Supplementary Material for "Abstracting Effect Systems for Algebraic Effect Handlers"

Takuma Yoshioka[1], Taro Sekiyama[2], and Atsushi Igarashi[1]

[1]Kyoto University, Japan
[2]National Institute of Informatics & SOKENDAI, Japan

1 Definitions

Remark 1.1 (Notation). *We write* $\boldsymbol{\alpha}^I$ *for a finite sequence* $\alpha_0, \ldots, \alpha_n$ *with an index set* $I = \{0, \ldots, n\}$*, where* α *is any metavariable. We also write* $\{\boldsymbol{\alpha}^I\}$ *for the set consisting of the elements of* $\boldsymbol{\alpha}^I$.

Definition 1.2 (Kinds). *Kinds are defined as* $K ::= \mathbf{Typ} \mid \mathbf{Lab} \mid \mathbf{Eff}$.

Definition 1.3 (Signatures). *Given a set* S *of label names, a label signature* Σ_{lab} *is a functional relation whose domain* $\mathrm{dom}(\Sigma_{\mathrm{lab}})$ *is* S*. The codomain of* Σ_{lab} *is the set of functional kinds of the form* $\Pi_{i\in I} K_i \to \mathbf{Lab}$ *for some* I *and* $K_i^{i\in I}$ *(if* $I = \emptyset$*, it means* $\mathbf{Lab}$ *simply). Similarly, given a set* S *of effect constructors, an effect signature* Σ_{eff} *is a function relation whose domain* $\mathrm{dom}(\Sigma_{\mathrm{eff}})$ *is* S *and its codomain is the set of functional kinds of the form* $\Pi_{i\in I} K_i \to \mathbf{Eff}$ *for some* I *and* $K_i^{i\in I}$*. A signature* Σ *is the disjoint union of a label signature and an effect signature. We write* $\Pi \boldsymbol{K}^I \to K$*, and more simply,* $\Pi\boldsymbol{K} \to K$ *as an abbreviation for* $\Pi_{i\in I} K_i \to K$.

Remark 1.4. *We write* $\mathcal{C} : \Pi\boldsymbol{K} \to K$ *to denote the pair* $\langle \mathcal{C}, \Pi\boldsymbol{K} \to K\rangle$ *for label name or effect constructor* C.

Definition 1.5 (The Syntax of λ_{EA}). *Given a signature* $\Sigma = \Sigma_{\mathrm{lab}} \uplus \Sigma_{\mathrm{eff}}$*, the syntax of* λ_{EA} *is defined as follows.*

I, J, N	(index sets)	i, j, n, r	(indices)
f, g, x, y, z, p, k	(variables)	$\alpha, \beta, \gamma, \tau, \iota, \rho$	(typelike variables)
op	(operation names)	$l \in \mathrm{dom}(\Sigma_{\mathrm{lab}})$	(label names)
$\mathcal{F} \in \mathrm{dom}(\Sigma_{\mathrm{eff}})$	(effect constructors)	$\mathcal{C} \in \mathrm{dom}(\Sigma_{\mathrm{lab}}) \cup \mathrm{dom}(\Sigma_{\mathrm{eff}})$	

$$
\begin{array}{rcll}
K & ::= & \mathbf{Typ} \mid \mathbf{Lab} \mid \mathbf{Eff} & \text{(kinds)} \\
S, T & ::= & A \mid L \mid \varepsilon & \text{(typelikes)} \\
A, B, C & ::= & \tau \mid A \to_\varepsilon B \mid \forall \alpha : K. A^\varepsilon & \text{(types)} \\
L & ::= & \iota \mid l\, \boldsymbol{S}^I & \text{(labels)} \\
\varepsilon & ::= & \rho \mid \mathcal{F}\, \boldsymbol{S}^I & \text{(effects)} \\
\sigma & ::= & \{\} \mid \sigma \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J. A \Rightarrow B\} & \text{(operation signatures)} \\
\Xi & ::= & \emptyset \mid \Xi, l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I. \sigma & \text{(effect contexts)} \\
\Gamma & ::= & \emptyset \mid \Gamma, x : A \mid \Gamma, \alpha : K & \text{(typing contexts)} \\
e & ::= & v \mid v_1\, v_2 \mid v\, S \mid \mathbf{let}\, x = e_1 \,\mathbf{in}\, e_2 \mid \mathbf{handle}_{l\, \boldsymbol{S}^I}\, e \,\mathbf{with}\, h & \text{(expressions)} \\
v & ::= & x \mid \mathbf{fun}\,(f, x, e) \mid \Lambda \alpha : K. e \mid \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J & \text{(values)} \\
h & ::= & \{\mathbf{return}\, x \mapsto e\} \mid h \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\} & \text{(handlers)} \\
E & ::= & \Box \mid \mathbf{let}\, x = E \,\mathbf{in}\, e \mid \mathbf{handle}_{l\, \boldsymbol{S}^I}\, E \,\mathbf{with}\, h & \text{(evaluation contexts)}
\end{array}
$$

Remark 1.6. *We write* $\lambda x.e$ *for* $\mathbf{fun}\,(f, x, e)$ *if variable* f *does not occur in expression* e.

Definition 1.7 (Free Variables). *The notion of free variables is defined as usual. We write* $\mathrm{FV}(e)$ *for the set of free variables in expression* e.

Definition 1.8 (Free Typelike Variables). *The notion of free typelike variables is defined as usual. We write* $\mathrm{FTV}(e)$ *and* $\mathrm{FTV}(S)$ *for the sets of free typelike variables in expression* e *and typelike* S*, respectively.*

Definition 1.9 (Value Substitution). *Substitution $e[v/x]$ and $h[v/x]$ of value v for variable x in expression e and handler h, respectively, are defined as follows:*

$$
\begin{aligned}
x[v/x] &= v \\
y[v/x] &= y \quad (\text{if } x \neq y) \\
\mathbf{fun}\,(f, y, e)[v/x] &= \mathbf{fun}\,(f, y, e[v/x]) \quad (\text{if } f, y \notin \mathrm{FV}(v) \cup \{x\}) \\
(\Lambda\alpha : K.e)[v/x] &= \Lambda\alpha : K.e[v/x] \quad (\text{if } \alpha \notin \mathrm{FTV}(v)) \\
\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J[v/x] &= \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J \\
(v_1\,v_2)[v/x] &= (v_1[v/x])\,(v_2[v/x]) \\
(v'\,S)[v/x] &= (v'[v/x])\,S \\
(\mathbf{handle}_{l\,\boldsymbol{S}^N}\,e\,\mathbf{with}\,h)[v/x] &= \mathbf{handle}_{l\,\boldsymbol{S}^N}\,e[v/x]\,\mathbf{with}\,(h[v/x]) \\
(\mathbf{let}\,y = e_1\,\mathbf{in}\,e_2)[v/x] &= \mathbf{let}\,y = e_1[v/x]\,\mathbf{in}\,e_2[v/x] \\
&\qquad (\text{if } y \neq x \text{ and } y \notin \mathrm{FV}(v)) \\
([e]_L)[v/x] &= [e[v/x]]_L \\
\{\,\mathbf{return}\,y \mapsto e_r\}[v/x] &= \{\,\mathbf{return}\,y \mapsto e_r[v/x]\} \\
&\qquad (\text{if } y \neq x \text{ and } y \notin \mathrm{FV}(v)) \\
(h \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\,p\,k \mapsto e\})[v/x] &= h[v/x] \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\,p\,k \mapsto e[v/x]\} \\
&\qquad (\text{if } x \neq p, k \text{ and } p, k \notin \mathrm{FV}(v) \text{ and } \{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(v) = \emptyset)
\end{aligned}
$$

Definition 1.10 (Typelike Substitution). *Substitution $e[S/\alpha]$, $h[S/\alpha]$, $T[S/\alpha]$, and $\Gamma[S/\alpha]$ of typelike S for typelike variable α in expression e, handler h, typelike T, and typing context Γ, respectively, are defined as follows:*

$$
\begin{aligned}
x[S/\alpha] &= x \\
(\mathbf{fun}\,(f, x, e))[S/\alpha] &= \mathbf{fun}\,(f, x, e[S/\alpha]) \\
(\Lambda\beta : K.e)[S/\alpha] &= \Lambda\beta : K.(e[S/\alpha]) \quad (\text{if } \alpha \neq \beta \text{ and } \beta \notin \mathrm{FTV}(S)) \\
(\mathsf{op}_{l\,\boldsymbol{S'}^I}\,\boldsymbol{T}^J)[S/\alpha] &= \mathsf{op}_{l\,\boldsymbol{S'[S/\alpha]}^I}\,\boldsymbol{T[S/\alpha]}^J \\
(v_1\,v_2)[S/\alpha] &= (v_1[S/\alpha])\,(v_2[S/\alpha]) \\
(v\,T)[S/\alpha] &= (v[S/\alpha])\,(T[S/\alpha]) \\
(\mathbf{handle}_{l\,\boldsymbol{T}^N}\,e\,\mathbf{with}\,h)[S/\alpha] &= \mathbf{handle}_{l\,\boldsymbol{T[S/\alpha]}^N}\,e[S/\alpha]\,\mathbf{with}\,(h[S/\alpha]) \\
(\mathbf{let}\,x = e_1\,\mathbf{in}\,e_2)[S/\alpha] &= \mathbf{let}\,x = e_1[S/\alpha]\,\mathbf{in}\,e_2[S/\alpha] \\
([e]_L)[S/\alpha] &= [e[S/\alpha]]_{L[S/\alpha]} \\
\{\,\mathbf{return}\,x \mapsto e_r\}[S/\alpha] &= \{\,\mathbf{return}\,x \mapsto e_r[S/\alpha]\} \\
(h \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\,p\,k \mapsto e\})[S/\alpha] &= h[S/\alpha] \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\,p\,k \mapsto e[S/\alpha]\} \\
&\qquad (\text{if } \{\boldsymbol{\beta}^J\} \cap (\{\alpha\} \cup \mathrm{FTV}(S)) = \emptyset)
\end{aligned}
$$

$$
\begin{aligned}
\alpha[S/\alpha] &= S \\
\beta[S/\alpha] &= \beta \quad (\text{if } \alpha \neq \beta) \\
(A \rightarrow_{\varepsilon} B)[S/\alpha] &= (A[S/\alpha]) \rightarrow_{\varepsilon[S/\alpha]} (B[S/\alpha]) \\
(\forall\beta : K.A^{\varepsilon})[S/\alpha] &= \forall\beta : K.A[S/\alpha]^{(\varepsilon[S/\alpha])} \\
&\qquad (\text{if } \alpha \neq \beta \text{ and } \beta \notin \mathrm{FTV}(S)) \\
(\mathcal{C}\,\boldsymbol{T}^I)[S/\alpha] &= \mathcal{C}\,\boldsymbol{T[S/\alpha]}^I
\end{aligned}
$$

$$
\begin{aligned}
\{\}[S/\alpha] &= \{\} \\
(\sigma \uplus \{\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K}^J.A \Rightarrow B\})[S/\alpha] &= \sigma[S/\alpha] \uplus \{\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K}^J.A[S/\alpha] \Rightarrow B[S/\alpha]\} \\
&\qquad (\text{if } \{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(S) = \emptyset)
\end{aligned}
$$

$$
\begin{aligned}
\emptyset[S/\alpha] &= \emptyset \\
(\Gamma, x : A)[S/\alpha] &= \Gamma[S/\alpha], x : A[S/\alpha]
\end{aligned}
$$

$$(\Gamma, \beta : K)[S/\alpha] = \Gamma[S/\alpha], \beta : K \quad (\text{if } \alpha \neq \beta)$$

Definition 1.11 (Typelike Extraction Function)**.** *A typelike context* $\Delta(\Gamma)$ *extracted from a typing context* Γ *is defined as follows:*

$$\Delta(\emptyset) = \emptyset \qquad \Delta(\Gamma, x : A) = \Delta(\Gamma) \qquad \Delta(\Gamma, \alpha : K) = \Delta(\Gamma), \alpha : K \ .$$

Definition 1.12 (Domains of Typing Contexts)**.** *The set* $\mathrm{dom}(\Gamma)$ *of variables and typelike variables bound by a typing context* Γ *is defined as follows:*

$$\mathrm{dom}(\emptyset) = \emptyset \qquad \mathrm{dom}(\Gamma, x : A) = \mathrm{dom}(\Gamma) \cup \{x\} \qquad \mathrm{dom}(\Gamma, \alpha : K) = \mathrm{dom}(\Gamma) \cup \{\alpha\} \ .$$

Definition 1.13 (Context Well-formedness and Kinding Rules)**.**
Contexts Well-formedness $\boxed{\vdash \Gamma}$

$$\frac{}{\vdash \emptyset}\ \textsc{C\_Empty} \qquad \frac{x \notin \mathrm{dom}(\Gamma) \quad \Gamma \vdash A : \mathbf{Typ}}{\vdash \Gamma, x : A}\ \textsc{C\_Var} \qquad \frac{\vdash \Gamma \quad \alpha \notin \mathrm{dom}(\Gamma)}{\vdash \Gamma, \alpha : K}\ \textsc{C\_TVar}$$

Kinding $\boxed{\Gamma \vdash S : K}$ $\boxed{\Gamma \vdash \boldsymbol{S}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}} \iff \forall i \in I. (\Gamma \vdash S_i : K_i)$

$$\frac{\vdash \Gamma \quad \alpha : K \in \Gamma}{\Gamma \vdash \alpha : K}\ \textsc{K\_Var} \qquad \frac{\vdash \Gamma \quad \mathcal{C} : \Pi \boldsymbol{K}^{\boldsymbol{I}} \to K \in \Sigma \quad \Gamma \vdash \boldsymbol{S}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}}{\Gamma \vdash \mathcal{C}\, \boldsymbol{S}^{\boldsymbol{I}} : K}\ \textsc{K\_Cons}$$

$$\frac{\Gamma \vdash A : \mathbf{Typ} \quad \Gamma \vdash \varepsilon : \mathbf{Eff} \quad \Gamma \vdash B : \mathbf{Typ}}{\Gamma \vdash A \to_\varepsilon B : \mathbf{Typ}}\ \textsc{K\_Fun} \qquad \frac{\Gamma, \alpha : K \vdash A : \mathbf{Typ} \quad \Gamma, \alpha : K \vdash \varepsilon : \mathbf{Eff}}{\Gamma \vdash \forall \alpha : K. A^\varepsilon : \mathbf{Typ}}\ \textsc{K\_Poly}$$

Definition 1.14 (Proper Effect Contexts)**.** *An effect context* Ξ *is proper if, for any* $l :: \forall \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}. \sigma \in \Xi$*, the following holds:*

- $l : \Pi \boldsymbol{K}^{\boldsymbol{I}} \to \mathbf{Lab} \in \Sigma_{\mathrm{lab}}$*;*
- *for any* $\boldsymbol{\alpha_0}^{I_0}$*,* $\boldsymbol{K_0}^{I_0}$*, and* σ_0*, if* $l :: \forall \boldsymbol{\alpha_0}^{I_0} : \boldsymbol{K_0}^{I_0}. \sigma_0 \in \Xi$*, then* $\boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}} = \boldsymbol{\alpha_0}^{I_0} : \boldsymbol{K_0}^{I_0}$ *and* $\sigma = \sigma_0$*; and*
- *for any* $\mathsf{op} : \forall \boldsymbol{\beta}^{\boldsymbol{J}} : \boldsymbol{K_0}^{\boldsymbol{J}}. A \Rightarrow B \in \sigma$*,*

$$\boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \boldsymbol{\beta}^{\boldsymbol{J}} : \boldsymbol{K_0}^{\boldsymbol{J}} \vdash A : \mathbf{Typ} \quad \text{and} \quad \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \boldsymbol{\beta}^{\boldsymbol{J}} : \boldsymbol{K_0}^{\boldsymbol{J}} \vdash B : \mathbf{Typ} \ .$$

Definition 1.15 (Well-Formedness-Preserving Functions)**.** *Given a signature* Σ*, a (possibly partial) function* $f \in K_i(\Sigma)^{i \in \{1,\dots,n\}} \rightharpoonup K(\Sigma)$ *preserves well-formedness if*

$$\forall \Gamma, S_1, \dots, S_n. \Gamma \vdash S_1 : K_1 \wedge \cdots \wedge \Gamma \vdash S_n : K_n \wedge f(S_1, \dots, S_n) \in K(\Sigma) \implies \Gamma \vdash f(S_1, \dots, S_n) : K \ .$$

Similarly, $f \in K(\Sigma)$ *preserves well-formedness if* $\Gamma \vdash f : K$ *for any* Γ*.*

Definition 1.16. *We write* $\alpha \mapsto T \vdash \boldsymbol{S} : K_0$ *for a quadruple* $\langle \alpha, T, \boldsymbol{S}, K_0 \rangle$ *such that* $\exists \Gamma_1, K, \Gamma_2. (\forall S_0 \in \boldsymbol{S}. \Gamma_1, \alpha : K, \Gamma_2 \vdash S_0 : K_0) \wedge \Gamma_1 \vdash T : K$*.*

Definition 1.17 (Effect algebras)**.** *Given a label signature* Σ_{lab}*, an effect algebra is a quintuple* $\langle \Sigma_{\mathrm{eff}}, \odot, \mathbb{0}, (-)^{\uparrow}, \sim \rangle$ *satisfying the following, where we let* $\Sigma = \Sigma_{\mathrm{lab}} \uplus \Sigma_{\mathrm{eff}}$*.*

- $\odot \in \mathbf{Eff}(\Sigma) \times \mathbf{Eff}(\Sigma) \rightharpoonup \mathbf{Eff}(\Sigma)$*,* $\mathbb{0} \in \mathbf{Eff}(\Sigma)$*, and* $(-)^{\uparrow} \in \mathbf{Lab}(\Sigma) \to \mathbf{Eff}(\Sigma)$ *preserve well-formedness. Furthermore,* $\sim$ *is an equivalence relation on* $\mathbf{Eff}(\Sigma)$ *and preserves well-formedness, that is,* $\forall \varepsilon_1, \varepsilon_2. \varepsilon_1 \sim \varepsilon_2 \implies (\forall \Gamma. \Gamma \vdash \varepsilon_1 : \mathbf{Eff} \iff \Gamma \vdash \varepsilon_2 : \mathbf{Eff})$.
- $\langle \mathbf{Eff}(\Sigma), \odot, \mathbb{0} \rangle$ *is a partial monoid under* $\sim$*, that is, the following holds:*
  - $\forall \varepsilon \in \mathbf{Eff}(\Sigma). \varepsilon \odot \mathbb{0} \sim \varepsilon \wedge \mathbb{0} \odot \varepsilon \sim \varepsilon$*; and*
  - $\forall \varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathbf{Eff}(\Sigma)$.
    $(\varepsilon_1 \odot \varepsilon_2) \odot \varepsilon_3 \in \mathbf{Eff}(\Sigma) \vee \varepsilon_1 \odot (\varepsilon_2 \odot \varepsilon_3) \in \mathbf{Eff}(\Sigma) \implies (\varepsilon_1 \odot \varepsilon_2) \odot \varepsilon_3 \sim \varepsilon_1 \odot (\varepsilon_2 \odot \varepsilon_3)$.
- *Typelike substitution respecting well-formedness is a homomorphism for* $\odot$*,* $(-)^{\uparrow}$*, and* $\sim$*, that is, the following holds:*
  - $\forall \alpha, S, \varepsilon_1, \varepsilon_2. \alpha \mapsto S \vdash \varepsilon_1, \varepsilon_2 : \mathbf{Eff} \wedge \varepsilon_1 \odot \varepsilon_2 \in \mathbf{Eff}(\Sigma) \implies (\varepsilon_1 \odot \varepsilon_2)[S/\alpha] = \varepsilon_1[S/\alpha] \odot \varepsilon_2[S/\alpha]$*;*

– $\forall \alpha, S, L.\, \alpha \mapsto S \vdash L : \mathbf{Lab} \implies (L)^{\uparrow}[S/\alpha] = (L[S/\alpha])^{\uparrow}$; *and*

– $\forall \alpha, S, \varepsilon_1, \varepsilon_2.\, \alpha \mapsto S \vdash \varepsilon_1, \varepsilon_2 : \mathbf{Eff} \wedge \varepsilon_1 \sim \varepsilon_2 \implies \varepsilon_1[S/\alpha] \sim \varepsilon[S/\alpha]$.

