rho''$,

$$(j, \mathbf{e}_1[\mathbf{v}^2_1/\mathbf{z}], TS^{\sigma'} \left(\begin{array}{l} \mathbf{let} \mathbf{x} = \sigma'' ST \mathbf{return}_0 (\mathbf{v}^2_2).2 \mathbf{in} \\ \mathbf{v}'_2 \mathbf{x} \end{array} \right)) \in \mathcal{E} \llbracket \sigma_2^\dagger \rrbracket \rho''.$$

$\mathbf{v}^2_1 = \langle \mathbf{v}^3_1, \mathbf{v}^3_1 \rangle$, $\mathbf{v}^2_2 = \langle \langle \rangle, \mathbf{v}^3_2 \rangle$ such that $(j, \mathbf{v}^3_1, \mathbf{v}^3_2) \in \mathcal{V} \llbracket \sigma_1^+ \rrbracket \rho''$.

Furthermore by Lemma 10.5 and Lemma 10.4, there exist $m, n, \mathbf{v}^4_2, \mathbf{v}^3_2$ such that $\sigma_1 ST$ **return** $\mathbf{v}^3_2 \mapsto^m$ \mathbf{v}^4_2 and $TS^{\sigma_1} \mathbf{v}^4_2 \mapsto^n$ **return** \mathbf{v}^4_2 .

$$TS^{\sigma'} \left(\begin{array}{l} \mathbf{let} \mathbf{x} = \sigma'' ST \mathbf{return}_0 (\mathbf{v}^2_2).2 \mathbf{in} \\ \mathbf{v}'_2 \mathbf{x} \end{array} \right) \mapsto^{m+2} \mathbf{v}'_2 \mathbf{v}^4_2 \mapsto^{n+8} \mathbf{e}_2[\langle \mathbf{v}^3_2, \mathbf{v}^4_2 \rangle / \mathbf{x}].$$

Thus by Lemma 7.11 and fact that $(k, \mathbf{v}_1'', \mathbf{v}'_2) \in \mathcal{V} \llbracket \langle \alpha, \sigma_1^+ \rangle \rightarrow \sigma_2^\dagger \rrbracket \rho'$ it is sufficient to show that $(j, \mathbf{v}^2_1, \langle \mathbf{v}^3_2, \mathbf{v}^4_2 \rangle) \in \mathcal{V} \llbracket \langle \alpha, \sigma_1^+ \rangle \rrbracket \rho'$.

Recalling that $\mathbf{v}^2_1 = \langle \mathbf{v}^3_1, \mathbf{v}^3_1 \rangle$, we get $(j, \mathbf{v}^3_1, \mathbf{v}'_2) \in \mathcal{V} \llbracket \alpha \rrbracket \rho'$ by Lemma 7.6. Finally we need to show that $(j, \mathbf{v}^3_1, \mathbf{v}^4_2) \in \mathcal{V} \llbracket \sigma_1^+ \rrbracket \rho'$. By inductive hypothesis and the fact that $(j, \mathbf{v}^3_1, \mathbf{v}^3_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket \rho''$, $(j, \mathbf{v}^3_1, \mathbf{v}^4_2) \in \mathcal{V} \llbracket \sigma^+ \rrbracket \rho''$. Then the property holds by two applications of Lemma 7.4

Case $\sigma = \mu\alpha. \sigma'$, $(\mu\alpha. \sigma')^+ = \mu\alpha. \sigma'^+$: the proof is directly analogous to the case in Lemma 10.6. □

Lemma 10.8 (Boundary Cancellation Equivalence)

1. If $\Delta; \Gamma \vdash \mathbf{e} : \sigma$, then $\Delta; \Gamma \vdash \mathbf{e} \approx_{\text{ST}}^{\text{ctx}} \sigma ST (TS^\sigma \mathbf{e}) : \sigma$.
2. If $\Delta; \Gamma \vdash \mathbf{e} : \sigma^\dagger$, then $\Delta; \Gamma \vdash \mathbf{e} \approx_{\text{ST}}^{\text{ctx}} TS^\sigma (\sigma ST \mathbf{e}) : \sigma^\dagger$.

Proof

By Theorem 7.43, induction on the step index, Lemma 10.6 and Lemma 10.7. □

Lemma 10.9 (Cross Language Relation Alternative)

1. $(k, \mathbf{e}, \mathbf{e}) \in \mathcal{E}^\dagger \llbracket \sigma \rrbracket$ iff $(k, \mathbf{e}, \sigma ST \mathbf{e}) \in \mathcal{E} \llbracket \sigma \rrbracket \emptyset$
2. $(k, \mathbf{e}, \mathbf{e}) \in \mathcal{E}^\dagger \llbracket \sigma \rrbracket$ iff $(k, TS^\sigma \mathbf{e}, \mathbf{e}) \in \mathcal{E} \llbracket \sigma^\dagger \rrbracket \emptyset$
3. $\cdot \vdash \mathbf{e} \approx_{\dagger} \mathbf{e} : \sigma$ iff $\cdot \vdash \mathbf{e} \approx_{\text{ST}}^{\text{ctx}} \sigma ST \mathbf{e} : \sigma$

4. $\cdot \vdash e \approx_{\div} e : \sigma \text{ iff } ; \cdot \vdash \mathcal{TS}^{\sigma} e \approx_{\text{ST}}^{\text{ctx}} e : \sigma^{\div}$

Proof

Case Expansion of definition.

Case By previous case, Lemma 7.35 and Lemma 10.7.

Case By above and induction on k .