Remark 1.18. *For readability, we introduce the following abbreviations.*

- $\varepsilon_1 \oslash \varepsilon_2 \stackrel{\text{def}}{=} \exists \varepsilon.\, \varepsilon_1 \odot \varepsilon \sim \varepsilon_2$ *and*
- $\Gamma \vdash \varepsilon_1 \oslash \varepsilon_2 \stackrel{\text{def}}{=} \exists \varepsilon.\, \varepsilon_1 \odot \varepsilon \sim \varepsilon_2 \wedge (\forall \varepsilon' \in \{\varepsilon_1, \varepsilon_2, \varepsilon\}.\, \Gamma \vdash \varepsilon' : \mathbf{Eff})$.

Remark 1.19 (Parameters of λ_{EA})**.** λ_{EA} *takes a label signature in Definition 1.3, an effect algebra over that label signature in Definition 1.17, and an effect context as parameters.*

Example 1.20 (Effect Signature for Effect Sets)**.** The effect signature $\Sigma_{\text{eff}}^{\text{Set}}$ for effect sets consists of the pairs $\{\} : \mathbf{Eff}$, $\{-\} : \mathbf{Lab} \to \mathbf{Eff}$, and $- \cup - : \mathbf{Eff} \times \mathbf{Eff} \to \mathbf{Eff}$.

Example 1.21 (Effect Signature for Effect Multisets)**.** The effect signature $\Sigma_{\text{eff}}^{\text{MSet}}$ for effect multisets consists of the pairs $\{\} : \mathbf{Eff}$, $\{-\} : \mathbf{Lab} \to \mathbf{Eff}$, and $- \sqcup - : \mathbf{Eff} \times \mathbf{Eff} \to \mathbf{Eff}$.

Example 1.22 (Effect Signature for Rows)**.** The effect signature $\Sigma_{\text{eff}}^{\text{Row}}$ for both simple rows and scoped rows consists of the pairs $\langle\rangle : \mathbf{Eff}$ and $\langle - \mid - \rangle : \mathbf{Lab} \times \mathbf{Eff} \to \mathbf{Eff}$.

Example 1.23 (Effect Sets)**.** An effect algebra EA_{Set} for effect sets is defined by $\langle \Sigma_{\text{eff}}^{\text{Set}}, - \cup -, \{\}, \{-\}, \sim_{\text{Set}} \rangle$ where $\sim_{\text{Set}}$ is the least equivalence relation satisfying the following rules:

$$\frac{}{\varepsilon \cup \{\} \sim_{\text{Set}} \varepsilon} \qquad \frac{}{\varepsilon_1 \cup \varepsilon_2 \sim_{\text{Set}} \varepsilon_2 \cup \varepsilon_1} \qquad \frac{}{\varepsilon \cup \varepsilon \sim_{\text{Set}} \varepsilon} \qquad \frac{}{(\varepsilon_1 \cup \varepsilon_2) \cup \varepsilon_3 \sim_{\text{Set}} \varepsilon_1 \cup (\varepsilon_2 \cup \varepsilon_3)}$$

$$\frac{\varepsilon_1 \sim_{\text{Set}} \varepsilon_2 \quad \varepsilon_3 \sim_{\text{Set}} \varepsilon_4}{\varepsilon_1 \cup \varepsilon_3 \sim_{\text{Set}} \varepsilon_2 \cup \varepsilon_4}$$

Example 1.24 (Effect Multisets)**.** An effect algebra EA_{MSet} for effect multisets is defined by $\langle \Sigma_{\text{eff}}^{\text{MSet}}, - \sqcup -, \{\}, \{-\}, \sim_{\text{MSet}} \rangle$ where $\sim_{\text{MSet}}$ is the least equivalence relation satisfying the following rules:

$$\frac{}{\varepsilon \sqcup \{\} \sim_{\text{MSet}} \varepsilon} \qquad \frac{}{\varepsilon_1 \sqcup \varepsilon_2 \sim_{\text{MSet}} \varepsilon_2 \sqcup \varepsilon_1} \qquad \frac{}{(\varepsilon_1 \sqcup \varepsilon_2) \sqcup \varepsilon_3 \sim_{\text{MSet}} \varepsilon_1 \sqcup (\varepsilon_2 \sqcup \varepsilon_3)} \qquad \frac{\varepsilon_1 \sim_{\text{MSet}} \varepsilon_2 \quad \varepsilon_3 \sim_{\text{MSet}} \varepsilon_4}{\varepsilon_1 \sqcup \varepsilon_3 \sim_{\text{MSet}} \varepsilon_2 \sqcup \varepsilon_4}$$

Example 1.25 (Simple Rows)**.** An effect algebra EA_{SimpR} for simple rows is defined by $\langle \Sigma_{\text{eff}}^{\text{Row}}, \odot_{\text{SimpR}}, \langle\rangle, \langle - \mid \langle\rangle\rangle, \sim_{\text{SimpR}} \rangle$ where

$$\varepsilon_1 \odot_{\text{SimpR}} \varepsilon_2 \stackrel{\text{def}}{=} \begin{cases} \langle L_1 \mid \langle \cdots \langle L_n \mid \varepsilon_2 \rangle\rangle\rangle & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \langle\rangle\rangle\rangle\rangle) \\ \varepsilon_1 & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \rho \rangle\rangle\rangle \text{ and } \varepsilon_2 = \langle\rangle) \\ \mathit{undefined} & (\text{otherwise}) \end{cases}$$

and $\sim_{\text{SimpR}}$ is the least equivalence relation satisfying the following.

$$\frac{\varepsilon_1 \sim_{\text{SimpR}} \varepsilon_2}{\langle L \mid \varepsilon_1 \rangle \sim_{\text{SimpR}} \langle L \mid \varepsilon_2 \rangle} \qquad \frac{L_1 \neq L_2}{\langle L_1 \mid \langle L_2 \mid \varepsilon \rangle\rangle \sim_{\text{SimpR}} \langle L_2 \mid \langle L_1 \mid \varepsilon \rangle\rangle} \qquad \frac{}{\langle L \mid \varepsilon \rangle \sim_{\text{SimpR}} \langle L \mid \langle L \mid \varepsilon \rangle\rangle}$$

Example 1.26 (Scoped Rows)**.** An effect algebra EA_{ScpR} for scoped rows is defined by $\langle \Sigma_{\text{eff}}^{\text{Row}}, \odot_{\text{ScpR}}, \langle\rangle, \langle - \mid \langle\rangle\rangle, \sim_{\text{ScpR}} \rangle$ where

$$\varepsilon_1 \odot_{\text{ScpR}} \varepsilon_2 \stackrel{\text{def}}{=} \begin{cases} \langle L_1 \mid \langle \cdots \langle L_n \mid \varepsilon_2 \rangle\rangle\rangle & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \langle\rangle\rangle\rangle\rangle) \\ \varepsilon_1 & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \rho \rangle\rangle\rangle \text{ and } \varepsilon_2 = \langle\rangle) \\ \mathit{undefined} & (\text{otherwise}) \end{cases}$$

and $\sim_{\text{ScpR}}$ is the least equivalence relation satisfying the following.

$$\frac{\varepsilon_1 \sim_{\text{ScpR}} \varepsilon_2}{\langle L \mid \varepsilon_1 \rangle \sim_{\text{ScpR}} \langle L \mid \varepsilon_2 \rangle} \qquad \frac{L_1 \neq L_2}{\langle L_1 \mid \langle L_2 \mid \varepsilon \rangle\rangle \sim_{\text{ScpR}} \langle L_2 \mid \langle L_1 \mid \varepsilon \rangle\rangle}$$

Example 1.27 (Erasable Sets)**.** An effect algebra $\mathrm{EA}_{\mathrm{ESet}}$ for effect sets is defined by $\langle \Sigma_{\mathrm{eff}}^{\mathrm{Set}}, - \cup -, \{\}, \{-\}, \sim_{\mathrm{ESet}} \rangle$ where $\sim_{\mathrm{ESet}}$ is the least equivalence relation satisfying the following rules:

$$\frac{l_1 \neq l_2}{\{l_1\, \boldsymbol{S_1}^{I_1}\} \cup \{l_2\, \boldsymbol{S_2}^{I_2}\} \sim_{\mathrm{ESet}} \{l_2\, \boldsymbol{S_2}^{I_2}\} \cup \{l_1\, \boldsymbol{S_1}^{I_1}\}} \qquad \frac{}{\{l\, \boldsymbol{S_1}^{I_1}\} \cup \{l\, \boldsymbol{S_2}^{I_2}\} \sim_{\mathrm{ESet}} \{l\, \boldsymbol{S_1}^{I_1}\}} \qquad \frac{}{\varepsilon \cup \{\} \sim_{\mathrm{ESet}} \varepsilon}$$

$$\frac{}{\{\} \cup \varepsilon \sim_{\mathrm{ESet}} \varepsilon} \qquad \frac{}{(\varepsilon_1 \cup \varepsilon_2) \cup \varepsilon_3 \sim_{\mathrm{ESet}} \varepsilon_1 \cup (\varepsilon_2 \cup \varepsilon_3)} \qquad \frac{\varepsilon_1 \sim_{\mathrm{ESet}} \varepsilon_2 \quad \varepsilon_3 \sim_{\mathrm{ESet}} \varepsilon_4}{\varepsilon_1 \cup \varepsilon_3 \sim_{\mathrm{ESet}} \varepsilon_2 \cup \varepsilon_4}$$

Example 1.28 (Erasable Multisets)**.** An effect algebra $\mathrm{EA}_{\mathrm{EMSet}}$ for effect multisets is defined by $\langle \Sigma_{\mathrm{eff}}^{\mathrm{MSet}}, - \sqcup -, \{\}, \{-\}, \sim_{\mathrm{EMSet}} \rangle$ where $\sim_{\mathrm{EMSet}}$ is the least equivalence relation satisfying the following rules:

$$\frac{l_1 \neq l_2}{\{l_1\, \boldsymbol{S_1}^{I_1}\} \sqcup \{l_2\, \boldsymbol{S_2}^{I_2}\} \sim_{\mathrm{EMSet}} \{l_2\, \boldsymbol{S_2}^{I_2}\} \sqcup \{l_1\, \boldsymbol{S_1}^{I_1}\}} \qquad \frac{}{\varepsilon \sqcup \{\} \sim_{\mathrm{EMSet}} \varepsilon} \qquad \frac{}{\{\} \sqcup \varepsilon \sim_{\mathrm{EMSet}} \varepsilon}$$

$$\frac{}{(\varepsilon_1 \sqcup \varepsilon_2) \sqcup \varepsilon_3 \sim_{\mathrm{EMSet}} \varepsilon_1 \sqcup (\varepsilon_2 \sqcup \varepsilon_3)} \qquad \frac{\varepsilon_1 \sim_{\mathrm{EMSet}} \varepsilon_2 \quad \varepsilon_3 \sim_{\mathrm{EMSet}} \varepsilon_4}{\varepsilon_1 \sqcup \varepsilon_3 \sim_{\mathrm{EMSet}} \varepsilon_2 \sqcup \varepsilon_4}$$

Example 1.29 (Erasable Simple Rows)**.** An effect algebra $\mathrm{EA}_{\mathrm{ESimpR}}$ for erasable simple rows is defined by $\langle \Sigma_{\mathrm{eff}}^{\mathrm{Row}}, \odot_{\mathrm{ESimpR}}, \langle\rangle, \langle - \mid \langle\rangle\rangle, \sim_{\mathrm{ESimpR}} \rangle$ where

$$\varepsilon_1 \odot_{\mathrm{ESimpR}} \varepsilon_2 \stackrel{\mathrm{def}}{=} \begin{cases} \langle L_1 \mid \langle \cdots \langle L_n \mid \varepsilon_2 \rangle\rangle\rangle & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \langle\rangle\rangle\rangle\rangle) \\ \varepsilon_1 & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \rho \rangle\rangle\rangle \text{ and } \varepsilon_2 = \langle\rangle) \\ \textit{undefined} & (\text{otherwise}) \end{cases}$$

and $\sim_{\mathrm{ESimpR}}$ is the least equivalence relation satisfying the following.

$$\frac{\varepsilon_1 \sim_{\mathrm{SimpR}} \varepsilon_2}{\langle L \mid \varepsilon_1 \rangle \sim_{\mathrm{SimpR}} \langle L \mid \varepsilon_2 \rangle} \qquad \frac{l_1 \neq l_2}{\langle l_1\, \boldsymbol{S_1}^{I_1} \mid \langle l_2\, \boldsymbol{S_2}^{I_2} \mid \varepsilon \rangle\rangle \sim_{\mathrm{SimpR}} \langle l_2\, \boldsymbol{S_2}^{I_2} \mid \langle l_1\, \boldsymbol{S_1}^{I_1} \mid \varepsilon \rangle\rangle}$$

$$\frac{}{\langle l\, \boldsymbol{S_1}^{I_1} \mid \varepsilon \rangle \sim_{\mathrm{SimpR}} \langle l\, \boldsymbol{S_1}^{I_1} \mid \langle l\, \boldsymbol{S_2}^{I_2} \mid \varepsilon \rangle\rangle}$$

Example 1.30 (Erasable Scoped Rows)**.** An effect algebra $\mathrm{EA}_{\mathrm{EScpR}}$ for scoped rows is defined by $\langle \Sigma_{\mathrm{eff}}^{\mathrm{Row}}, \odot_{\mathrm{EScpR}}, \langle\rangle, \langle - \mid \langle\rangle\rangle, \sim_{\mathrm{EScpR}} \rangle$ where

$$\varepsilon_1 \odot_{\mathrm{EScpR}} \varepsilon_2 \stackrel{\mathrm{def}}{=} \begin{cases} \langle L_1 \mid \langle \cdots \langle L_n \mid \varepsilon_2 \rangle\rangle\rangle & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \langle\rangle\rangle\rangle\rangle) \\ \varepsilon_1 & (\text{if } \varepsilon_1 = \langle L_1 \mid \langle \cdots \langle L_n \mid \rho \rangle\rangle\rangle \text{ and } \varepsilon_2 = \langle\rangle) \\ \textit{undefined} & (\text{otherwise}) \end{cases}$$

and $\sim_{\mathrm{EScpR}}$ is the least equivalence relation satisfying the following.

$$\frac{\varepsilon_1 \sim_{\mathrm{EScpR}} \varepsilon_2}{\langle L \mid \varepsilon_1 \rangle \sim_{\mathrm{EScpR}} \langle L \mid \varepsilon_2 \rangle} \qquad \frac{l_1 \neq l_2}{\langle l_1\, \boldsymbol{S_1}^{I_1} \mid \langle l_2\, \boldsymbol{S_2}^{I_2} \mid \varepsilon \rangle\rangle \sim_{\mathrm{EScpR}} \langle l_2\, \boldsymbol{S_2}^{I_2} \mid \langle l_1\, \boldsymbol{S_1}^{I_1} \mid \varepsilon \rangle\rangle}$$

Definition 1.31 (Freeness of Labels)**.**
Freeness of labels $\boxed{n\text{−free}(L, E)}$

$$\frac{}{0\text{−free}(L, \square)} \qquad \frac{n\text{−free}(L, E)}{n\text{−free}(L, \mathbf{let}\ x = E\ \mathbf{in}\ e)} \qquad \frac{n+1\text{−free}(L, E)}{n\text{−free}(L, \mathbf{handle}_L\ E\ \mathbf{with}\ h)} \qquad \frac{n\text{−free}(L, E) \quad L \neq L'}{n\text{−free}(L, \mathbf{handle}_{L'}\ E\ \mathbf{with}\ h)}$$

Definition 1.32 (Operational Semantics)**.**
Reduction $\boxed{e \longmapsto e'}$

$$\frac{}{\mathbf{fun}\,(f, x, e)\ v \longmapsto e[\mathbf{fun}\,(f, x, e)/f][v/x]}\ \text{R\_App} \qquad \frac{}{(\Lambda\alpha : K.e)\ S \longmapsto e[S/\alpha]}\ \text{R\_TApp}$$

$$\frac{}{\mathbf{let}\ x = v\ \mathbf{in}\ e \longmapsto e[v/x]}\ \text{R\_Let} \qquad \frac{\mathbf{return}\ x \mapsto e_r \in h}{\mathbf{handle}_{l\, \boldsymbol{S}^I}\ v\ \mathbf{with}\ h \longmapsto e_r[v/x]}\ \text{R\_Handle1}$$

$$\frac{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\ p\ k \mapsto e \in h \quad v_{cont} = \lambda z.\mathbf{handle}_{l\, \boldsymbol{S}^I}\ E[z]\ \mathbf{with}\ h \quad 0\text{−free}(l\, \boldsymbol{S}^I, E)}{\mathbf{handle}_{l\, \boldsymbol{S}^I}\ E[\mathsf{op}_{l\, \boldsymbol{S}^I}\ \boldsymbol{T}^J\ v]\ \mathbf{with}\ h \longmapsto e[\boldsymbol{T}^J/\boldsymbol{\beta}^J][v/p][v_{cont}/k]}\ \text{R\_Handle2}$$

Evaluation $\boxed{e \longrightarrow e'}$

$$\frac{e_1 \longmapsto e_2}{E[e_1] \longrightarrow E[e_2]}\ \text{E\_EVAL}$$

Definition 1.33. *We write* $\longrightarrow^*$ *for the reflexive, transitive closure of* $\longrightarrow$*. We also write* $e \not\longrightarrow$ *to denote that there is no* e' *such that* $e \longrightarrow e'$*.*

Definition 1.34 (Typing and Subtyping Rules)**.**

Typing $\boxed{\Gamma \vdash e : A \mid \varepsilon}$

$$\frac{\vdash \Gamma \quad x : A \in \Gamma}{\Gamma \vdash x : A \mid \mathbb{0}}\ \text{T\_VAR} \qquad \frac{\Gamma, f : A \to_\varepsilon B, x : A \vdash e : B \mid \varepsilon}{\Gamma \vdash \mathbf{fun}\,(f, x, e) : A \to_\varepsilon B \mid \mathbb{0}}\ \text{T\_ABS}$$

$$\frac{\Gamma \vdash v_1 : A \to_\varepsilon B \mid \mathbb{0} \quad \Gamma \vdash v_2 : A \mid \mathbb{0}}{\Gamma \vdash v_1\, v_2 : B \mid \varepsilon}\ \text{T\_APP} \qquad \frac{\Gamma, \alpha : K \vdash e : A \mid \varepsilon}{\Gamma \vdash \Lambda\alpha : K.e : \forall\alpha : K.A^\varepsilon \mid \mathbb{0}}\ \text{T\_TABS}$$

$$\frac{\Gamma \vdash v : \forall\alpha : K.A^\varepsilon \mid \mathbb{0} \quad \Gamma \vdash S : K}{\Gamma \vdash v\, S : A[S/\alpha] \mid \varepsilon[S/\alpha]}\ \text{T\_TAPP} \qquad \frac{\Gamma \vdash e_1 : A \mid \varepsilon \quad \Gamma, x : A \vdash e_2 : B \mid \varepsilon}{\Gamma \vdash \mathbf{let}\, x = e_1\, \mathbf{in}\, e_2 : B \mid \varepsilon}\ \text{T\_LET}$$

$$\frac{\Gamma \vdash e : A \mid \varepsilon \quad \Gamma \vdash A \mid \varepsilon <: A' \mid \varepsilon'}{\Gamma \vdash e : A' \mid \varepsilon'}\ \text{T\_SUB} \qquad \frac{\begin{array}{c} l :: \forall\boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi \quad \mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \\ \vdash \Gamma \quad \Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I \quad \Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J \end{array}}{\Gamma \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J : (A[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \to_{(l\,\boldsymbol{S}^I)^\uparrow} (B[\boldsymbol{T}^I/\boldsymbol{\beta}^I]) \mid \mathbb{0}}\ \text{T\_OP}$$

$$\frac{\begin{array}{c} \Gamma \vdash e : A \mid \varepsilon' \qquad l :: \forall\boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi \qquad \Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I \\ \Gamma \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h : A \Rightarrow^\varepsilon B \quad (l\,\boldsymbol{S}^I)^\uparrow \odot \varepsilon \sim \varepsilon' \end{array}}{\Gamma \vdash \mathbf{handle}_{l\,\boldsymbol{S}^I}\, e\, \mathbf{with}\, h : B \mid \varepsilon}\ \text{T\_HANDLING}$$

Handler Typing $\boxed{\Gamma \vdash_\sigma h : A \Rightarrow^\varepsilon B}$

$$\frac{\Gamma, x : A \vdash e_r : B \mid \varepsilon}{\Gamma \vdash_{\{\}} \{\mathbf{return}\, x \mapsto e_r\} : A \Rightarrow^\varepsilon B}\ \text{H\_RETURN}$$

$$\frac{\begin{array}{c} \sigma = \sigma' \uplus \{\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B'\} \\ \Gamma \vdash_{\sigma'} h : A \Rightarrow^\varepsilon B \quad \Gamma, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \to_\varepsilon B \vdash e : B \mid \varepsilon \end{array}}{\Gamma \vdash_\sigma h \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\} : A \Rightarrow^\varepsilon B}\ \text{H\_OP}$$

Subtyping $\boxed{\Gamma \vdash A <: B}$

$$\frac{\Gamma \vdash A : \mathbf{Typ}}{\Gamma \vdash A <: A}\ \text{ST\_REFL} \qquad \frac{\Gamma \vdash A_2 <: A_1 \quad \Gamma \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2}{\Gamma \vdash A_1 \to_{\varepsilon_1} B_1 <: A_2 \to_{\varepsilon_2} B_2}\ \text{ST\_FUN}$$

$$\frac{\Gamma, \alpha : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2}{\Gamma \vdash \forall\alpha : K.{A_1}^{\varepsilon_1} <: \forall\alpha : K.{A_2}^{\varepsilon_2}}\ \text{ST\_POLY} \qquad \frac{\Gamma \vdash A_1 <: B \quad \Gamma \vdash \varepsilon_1 \otimes \varepsilon_2}{\Gamma \vdash A \mid \varepsilon_1 <: B \mid \varepsilon_2}\ \text{ST\_COMP}$$

Definition 1.35 (Semantics of Shallow Handlers)**.** *The semantics for shallow handlers consists of the reduction and evaluation relations defined by the following rule* R_SHANDLE *and those in Definition 1.32 except for* R_HANDLE2*.*

$$\frac{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e \in h \quad v_{cont} = \lambda z.E[z] \quad 0\text{–free}(l\,\boldsymbol{S}^I, E)}{\mathbf{handle}_{l\,\boldsymbol{S}^I}\, E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v]\, \mathbf{with}\, h \longmapsto e[\boldsymbol{T}^J/\boldsymbol{\beta}^J][v/p][v_{cont}/k]}\ \text{R\_SHANDLE}$$

Definition 1.36 (Typing of Shallow Handlers)**.** *The typing rules of shallow handlers consist of the rules defined by the following rules* T_SHANDLING, SH_RETURN, *and* SH_OP, *and those in Definition 1.34 except for* T_HANDLING, H_RETURN, *and* H_OP*.*

Typing $\boxed{\Gamma \vdash e : A \mid \varepsilon}$

$$\frac{\begin{array}{c} \Gamma \vdash e : A \mid \varepsilon' \qquad l :: \forall\boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi \qquad \Gamma \vdash \boldsymbol{S}^N : \boldsymbol{K}^N \\ \Gamma \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A^{\varepsilon'} \Rightarrow^\varepsilon B \quad (l\,\boldsymbol{S}^N)^\uparrow \odot \varepsilon \sim \varepsilon' \end{array}}{\Gamma \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e\, \mathbf{with}\, h : B \mid \varepsilon}\ \text{T\_SHANDLING}$$

Shallow Handler Typing $\boxed{\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^{\varepsilon} B}$

$$\frac{\Gamma, x : A \vdash e_r : B \mid \varepsilon \quad \Gamma \vdash \varepsilon' : \mathbf{Eff}}{\Gamma \vdash_{\{\}} \{\mathbf{return}\, x \mapsto e_r\} : A^{\varepsilon'} \Rightarrow^{\varepsilon} B} \text{ SH\_RETURN}$$

$$\frac{\begin{array}{c}\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J . A' \Rightarrow B'\} \\ \Gamma \vdash_{\sigma'} h : A^{\varepsilon'} \Rightarrow^{\varepsilon} B \quad \Gamma, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} A \vdash e : B \mid \varepsilon\end{array}}{\Gamma \vdash_\sigma h \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\} : A^{\varepsilon'} \Rightarrow^{\varepsilon} B} \text{ SH\_OP}$$

Definition 1.37 (The Syntax of λ_{EA} with Lift Coercions)**.** *The syntax of λ_{EA} extended by lift coercions is the same as Definition 1.5 except for the following.*

$$e \;::=\; \cdots \mid [e]_L \quad \text{(expressions)} \qquad E \;::=\; \cdots \mid [E]_L \quad \text{(evaluation contexts)}$$

Definition 1.38 (Freeness of Labels with Lift Coercions)**.** *The rules of freeness of labels for λ_{EA} extended by lift coercions consist of the rules in Definition 1.31 and the following rules.*

Freeness of labels $\boxed{n\text{−free}(L, E)}$

$$\frac{n\text{−free}(L, E)}{n+1\text{−free}(L, [E]_L)} \qquad \frac{n\text{−free}(L, E) \quad L \neq L'}{n\text{−free}(L, [E]_{L'})}$$

Definition 1.39 (Semantics with Lift Coercions)**.** *The semantics for λ_{EA} extended by lift coercions consists of the reduction and evaluation relations defined by the following rule* R_LIFT *and those in Definition 1.32 except for* R_HANDLE2*.*

Reduction $\boxed{e \longmapsto e'}$

$$\frac{}{[v]_L \longmapsto v} \text{ R\_LIFT}$$

Definition 1.40 (Typing of Lift Coercions)**.** *The typing rules of λ_{EA} extended by lift coercions consist of the rules in Definition 1.34 and the following rule.*

$$\frac{\Gamma \vdash e : A \mid \varepsilon' \quad \Gamma \vdash L : \mathbf{Lab} \quad (L)^{\uparrow} \odot \varepsilon' \sim \varepsilon}{\Gamma \vdash [e]_L : A \mid \varepsilon} \text{ T\_LIFT}$$

Definition 1.41 (Freeness of Label Names)**.**

Freeness of label names $\boxed{n\text{−free}(L, E)}$

$$\frac{}{0\text{−free}(l, \square)} \qquad \frac{n\text{−free}(l, E)}{n\text{−free}(l, \mathbf{let}\, x = E\, \mathbf{in}\, e)} \qquad \frac{n+1\text{−free}(l, E)}{n\text{−free}(l, \mathbf{handle}_{l\, \boldsymbol{S}^I}\, E\, \mathbf{with}\, h)}$$

$$\frac{n\text{−free}(l, E) \quad l \neq l'}{n\text{−free}(l, \mathbf{handle}_{l'\, \boldsymbol{S}^I}\, E\, \mathbf{with}\, h)}$$

Definition 1.42 (Operational Semantics with Type-Erasure)**.** *The type-erasure semantics consists of the reduction and evaluation relations defined by the following rule* R_HANDLE2' *and those in Definition 1.32 except for* R_HANDLE2*.*

$$\frac{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e \in h \quad v_{cont} = \lambda z.\mathbf{handle}_{l\, \boldsymbol{S}^I}\, E[z]\, \mathbf{with}\, h \quad 0\text{−free}(l, E)}{\mathbf{handle}_{l\, \boldsymbol{S}^I}\, E[\mathsf{op}_{l\, \boldsymbol{S}'^I}\, \boldsymbol{T}^J\, v]\, \mathbf{with}\, h \longmapsto e[\boldsymbol{T}^J / \boldsymbol{\beta}^J][v/p][v_{cont}/k]} \text{ R\_HANDLE2'}$$

Definition 1.43 (Freeness of Label Names with Lift Coercions)**.**
The rules of freeness of label names for λ_{EA} extended by lift coercions consist of the rules in Definition 1.41 and the following rules.

Freeness of label names $\boxed{n\text{−free}(l, E)}$

$$\frac{n\text{−free}(l, E)}{n+1\text{−free}(l, [E]_{l\, \boldsymbol{S}^I})} \qquad \frac{n\text{−free}(l, E) \quad L \neq l\, \boldsymbol{S}^I}{n\text{−free}(l, [E]_L)}$$

Definition 1.44 (Semantics with Lift Coercions and Type-Erasure). *The semantics for lift coercions consists of the reduction and evaluation relations defined by the rule* R_HANDLE2' *defined in Definition 1.42 and those in Definition 1.39 except for* R_HANDLE2.

Definition 1.45 (Safety Conditions).

(1) For any L, $(L)^{\uparrow} \otimes \mathbb{0}$ does not hold.

(2) If $(L)^{\uparrow} \otimes \varepsilon$ and $(L')^{\uparrow} \odot \varepsilon' \sim \varepsilon$ and $L \neq L'$, then $(L)^{\uparrow} \otimes \varepsilon'$.

Definition 1.46 (Safety Condition for Lift Coercions). *The safety condition added for lift coercions is the following:*

(3) If $(L)^{\uparrow} \odot \varepsilon_1 \sim (L_1)^{\uparrow} \odot \cdots \odot (L_n)^{\uparrow} \odot (L)^{\uparrow} \odot \varepsilon_2$ and $L \notin \{L_1, \ldots, L_n\}$, then $\varepsilon_1 \sim (L_1)^{\uparrow} \odot \cdots \odot (L_n)^{\uparrow} \odot \varepsilon_2$.

Definition 1.47 (Safety Condition for Type-Erasure). *The safety condition added for the type-erasure semantics is the following:*

(4) If $(l\, \boldsymbol{S_1}^{I_1})^{\uparrow} \otimes \varepsilon$ and $(l\, \boldsymbol{S_2}^{I_2})^{\uparrow} \otimes \varepsilon$, then $\boldsymbol{S_1}^{I_1} = \boldsymbol{S_2}^{I_2}$.

Example 1.48 (Unsafe Effect Algebras).

Effect algebra violating safety condition (1) Consider an effect algebra such that $\emptyset \vdash (l)^{\uparrow} \otimes \mathbb{0}$ holds for some l. Clearly, this effect algebra violates safety condition (1). In this case, $\emptyset \vdash \mathsf{op}_l\, v : A \mid \mathbb{0}$ can be derived for some A (if $\mathsf{op}_l\, v$ is well typed) because $\mathsf{op}_l\, v$ is given the effect $(l)^{\uparrow}$ and the subeffecting $\emptyset \vdash (l)^{\uparrow} \otimes \mathbb{0}$ holds. However, the operation call is not handled.

Effect algebra violating safety condition (2) Consider an effect algebra such that safety condition (1), $(l)^{\uparrow} \otimes (l')^{\uparrow}$, and $(l')^{\uparrow} \odot \mathbb{0} \sim (l')^{\uparrow}$ hold for some l and l' such that $l \neq l'$. This effect algebra violates safety condition (2): if safety condition (2) is met, we would have $(l)^{\uparrow} \otimes \mathbb{0}$, but it is contradictory with safety condition (1).

This effect algebra allows assigning the empty effect $\mathbb{0}$ to the expression $\mathbf{handle}_{l'}\, \mathsf{op}_l\ v\, \mathbf{with}\, h$ as illustrated by the following typing derivation:

$$\dfrac{\cdots \quad (l')^{\uparrow} \odot \mathbb{0} \sim (l')^{\uparrow} \quad \dfrac{\emptyset \vdash \mathsf{op}_l\ v : A \mid (l)^{\uparrow} \quad \emptyset \vdash A \mid (l)^{\uparrow} <: A \mid (l')^{\uparrow}}{\emptyset \vdash \mathsf{op}_l\ v : A \mid (l')^{\uparrow}}\ \text{T\_SUB}}{\emptyset \vdash \mathbf{handle}_{l'}\, \mathsf{op}_l\ v\, \mathbf{with}\, h : B \mid \mathbb{0}}\ \text{T\_HANDLING}$$

However, the operation call in it is not handled.

$$\exists\alpha : \textbf{Typ}.\exists\rho : \textbf{Eff}.\{$$
$$\mathit{empty} : \alpha,\quad \mathit{add} : \mathsf{Int} \rightarrow_{\{\}} \alpha \rightarrow_{\{\}} \alpha,\quad \mathit{size} : \alpha \rightarrow_{\{\}} \mathsf{Int},\quad \mathit{find} : \mathsf{Int} \rightarrow_{\{\}} \alpha \rightarrow_{\{\}} \mathsf{Bool},$$
$$\mathit{filter} : (\mathsf{Int} \rightarrow_{\{\}} \mathsf{Bool}) \rightarrow_{\{\}} \alpha \rightarrow_{\{\}} \alpha,\quad \mathit{choose} : \alpha \rightarrow_{\rho} \mathsf{Int},$$
$$\mathit{accumulate} : \forall\beta : \textbf{Typ}.\forall\rho' : \textbf{Eff}.(\mathsf{Unit} \rightarrow_{\rho \cup \rho'} \beta) \rightarrow_{\rho'} \beta\,\mathsf{List}$$
$$\}$$

Figure 1: Module Interface IntSet

$$\textbf{pack}(\mathsf{Int\,List}, \{\mathsf{Selection\,Int}\}, \{\cdots$$
$$\mathit{choose} = \mathsf{select}_{\mathsf{Selection\,Int}}$$
$$\mathit{accumulate} = \Lambda\beta : \textbf{Typ}.\Lambda\rho' : \textbf{Eff}.\lambda f : \mathsf{Unit} \rightarrow_{\{\mathsf{Selection\,Int}\} \cup \rho'} \beta.$$
$$\textbf{handle}_{\mathsf{Selection\,Int}}\, f\,()\,\textbf{with}\,\{\,\textbf{return}\,x \mapsto [x]\} \uplus \{\mathsf{select}\; xs\; k \mapsto \mathtt{concat}\,(\mathtt{map}\,k\,xs)\}$$
$$\})$$

$$\textbf{pack}(\mathsf{Int\,List}, \{\mathsf{Fail}\} \cup \{\mathsf{Choice}\}, \{\cdots$$
$$\mathit{choose} = \textbf{fun}(\mathit{aux}, \mathit{xs}, \textbf{match}\; \mathit{xs}\; \textbf{with}$$
$$\mid [\,] \rightarrow \mathsf{fail}_{\mathsf{Fail}}\,\mathsf{Int}\,()$$
$$\mid y :: \mathit{ys} \rightarrow \textbf{if}\;\mathsf{decide}_{\mathsf{Choice}}\;()\;\textbf{then}\;y\;\textbf{else}\;\mathit{aux}\;\mathit{ys}),$$
$$\mathit{accumulate} = \Lambda\beta : \textbf{Typ}.\Lambda\rho' : \textbf{Eff}.\lambda f : \mathsf{Unit} \rightarrow_{\{\mathsf{Fail}\} \cup \{\mathsf{Choice}\} \cup \rho'} \beta.$$
$$\textbf{handle}_{\mathsf{Choice}}$$
$$\textbf{handle}_{\mathsf{Fail}}$$
$$f\,()$$
$$\textbf{with}\,\{\,\textbf{return}\,x \mapsto [x]\} \uplus \{\mathsf{fail}\,\alpha : \textbf{Typ}\,\_\,\_ \mapsto [\,]\}$$
$$\textbf{with}\,\{\,\textbf{return}\,x \mapsto x\} \uplus \{\mathsf{decide}\;\_\,k \mapsto k\,\mathsf{true} \mathbin{@} k\,\mathsf{false}\}$$
$$\})$$

Figure 2: Two implementation of IntSet

2 Example

We present a motivating example of allowing multiple effect variables in one effect collection. In this example, we use $\mathrm{EA}_{\mathsf{Set}}$ and offer two modules of type IntSet, which is an interface of implementations for integer sets defined in Figure 1.