Case By above and induction on k . □

Lemma 10.10 (Contextual Boundary Cancellation)

1. $(k, e_1, C[v_2]) \in \mathcal{E}[\theta] \rho$ iff $(k, e_1, C[v'_2]) \in \mathcal{E}[\theta] \rho$ where ${}^{\sigma}ST \mathcal{TS}^{\sigma} v_2 \mapsto^* v'_2$.
2. $(k, e_1, C[e_2]) \in \mathcal{E}[\theta] \rho$ iff $(k, e_1, C[{}^{\sigma}ST \mathcal{TS}^{\sigma} e_2]) \in \mathcal{E}[\theta] \rho$
3. $(k, e_1, C[v_2]) \in \mathcal{E}[\theta] \rho$ iff $(k, e_1, C[v'_2]) \in \mathcal{E}[\theta] \rho$ where $\mathcal{TS}^{\sigma} {}^{\sigma}ST \text{return}_0 v_2 \mapsto^* \text{return}_0 v'_2$.
4. $(k, e_1, C[e_2]) \in \mathcal{E}[\theta] \rho$ iff $(k, e_1, C[\mathcal{TS}^{\sigma} {}^{\sigma}ST e_2]) \in \mathcal{E}[\theta] \rho$

Proof

By Lemma 7.39, Lemma 10.6 and Lemma 10.7. □

Lemma 10.11 (Cross-Language Monadic Bind)

If $(k, e, e) \in \mathcal{E}^{\div}[\sigma]$ and for all $j \leq k$, if $(j, v, v) \in \mathcal{V}^+[\sigma]$ then $(j, K[v], K[\text{return}_0 v]) \in \mathcal{E}^{\div}[\sigma']$, then $(k, K[e], K[e]) \in \mathcal{E}^{\div}[\sigma']$.

Proof

Applying Lemma 10.10 and definition of $\mathcal{E}^{\div}[\sigma']$, it is sufficient to prove that $(k, K[e], {}^{\sigma'}ST K[\mathcal{TS}^{\sigma} {}^{\sigma}ST e]) \in \mathcal{E}[\sigma'] \emptyset$.

By Lemma 7.9, it is sufficient to prove that for all $j \leq k$ and $(j, v_1, v_2) \in \mathcal{V}[\sigma] \emptyset$, $(j, K[v_1], {}^{\sigma'}ST K[\mathcal{TS}^{\sigma} v_2]) \in \mathcal{E}[\sigma'] \emptyset$.

By Lemma 10.4, there exists v_2 such that $\mathcal{TS}^{\sigma} v_2 \mapsto^* \text{return}_0 v_2$. Then by Lemma 7.11, it is sufficient to show that $(j, K[v_1], {}^{\sigma'}ST K[\text{return}_0 v_2]) \in \mathcal{E}[\sigma'] \emptyset$, which holds by hypothesis since $(j, v_1, v_2) \in \mathcal{V}^+[\sigma]$. □

Lemma 10.12 (Cross Language Expression Relation closed under Anti Reduction)

If $(k, e, {}^{\sigma}ST e) \in \text{Atom}[\sigma] \emptyset$, $e \mapsto^{k_1} e'$, $e \mapsto^{k_2} e'$, $(k', e', e') \in \mathcal{E}^{\div}[\sigma]$ and $k \leq k' + \min(k_1, k_2)$ then $(k, e, e) \in \mathcal{E}^{\div}[\sigma]$

Proof

Immediate by definition of the operational semantics and Lemma 7.11 □

Lemma 10.13 (Cross Language Value Relation Embeds in Expression Relation)

If $(k, v, v) \in \mathcal{V}^+[\sigma]$ then $(k, v, \text{return}_0 v) \in \mathcal{E}^{\div}[\sigma]$

Proof

We need to show

$$(k, v, \text{return}_0 v) \in \mathcal{E}^{\div}[\sigma]$$

that is

$$(k, v, {}^{\sigma}ST \text{return}_0 v) \in \mathcal{E}[\sigma] \emptyset$$

By definition of $\mathcal{V}^+[\sigma]$, ${}^{\sigma}ST \text{return}_0 v \mapsto^* v'$ such that $(k, v, v') \in \mathcal{V}[\sigma] \emptyset$. Thus the result holds by Lemma 10.12 and Lemma 7.8. □

Theorem 10.14 (Translation preserves Semantics)

1. If $\Gamma \vdash v : \sigma$, and $\Gamma \vdash v : \sigma \rightsquigarrow_v \mathbf{v}$ then $\Gamma \vdash v \approx_+ \mathbf{v} : \sigma$.
2. If $\Gamma \vdash e : \sigma$, and $\Gamma \vdash e : \sigma \rightsquigarrow_e \mathbf{e}$ then $\Gamma \vdash e \approx_{\div} \mathbf{e} : \sigma$.

Proof

We proceed by mutual induction on the structure of the translation judgments. We omit the cases for unit, sums, pairs, projections, and case. For each case, suppose $(k, \gamma, \gamma) \in \mathcal{G}^+[\Gamma]$.

1. Values:

Case $v = x, \Gamma \vdash x : \sigma \rightsquigarrow_v \mathbf{x}$:

We need to show that there exists v' such that ${}^\sigma \mathcal{ST} \text{ return } \gamma(v) \mapsto^* v'$ such that for any $k \geq 0$, $(k, \gamma(x), v') \in \mathcal{V}[\sigma] \emptyset$. This holds directly by definition of $\mathcal{G}^+[\Gamma]$.

Case $v = \lambda(x : \sigma').e$. Then $\sigma = \sigma' \rightarrow \sigma''$ and $\Gamma \vdash v : \sigma' \rightarrow \sigma'' \rightsquigarrow_v \mathbf{v}$ where

$$\begin{aligned} v = & \text{pack}(\tau_{\text{env}}, \langle \lambda(z : \langle \tau_{\text{env}}, \sigma'^+ \rangle). \dots, \langle y_1, \dots, y_n \rangle \rangle) \text{ as } \exists \alpha. \langle \langle \langle \alpha, \sigma'^+ \rangle \rightarrow \sigma''^{\div} \rangle, \alpha \rangle, \\ & \text{let } y_{\text{env}} = \text{return}_0 \mathbf{z}.1 \text{ in} \\ & \text{let } y_1 = \text{return}_0 y_{\text{env}}.1 \text{ in} \\ & \vdots \\ & \text{let } y_n = \text{return}_0 y_{\text{env}}.n \text{ in} \\ & \text{let } \mathbf{x} = \text{return}_0 \mathbf{z}.2 \text{ in } e \end{aligned}$$

$(y_1, \dots, y_n) = \text{fv}(\lambda(x : \sigma').e)$, $\Gamma(y_i) = \sigma_i$ for each $i \in \{1, \dots, n\}$, $\Gamma' = (y_1 : \sigma_1, \dots, y_n : \sigma_n)$, $\tau_{\text{env}} = \langle \sigma_1^+, \dots, \sigma_n^+ \rangle$, and $\Gamma', x : \sigma \vdash e : \sigma'' \rightsquigarrow_e \mathbf{e}$.

Next, $\sigma' \rightarrow \sigma'' \mathcal{ST} \gamma(v) \mapsto v_2$, where

$$\begin{aligned} v_2 = & \lambda(x : \sigma'). {}^{\sigma''} \mathcal{ST} (\text{unpack}(\alpha, z') = \gamma(v) \text{ in let } x_f = \text{return } z'.1 \text{ in} \quad) \\ & \text{let } x_{\text{env}} = \text{return } z'.2 \text{ in} \\ & \text{let } x' = \mathcal{TS}^{\sigma'} x \text{ in } x_f[\alpha] \langle x_{\text{env}}, x' \rangle \end{aligned}$$

We need to show that $(k, \gamma(v), v_2) \in \mathcal{V}[\sigma' \rightarrow \sigma''] \emptyset$.

Let $j \leq k$, $(j, v'_1, v'_2) \in \mathcal{V}[\sigma']$. We need to show that

$$(j, \gamma(e)[v'_1/x], {}^{\sigma''} \mathcal{ST} (\text{unpack}(\alpha, z') = \gamma(v) \text{ in let } x_f = \text{return } z'.1 \text{ in} \quad)) \in \mathcal{E}[\sigma''] \emptyset$$

$$\begin{aligned} & \text{let } x_{\text{env}} = \text{return } z'.2 \text{ in} \\ & \text{let } x' = \mathcal{TS}^{\sigma'} v'_2 \text{ in } x_f[\alpha] \langle x_{\text{env}}, x' \rangle \end{aligned}$$

By Lemma 10.4, $\mathcal{TS}^{\sigma'} v'_2 \mapsto^* \text{return } v'_2$ for some v'_2 . Then by Lemma 10.6, $(j, v'_1, v'_2) \in \mathcal{V}^+[\sigma']$.

Now define $\gamma'(y_i) = \gamma(y_i)$ for each $y_i \in \text{fv}(v)$ and $\gamma'(x) = v'_1$. Then $\gamma(e)[v'_1/x] = \gamma'(e)$ since $y_1, \dots, y_n, x = \text{fv}(e)$.

Next, define $\gamma'(y_i) = \gamma(y_i)$ for each $y_i \in \text{fv}(v)$ and $\gamma'(x) = v'_2$. Then $(j, \gamma', \gamma') \in \mathcal{G}^+[\Gamma', x : \sigma']$ by Lemma 7.6.

$$\begin{aligned} \text{Next, } & {}^{\sigma''} \mathcal{ST} (\text{unpack}(\alpha, z') = \gamma(v) \text{ in let } x_f = \text{return } z'.1 \text{ in} \quad) \\ & \text{let } x_{\text{env}} = \text{return } z'.2 \text{ in} \\ & \text{let } x' = \mathcal{TS}^{\sigma'} v'_2 \text{ in } x_f[\alpha] \langle x_{\text{env}}, x' \rangle \end{aligned}$$

$$\begin{aligned} & \mapsto^* \gamma'(e[\tau/\alpha][\dots/z'][\dots/x_f][\dots/x_{\text{env}}][\dots/x'][\dots/z][\dots/y_{\text{env}}]) \\ & = \gamma'(e) \end{aligned}$$

The last equality is justified by the fact that $\alpha, z', x_f, x_{\text{env}}, x', z, y_{\text{env}} \notin \text{fv}(e)$ which we know by Theorem 10.1.

Finally, by Lemma 7.11, we need to show that $(j, \gamma'(e), \gamma'(e)) \in \mathcal{E}^{\div}[\sigma'']$ which holds by inductive hypothesis.

Case $v = \text{fold}_{\mu\alpha.\sigma'} v'$: Then $\Gamma \vdash \text{fold}_{\mu\alpha.\sigma} v : \mu\alpha.\sigma \rightsquigarrow_v \text{fold}_{\mu\alpha.\sigma'} v$ where $\Gamma \vdash v : \sigma[\mu\alpha.\sigma/\alpha] \rightsquigarrow_v v$.