We show two implementations of IntSet in Figure 2. The former implementation assumes the effect context $\mathsf{Selection} :: \forall\alpha : \textbf{Typ}.\{\mathsf{select} : \alpha\,\mathsf{List} \Rightarrow \alpha\}$, and concretizes ρ by Selection Int. The latter implementation assumes the effect context $\mathsf{Choice} :: \{\mathsf{decide} : \mathsf{Unit} \Rightarrow \mathsf{Bool}\}, \mathsf{Fail} :: \{\mathsf{fail} : \forall\alpha : \textbf{Typ}.\mathsf{Unit} \Rightarrow \alpha\}$, and concretizes ρ by $\{\mathsf{Fail}\} \cup \{\mathsf{Choice}\}$. Because the concrete effect of the latter implementation consists of two labels, it needs to be abstracted by a row variable, not by a label variable.

We define the function *search_path* using this package as follows.

$$\mathit{search\_path} = \lambda \mathit{sets} : \mathsf{IntSet\,List}.\lambda s : \mathsf{Int}.\lambda t : \mathsf{Int}.$$
$$\textbf{fun}(\mathit{aux}, p, \lambda \mathit{path} : \mathsf{Int\,List}.$$
$$\textbf{if}\; p = t\;\textbf{then}\;\mathit{path}$$
$$\textbf{else}\;\textbf{let}\; x = \mathit{choose}\,(\mathit{filter}\,(\lambda y : \mathsf{Int}.\mathit{not}\,(\mathtt{exists}\,(\lambda z.z = y)\,\mathit{path}))\,(\mathit{nth}\,p\,\mathit{sets}))$$
$$\textbf{in}\;\mathit{aux}\,x\,(x :: \mathit{path}))$$
$$s\,[s]$$

We show the example program using *search_path* as follows.

$$graph = [add\,1\,(add\,2\,empty);\ add\,0\,(add\,2\,(add\,3\,empty));\ add\,0\,(add\,1\,(add\,4\,empty));\ add\,1\,(add\,4\,(add\,5\,empty));\ add\,2\,(add\,3\,(add\,6\,empty));\ add\,3\,empty;\ add\,4\,empty]$$

$clean\,(\lambda\_ : \mathsf{Unit}.search\_path\ graph\,0\,5)$

$clean\,(\lambda\_ : \mathsf{Unit}.search\_path\ graph\,0\,5)$

The evaluation results are as follows.

[[5; 3; 4; 2; 1; 0]; [5; 3; 1; 0]; [5; 3; 4; 2; 0]; [5; 3; 1; 2; 0]]
[[6; 4; 2; 0; 1]; [6; 4; 2; 1]; [6; 4; 3; 1]]

3 Properties

3.1 Properties with Deep Handlers

This section assumes that the safety conditions in Definition 1.45 hold.

Lemma 3.1 (Well-formedness of context in judgement)**.** *If* $\Gamma \vdash S : K$*, then* $\vdash \Gamma$*.*

Proof. By induction on a derivation of $\Gamma \vdash S : K$. We proceed by case analysis on the kinding rule applied lastly to this derivation.
Case K_Var**:** Clearly.

Case K_Fun**:** $S = A \rightarrow_\varepsilon B$, $\Gamma \vdash A : \textbf{Typ}$, $\Gamma \vdash \varepsilon : \textbf{Eff}$, and $\Gamma \vdash B : \textbf{Typ}$ are given. By the induction hypothesis, we have $\vdash \Gamma$.

Case K_Poly**:** $S = \forall \alpha : K.A^\varepsilon$, $\Gamma, \alpha : K \vdash A : \textbf{Typ}$, and $\Gamma, \alpha : K \vdash \varepsilon : \textbf{Eff}$ are given. By the induction hypothesis, we have $\vdash \Gamma, \alpha : K$. Since only C_TVar can derive $\vdash \Gamma, \alpha : K$, the required result $\vdash \Gamma$ is achieved.

Case K_Cons**:** Clearly.

■

Lemma 3.2.

(1) If $\vdash \Gamma$*, then* $\vdash \Delta(\Gamma)$*.*

(2) If $\Gamma \vdash S : K$*, then* $\Delta(\Gamma) \vdash S : K$*.*

Proof.
By mutual induction on the derivations. We proceed by case analysis on the rule applied lastly to the derivation.
Case C_Empty**:** Clearly because of $\Delta(\emptyset) = \emptyset$.

Case C_Var**:** For some Γ', x, and A, the following are given:

- $\Gamma = \Gamma', x : A$ and
- $\Gamma' \vdash A : \textbf{Typ}$.

By the induction hypothesis, we have $\Delta(\Gamma') \vdash A : \textbf{Typ}$. By Lemma 3.1, we have $\vdash \Delta(\Gamma')$. Thus, we get $\vdash \Delta(\Gamma)$ as required because of $\Delta(\Gamma', x : A) = \Delta(\Gamma')$.

Case C_TVar**:** For some Γ', α, and K, the following are given:

- $\Gamma = \Gamma', \alpha : K$,
- $\vdash \Gamma'$, and
- $\alpha \notin \mathrm{dom}(\Gamma')$.

By the induction hypothesis, we have $\vdash \Delta(\Gamma')$. By $\alpha \notin \mathrm{dom}(\Gamma')$, we have $\alpha \notin \mathrm{dom}(\Delta(\Gamma'))$ because $\mathrm{dom}(\Delta(\Gamma')) \subseteq \mathrm{dom}(\Gamma')$. Thus, C_TVar derives $\vdash \Delta(\Gamma'), \alpha : K$ as required.

Case K_Var**:** $\vdash \Gamma$, $\alpha : K \in \Gamma$, and $S = \alpha$ are given for some α. By the induction hypothesis, we have $\vdash \Delta(\Gamma)$. By Definition 1.11, we have $\alpha : K \in \Delta(\Gamma)$. Thus, K_Var derives $\Delta(\Gamma) \vdash \alpha : K$.

Case K_Cons**:** For some $\mathcal{C}$, $\boldsymbol{S}^I$, and $\boldsymbol{K}^I$, the following are given:

- $S = \mathcal{C}\, \boldsymbol{S}^I$,
- $\vdash \Gamma$,
- $\mathcal{C} : \Pi \boldsymbol{K}^I \rightarrow K \in \Sigma$, and
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$.

By the induction hypothesis, we have $\vdash \Delta(\Gamma)$ and $\Delta(\Gamma) \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$. Thus, K_Cons derives $\Delta(\Gamma) \vdash \mathcal{C}\, \boldsymbol{S}^I : K$ as required.

Case K_Fun**:** For some A, ε, and B, the following are given:

- $S = A \rightarrow_\varepsilon B$,
- $K = \textbf{Typ}$,

- $\Gamma \vdash A : \mathbf{Typ}$,
- $\Gamma \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma \vdash B : \mathbf{Typ}$.

By the induction hypothesis, we have

- $\Delta(\Gamma) \vdash A : \mathbf{Typ}$,
- $\Delta(\Gamma) \vdash \varepsilon : \mathbf{Eff}$, and
- $\Delta(\Gamma) \vdash B : \mathbf{Typ}$.

Thus, K_Fun derives $\Delta(\Gamma) \vdash A \rightarrow_{\varepsilon} B : \mathbf{Typ}$.

Case K_Poly**:** For some α, K', A, and ε, the following are given:

- $S = \forall \alpha : K'.A^{\varepsilon}$,
- $K = \mathbf{Typ}$,
- $\Gamma, \alpha : K' \vdash A : \mathbf{Typ}$, and
- $\Gamma, \alpha : K' \vdash \varepsilon : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Delta(\Gamma, \alpha : K') \vdash A : \mathbf{Typ}$ and
- $\Delta(\Gamma, \alpha : K') \vdash \varepsilon : \mathbf{Eff}$.

By Definition 1.11, we have $\Delta(\Gamma, \alpha : K') = \Delta(\Gamma), \alpha : K'$. Thus, K_Poly derives $\Delta(\Gamma) \vdash \forall \alpha : K'.A^{\varepsilon} : \mathbf{Typ}$ as required.

■

Lemma 3.3.

(1) For any Γ and ε, if $\Gamma \vdash \varepsilon : \mathbf{Eff}$, then $\Gamma \vdash \varepsilon \otimes \varepsilon$ holds.

(2) For any Γ, ε_1, ε_2, and ε_3, if $\Gamma \vdash \varepsilon_1 \otimes \varepsilon_2$ and $\Gamma \vdash \varepsilon_2 \otimes \varepsilon_3$, then $\Gamma \vdash \varepsilon_1 \otimes \varepsilon_3$.

Proof.

(1) Clearly because of Lemma 3.2(2) and because $\mathbb{0}$ is a unit element.

(2) Clearly because $\odot$ is associative and preserves well-formendness.

■

Lemma 3.4 (Transitivity of Subtyping)**.**

(1) If $\Gamma \vdash A_1 <: A_2$ and $\Gamma \vdash A_2 <: A_3$, then $\Gamma \vdash A_1 <: A_3$.

(2) If $\Gamma \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$ and $\Gamma \vdash A_2 \mid \varepsilon_2 <: A_3 \mid \varepsilon_3$, then $\Gamma \vdash A_1 \mid \varepsilon_1 <: A_3 \mid \varepsilon_3$.

Proof. By the structural induction on the summation of the sizes of A_1, A_2, and A_3. If either $\Gamma \vdash A_1 <: A_2$ or $\Gamma \vdash A_2 <: A_3$ is derived by ST_Refl, then we have $\Gamma \vdash A_1 <: A_3$ immediately. Thus, we suppose that neither $\Gamma \vdash A_1 <: A_2$ nor $\Gamma \vdash A_2 <: A_3$ is derived by ST_Refl in the following. We proceed by case analysis on what form A_1 has.

Case $A_1 = \tau$**:** No rules other than ST_Refl can derive $\Gamma \vdash A_1 <: A_2$.

Case $A_1 = B_1 \rightarrow_{\varepsilon_1} C_1$**:** Since only ST_Fun can derive $\Gamma \vdash B_1 \rightarrow_{\varepsilon_1} C_1 <: A_2$, we have $A_2 = B_2 \rightarrow_{\varepsilon_2} C_2$ for some B_2, ε_2, and C_2 such that

- $\Gamma \vdash B_2 <: B_1$ and
- $\Gamma \vdash C_1 \mid \varepsilon_1 <: C_2 \mid \varepsilon_2$.

Since only ST_Fun can derive $\Gamma \vdash B_2 \rightarrow_{\varepsilon_2} C_2 <: A_3$, we have $A_3 = B_3 \rightarrow_{\varepsilon_3} C_3$ for some B_3, ε_3, and C_3 such that

- $\Gamma \vdash B_3 <: B_2$ and
- $\Gamma \vdash C_2 \mid \varepsilon_2 <: C_3 \mid \varepsilon_3$.

Since only ST_Comp can derive $\Gamma \vdash C_2 \mid \varepsilon_2 <: C_3 \mid \varepsilon_3$ and $\Gamma \vdash C_1 \mid \varepsilon_1 <: C_2 \mid \varepsilon_2$, we have

- $\Gamma \vdash C_1 <: C_2$,

- $\Gamma \vdash C_2 <: C_3$,
- $\Gamma \vdash \varepsilon_1 \otimes \varepsilon_2$, and
- $\Gamma \vdash \varepsilon_2 \otimes \varepsilon_3$.

By the induction hypothesis and Lemma 3.3(2), we have

- $\Gamma \vdash B_3 <: B_1$,
- $\Gamma \vdash \varepsilon_1 \otimes \varepsilon_3$, and
- $\Gamma \vdash C_1 <: C_3$.

Thus, we have $\Gamma \vdash A_1 <: A_3$ by ST_Fun as required.

Case $A_1 = \forall\alpha : K.{B_1}^{\varepsilon_1}$**:** Since only ST_Poly can derive $\Gamma \vdash \forall\alpha : K.{B_1}^{\varepsilon_1} <: A_2$, we have $A_2 = \forall\alpha : K.{B_2}^{\varepsilon_2}$ for some B_2 and ε_2 such that $\Gamma, \alpha : K \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$. Since only ST_Poly can derive $\Gamma \vdash \forall\alpha : K.{B_2}^{\varepsilon_2} <: A_3$, we have $A_3 = \forall\alpha : K.{B_3}^{\varepsilon_3}$ for some B_3 and ε_3 such that $\Gamma, \alpha : K \vdash B_2 \mid \varepsilon_2 <: B_3 \mid \varepsilon_3$. Since only ST_Comp can derive $\Gamma, \alpha : K \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$ and $\Gamma, \alpha : K \vdash B_2 \mid \varepsilon_2 <: B_3 \mid \varepsilon_3$, we have

- $\Gamma, \alpha : K \vdash B_1 <: B_2$,
- $\Gamma, \alpha : K \vdash \varepsilon_1 \otimes \varepsilon_2$,
- $\Gamma, \alpha : K \vdash B_2 <: B_3$, and
- $\Gamma, \alpha : K \vdash \varepsilon_2 \otimes \varepsilon_3$.

By the induction hypothesis and Lemma 3.3(2), we have

- $\Gamma, \alpha : K \vdash B_1 <: B_3$ and
- $\Gamma, \alpha : K \vdash \varepsilon_1 \otimes \varepsilon_3$.

Thus, we have $\Gamma \vdash A_1 <: A_3$ by ST_Poly as required.

■

Lemma 3.5 (Weakening)**.** *Suppose that* $\vdash \Gamma_1, \Gamma_2$ *and* $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3) = \emptyset$.

(1) If $\vdash \Gamma_1, \Gamma_3$*, then* $\vdash \Gamma_1, \Gamma_2, \Gamma_3$.

(2) If $\Gamma_1, \Gamma_3 \vdash S : K$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash S : K$.

(3) If $\Gamma_1, \Gamma_3 \vdash A <: B$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A <: B$.

(4) If $\Gamma_1, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$.

(5) If $\Gamma_1, \Gamma_3 \vdash e : A \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e : A \mid \varepsilon$.

(6) If $\Gamma_1, \Gamma_3 \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_\sigma h : A \Rightarrow^\varepsilon B$.

Proof.

(1)(2) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case C_Empty**:** Clearly because of $\vdash \Gamma_1, \Gamma_2$ and $\Gamma_1 = \Gamma_3 = \emptyset$.

Case C_Var**:** If $\Gamma_3 = \emptyset$, then $\vdash \Gamma_1, \Gamma_2, \Gamma_3$ holds immediately. If $\Gamma_3 \neq \emptyset$, then for some Γ_3', x, and A, the following are given:

- $\Gamma_3 = \Gamma_3', x : A$,
- $x \notin \mathrm{dom}(\Gamma_1, \Gamma_3')$, and
- $\Gamma_1, \Gamma_3' \vdash A : \mathbf{Typ}$.

Since $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3') = \emptyset$ holds, we have $\Gamma_1, \Gamma_2, \Gamma_3' \vdash A : \mathbf{Typ}$ by the induction hypothesis. By $x \notin \mathrm{dom}(\Gamma_1, \Gamma_3')$ and $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3', x : A) = \emptyset$, we have $x \notin \mathrm{dom}(\Gamma_1, \Gamma_2, \Gamma_3')$. Thus, C_Var derives $\vdash \Gamma_1, \Gamma_2, \Gamma_3', x : A$.

Case C_TVar**:** If $\Gamma_3 = \emptyset$, then $\vdash \Gamma_1, \Gamma_2$ holds immediately. If $\Gamma_3 \neq \emptyset$, then for some Γ_3', α, and K, the following are given:

- $\Gamma_3 = \Gamma_3', \alpha : K$,
- $\alpha \notin \mathrm{dom}(\Gamma_1, \Gamma_3')$, and
- $\vdash \Gamma_1, \Gamma_3'$.

Since $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3') = \emptyset$, we have $\vdash \Gamma_1, \Gamma_2, \Gamma_3'$ by the induction hypothesis. By $\alpha \notin \mathrm{dom}(\Gamma_1, \Gamma_3')$ and $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3', \alpha : K) = \emptyset$, we have $\alpha \notin \mathrm{dom}(\Gamma_1, \Gamma_2, \Gamma_3')$ Thus, C_TVar derives $\vdash \Gamma_1, \Gamma_2, \Gamma_3', \alpha : K$.

Case K_Var**:** For some α, the following are given:

- $S = \alpha$,
- $\vdash \Gamma_1, \Gamma_3$, and
- $\alpha : K \in \Gamma_1, \Gamma_3$.