By inductive hypothesis, $\exists v_2. \gamma(\mathcal{TS}^{\sigma[\mu\alpha.\sigma/\alpha]} v) \mapsto^* v_2 \wedge (k, \gamma(v'), v_2) \in \mathcal{V}[\sigma[\mu\alpha.\sigma/\alpha]] \emptyset$.

Then by the operational semantics, $\mathcal{TS}^{\mu\alpha.\sigma} \text{fold}_{\mu\alpha.\sigma'} v \mapsto^* \text{fold}_{\mu\alpha.\sigma'} v_2$.

We need to show $(k, \text{fold}_{\mu\alpha.\sigma'} \gamma(v'), \text{fold}_{\mu\alpha.\sigma'} v_2) \in \mathcal{V}[\mu\alpha.\sigma'] \emptyset$.

If $k = 0$, this is trivial. Otherwise the result follows by Lemma 7.6.

2. Expressions:

Case $e = v, \Gamma \vdash v : \sigma \rightsquigarrow_e \text{return } v$ where $\Gamma \vdash v : \sigma \rightsquigarrow_v v$: we need to show that

$$(k, \gamma(v), {}^\sigma\mathcal{ST} \text{return } \gamma(v)) \in \mathcal{E}[\sigma] \emptyset.$$

By inductive hypothesis there is a v' such that ${}^\sigma\mathcal{ST} \gamma(v) \mapsto^* v'$ and $(k, \gamma(v), v') \in \mathcal{V}[\sigma] \emptyset$, so the result holds by Lemma 7.11 and Lemma 7.8.

Case $e = v_1 v_2$: Then

$$\Gamma \vdash v_1 v_2 : \sigma_2 \rightsquigarrow_e \text{unpack } (\alpha, z) = v_1 \text{ in} \\ \text{let } y_1 = \text{return } z.1 \text{ in} \\ \text{let } y_2 = \text{return } z.2 \text{ in} \\ y_1 \langle y_2, v_2 \rangle$$

where $\Gamma \vdash v_1 : \sigma_1 \rightsquigarrow_v v_1$ and $\Gamma \vdash v_2 : \sigma_1 \rightsquigarrow_v v_2$.

We need to show that

$$\left(\begin{array}{l} k, \gamma(v_1) \gamma(v_2), \text{unpack } (\alpha, z) = \gamma(v_1) \text{ in} \\ \text{let } y_1 = \text{return } z.1 \text{ in} \\ \text{let } y_2 = \text{return } z.2 \text{ in} \\ y_1 \langle y_2, \gamma(v_2) \rangle \end{array} \right) \in \mathcal{E}^\dagger[\sigma_2]$$

By Lemma 10.10, it is sufficient to show that

$$\left(\begin{array}{l} k, \gamma(v_1) \gamma(v_2), \text{unpack } (\alpha, z) = v'_1 \text{ in} \\ \text{let } y_1 = \text{return } z.1 \text{ in} \\ \text{let } y_2 = \text{return } z.2 \text{ in} \\ y_1 \langle y_2, \gamma(v_2) \rangle \end{array} \right) \in \mathcal{E}^\dagger[\sigma_2]$$

where $\mathcal{TS}^{\sigma_1 \rightarrow \sigma_2} \sigma_1 \rightarrow \sigma_2 \mathcal{ST} \text{return } \gamma(v_1) \mapsto^* \text{return } v'_1$. By definition of the operational semantics we see that

$$v'_1 = \text{pack } (1, \langle \lambda(z : \langle 1, \sigma''^+ \rangle). \mathcal{TS}^{\sigma'} \left(\begin{array}{l} \text{let } x = \sigma'' \mathcal{ST} \text{return}_0 z.2 \text{ in} \\ v'_1 x \end{array} \right), \langle \rangle \rangle) \text{ as } (\sigma_1 \rightarrow \sigma_2)^+$$

where

$$v'_1 = \lambda(x : \sigma_1). {}^{\sigma_2}\mathcal{ST} (\text{unpack } (\alpha, z) = v_1 \text{ in let } x_f = \text{return } z.1 \text{ in} \\ \text{let } x_{\text{env}} = \text{return } z.2 \text{ in} \\ \text{let } x = \mathcal{TS}^{\sigma_1} x \text{ in } x_f [\alpha] \langle x_{\text{env}}, x \rangle)$$

By definition of $\mathcal{V}^+[\![\]\!]_{\emptyset}$ and inductive hypothesis, $(k, \gamma(v_1), v'_1) \in \mathcal{V}[\![\sigma_1 \rightarrow \sigma_2]\!]_{\emptyset}$.
Next,

$$\begin{aligned} \sigma_2 \mathcal{ST} \text{unpack } (\alpha, z) = v'_1 \text{ in} \\ \text{let } y_1 = \text{return } z.1 \text{ in} \\ \text{let } y_2 = \text{return } z.2 \text{ in} \\ y_1 \langle y_2, \gamma(v_2) \rangle &\xrightarrow{5} \sigma_2 \mathcal{ST} \left(\left(\begin{array}{l} \lambda(z : \langle 1, \sigma_1^+ \rangle). \\ \mathcal{TS}^{\sigma_2} \left(\text{let } x = \sigma_1 \mathcal{ST} \text{return}_0 z.2 \text{ in} \\ v'_1 x \right) \end{array} \right) \langle \langle \rangle, \gamma(v_2) \rangle \right) \\ &\xrightarrow{2} \sigma_2 \mathcal{ST} \mathcal{TS}^{\sigma_2} (\text{let } x = \sigma_1 \mathcal{ST} \text{return}_0 \gamma(v_2) \text{ in } v'_1 x) \end{aligned}$$

Therefore by Lemma 7.11 and Lemma 10.6, it is sufficient to show that

$$(k, \gamma(v_1) \gamma(v_2), \text{let } x = \sigma_1 \mathcal{ST} \text{return}_0 \gamma(v_2) \text{ in } v'_1 x) \in \mathcal{E}[\![\sigma_2]\!]_{\emptyset}.$$

Next, by Lemma 10.5, $\sigma_1 \mathcal{ST} \text{return}_0 \gamma(v_2) \xrightarrow{*} v''_2$ and by inductive hypothesis $(k, \gamma(v_2), v''_2) \in \mathcal{V}[\![\sigma_1]\!]_{\emptyset}$, so $\text{let } x = \sigma_1 \mathcal{ST} \text{return}_0 \gamma(v_2) \text{ in } v'_1 x \xrightarrow{*} v'_1 v''_2$, so by Lemma 7.11 it is sufficient to show that

$$(k, \gamma(v_1) \gamma(v_2), v'_1 v''_2) \in \mathcal{E}[\![\sigma_2]\!]_{\emptyset},$$

which holds by similar reasoning to Lemma 7.19.

Case $e = \text{unfold } v, \Gamma \vdash e : \sigma[\mu\alpha. \sigma/\alpha] \rightsquigarrow_e \text{return unfold } v$, where $\Gamma \vdash v : \mu\alpha. \sigma \rightsquigarrow_v v$.

We need to show that for all $k \geq 0$,

$$(k, \text{unfold } \gamma(v), \sigma[\mu\alpha. \sigma/\alpha] \mathcal{ST} \text{return unfold } \gamma(v)) \in \mathcal{E}[\![\sigma[\mu\alpha. \sigma/\alpha]^{\dagger}]\!]_{\emptyset}$$

and by inductive hypothesis and Lemma 10.5, $\mu\alpha. \sigma \mathcal{ST} \text{return } \gamma(v) \xrightarrow{*} v'$ and $(k, \gamma(v), v') \in \mathcal{V}[\![\mu\alpha. \sigma]\!]_{\emptyset}$. By definition of $\mathcal{V}[\![\mu\alpha. \sigma]\!]_{\emptyset}$, this means $\gamma(v) = \text{fold}_{\mu\alpha. \sigma} v_1$ and $v' = \text{fold}_{\mu\alpha. \sigma} v_2$ where for all $j < k$, $(j, v_1, v_2) \in \mathcal{V}[\![\sigma[\mu\alpha. \sigma/\alpha]]\!]_{\emptyset}$.

Then by definition of the operational semantics, $\gamma(v) = \text{fold}_{(\mu\alpha. \sigma)^+} v_2$ where $\sigma[\mu\alpha. \sigma/\alpha] \mathcal{ST} \text{return } v_2 \xrightarrow{*} v_2$.

Therefore

$$\text{unfold } \gamma(v) = \text{unfold fold}_{\mu\alpha. \sigma} v_1 \xrightarrow{*} v_1$$

and

$$\begin{aligned} \sigma[\mu\alpha. \sigma/\alpha] \mathcal{ST} \text{return unfold } \gamma(v) &= \sigma[\mu\alpha. \sigma/\alpha] \mathcal{ST} \text{return unfold } (\text{fold}_{(\mu\alpha. \sigma)^+} v_2) \\ &\xrightarrow{*} \sigma[\mu\alpha. \sigma/\alpha] \mathcal{ST} \text{return } v_2 \\ &\xrightarrow{*} v_2 \end{aligned}$$

so by Lemma 7.11, it is sufficient to show that $(k-1, v_1, v_2) \in \mathcal{E}[\![\sigma[\mu\alpha. \sigma/\alpha]]\!]_{\emptyset}$, which holds by inductive hypothesis and Lemma 7.8.

Case $e = \text{let } x = e_1 \text{ in } e_2, \Gamma \vdash e : \sigma \rightsquigarrow_e \text{handle } e_1 \text{ with } (x. e_2) (y. \text{raise } y)$, where $\Gamma \vdash e_1 : \sigma' \rightsquigarrow_e e_1$ and $\Gamma, x : \sigma' \vdash e_2 : \sigma \rightsquigarrow_e e_2$.

We need to show that for all $k \geq 0$,

$$(k, \text{let } x = \gamma(e_1) \text{ in } \gamma(e_2), \sigma \mathcal{ST} (\text{handle } \gamma(e_1) \text{ with } (x. \gamma(e_2)) (y. \text{raise } y))) \in \mathcal{E}[\![\sigma^{\dagger}]\!]_{\emptyset}.$$

By Lemma 10.10, it is sufficient to show that

$$(k, \text{let } x = \gamma(e_1) \text{ in } \gamma(e_2), \sigma \mathcal{ST} (\text{handle } \mathcal{TS}^{\sigma'} \sigma' \mathcal{ST} \gamma(e_1) \text{ with } (x. \gamma(e_2)) (y. \text{raise } y))) \in \mathcal{E}[\![\sigma^{\dagger}]\!]_{\emptyset}$$

By inductive hypothesis, $(k, \gamma(e_1), {}^\sigma ST \gamma(e_1)) \in \mathcal{E} \llbracket \sigma^{\dot{\div}} \rrbracket \emptyset$. By Lemma 7.9, it is sufficient to show that for all $j \leq k$, $(j, \mathbf{v}_1, \mathbf{v}_2) \in \mathcal{V} \llbracket \sigma' \rrbracket \emptyset$,

$$(j, \text{let } x = \mathbf{v}_1 \text{ in } \gamma(e_2), {}^\sigma ST (\text{handle } \mathcal{TS}^{\sigma'} \mathbf{v}_2 \text{ with } (x, \gamma(e_2)) (y, \text{raise } y))) \in \mathcal{E} \llbracket \sigma'^{\dot{\div}} \rrbracket \emptyset.$$