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2, \Gamma_3$. Thus, $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \alpha : K$ holds because of $\alpha : K \in \Gamma_1, \Gamma_2, \Gamma_3$.

Case K_Fun**:** For some A, B, and ε, the following are given:

- $S = A \rightarrow_\varepsilon B$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, \Gamma_3 \vdash A : \mathbf{Typ}$,
- $\Gamma_1, \Gamma_3 \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma_1, \Gamma_3 \vdash B : \mathbf{Typ}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A : \mathbf{Typ}$,
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash B : \mathbf{Typ}$.

Thus, K_Fun derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A \rightarrow_\varepsilon B : \mathbf{Typ}$.

Case K_Poly**:** Without loss of generality, we can choose α such that $\alpha \notin \mathrm{dom}(\Gamma_2)$. For some K', A, and ε, the following are given:

- $S = \forall \alpha : K'.A^\varepsilon$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, \Gamma_3, \alpha : K' \vdash A : \mathbf{Typ}$, and
- $\Gamma_1, \Gamma_3, \alpha : K' \vdash \varepsilon : \mathbf{Eff}$.

Since $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3, \alpha : K') = \emptyset$, we have

- $\Gamma_1, \Gamma_2, \Gamma_3, \alpha : K' \vdash A : \mathbf{Typ}$ and
- $\Gamma_1, \Gamma_2, \Gamma_3, \alpha : K' \vdash \varepsilon : \mathbf{Eff}$

by the induction hypothesis. Thus, K_Poly derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \forall \alpha : K'.A^\varepsilon : \mathbf{Typ}$.

Case K_Cons**:** For some $\mathcal{C}$, $\boldsymbol{S}^{\boldsymbol{I}}$, and $\boldsymbol{K}^{\boldsymbol{I}}$, the following are given:

- $S = \mathcal{C}\, \boldsymbol{S}^{\boldsymbol{I}}$,
- $\mathcal{C} : \Pi \boldsymbol{K}^{\boldsymbol{I}} \rightarrow K \in \Sigma$,
- $\vdash \Gamma_1, \Gamma_3$, and
- $\Gamma_1, \Gamma_3 \vdash \boldsymbol{S}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}$.

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2, \Gamma_3$ and $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \boldsymbol{S}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}$. Thus, K_Cons derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \mathcal{C}\, \boldsymbol{S}^{\boldsymbol{I}} : K$.

(3)(4) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case ST_Refl**:** $A = B$ and $\Gamma_1, \Gamma_3 \vdash A : \mathbf{Typ}$ are given. By case (2), we have $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A : \mathbf{Typ}$. Thus, ST_Refl derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A <: A$.

Case ST_Fun**:** For some A_1, ε_1, B_1, A_2, ε_2, and B_2, the following are given:

- $A = A_1 \rightarrow_{\varepsilon_1} B_1$,
- $B = A_2 \rightarrow_{\varepsilon_2} B_2$,
- $\Gamma_1, \Gamma_3 \vdash A_2 <: A_1$, and
- $\Gamma_1, \Gamma_3 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$. Thus, ST_Fun derives

$$\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 \rightarrow_{\varepsilon_1} B_1 <: A_2 \rightarrow_{\varepsilon_2} B_2$$

as required.

Case ST_POLY**:** Without loss of generality, we can choose α such that $\alpha \notin \mathrm{dom}(\Gamma_2)$. For some K, A_1, ε_1, A_2, and ε_2, the following are given:

- $A = \forall\alpha : K.{A_1}^{\varepsilon_1}$,
- $B = \forall\alpha : K.{A_2}^{\varepsilon_2}$, and
- $\Gamma_1, \Gamma_3, \alpha : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3, \alpha : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$. Thus, ST_POLY derives

$$\Gamma_1, \Gamma_2, \Gamma_3 \vdash \forall\alpha : K.{A_1}^{\varepsilon_1} <: \forall\alpha : K.{A_2}^{\varepsilon_2}$$

as required.

Case ST_COMP**:** We have $\Gamma_1, \Gamma_3 \vdash A_1 <: A_2$ and $\Gamma_1, \Gamma_3 \vdash \varepsilon_1 \otimes \varepsilon_2$. By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 <: A_2$. By case (2), we have $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \varepsilon_1 \otimes \varepsilon_2$. Thus, ST_COMP derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$ as required.

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_VAR**:** For some x, the following are given:

- $e = x$,
- $\varepsilon = \mathbb{0}$,
- $\vdash \Gamma_1, \Gamma_3$, and
- $x : A \in \Gamma_1, \Gamma_3$.

By case (1), we have $\vdash \Gamma_1, \Gamma_2, \Gamma_3$. Thus, T_VAR derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash x : A \mid \mathbb{0}$ because of $x : A \in \Gamma_1, \Gamma_2, \Gamma_3$.

Case T_ABS**:** Without loss of generality, we can choose f and x such that $f \notin \mathrm{dom}(\Gamma_2)$ and $x \notin \mathrm{dom}(\Gamma_2)$. For some e', A', B', and ε', the following are given:

- $e = \mathbf{fun}\,(f, x, e')$,
- $A = A' \rightarrow_{\varepsilon'} B'$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma_1, \Gamma_3, f : A' \rightarrow_{\varepsilon'} B', x : A' \vdash e' : B' \mid \varepsilon'$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3, f : A' \rightarrow_{\varepsilon'} B', x : A' \vdash e' : B' \mid \varepsilon'$ because of $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3, f : A' \rightarrow_{\varepsilon'} B', x : A') = \emptyset$. Thus, T_ABS derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \mathbf{fun}\,(f, x, e') : A' \rightarrow_{\varepsilon'} B' \mid \mathbb{0}$.

Case T_APP**:** For some v_1, v_2, and C, the following are given:

- $e = v_1\, v_2$,
- $\Gamma_1, \Gamma_3 \vdash v_1 : B \rightarrow_{\varepsilon} A \mid \mathbb{0}$, and
- $\Gamma_1, \Gamma_3 \vdash v_2 : B \mid \mathbb{0}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash v_1 : B \rightarrow_{\varepsilon} A \mid \mathbb{0}$ and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash v_2 : B \mid \mathbb{0}$.

Thus, T_APP derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash v_1\, v_2 : A \mid \varepsilon$.

Case T_TABS**:** Without loss of generality, we can choose α such that $\alpha \notin \mathrm{dom}(\Gamma_2)$. For some K, e', B', and ε', the following are given:

- $e = \Lambda\alpha : K.e'$,
- $A = \forall\alpha : K.{A'}^{\varepsilon'}$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma_1, \Gamma_3, \alpha : K \vdash e' : A' \mid \varepsilon'$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3, \alpha : K \vdash e' : A' \mid \varepsilon'$ because of $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3, \alpha : K) = \emptyset$. Thus, T_TABS derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \Lambda\alpha : K.e' : \forall\alpha : K.{A'}^{\varepsilon'} \mid \mathbb{0}$.

Case T_TAPP**:** For some v, S, α, A', ε', and K, the following are given:

- $e = v\, S$,
- $A = A'[S/\alpha]$,
- $\varepsilon = \varepsilon'[S/\alpha]$,

- $\Gamma_1, \Gamma_3 \vdash v : \forall \alpha : K. A'^{\varepsilon'} \mid \mathbb{0}$, and
- $\Gamma_1, \Gamma_3 \vdash S : K$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash v : \forall \alpha : K. A'^{\varepsilon'} \mid \mathbb{0}$ and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash S : K$.

Thus, T_TApp derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash v\, S : A'[S/\alpha] \mid \varepsilon'[S/\alpha]$.

Case T_Let**:** Without loss of generality, we can choose x such that $x \notin \mathrm{dom}(\Gamma_2)$. For some e_1, e_2, and B, the following are given:

- $e = (\textbf{let}\, x = e_1\, \textbf{in}\, e_2)$,
- $\Gamma_1, \Gamma_3 \vdash e_1 : B \mid \varepsilon$, and
- $\Gamma_1, \Gamma_3, x : B \vdash e_2 : A \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e_1 : B \mid \varepsilon$ and
- $\Gamma_1, \Gamma_2, \Gamma_3, x : B \vdash e_2 : A \mid \varepsilon$

because of $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3, x : B) = \emptyset$. Thus, T_Let derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \textbf{let}\, x = e_1\, \textbf{in}\, e_2 : A \mid \varepsilon$.

Case T_Sub**:** For some A' and ε', the following are given:

- $\Gamma_1, \Gamma_3 \vdash e : A' \mid \varepsilon'$ and
- $\Gamma_1, \Gamma_3 \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis and case (4), we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e : A' \mid \varepsilon'$ and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

Thus, T_Sub derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e : A \mid \varepsilon$.

Case T_Op**:** For some op, l, A', B', I, and J, the following are given:

- $e = \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J$,
- $A = (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \to_{(l\, \boldsymbol{S}^I)^\uparrow} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J])$,
- $\varepsilon = \mathbb{0}$,
- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I. \sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J. A' \Rightarrow B' \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\vdash \Gamma_1, \Gamma_3$,
- $\Gamma_1, \Gamma_3 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$, and
- $\Gamma_1, \Gamma_3 \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$.

By cases (1) and (2), we have

- $\vdash \Gamma_1, \Gamma_2, \Gamma_3$,
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$, and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$.

Thus, T_Op derives

$$\Gamma_1, \Gamma_2, \Gamma_3 \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^I : (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \to_{(l\, \boldsymbol{S}^I)^\uparrow} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \mid \mathbb{0}.$$

Case T_Handling**:** For some N, e', A', ε', l, $\boldsymbol{S}^N$, $\boldsymbol{K}^N$, h, and σ, the following are given:

- $e = \textbf{handle}_{l\, \boldsymbol{S}^N}\, e'\, \textbf{with}\, h$,
- $\Gamma_1, \Gamma_3 \vdash e' : A' \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N. \sigma \in \Xi$,
- $\Gamma_1, \Gamma_3 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A' \Rightarrow^{\varepsilon} A$,
- $\Gamma_1, \Gamma_3 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$, and
- $(l\, \boldsymbol{S}^N)^\uparrow \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e' : A' \mid \varepsilon'$,
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A' \Rightarrow^{\varepsilon} A$, and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$.

Thus, T_Handling derives

$$\Gamma_1, \Gamma_2, \Gamma_3 \vdash \textbf{handle}_{l\,\boldsymbol{S}^N}\, e\, \textbf{with}\, h : A \mid \varepsilon.$$

Case H_Return**:** Without loss of generality, we can choose x such that $x \notin \mathrm{dom}(\Gamma_2)$. For some e_r, the following are given:

- $h = \{\textbf{return}\, x \mapsto e_r\}$,
- $\sigma = \{\}$, and
- $\Gamma_1, \Gamma_3, x : A \vdash e_r : B \mid \varepsilon$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3, x : A \vdash e_r : B \mid \varepsilon$. Thus, H_Return derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\{\}} \{\textbf{return}\, x \mapsto e_r\} : A \Rightarrow^{\varepsilon} B$.

Case H_Op**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ and p and k such that:

- $\{\boldsymbol{\beta}^J\} \cap \mathrm{dom}(\Gamma_2) = \emptyset$,
- $p \notin \mathrm{dom}(\Gamma_2)$, and
- $k \notin \mathrm{dom}(\Gamma_2)$.

For some h', σ', op, A', B', and e, the following are given:

- $h = h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J . A' \Rightarrow B'\}$,
- $\Gamma_1, \Gamma_3 \vdash_{\sigma'} h' : A \Rightarrow^{\varepsilon} B$, and
- $\Gamma_1, \Gamma_3, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \to_{\varepsilon} B \vdash e : B \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma'} h' : A \Rightarrow^{\varepsilon} B$ and
- $\Gamma_1, \Gamma_2, \Gamma_3, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \to_{\varepsilon} B \vdash e : B \mid \varepsilon$.

Thus, H_Op derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma} h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\} : A \Rightarrow^{\varepsilon} B$.

■

Lemma 3.6. *For any Γ_1, Γ_2, S, and K, if $\Delta(\Gamma_1), \Gamma_2 \vdash S : K$ and $\vdash \Gamma_1$ and $\mathrm{dom}(\Gamma_1) \cap \mathrm{dom}(\Gamma_2) = \emptyset$, then $\Gamma_1, \Gamma_2 \vdash S : K$.*

Proof. By induction on the size of Γ_1. We proceed by case analysis on the rule lastly applied to this derivation.
Case C_Empty**:** Clearly.

Case C_Var**:** For some Γ'_1, x, and A, we have

- $\Gamma_1 = \Gamma'_1, x : A$,
- $x \notin \mathrm{dom}(\Gamma'_1)$, and
- $\Gamma'_1 \vdash A : \mathbf{Typ}$.

By Lemma 3.1, we have $\vdash \Gamma'_1$. By Definition 1.11, we have $\Delta(\Gamma'_1), \Gamma_2 \vdash S : K$. By $\mathrm{dom}(\Gamma'_1) \subseteq \mathrm{dom}(\Gamma_1)$, we have $\mathrm{dom}(\Gamma'_1) \cap \mathrm{dom}(\Gamma_2) = \emptyset$. By the induction hypothesis, we have $\Gamma'_1, \Gamma_2 \vdash S : K$. By Lemma 3.5(2), we have $\Gamma'_1, x : A, \Gamma_2 \vdash S : K$ as required.

Case C_TVar**:** For some Γ'_1, α, and K', we have

- $\Gamma_1 = \Gamma'_1, \alpha : K'$,
- $\vdash \Gamma'_1$, and
- $\alpha \notin \mathrm{dom}(\Gamma'_1)$.

By Definition 1.11, we have $\Delta(\Gamma'_1), \alpha : K', \Gamma_2 \vdash S : K$. By $\alpha \notin \mathrm{dom}(\Gamma'_1)$ and $\mathrm{dom}(\Gamma'_1) \subseteq \mathrm{dom}(\Gamma_1)$, we have $\mathrm{dom}(\Gamma'_1) \cap \mathrm{dom}(\alpha : K', \Gamma'_2) = \emptyset$. By the induction hypothesis, we have $\Gamma'_1, \alpha : K', \Gamma_2 \vdash S : K$ as required.

■

Lemma 3.7 (Substitution of values)**.** *Suppose that $\Gamma_1 \vdash v : A \mid \mathbb{0}$.*

(1) If $\vdash \Gamma_1, x : A, \Gamma_2$, then $\vdash \Gamma_1, \Gamma_2$.

(2) If $\Gamma_1, x : A, \Gamma_2 \vdash S : K$, then $\Gamma_1, \Gamma_2 \vdash S : K$.

(3) If $\Gamma_1, x : A, \Gamma_2 \vdash B <: C$, then $\Gamma_1, \Gamma_2 \vdash B <: C$.

(4) If $\Gamma_1, x : A, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$, then $\Gamma_1, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$.

(5) *If* $\Gamma_1, x : A, \Gamma_2 \vdash e : B \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2 \vdash e[v/x] : B \mid \varepsilon$*.*

(6) *If* $\Gamma_1, x : A, \Gamma_2 \vdash_\sigma h : B \Rightarrow^\varepsilon C$*, then* $\Gamma_1, \Gamma_2 \vdash_\sigma h[v/x] : B \Rightarrow^\varepsilon C$*.*

Proof.

(1)(2) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case C_Empty**:** Cannot happen.

Case C_Var**:** If $\Gamma_2 = \emptyset$, then we have $\Gamma_1 \vdash A : \mathbf{Typ}$. By Lemma 3.1, $\vdash \Gamma_1$ holds. If $\Gamma_2 \neq \emptyset$, then we have

- $\Gamma_2 = \Gamma_2', y : B$,
- $\Gamma_1, x : A, \Gamma_2' \vdash B : \mathbf{Typ}$, and
- $y \notin \mathrm{dom}(\Gamma_1, x : A, \Gamma_2')$,

for some Γ_2', y, and B. By the induction hypothesis, we have $\Gamma_1, \Gamma_2' \vdash B : \mathbf{Typ}$. Thus, C_Var derives $\vdash \Gamma_1, \Gamma_2$ because $y \notin \mathrm{dom}(\Gamma_1, \Gamma_2')$.

Case C_TVar**:** Since Γ_2 cannot be $\emptyset$, we have

- $\Gamma_2 = \Gamma_2', \alpha : K$,
- $\vdash \Gamma_1, x : A, \Gamma_2'$, and
- $\alpha \notin \mathrm{dom}(\Gamma_1, x : A, \Gamma_2')$,

for some Γ_2, α, and K. By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2'$. Thus, C_TVar derives $\vdash \Gamma_1, \Gamma_2$ because $\alpha \notin \mathrm{dom}(\Gamma_1, \Gamma_2')$.

Case K_Var**:** For some α, the following are given:

- $S = \alpha$,
- $\vdash \Gamma_1, x : A, \Gamma_2$, and
- $\alpha : K \in \Gamma_1, x : A, \Gamma_2$.

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2$. Thus, K_Var derives $\Gamma_1, \Gamma_2 \vdash \alpha : K$ because of $\alpha : K \in \Gamma_1, \Gamma_2$.

Case K_Fun**:** For some B, C, and ε, the following are given:

- $S = B \to_\varepsilon C$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, x : A, \Gamma_2 \vdash B : \mathbf{Typ}$,
- $\Gamma_1, x : A, \Gamma_2 \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash C : \mathbf{Typ}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2 \vdash B : \mathbf{Typ}$,
- $\Gamma_1, \Gamma_2 \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma_1, \Gamma_2 \vdash C : \mathbf{Typ}$.

Thus, K_Fun derives $\Gamma_1, \Gamma_2 \vdash B \to_\varepsilon C : \mathbf{Typ}$.

Case K_Poly**:** For some α, K', A', and ε, the following are given:

- $S = \forall \alpha : K'.A'^{\varepsilon}$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, x : A, \Gamma_2, \alpha : K' \vdash A' : \mathbf{Typ}$, and
- $\Gamma_1, x : A, \Gamma_2, \alpha : K' \vdash \varepsilon : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \alpha : K' \vdash A' : \mathbf{Typ}$, and
- $\Gamma_1, \Gamma_2, \alpha : K' \vdash \varepsilon : \mathbf{Eff}$.