By Lemma 10.4, there exists \mathbf{v}_2 such that $\mathcal{TS}^{\sigma'} \mathbf{v}_2 \mapsto^* \text{return } \mathbf{v}_2$. Define $\gamma' = \gamma[x \mapsto \mathbf{v}_1]$, $\gamma' = \gamma[x \mapsto \mathbf{v}_2]$. Then by Lemma 10.9, $(k, \gamma', \gamma') \in \mathcal{G}^+ \llbracket \Gamma, x : \sigma' \rrbracket$. Finally,

$$\text{let } x = \mathbf{v}_1 \text{ in } \gamma(e_2) \mapsto \gamma'(e_2)$$

and

$${}^\sigma ST (\text{handle } \mathcal{TS}^{\sigma'} \mathbf{v}_2 \text{ with } (x, \gamma(e_2)) (y, \text{raise } y)) \mapsto {}^\sigma ST \gamma'(e_2)$$

So by Lemma 7.11, it is sufficient to show that $(j, \gamma'(e_2), {}^\sigma ST \gamma'(e_2)) \in \mathcal{E} \llbracket \sigma^+ \rrbracket \emptyset$, which holds by inductive hypothesis. □

Lemma 10.15 (Translation and Back-Translation Preserves and Reflects Termination)

1. If $\cdot \vdash e : \sigma \rightsquigarrow_e e$ then $e \Downarrow$ iff $e \Downarrow$.
2. If $\cdot; \cdot \vdash^{\dot{\div}} e : \theta \rightarrow e_u$ then $e \Downarrow$ iff $e_u \Downarrow$

Proof

By Lemma 10.14, $\cdot \vdash e \approx_{\dot{\div}} e : \sigma$. Unfolding definitions, we get $\forall k, (k, e, {}^\sigma ST e) \in \mathcal{E} \llbracket \sigma \rrbracket \emptyset$. Choosing $(k, [\cdot], [\cdot]) \in \mathcal{K} \llbracket \sigma \rrbracket \emptyset$, we get that $\forall k, (k, e, {}^\sigma ST e) \in \mathcal{O}$.

Then if $e \mapsto^j v$, since $(j+1, e, {}^\sigma ST e) \in \mathcal{O}$, ${}^\sigma ST e \Downarrow$. Furthermore, if ${}^\sigma ST e \Downarrow$ then $e \Downarrow$.

The other direction can be proved by a symmetric argument by starting with $\forall k, (k, \mathcal{TS}^\sigma e, e) \in \mathcal{E} \llbracket \sigma^{\dot{\div}} \rrbracket \emptyset$.

By Theorem 9.11, $\cdot; \cdot \vdash e_u \approx_{\mathcal{E}^U}^{log} e : \theta$. Unfolding definitions we get $\forall k, (k, e_u, e) \in \mathcal{E}^U \llbracket \theta \rrbracket \emptyset$.

Then we have $\forall k, (k, \text{let } x = [\cdot] \text{ in } \langle \rangle, \mathcal{TS}^{\langle \rangle} \text{handle } [\cdot] \text{ with } (x, \text{return } \langle \rangle) (y, \text{return } \langle \rangle)) \in \mathcal{K}^U \llbracket \theta \rrbracket \emptyset$.

Then if $e_u \mapsto^j v_u$, $(j+2, \text{let } x = e_u \text{ in } \langle \rangle, \mathcal{TS}^{\langle \rangle} \text{let } x = e \text{ in } \langle \rangle) \in \mathcal{O}$ and $\text{let } x = e_u \text{ in } \langle \rangle \not\mapsto^{j+2}$, so $\mathcal{TS}^{\langle \rangle} \text{let } x = e \text{ in } \langle \rangle \Downarrow$, and therefore $e \Downarrow$.

A similar argument gives the reverse implication. □

10.2 Full Abstraction

Lemma 10.16 (Translation is Equivalent to Embedding)

If $e \in \lambda^S$ and $\Gamma \vdash e : \sigma$ and $\Gamma \vdash e : \sigma \rightsquigarrow_e e$, and $\Gamma = x_1 : \sigma_1, \dots, x_n : \sigma_n$ then

$$\begin{aligned} \cdot; \Gamma^+ \vdash e \approx_{\text{ST}}^{ctx} \mathcal{TS}^\sigma \text{let } x_1 = \sigma_1 ST \text{return } x_1 \text{ in } : \sigma^{\dot{\div}}. \\ \vdots \\ \text{let } x_n = \sigma_n ST \text{return } x_n \text{ in} \\ e \end{aligned}$$

We denote the term on the right as $\mathcal{TS}^\sigma \text{let } \Gamma = ST \Gamma^+ \text{ in } e$.

Proof

By Theorem 7.43, it is sufficient to show that $\cdot; \Gamma^+ \vdash \mathbf{e} \approx_{\mathcal{E}}^{\text{log}} \mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \mathbf{e} : \sigma^\dagger$.

Suppose $(k, \gamma) \in \mathcal{G}[\Gamma^+] \emptyset$. Then by Lemma 10.5, for each $\mathbf{x}_i : \sigma_i \in \Gamma$,

$$\sigma_i \mathcal{ST} \text{ return } \gamma_2(\mathbf{x}_i) \mapsto^* \mathbf{v}_i$$

for some \mathbf{v}_i and $(k, \mathbf{v}_i, \gamma_2(\mathbf{x}_i)) \in \mathcal{V}^+[\sigma_i]$.