Thus, K_Poly derives $\Gamma_1, \Gamma_2 \vdash \forall \alpha : K'.A'^{\varepsilon} : \mathbf{Typ}$.

Case K_Cons**:** For some $\mathcal{C}$, $\boldsymbol{S}^I$ and $\boldsymbol{K}^I$, the following are given:

- $S = \mathcal{C}\, \boldsymbol{S}^I$,
- $\mathcal{C} : \Pi \boldsymbol{K}^I \to K \in \Sigma$,
- $\vdash \Gamma_1, x : A, \Gamma_2$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$.

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2$ and $\Gamma_1, \Gamma_2 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$. Thus, K_Cons derives $\Gamma_1, \Gamma_2 \vdash \mathcal{C}\, \boldsymbol{S}^I : K$.

(3)(4) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case ST_Refl**:** $B = C$ and $\Gamma_1, x : A, \Gamma_2 \vdash B : \mathbf{Typ}$ are given. By case (2), we have $\Gamma_1, \Gamma_2 \vdash B : \mathbf{Typ}$. Thus, ST_Refl derives $\Gamma_1, \Gamma_2 \vdash B <: B$.

Case ST_Fun**:** For some A_{11}, ε_1, A_{12}, A_{21}, ε_2, and A_{22}, the following are given:

- $B = A_{11} \rightarrow_{\varepsilon_1} A_{12}$,
- $C = A_{21} \rightarrow_{\varepsilon_2} A_{22}$,
- $\Gamma_1, x : A, \Gamma_2 \vdash A_{21} <: A_{11}$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash A_{12} \mid \varepsilon_1 <: A_{22} \mid \varepsilon_2$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2 \vdash A_{21} <: A_{11}$ and $\Gamma_1, \Gamma_2 \vdash A_{12} \mid \varepsilon_1 <: A_{22} \mid \varepsilon_2$. Thus, ST_Fun derives $\Gamma_1, \Gamma_2 \vdash A_{11} \rightarrow_{\varepsilon_1} A_{12} <: A_{21} \rightarrow_{\varepsilon_2} A_{22}$.

Case ST_Poly**:** For some α, K, A_1, ε_1, A_2, and ε_2, the following are given:

- $B = \forall \alpha : K.A_1{}^{\varepsilon_1}$,
- $C = \forall \alpha : K.A_2{}^{\varepsilon_2}$, and
- $\Gamma_1, x : A, \Gamma_2, \alpha : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \alpha : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$. Thus, ST_Poly derives $\Gamma_1, \Gamma_2 \vdash \forall \alpha : K.A_1{}^{\varepsilon_1} <: \forall \alpha : K.A_2{}^{\varepsilon_2}$.

Case ST_Comp**:** We have $\Gamma_1, x : A, \Gamma_2 \vdash B_1 <: B_2$ and $\Gamma_1, x : A, \Gamma_2 \vdash \varepsilon_1 \otimes \varepsilon_2$. By the induction hypothesis, we have $\Gamma_1, \Gamma_2 \vdash B_1 <: B_2$. By case (2), we have $\Gamma_1, \Gamma_2 \vdash \varepsilon_1 \otimes \varepsilon_2$. Thus, ST_Comp derives $\Gamma_1, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$ as required.

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_Var**:** For some y, the following are given:

- $e = y$,
- $\varepsilon = \mathbb{0}$,
- $\vdash \Gamma_1, x : A, \Gamma_2$, and
- $y : B \in \Gamma_1, x : A, \Gamma_2$.

By case (1), we have $\vdash \Gamma_1, \Gamma_2$.

If $y = x$, then $\Gamma_1, \Gamma_2 \vdash v : A \mid \mathbb{0}$ holds because of $\Gamma_1 \vdash v : A \mid \mathbb{0}$ and Lemma 3.5(5).

If $y \neq x$, then we have $y : B \in \Gamma_1, \Gamma_2$. Thus, T_Var derives $\Gamma_1, \Gamma_2 \vdash y : B \mid \mathbb{0}$.

Case T_Abs**:** Without loss of generality, we can choose f and y such that $f, y \neq x$ and $f, y \notin \mathrm{FV}(v)$. For some e', A', B', and ε', the following are given:

- $e = \mathbf{fun}\,(f, y, e')$,
- $B = A' \rightarrow_{\varepsilon'} B'$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma_1, x : A, \Gamma_2, f : A' \rightarrow_{\varepsilon'} B', y : A' \vdash e' : B' \mid \varepsilon'$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, g : A' \rightarrow_{\varepsilon'} B', y : A' \vdash e'[v/x] : B' \mid \varepsilon'$. Thus, T_Abs derives $\Gamma_1, \Gamma_2 \vdash \mathbf{fun}\,(f, y, e'[v/x]) : A' \rightarrow_{\varepsilon'} B' \mid \mathbb{0}$, and since $(\mathbf{fun}\,(f, y, e'))[v/x] = \mathbf{fun}\,(f, y, e'[v/x])$, the required result is achieved.

Case T_App**:** For some v_1, v_2, and C, the following are given:

- $e = v_1\, v_2$,
- $\Gamma_1, x : A, \Gamma_2 \vdash v_1 : C \rightarrow_{\varepsilon} B \mid \mathbb{0}$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash v_2 : C \mid \mathbb{0}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2 \vdash v_1[v/x] : C \rightarrow_{\varepsilon} B \mid \mathbb{0}$
- and $\Gamma_1, \Gamma_2 \vdash v_2[v/x] : C \mid \mathbb{0}$.

Thus, T_App derives $\Gamma_1, \Gamma_2 \vdash (v_1[v/x])\,(v_2[v/x]) : B \mid \varepsilon$, and since $(v_1\, v_2)[v/x] = (v_1[v/x])\,(v_2[v/x])$, the required result is achieved.

Case T_TAbs**:** Without loss of generality, we can choose α such that $\alpha \notin \mathrm{FTV}(v)$. For some K, e', B', and ε', the following are given:

- $e = \Lambda\alpha : K.e'$,
- $B = \forall\alpha : K.B'^{\varepsilon'}$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma_1, x : A, \Gamma_2, \alpha : K \vdash e' : B' \mid \varepsilon'$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \alpha : K \vdash e'[v/x] : B' \mid \varepsilon'$. Thus, T_TAbs derives $\Gamma_1, \Gamma_2 \vdash \Lambda\alpha : K.e'[v/x] : \forall\alpha : K.B'^{\varepsilon'} \mid \mathbb{0}$, and since $(\Lambda\alpha : K.e')[v/x] = \Lambda\alpha : K.e'[v/x]$, the required result is achieved.

Case T_TApp**:** For some v', S, α, B', ε', and K, the following are given:

- $e = v'\,S$,
- $B = B'[S/\alpha]$,
- $\varepsilon = \varepsilon'[S/\alpha]$,
- $\Gamma_1, x : A, \Gamma_2 \vdash v' : \forall\alpha : K.B'^{\varepsilon'} \mid \mathbb{0}$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash S : K$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2 \vdash v'[v/x] : \forall\alpha : K.B'^{\varepsilon} \mid \mathbb{0}$ and
- $\Gamma_1, \Gamma_2 \vdash S : K$.

Thus, T_TApp derives $\Gamma_1, \Gamma_2 \vdash v'[v/x]\,S : B'[S/\alpha] \mid \varepsilon'[S/\alpha]$, and since $(v'\,S)[v/x] = v'[v/x]\,S$, the required result is achieved.

Case T_Let**:** Without loss of generality, we can choose y such that $y \neq x$ and $y \notin \mathrm{FV}(v)$. For some e_1, e_2, and C, the following are given:

- $e = (\mathbf{let}\,y = e_1\,\mathbf{in}\,e_2)$,
- $\Gamma_1, x : A, \Gamma_2 \vdash e_1 : C \mid \varepsilon$, and
- $\Gamma_1, x : A, \Gamma_2, y : C \vdash e_2 : B \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2 \vdash e_1[v/x] : C \mid \varepsilon$ and
- $\Gamma_1, \Gamma_2, y : C \vdash e_2[v/x] : B \mid \varepsilon$.

Thus, T_Let derives $\Gamma_1, \Gamma_2 \vdash \mathbf{let}\,y = e_1[v/x]\,\mathbf{in}\,e_2[v/x] : B \mid \varepsilon$, and since $(\mathbf{let}\,y = e_1\,\mathbf{in}\,e_2)[v/x] = \mathbf{let}\,y = e_1[v/x]\,\mathbf{in}\,e_2[v/x]$, the required result is achieved.

Case T_Sub**:** For some B' and ε', the following are given:

- $\Gamma_1, x : A, \Gamma_2 \vdash e : B' \mid \varepsilon'$ and
- $\Gamma_1, x : A, \Gamma_2 \vdash B' \mid \varepsilon' <: B \mid \varepsilon$.

By the induction hypothesis and case (4), we have $\Gamma_1, \Gamma_2 \vdash e[v/x] : B' \mid \varepsilon'$ and $\Gamma_1, \Gamma_2 \vdash B' \mid \varepsilon' <: B \mid \varepsilon$. Thus, T_Sub derives $\Gamma_1, \Gamma_2 \vdash e[v/x] : B \mid \varepsilon$.

Case T_Op**:** For some $\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J$, A', and B', the following are given:

- $e = \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J$,
- $B = (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \rightarrow_{(l\,\boldsymbol{S}^I)^{\uparrow}} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J])$,
- $\varepsilon = \mathbb{0}$,
- $l :: \forall\boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$
- $\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K'}^J.A' \Rightarrow B' \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\vdash \Gamma_1, x : A, \Gamma_2$,
- $\Gamma_1, x : A, \Gamma_2 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$.

By cases (1) and (2), we have

- $\vdash \Gamma_1, \Gamma_2$,
- $\Gamma_1, \Gamma_2 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$, and
- $\Gamma_1, \Gamma_2 \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$.

Thus, T_Op derives

$$\Gamma_1, \Gamma_2 \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J : (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \rightarrow_{(l\,\boldsymbol{S}^I)^{\uparrow}} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \mid \mathbb{0}.$$

Case T_HANDLING**:** For some N, e', A', ε', l, $\boldsymbol{S}^N$, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, h, and σ, the following are given:

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\; e'\;\mathbf{with}\; h$,
- $\Gamma_1, x : A, \Gamma_2 \vdash e' : A' \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\Gamma_1, x : A, \Gamma_2 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\Gamma_1, x : A, \Gamma_2 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A' \Rightarrow^{\varepsilon} B$, and
- $(l\,\boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\Gamma_1, \Gamma_2 \vdash e'[v/x] : A' \mid \varepsilon'$, and
- $\Gamma_1, \Gamma_2 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h[v/x] : A' \Rightarrow^{\varepsilon} A$.

Thus, T_HANDLING derives

$$\Gamma_1, \Gamma_2 \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\; e'[v/x]\;\mathbf{with}\; h[v/x] : B \mid \varepsilon.$$

Case H_RETURN**:** Without loss of generality, we can choose y such that $y \neq x$ and $y \notin \mathrm{FV}(v)$. For some e_r, the following are given:

- $h = \{\mathbf{return}\; y \mapsto e_r\}$,
- $\sigma = \{\}$, and
- $\Gamma_1, x : A, \Gamma_2, y : B \vdash e_r : C \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, y : B \vdash e_r[v/x] : C \mid \varepsilon$.

Thus, H_RETURN derives

$$\Gamma_1, \Gamma_2 \vdash_{\{\}} \{\mathbf{return}\; y \mapsto e_r[v/x]\} : B \Rightarrow^{\varepsilon} C.$$

Case H_OP**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ and p and k such that:

- $p \neq x$,
- $k \neq x$,
- $p \notin \mathrm{FV}(v)$,
- $k \notin \mathrm{FV}(v)$, and
- $\{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(v) = \emptyset$.

For some h', σ', op, A', B', and e, the following are given:

- $h = h' \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\; p\; k \mapsto e\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B'\}$,
- $\Gamma_1, x : A, \Gamma_2 \vdash_{\sigma'} h' : B \Rightarrow^{\varepsilon} C$, and
- $\Gamma_1, x : A, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon} C \vdash e : C \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2 \vdash_{\sigma'} h'[v/x] : A \Rightarrow^{\varepsilon} B$ and
- $\Gamma_1, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon} B \vdash e[v/x] : B \mid \varepsilon$.

Thus, H_OP derives

$$\Gamma_1, \Gamma_2 \vdash_{\sigma} h'[v/x] \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\; p\; k \mapsto e[v/x]\} : B \Rightarrow^{\varepsilon} C$$

.

■

Lemma 3.8 (Well-formedness of contexts in subtyping judgments)**.**

- *If* $\Gamma \vdash A_1 <: A_2$*, then* $\vdash \Gamma$.
- *If* $\Gamma \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\vdash \Gamma$.

Proof. Straightforward by mutual induction on the subtyping derivations with Lemma 3.1. ■

Lemma 3.9 (Well-formedness of contexts in typing judgments)**.**

- *If* $\Gamma \vdash e : A \mid \varepsilon$*, then* $\vdash \Gamma$.

- *If* $\Gamma \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\vdash \Gamma$.

Proof. Straightforward by mutual induction on the derivations with Lemma 3.1. ■

Lemma 3.10 (Substitution of Typelikes)**.** *Suppose that* $\Gamma_1 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$*.*

(1) If $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$*, then* $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(2) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash T : K$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : K$*.*

(3) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A <: B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(4) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(5) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e : A \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(6) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S}/\boldsymbol{\alpha}]} h[\boldsymbol{S}/\boldsymbol{\alpha}] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

Proof.

(1)(2) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivations.

Case C_Empty**:** Cannot happen.

Case C_Var**:** Since Γ_2 cannot be $\emptyset$, for some Γ'_2, x, and A, the following are given:
- $\Gamma_2 = \Gamma'_2, x : A$,
- $x \notin \mathrm{dom}(\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma'_2)$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma'_2 \vdash A : \mathbf{Typ}$.

By the induction hypothesis, we have $\Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Typ}$. By $x \notin \mathrm{dom}(\Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]))$, C_Var derives $\vdash \Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]), x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and since $\Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ holds, the required result is achieved.

Case C_TVar**:** If $\Gamma_2 = \emptyset$, we have
- $\boldsymbol{\alpha}^I : \boldsymbol{K}^I = \boldsymbol{\alpha}^J : \boldsymbol{K}^J, \alpha_i : K_i$,
- $\vdash \Gamma_1, \boldsymbol{\alpha}^J : \boldsymbol{K}^J$, and
- $\alpha_i \notin \mathrm{dom}(\Gamma_1, \boldsymbol{\alpha}^J : \boldsymbol{K}^J)$,

for some J, $\boldsymbol{\alpha}^J$, $\boldsymbol{K}^J$, i, α_i, and K_i. By the induction hypothesis, we have $\vdash \Gamma_1$.
If $\Gamma_2 \neq \emptyset$, for some Γ_2, β, and K', the following are given:
- $\Gamma_2 = \Gamma'_2, \beta : K'$,
- $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma'_2$, and
- $\beta \notin \mathrm{dom}(\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma'_2)$.

By the induction hypothesis, we have $\vdash \Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$. Thus, C_TVar derives $\vdash \Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]), \beta : K'$, and since

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \Gamma_1, (\Gamma'_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]), \beta : K'$$

holds, the required result is achieved.

Case K_Var**:** For some β, the following are given:
- $T = \beta$,
- $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$, and
- $\beta : K \in \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$.

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.
If $\beta = \alpha_i$ for some $i \in I$, then $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \beta[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : K$ holds because of the following:
- $\Gamma_1 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- Lemma 3.5(2),
- $S_i = \beta[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and
- $K_i = K$.

If $\beta \neq \alpha_i$ for any $i \in I$, then K_Var derives $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \beta : K$ because of $\beta : K \in \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$. Since $\beta = \beta[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, the required result is achieved.

Case K_FUN**:** For some A, B, and ε, the following are given:

- $S = A \rightarrow_{\varepsilon} B$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A : \mathbf{Typ}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \varepsilon : \mathbf{Eff}$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash B : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Typ}$,
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Eff}$, and
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Eff}$.

Thus, K_FUN derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash (A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \rightarrow_{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} (B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) : \mathbf{Typ},$$

and since

$$(A \rightarrow_{\varepsilon} B)[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = (A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \rightarrow_{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} (B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$$

holds, the required result is achieved.

Case K_POLY**:** For some β, K', A, and ε, the following are given:

- $S = \forall \beta : K'.A^{\varepsilon}$,
- $K = \mathbf{Typ}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, \beta : K' \vdash A : \mathbf{Typ}$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, \beta : K' \vdash \varepsilon : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], \beta : K' \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Typ}$ and
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], \beta : K' \vdash \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Eff}$.

Thus, K_POLY derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \forall \beta : K'.A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])} : \mathbf{Typ}\ .$$

Since we can assume that β does not occur in $\boldsymbol{S}^I$ and $\boldsymbol{\alpha}^I$ without loss of generality, we have

$$(\forall \beta : K'.A^{\varepsilon})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \forall \beta : K'.A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])}\ .$$

Therefore, the required result is achieved.

Case K_CONS**:** For some $\mathcal{C}$, $\boldsymbol{S'}^J$, and $\boldsymbol{K'}^J$, the following are given:

- $S = \mathcal{C}\,\boldsymbol{S'}^J$,
- $\mathcal{C} : \Pi \boldsymbol{K'}^J \rightarrow K \in \Sigma$,
- $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \boldsymbol{S'}^J : \boldsymbol{K'}^J$

By the induction hypothesis, we have $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \boldsymbol{S'[S^I/\alpha^I]}^J : \boldsymbol{K'}^J$. Thus, K_CONS derives $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \mathcal{C}\,\boldsymbol{S'[S^I/\alpha^I]}^J : K$.

(3)(4) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case ST_REFL**:** $A = B$ and $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A : \mathbf{Typ}$ are given. By case (2), we have $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Typ}$. Thus, ST_REFL derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Case ST_FUN**:** For some A_{11}, ε_1, A_{12}, A_{21}, ε_2, B_{22}, the following are given:

- $A = A_{11} \rightarrow_{\varepsilon_1} A_{12}$,
- $B = A_{21} \rightarrow_{\varepsilon_2} A_{22}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_{21} <: A_{11}$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_{12} \mid \varepsilon_1 <: A_{22} \mid \varepsilon_2$.