Therefore

$$\mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \gamma_2(\Gamma^+) \text{ in } \gamma_2(\mathbf{e}) \mapsto^* \mathcal{TS}^\sigma (\dots \gamma_2(\mathbf{e})[\mathbf{v}_1/\mathbf{x}_1] \dots)[\mathbf{v}_n/\mathbf{x}_n]$$

Next, $\gamma_2(\mathbf{x}_i) = \mathbf{v}_i$ since $\Gamma \cap \Gamma^+ = \emptyset$. Define $\gamma(\mathbf{x}_i) = \mathbf{v}_i$ for each $\mathbf{x}_i \in \Gamma$.

Next we want to show that $(k, \gamma, \gamma_1) \in \mathcal{G}^U[\Gamma]$. For any $\mathbf{x}_i : \sigma_i$, we have $(k, \gamma(\mathbf{x}_i) = \mathbf{v}_i, \gamma_2(\mathbf{x}_i)) \in \mathcal{V}^+[\sigma_i]$ and $(k, \gamma_2(\mathbf{x}_i), \gamma_1(\mathbf{x}_i)) \in \mathcal{V}[\sigma_i^+] \emptyset$. But by Lemma 7.35, $\sigma_i \mathcal{ST} \text{ return } \gamma_1(\mathbf{x}_i) \mapsto^* \mathbf{v}'_i$ and $(k, \mathbf{v}_i, \mathbf{v}'_i) \in \mathcal{V}[\sigma_i] \emptyset$, that is, $(k, \mathbf{v}_i, \gamma_1(\mathbf{x}_i)) \in \mathcal{V}^+[\sigma_i]$. Then we have $(k, \gamma, \gamma_1) \in \mathcal{G}^U[\Gamma]$.

Then by Lemma 7.11, it is sufficient to show that

$$(k, \mathcal{TS}^\sigma \gamma(\mathbf{e}), \gamma(\mathbf{e})) \in \mathcal{E}[\sigma^\dagger] \emptyset$$

which by Lemma 10.9 is equivalent to showing

$$(k, \gamma(\mathbf{e}), \gamma(\mathbf{e})) \in \mathcal{E}^\dagger[\sigma]$$

which follows from Lemma 10.14. □

Theorem 10.17 (Source Equivalence Implies Multi-language Equivalence)

If $\mathbf{e}_1, \mathbf{e}_2 \in \lambda^S$ and $\Gamma \vdash \mathbf{e}_1 \approx_5^{\text{ctx}} \mathbf{e}_2 : \sigma$, then $\cdot; \Gamma \vdash \mathbf{e}_1 \approx_{\text{ST}}^{\text{ctx}} \mathbf{e}_2 : \sigma$.

Proof

We show one direction of the equivalence, the other follows by symmetry.

Suppose $C \in \lambda^{\text{ST}}$ is an appropriate closing context and $C[\mathbf{e}_1] \Downarrow$. We need to show that $C[\mathbf{e}_2] \Downarrow$.

By Lemma 9.14 and Lemma 9.15, we back-translate $\cdot; \cdot \vdash C[\mathbf{e}_1] : \sigma' \rightarrow C[\mathbf{e}_1]$ and $\cdot; \cdot \vdash C[\mathbf{e}_2] : \sigma' \rightarrow C[\mathbf{e}_2]$ where $C \in \lambda^S$.

By Lemma 10.15, $C[\mathbf{e}_1] \Downarrow$ iff $C[\mathbf{e}_1] \Downarrow$ and $C[\mathbf{e}_2] \Downarrow$ iff $C[\mathbf{e}_2] \Downarrow$.

Since $C \in \lambda^S$ and $\Gamma \vdash \mathbf{e}_1 \approx_5^{\text{ctx}} \mathbf{e}_2 : \sigma$, $C[\mathbf{e}_1] \Downarrow$ iff $C[\mathbf{e}_2] \Downarrow$.

Then we compose the iffs, to get the result:

$$C[\mathbf{e}_1] \Downarrow \text{ iff } C[\mathbf{e}_1] \Downarrow \text{ iff } C[\mathbf{e}_2] \Downarrow \text{ iff } C[\mathbf{e}_2] \Downarrow.$$

□

Theorem 10.18 (Translation Preserves Multi-language Equivalence)

If $\cdot; \Gamma \vdash \mathbf{e}_1 \approx_{\text{ST}}^{\text{ctx}} \mathbf{e}_2 : \sigma$, $\Gamma \vdash \mathbf{e}_1 : \sigma \rightsquigarrow_e \mathbf{e}_1$ and $\Gamma \vdash \mathbf{e}_2 : \sigma \rightsquigarrow_e \mathbf{e}_2$, then $\cdot; \Gamma^+ \vdash \mathbf{e}_1 \approx_{\text{ST}}^{\text{ctx}} \mathbf{e}_2 : \sigma^\dagger$.

Proof

By Lemma 10.16,

$$\cdot; \Gamma^+ \vdash \mathbf{e} \approx_{\text{ST}}^{\text{ctx}} \mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \mathbf{e} : \sigma^\dagger$$

and

$$\cdot; \Gamma^+ \vdash \mathbf{e}' \approx_{\text{ST}}^{\text{ctx}} \mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \mathbf{e}' : \sigma^\dagger.$$

Since $\cdot; \Gamma \vdash \mathbf{e}_1 \approx_{\text{ST}}^{\text{ctx}} \mathbf{e}_2 : \sigma$,

$$\cdot; \Gamma^+ \vdash \mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \mathbf{e} \approx_{\text{ST}}^{\text{ctx}} \mathcal{TS}^\sigma \text{let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \mathbf{e}' : \sigma^\dagger.$$

The result then holds by transitivity of contextual equivalence. □

Theorem 10.19 (Multi-language Equivalence Implies Target Equivalence)

If $\cdot; \Gamma^+ \vdash e_1 \approx_{\text{ST}}^{ctx} e_2 : \sigma^\dagger$, then $\cdot; \Gamma^+ \vdash e_1 \approx_{\mathbf{T}}^{ctx} e_2 : \sigma^\dagger$.

Proof

Trivial, since every target context is a multi-language context. □

Theorem 10.20 (Translation is Equivalence Preserving)

If $\Gamma \vdash e \approx_S^{ctx} e' : \sigma$, $\Gamma \vdash e : \sigma \rightsquigarrow_e e$ and $\Gamma \vdash e' : \sigma \rightsquigarrow_e e'$ then $\cdot; \Gamma^+ \vdash e \approx_{\mathbf{T}}^{ctx} e' : \sigma^\dagger$.

Proof (1: Decomposed)

By composition of Theorem 10.17, Theorem 10.18 and Theorem 10.19. □

Proof (2: Direct)

We prove one direction, the other case holds by symmetry. Suppose $\mathbf{C} \in \lambda^{\mathbf{T}}$ appropriately typed.

By Lemma 10.16,

$$\cdot; \Gamma^+ \vdash e \approx_{\text{ST}}^{ctx} \mathcal{TS}^\sigma \text{ let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } e : \sigma^\dagger$$

and

$$\cdot; \Gamma^+ \vdash e' \approx_{\text{ST}}^{ctx} \mathcal{TS}^\sigma \text{ let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } e' : \sigma^\dagger.$$

Let $C = \mathbf{C}[\mathcal{TS}^\sigma \text{ let } \Gamma = \mathcal{ST} \Gamma^+ \text{ in } \cdot]$. Then $\mathbf{C}[e] \Downarrow$ iff $C[e] \Downarrow$ and $\mathbf{C}[e'] \Downarrow$ iff $C[e'] \Downarrow$.

Next, by Lemma 9.15 and Lemma 9.14, we back-translate,

$$\cdot; \cdot \vdash C[e] : \theta \Rightarrow \mathbf{C}[e]$$

and

$$\cdot; \cdot \vdash C[e'] : \theta \Rightarrow \mathbf{C}[e'].$$

Then by Lemma 10.15, $C[e] \Downarrow$ iff $\mathbf{C}[e] \Downarrow$ and $C[e'] \Downarrow$ iff $\mathbf{C}[e'] \Downarrow$.

Then since $\Gamma \vdash e \approx_S^{ctx} e' : \sigma$, $\mathbf{C}[e] \Downarrow$ iff $\mathbf{C}[e'] \Downarrow$.

Then we can compose the above ifs to get the result. In summary:

$$\mathbf{C}[e] \Downarrow \text{ iff } C[e] \Downarrow \text{ iff } \mathbf{C}[e] \Downarrow \text{ iff } \mathbf{C}[e'] \Downarrow \text{ iff } C[e'] \Downarrow \text{ iff } \mathbf{C}[e'] \Downarrow.$$

□

Theorem 10.21 (Translation is Equivalence Reflecting)

If $\Gamma \vdash e : \sigma \rightsquigarrow_e e$, $\Gamma \vdash e' : \sigma \rightsquigarrow_e e'$ and $\cdot; \Gamma^+ \vdash e \approx_{\mathbf{T}}^{ctx} e' : \sigma^\dagger$ then $\Gamma \vdash e \approx_S^{ctx} e' : \sigma$.

Proof

Assume $\vdash \mathbf{C} : (\cdot; \Gamma \vdash \sigma) \Rightarrow (\cdot; \cdot \vdash \sigma')$ We need to show that $\mathbf{C}[e] \Downarrow$ iff $\mathbf{C}[e'] \Downarrow$.

First by Lemma 10.3, $\cdot \vdash \mathbf{C}[e] : \sigma' \rightsquigarrow_e \mathbf{C}[e]$ and $\cdot \vdash \mathbf{C}[e'] : \sigma' \rightsquigarrow_e \mathbf{C}[e']$.

Then by Lemma 10.15, $\mathbf{C}[e] \Downarrow$ iff $\mathbf{C}[e] \Downarrow$ and $\mathbf{C}[e'] \Downarrow$ iff $\mathbf{C}[e'] \Downarrow$.

Then since $\cdot; \Gamma^+ \vdash e \approx_{\mathbf{T}}^{ctx} e' : \sigma^\dagger$ and $\mathbf{C} \in \lambda^{\mathbf{T}}$, $\mathbf{C}[e] \Downarrow$ iff $\mathbf{C}[e'] \Downarrow$.

Finally, we compose the ifs to obtain our result:

$$\mathbf{C}[e] \Downarrow \text{ iff } \mathbf{C}[e] \Downarrow \text{ iff } \mathbf{C}[e'] \Downarrow \text{ iff } \mathbf{C}[e'] \Downarrow$$

□

Theorem 10.22 (Translation is Fully Abstract)

If $\Gamma \vdash e : \sigma \rightsquigarrow_e e$ and $\Gamma \vdash e' : \sigma \rightsquigarrow_e e'$ then $\Gamma \vdash e \approx_S^{ctx} e' : \sigma$ if and only if $\cdot; \Gamma^+ \vdash e \approx_{\mathbf{T}}^{ctx} e' : \sigma^\dagger$.

Proof

Immediate by Theorem 10.20 and Theorem 10.21 □