By the induction hypothesis, we have

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_{21}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_{11}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_{12}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_{22}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, ST_Fun drives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash (A_{11}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \rightarrow_{\varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} (A_{12}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) <: (A_{21}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \rightarrow_{\varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} (A_{22}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$$

and since, for any $i \in \{1, 2\}$,

$$(A_{i\,1} \rightarrow_{\varepsilon_i} A_{i\,2})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = (A_{i\,1}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \rightarrow_{\varepsilon_i[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} (A_{i\,2}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$$

holds, the required result is achieved.

Case ST_Poly**:** For some β, K, A_1, ε_1, A_2, and ε_2, the following are given:

– $A = \forall \beta : K.A_1{}^{\varepsilon_1}$,

– $B = \forall \beta : K.A_2{}^{\varepsilon_2}$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, \beta : K \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], \beta : K \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$. Thus, ST_Poly derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \forall \beta : K.A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])} <: \forall \beta : K.A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])}$$

and since

– $(\forall \beta : K.A_1{}^{\varepsilon_1})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \forall \beta : K.A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])}$ and

– $(\forall \beta : K.A_2{}^{\varepsilon_2})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \forall \beta : K.A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])}$

hold, the required result is achieved.

Case ST_Comp**:** We have $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_1 <: A_2$ and $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \varepsilon_1 \otimes \varepsilon_2$. By the induction hypothesis, we have $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$. By Lemma 3.8, $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$. Then, by case (2) and the fact that a typelike substitution is homomorphism for $\odot$ and $\sim$, we have $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash (\varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) \otimes (\varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$. Thus, ST_Comp derives $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_Var**:** For some x, the following are given:

– $e = x$,

– $\varepsilon = \mathbb{0}$,

– $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$, and

– $x : A \in \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$.

By case (1), we have $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Case $x : A \in \Gamma_1$**:** Since $x : A \in \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and $A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = A$ hold, T_Var derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \mathbb{0}.$$

Case $x : A \in \boldsymbol{\alpha}^I : \boldsymbol{K}^I$**:** Cannot happen.

Case $x : A \in \Gamma_2$**:** Since $x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \in \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ holds, T_Var derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \mathbb{0}.$$

Thus, the required result is achieved because of $x[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = x$.

Case T_Abs**:** For some f, x, e', A', B', and ε', the following are given:

– $e = \mathbf{fun}\,(f, x, e')$,

– $A = A' \rightarrow_{\varepsilon'} B'$,

– $\varepsilon = \mathbb{0}$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, f : A' \rightarrow_{\varepsilon'} B', x : A' \vdash e' : B' \mid \varepsilon'$.

By the induction hypothesis, we have

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}], f : (A' \to_{\varepsilon'} B')[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}], x : A'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] : B'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \mid \varepsilon'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}].$$

Since

$$(A' \to_{\varepsilon'} B')[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] = (A'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) \to_{\varepsilon'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]} (B'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])$$

holds, T_Abs derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash \mathbf{fun}\,(f, x, e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) : (A'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) \to_{\varepsilon'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]} (B'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) \mid \mathbb{0}.$$

Thus, the required result is achieved because

$$(\mathbf{fun}\,(f, x, e'))[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] = \mathbf{fun}\,(f, x, e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])$$

holds.

Case T_App**:** For some v_1, v_2, and B, the following are given:

- $e = v_1\, v_2$,
- $\Gamma_1, \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \Gamma_2 \vdash v_1 : B \to_\varepsilon A \mid \mathbb{0}$, and
- $\Gamma_1, \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \Gamma_2 \vdash v_2 : B \mid \mathbb{0}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash v_1[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] : (B \to_\varepsilon A)[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \mid \mathbb{0}[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]$ and
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash v_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] : B[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \mid \mathbb{0}[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]$.

Since

- $(B \to_\varepsilon A)[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] = (B[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) \to_{\varepsilon[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]} (A[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])$ and
- $\mathbb{0}[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] = \mathbb{0}$

hold, T_App derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash (v_1[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])\,(v_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) : A[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \mid \varepsilon[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]$$

as required.

Case T_TAbs**:** Without loss of generality, we can choose β such that $\beta \neq \alpha_i$ and $\beta \notin \mathrm{FTV}(S_i)$ for any $i \in I$. For some K, e', A', and ε', the following are given:

- $e = \Lambda\beta : K.e'$,
- $A = \forall\beta : K.A'^{\varepsilon'}$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma_1, \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \Gamma_2, \beta : K \vdash e' : A' \mid \varepsilon'$.

By the induction hypothesis, we have

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}], \beta : K \vdash e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] : A'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \mid \varepsilon'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]$$

Thus, T_TAbs derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] \vdash \Lambda\beta : K.(e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]) : \forall\beta : K.A'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}]^{(\varepsilon'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])} \mid \mathbb{0}$$

and since

$$(\Lambda\beta : K.e')[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}] = \Lambda\beta : K.(e'[\boldsymbol{S}^{\boldsymbol{I}}/\boldsymbol{\alpha}^{\boldsymbol{I}}])$$

holds, the required result is achieved.

Case T_TApp**:** Without loss of generality, we can choose β such that $\beta \neq \alpha_i$ and $\beta \notin \mathrm{FTV}(S_i)$ for any $i \in I$. For some v, T, β, A', ε', and K, the following are given:

- $e = v\,T$,
- $A = A'[T/\beta]$,
- $\varepsilon = \varepsilon'[T/\beta]$,
- $\Gamma_1, \boldsymbol{\alpha}^{\boldsymbol{I}} : \boldsymbol{K}^{\boldsymbol{I}}, \Gamma_2 \vdash v : \forall\beta : K.A'^{\varepsilon'} \mid \mathbb{0}$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash T : K$.

By the induction hypothesis, we have

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash v[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : (\forall \beta : K.A'^{\varepsilon'})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \mathbb{0}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : K$.

Since

– $(\forall \beta : K.A'^{\varepsilon'})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \forall \beta : K.A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{(\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])}$ and

– $\mathbb{0}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \mathbb{0}$

hold, T_TApp derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash (v[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])\,(T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) : (A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])[T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]/\beta] \mid (\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])[T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]/\beta].$$

Finally, we have

– $(v\,T)[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = (v[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])\,(T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$,

– $(A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])[T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]/\beta] = (A'[T/\beta])[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and

– $(\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])[T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]/\beta] = (\varepsilon'[T/\beta])[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$

because $\forall i \in I.(\beta \notin \mathrm{FTV}(S_i))$. Thus, the required result is achieved.

Case T_Let**:** For some x, e_1, e_2, and B, the following are given:

– $e = (\mathbf{let}\, x = e_1 \,\mathbf{in}\, e_2)$

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e_1 : B \mid \varepsilon$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, x : B \vdash e_2 : A \mid \varepsilon$.

By the induction hypothesis, we have

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], x : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, T_Let derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \mathbf{let}\, x = e_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \,\mathbf{in}\, (e_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]) : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$$

and since

$$(\mathbf{let}\, x = e_1 \,\mathbf{in}\, e_2)[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \mathbf{let}\, x = e_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \,\mathbf{in}\, (e_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I])$$

holds, the required result is achieved.

Case T_Sub**:** For some A' and ε', the following are given:

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e : A' \mid \varepsilon'$ and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis and case (4), we have

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$ and

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, T_Sub derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$$

as required.

Case T_Op**:** For some op, l, $\boldsymbol{S_0}^{I_0}$, $\boldsymbol{T}^J$, σ, $\boldsymbol{\alpha_0}^{I_0}$, $\boldsymbol{K_0}^{I_0}$, $\boldsymbol{\beta}^J$, $\boldsymbol{K''}^J$, A', and B', the following are given:

– $e = \mathsf{op}_{l\,\boldsymbol{S_0}^{I_0}}\,\boldsymbol{T}^J$,

– $A = (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \rightarrow_{(l\,\boldsymbol{S_0}^{I_0})^{\uparrow}} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J])$,

– $\varepsilon = \mathbb{0}$,

– $l :: \forall \boldsymbol{\alpha_0}^{I_0} : \boldsymbol{K_0}^{I_0}.\sigma \in \Xi$,

– $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K''}^J.A' \Rightarrow B' \in \sigma[\boldsymbol{S_0}^{I_0}/\boldsymbol{\alpha_0}^{I_0}]$,

– $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$,

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \boldsymbol{S_0}^{I_0} : \boldsymbol{K_0}^{I_0}$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \boldsymbol{T}^J : \boldsymbol{K''}^J$.

By cases (1) and (2), we have

– $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{I_0} : \boldsymbol{K_0}^{I_0}$, and
– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J} : \boldsymbol{K''}^J$.

Since
– $((l\,\boldsymbol{S_0}^{I_0})^{\uparrow})[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = (l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{I_0})^{\uparrow}$ and
– $\mathbb{0}[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \mathbb{0}$,

T_Op derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \mathsf{op}_{l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{I_0}}\, \boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J} : A_0'[\boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J}/\boldsymbol{\beta}^J] \rightarrow_{(l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{I_0})^{\uparrow}} B_0'[\boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J}/\boldsymbol{\beta}^J] \mid \mathbb{0}$$

where

$$\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K''}^J . A_0' \Rightarrow B_0' \in \sigma[\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{I_0}/\boldsymbol{\alpha_0}^{I_0}].$$

Without loss of generality, we can assume that, for any $i \in I$, $\alpha_i \notin \mathrm{FTV}(A') \cup \mathrm{FTV}(B')$, and $(\{\alpha_i\} \cup \mathrm{FTV}(S_i)) \cap (\{\boldsymbol{\alpha_0}^{I_0}\} \cup \{\boldsymbol{\beta}^J\}) = \emptyset$. Then,
– $A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J][\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = A_0'[\boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J}/\boldsymbol{\beta}^J]$ and
– $B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J][\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = B_0'[\boldsymbol{T}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{J}/\boldsymbol{\beta}^J]$

hold. Therefore, the required result is achieved.

Case T_Handling**:** For some N, e', A', ε', l, $\boldsymbol{S_0}^N$, $\boldsymbol{\alpha_0}^N$, $\boldsymbol{K_0}^N$, h, and σ, the following are given:
– $e = \mathbf{handle}_{l\,\boldsymbol{S_0}^N}\, e'\, \mathbf{with}\, h$,
– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e' : A' \mid \varepsilon'$,
– $l :: \forall \boldsymbol{\alpha_0}^N : \boldsymbol{K_0}^N . \sigma \in \Xi$,
– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \boldsymbol{S_0}^N : \boldsymbol{K_0}^N$,
– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_{\sigma[\boldsymbol{S_0}^N/\boldsymbol{\alpha_0}^N]} h : A' \Rightarrow^{\varepsilon} A$, and
– $(l\,\boldsymbol{S_0}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis, case (2), and the fact that a typelike substitution is homomorphism for $\odot$ and $\sim$,we have
– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{N} : \boldsymbol{K_0}^N$,
– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S_0}^N/\boldsymbol{\alpha_0}^N][\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and
– $(l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^{N})^{\uparrow} \odot \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \sim \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Now, because we can assume that
– $\{\boldsymbol{\alpha}^I\} \cap \{\boldsymbol{\alpha_0}^N\} = \emptyset$ and
– $\{\boldsymbol{\alpha_0}^N\} \cap \mathrm{FTV}(\boldsymbol{S}^I) = \emptyset$

without loss of generality, we have

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N/\boldsymbol{\alpha_0}^N]} h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Thus, T_Handling derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \mathbf{handle}_{l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N}\, e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\, \mathbf{with}\, h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Case H_Return**:** For some x and e_r, the following are given:
– $h = \{\mathbf{return}\, y \mapsto e_r\}$,
– $\sigma = \{\}$, and
– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, x : A \vdash e_r : B \mid \varepsilon$.

By the induction hypothesis, we have
– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e_r[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, H_Return derives

$$\Gamma_1, \Gamma_2 \vdash_{\{\}} \{\mathbf{return}\, x \mapsto e_r[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\} : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Case H_Op**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ such that:
– $\{\boldsymbol{\beta}^J\} \cap \{\boldsymbol{\alpha}^I\} = \emptyset$ and

– $\{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(\boldsymbol{S}^I) = \emptyset$.

For some h', σ', op, A', B', and e, the following are given:

– $h = h' \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\ p\ k \mapsto e\}$,

– $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B'\}$,

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_{\sigma'} h' : A \Rightarrow^{\varepsilon} B$, and

– $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon} B \vdash e : B \mid \varepsilon$.

By the induction hypothesis and Definition 1.10, we have

– $\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \sigma'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow B'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\}$,

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and

– $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], k : B'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \rightarrow_{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, H_Op derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \uplus \{\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\ p\ k \mapsto e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\} : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

■

Lemma 3.11 (Well-kinded of Subtyping)**.**

- *If* $\Gamma \vdash A <: B$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash B : \mathbf{Typ}$.
- *If* $\Gamma \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon$*, then* $\Gamma \vdash A_i : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon_i : \mathbf{Eff}$ *for* $i \in \{1, 2\}$.

Proof. Straightforward by mutual induction on the subtyping derivations with Lemma 3.6. ■

Lemma 3.12 (Well-kinded of Typing)**.**

(1) If $\Gamma \vdash e : A \mid \varepsilon$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

(2) If $\Gamma \vdash_\sigma h : A \Rightarrow^{\varepsilon} B$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash B : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Proof. By mutual induction on derivations of the judgments. We proceed by cases on the typing rule applied lastly to the derivation.

Case T_Var**:** We are given $\varepsilon = \mathbb{0}$ and $\vdash \Gamma$ and $\Gamma = \Gamma_1, x : A, \Gamma_2$ for some x, Γ_1, and Γ_2. Because $\vdash \Gamma$, it is easy to prove that $\Gamma_1 \vdash A : \mathbf{Typ}$ using Lemma 3.1. Then, by Lemma 3.5, $\Gamma_1, x : A, \Gamma_2 \vdash A : \mathbf{Typ}$. We also have $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$ because $\mathbb{0}$ is well-formedness-preserving.

Case T_Abs**:** For some f, x, e', B, C, and ε', the following are given:

- $e = \mathbf{fun}\,(f, x, e')$,
- $A = B \rightarrow_{\varepsilon'} C$,
- $\varepsilon = \mathbb{0}$, and
- $\Gamma, f : B \rightarrow_{\varepsilon'} C, x : B \vdash e' : C \mid \varepsilon'$.

Since $\mathbb{0}$ is well-formedness-preserving, we have $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$. By the induction hypothesis, we have $\Gamma, f : B \rightarrow_{\varepsilon'} C, x : B \vdash C : \mathbf{Typ}$. By Lemma 3.1, we have $\vdash \Gamma, f : B \rightarrow_{\varepsilon'} C, x : B$. Since only C_Var can derive $\vdash \Gamma, f : B \rightarrow_{\varepsilon'} C, x : B$, we have $\Gamma, f : B \rightarrow_{\varepsilon'} C \vdash B : \mathbf{Typ}$. By Lemma 3.1, we have $\vdash \Gamma, f : B \rightarrow_{\varepsilon'} C$. Since only C_Var can derive $\vdash \Gamma, f : B \rightarrow_{\varepsilon'} C$, we have $\Gamma \vdash B \rightarrow_{\varepsilon'} C : \mathbf{Typ}$.

Case T_App**:** For some v_1, v_2, and B, the following are given:

- $e = v_1\ v_2$,
- $\Gamma \vdash v_1 : B \rightarrow_{\varepsilon} A \mid \mathbb{0}$, and
- $\Gamma \vdash v_2 : B \mid \mathbb{0}$.

By the induction hypothesis, we have $\Gamma \vdash B \rightarrow_{\varepsilon} A : \mathbf{Typ}$ and $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$. Since only K_Fun can derive $\Gamma \vdash B \rightarrow_{\varepsilon} A : \mathbf{Typ}$, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$ as required.

Case T_TAbs**:** For some α, K, e', B, and ε', the following are given:

- $e = \Lambda\alpha : K.e'$,
- $A = \forall \alpha : K.B^{\varepsilon'}$,
- $\varepsilon = \mathbb{0}$, and

- $\Gamma, \alpha : K \vdash e' : B \mid \varepsilon'$.

Since $\mathbb{0}$ is well-formedness-preserving, we have $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$. By the induction hypothesis, we have $\Gamma, \alpha : K \vdash B : \mathbf{Typ}$ and $\Gamma, \alpha : K \vdash \varepsilon' : \mathbf{Eff}$. Thus, K_Poly derives $\Gamma \vdash \forall \alpha : K.B^{\varepsilon'} : \mathbf{Typ}$.

Case T_TApp**:** For some v, S, A', ε', α, and K, the following are given:

- $e = v\, S$,
- $A = A'[S/\alpha]$,
- $\varepsilon = \varepsilon'[S/\alpha]$,
- $\Gamma \vdash v : \forall \alpha : K.A'^{\varepsilon'} \mid \mathbb{0}$, and
- $\Gamma \vdash S : K$.

By the induction hypothesis, we have $\Gamma \vdash \forall \alpha : K.A'^{\varepsilon'} : \mathbf{Typ}$. Since only K_Poly can derive $\Gamma \vdash \forall \alpha : K.A'^{\varepsilon'} : \mathbf{Typ}$, we have $\Gamma, \alpha : K \vdash A' : \mathbf{Typ}$ and $\Gamma, \alpha : K \vdash \varepsilon' : \mathbf{Eff}$. By Lemma 3.10(2), we have $\Gamma \vdash A'[S/\alpha] : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon'[S/\alpha] : \mathbf{Eff}$ as required.

Case T_Let**:** For some x, e_1, e_2, and B, the following are given:

- $e = (\mathbf{let}\, x = e_1 \,\mathbf{in}\, e_2)$,
- $\Gamma \vdash e_1 : B \mid \varepsilon$, and
- $\Gamma, x : B \vdash e_2 : A \mid \varepsilon$.

By the induction hypothesis, we have $\Gamma, x : B \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$. By $\Delta(\Gamma, x : B) = \Delta(\Gamma)$ and Lemma 3.2(2) and Lemma 3.6, we have $\Gamma \vdash A : \mathbf{Typ}$ as required.

Case T_Sub**:** For some A' and ε', the following are given:

- $\Gamma \vdash e : A' \mid \varepsilon'$ and
- $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By Lemma 3.11, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Typ}$.

Case T_Op**:** For some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, σ, $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, $\boldsymbol{\beta}^J$, $\boldsymbol{K'}^J$, A', B', the following are given:

- $e = \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J$,
- $A = (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \to_{(l\, \boldsymbol{S}^I)^\uparrow} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J])$,
- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J.A' \Rightarrow B' \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$, and
- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$.

Since $\mathbb{0}$ is well-formedness-preserving, we have $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$. Without loss of generality, we can assume that $\boldsymbol{\alpha}^I$ and $\boldsymbol{\beta}^J$ do not occur in Γ. Then, because there exist some A'' and B'' such that

- $\boldsymbol{\alpha}^I : \boldsymbol{K}^I, \boldsymbol{\beta}^J : \boldsymbol{K'}^J \vdash A'' : \mathbf{Typ}$,
- $\boldsymbol{\alpha}^I : \boldsymbol{K}^I, \boldsymbol{\beta}^J : \boldsymbol{K'}^J \vdash B'' : \mathbf{Typ}$,
- $A''[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = A'$, and
- $B''[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = B'$,

Lemma 3.5 and 3.10(2) imply $\Gamma \vdash A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J] : \mathbf{Typ}$ and $\Gamma \vdash B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J] : \mathbf{Typ}$. Thus, K_Fun derives $\Gamma \vdash (A'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) \to_{(l\, \boldsymbol{S}^I)^\uparrow} (B'[\boldsymbol{T}^J/\boldsymbol{\beta}^J]) : \mathbf{Typ}$.

Case T_Handling**:** For some A', σ, N, $\boldsymbol{\alpha}^N$, and $\boldsymbol{S}^N$, we have

$$\Gamma \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A' \Rightarrow^{\varepsilon} A.$$

By the induction hypothesis, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Case H_Return**:** For some x and e_r, we have

$$\Gamma, x : A \vdash e_r : B \mid \varepsilon.$$

By the induction hypothesis, we have

- $\Gamma, x : A \vdash B : \mathbf{Typ}$ and
- $\Gamma, x : A \vdash \varepsilon : \mathbf{Eff}$.

By Lemma 3.2(2), we have

- $\Delta(\Gamma) \vdash B : \mathbf{Typ}$ and
- $\Delta(\Gamma) \vdash \varepsilon : \mathbf{Eff}$.

By Lemma 3.6, we have

- $\Gamma \vdash B : \mathbf{Typ}$ and
- $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Now, we have $\vdash \Gamma, x : A$ by Lemma 3.9. Since only C_VAR can derive $\vdash \Gamma, x : A$, we have $\Gamma \vdash A : \mathbf{Typ}$.

***Case* H_OP:** For some h' and σ', we have $\Gamma \vdash_{\sigma'} h' : A \Rightarrow^{\varepsilon} B$. By the induction hypothesis, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash B : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

■

Lemma 3.13 (Inversion of Subtyping)**.**

(1) If $\Gamma \vdash C <: A_1 \rightarrow_{\varepsilon_1} B_1$ and $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$, then $C = A_2 \rightarrow_{\varepsilon_2} B_2$ such that $\Gamma \vdash A_1 <: A_2$, $\Gamma \vdash B_2 <: B_1$, and $\Gamma \vdash \varepsilon_2 \otimes \varepsilon_1$.

(2) If $\Gamma \vdash C <: \forall \alpha : K.A_1{}^{\varepsilon_1}$ and $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$, then $C = \forall \alpha : K.A_2{}^{\varepsilon_2}$ such that $\Gamma, \alpha : K \vdash A_2 <: A_1$ and $\Gamma, \alpha : K \vdash \varepsilon_2 \otimes \varepsilon_1$.

Proof.

(1) By induction on a derivation of $\Gamma \vdash C <: A_1 \rightarrow_{\varepsilon_1} B_1$. We proceed by case analysis on the subtyping rule applied lastly to this derivation.

***Case* ST_REFL:** $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 : \mathbf{Typ}$ and $C = A_1 \rightarrow_{\varepsilon_1} B_1$ are given. Because only K_FUN can derive $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 : \mathbf{Typ}$, we have $\Gamma \vdash A_1 : \mathbf{Typ}$, $\Gamma \vdash \varepsilon_1 : \mathbf{Eff}$, and $\Gamma \vdash B_1 : \mathbf{Typ}$. By ST_REFL, $\Gamma \vdash A_1 <: A_1$ and $\Gamma \vdash B_1 <: B_1$ hold. By Lemma 3.3(1), $\Gamma \vdash \varepsilon_1 \otimes \varepsilon_1$ holds.

***Case* ST_FUN:** Clearly.

***Case* others:** Cannot happen.

(2) By induction on a derivation of $\Gamma \vdash C <: \forall \alpha : K.A_1{}^{\varepsilon_1}$. We proceed by case analysis on the subtyping rule applied lastly to this derivation.

***Case* ST_REFL:** $\Gamma \vdash \forall \alpha : K.A_1{}^{\varepsilon_1} : \mathbf{Typ}$ and $C = \forall \alpha : K.A_1{}^{\varepsilon_1}$ are given. Because only K_POLY can derive $\Gamma \vdash \forall \alpha : K.A_1{}^{\varepsilon_1} : \mathbf{Typ}$, we have $\Gamma, \alpha : K \vdash A_1 : \mathbf{Typ}$ and $\Gamma, \alpha : K \vdash \varepsilon_1 : \mathbf{Eff}$. By ST_REFL, $\Gamma, \alpha : K \vdash A_1 <: A_1$ holds. By Lemma 3.3(1), $\Gamma, \alpha : K \vdash \varepsilon_1 \otimes \varepsilon_1$.

***Case* ST_POLY:** Clearly.

***Case* others:** Cannot happen.

■

Lemma 3.14 (Inversion)**.**

(1) If $\Gamma \vdash v : A \mid \varepsilon$, then $\Gamma \vdash v : A \mid \mathbb{0}$.

(2) If $\Gamma \vdash \mathbf{fun}\,(f, x, e) : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$, then $\Gamma, f : A_2 \rightarrow_{\varepsilon_2} B_2, x : A_2 \vdash e : B_2 \mid \varepsilon_2$ for some A_2, ε_2, and B_2 such that $\Gamma \vdash A_2 \rightarrow_{\varepsilon_2} B_2 <: A_1 \rightarrow_{\varepsilon_1} B_1$.

(3) If $\Gamma \vdash \Lambda \alpha : K.e : \forall \alpha : K.A_1{}^{\varepsilon_1} \mid \varepsilon$, then $\Gamma, \alpha : K \vdash e : A_1 \mid \varepsilon_1$.

(4) If $\Gamma \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$, then the following hold:

- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma[\boldsymbol{S}^I / \boldsymbol{\alpha}^I]$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$,
- $\Gamma \vdash A_1 <: A[\boldsymbol{T}^J / \boldsymbol{\beta}^J]$,
- $\Gamma \vdash B[\boldsymbol{T}^J / \boldsymbol{\beta}^J] <: B_1$, *and*

- $\Gamma \vdash (l\, \boldsymbol{S}^I)^\uparrow \olessthan \varepsilon_1$

for some $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, σ, $\boldsymbol{\beta}^J$, $\boldsymbol{K'}^J$, A, *and* B.

(5) *If* $\Gamma \vdash v_1\, v_2 : B \mid \varepsilon$*, then there exists some type* A *such that* $\Gamma \vdash v_1 : A \rightarrow_\varepsilon B \mid \mathbb{0}$ *and* $\Gamma \vdash v_2 : A \mid \mathbb{0}$.

Proof.

(1) By induction on a derivation of $\Gamma \vdash v : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case **T_Var:** Clearly because of $\varepsilon = \mathbb{0}$.

Case **T_Abs:** Clearly because of $\varepsilon = \mathbb{0}$.

Case **T_TAbs:** Clearly because of $\varepsilon = \mathbb{0}$.

Case **T_Op:** Clearly because of $\varepsilon = \mathbb{0}$.

Case **T_Sub:** For some A' and ε', the following are given:

- $\Gamma \vdash v : A' \mid \varepsilon'$ and
- $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis, $\Gamma \vdash v : A' \mid \mathbb{0}$. Since only ST_Comp derives $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\Gamma \vdash A' <: A$ and $\Gamma \vdash \varepsilon' \olessthan \varepsilon$. Because of Lemma 3.3(1), $\Gamma \vdash \mathbb{0} \olessthan \mathbb{0}$ holds. By T_Sub, we have $\Gamma \vdash v : A \mid \mathbb{0}$ as required.

Case **others:** Cannot happen.

(2) By induction on a derivation of $\Gamma \vdash \mathbf{fun}\,(f, x, e) : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case **T_Abs:** $\Gamma, f : A_1 \rightarrow_{\varepsilon_1} B_1, x : A_1 \vdash e : B_1 \mid \varepsilon_1$ is given. By Lemma 3.12, we have $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 : \mathbf{Typ}$. Thus, ST_Refl derives $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 <: A_1 \rightarrow_{\varepsilon_1} B_1$.

Case **T_Sub:** For some C and ε', the following are given:

- $\Gamma \vdash \mathbf{fun}\,(f, x, e) : C \mid \varepsilon'$ and
- $\Gamma \vdash C \mid \varepsilon' <: A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$.

Since only ST_Comp derives $\Gamma \vdash C \mid \varepsilon' <: A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$, we have $\Gamma \vdash C <: A_1 \rightarrow_{\varepsilon_1} B_1$. By Lemma 3.13(1), $C = A_2 \rightarrow_{\varepsilon_2} B_2$ for some A_2, ε_2, and B_2. By the induction hypothesis and Lemma 3.4, the required results are achieved.

Case **others:** Cannot happen.

(3) By induction on a derivation of $\Gamma \vdash \Lambda\alpha : K.e : \forall \alpha : K.{A_1}^{\varepsilon_1} \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case **T_TAbs:** Clearly.

Case **T_Sub:** For some B and ε', the following are given:

- $\Gamma \vdash \Lambda\alpha : K.e : B \mid \varepsilon'$ and
- $\Gamma \vdash B \mid \varepsilon' <: \forall \alpha : K.{A_1}^{\varepsilon_1} \mid \varepsilon$.

Since only ST_Comp derives $\Gamma \vdash B \mid \varepsilon' <: \forall \alpha : K.{A_1}^{\varepsilon_1} \mid \varepsilon$, we have $\Gamma \vdash B <: \forall \alpha : K.{A_1}^{\varepsilon_1}$. By Lemma 3.13(2), we have $B = \forall \alpha : K.{A_2}^{\varepsilon_2}$ for some A_2 and ε_2 such that

- $\Gamma, \alpha : K \vdash A_2 <: A_1$ and
- $\Gamma, \alpha : K \vdash \varepsilon_2 \olessthan \varepsilon_1$.

By the induction hypothesis, we have $\Gamma, \alpha : K \vdash e : A_2 \mid \varepsilon_2$. Thus, T_Sub derives $\Gamma, \alpha : K \vdash e : A_1 \mid \varepsilon_1$, because ST_Comp derives $\Gamma, \alpha : K \vdash A_2 \mid \varepsilon_2 <: A_1 \mid \varepsilon_1$.

Case **others:** Cannot happen.

(4) By induction on a derivation of $\Gamma \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case **T_Op:** For some $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, σ, $\boldsymbol{\beta}^J$, $\boldsymbol{K'}^J$, A, and B, the following are given:

- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma[\boldsymbol{S}^I / \boldsymbol{\alpha}^I]$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,

- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$,
- $A_1 = A[\boldsymbol{T}^J/\boldsymbol{\beta}^J]$,
- $B_1 = B[\boldsymbol{T}^J/\boldsymbol{\beta}^J]$, and
- $\varepsilon_1 = (l\, \boldsymbol{S}^I)^\uparrow$.

By Lemma 3.12, we have $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 : \mathbf{Typ}$. Since only K_Fun can derive $\Gamma \vdash A_1 \rightarrow_{\varepsilon_1} B_1 : \mathbf{Typ}$, we have

- $\Gamma \vdash A[\boldsymbol{T}^J/\boldsymbol{\beta}^J] : \mathbf{Typ}$,
- $\Gamma \vdash (l\, \boldsymbol{S}^I)^\uparrow : \mathbf{Eff}$, and
- $\Gamma \vdash B[\boldsymbol{T}^J/\boldsymbol{\beta}^J] : \mathbf{Typ}$.

Thus, the required results are achieved by ST_Refl and Lemma 3.3(1).

Case T_Sub**:** For some C and ε', the following are given:

- $\Gamma \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J : C \mid \varepsilon'$ and
- $\Gamma \vdash C \mid \varepsilon' <: A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$.

Since only ST_Comp can derive $\Gamma \vdash C \mid \varepsilon' <: A_1 \rightarrow_{\varepsilon_1} B_1 \mid \varepsilon$, we have $\Gamma \vdash C <: A_1 \rightarrow_{\varepsilon_1} B_1$. By Lemma 3.13(1), we have $C = A_2 \rightarrow_{\varepsilon_2} B_2$ such that

- $\Gamma \vdash A_1 <: A_2$,
- $\Gamma \vdash B_2 <: B_1$, and
- $\Gamma \vdash \varepsilon_2 \otimes \varepsilon_1$.

By the induction hypothesis,

- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$,
- $\Gamma \vdash A_2 <: A[\boldsymbol{T}^J/\boldsymbol{\beta}^J]$,
- $\Gamma \vdash B[\boldsymbol{T}^J/\boldsymbol{\beta}^J] <: B_2$, and
- $\Gamma \vdash (l\, \boldsymbol{S}^I)^\uparrow \otimes \varepsilon_2$.

By Lemma 3.4 and Lemma 3.3(2), the required result is achieved.

***Case* others:** Cannot happen.

(5) By induction on a derivation of $\Gamma \vdash v_1\, v_2 : B \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_App**:** Clearly.

Case T_Sub**:** For some B' and ε', the following are given:

- $\Gamma \vdash v_1\, v_2 : B' \mid \varepsilon'$ and
- $\Gamma \vdash B' \mid \varepsilon' <: B \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma \vdash v_1 : A \rightarrow_{\varepsilon'} B' \mid \mathbb{0}$ and
- $\Gamma \vdash v_2 : A \mid \mathbb{0}$

for some A. By Lemma 3.12, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \mathbb{0} : \mathbf{Eff}$. Thus, ST_Refl derives $\Gamma \vdash A <: A$ and Lemma 3.3(1) derives $\Gamma \vdash \mathbb{0} \otimes \mathbb{0}$. Therefore, by ST_Fun and ST_Comp, $\Gamma \vdash A \rightarrow_{\varepsilon'} B' \mid \mathbb{0} <: A \rightarrow_\varepsilon B \mid \mathbb{0}$. Then, by T_Sub, $\Gamma \vdash v_1 : A \rightarrow_\varepsilon B \mid \mathbb{0}$.

***Case* others:** Cannot happen.

■

Lemma 3.15 (Canonical Form)**.**

(1) If $\emptyset \vdash v : A \rightarrow_\varepsilon B \mid \varepsilon'$, *then either of the following holds:*

- $v = \mathbf{fun}\,(f, x, e)$ *for some* f, x, *and* e, *or*
- $v = \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J$ *for some* op, l, $\boldsymbol{S}^I$, *and* $\boldsymbol{T}^J$.

(2) If $\emptyset \vdash v : \forall \alpha : K.A^\varepsilon \mid \varepsilon'$, *then* $v = \Lambda \alpha : K.e$ *for some* e.

Proof.

(1) By induction on a derivation of $\Gamma \vdash v : A \rightarrow_{\varepsilon} B \mid \varepsilon'$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_VAR**:** Cannot happen.

Case T_ABS**:** Clearly.

Case T_SUB**:** For some C, the following are given:

- $\Gamma \vdash v : C \mid \varepsilon''$ and
- $\Gamma \vdash C \mid \varepsilon'' <: A \rightarrow_{\varepsilon} B \mid \varepsilon'$.

By Lemma 3.13(1), we have $C = A_1 \rightarrow_{\varepsilon_1} B_1$ for some A_1, ε_1, and B_1. By the induction hypothesis, the required result is achieved.

Case T_OP**:** Clearly.

***Case* others:** Cannot happen.

(2) By induction on a derivation of $\Gamma \vdash v : \forall \alpha : K.A^{\varepsilon} \mid \varepsilon'$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_VAR**:** Cannot happen.

Case T_TABS**:** Clearly.

Case T_SUB**:** For some B, the following are given:

- $\Gamma \vdash v : B \mid \varepsilon''$ and
- $\Gamma \vdash B \mid \varepsilon'' <: \forall \alpha : K.A^{\varepsilon} \mid \varepsilon'$.

By Lemma 3.13(2), we have $B = \forall \alpha : K.{A_1}^{\varepsilon_1}$ for some A_1 and ε_1. By the induction hypothesis, the required result is achieved.

***Case* others:** Cannot happen.

■

Lemma 3.16 (Inversion of Handler Typing)**.**

(1) If $\Gamma \vdash_{\sigma} h : A \Rightarrow^{\varepsilon} B$, then there exist some x and e_r such that **return** $x \mapsto e_r \in h$ *and* $\Gamma, x : A \vdash e_r : B \mid \varepsilon$.

(2) If $\Gamma \vdash_{\sigma} h : A \Rightarrow^{\varepsilon} B$ and $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B' \in \sigma$, then

- $\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e \in h$ *and*
- $\Gamma, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon} B \vdash e : B \mid \varepsilon$

for some p, k, and e.

Proof. (1) By induction on a derivation of $\Gamma \vdash_{\sigma} h : A \Rightarrow^{\varepsilon} B$. We proceed by cases on the typing rule applied lastly to this derivation.

Case H_RETURN**:** Clearly.

Case H_OP**:** Clearly by the induction hypothesis.

(2) By induction on a derivation of $\Gamma \vdash_{\sigma} h : A \Rightarrow^{\varepsilon} B$. We proceed by cases on the typing rule applied lastly to this derivation.

Case H_RETURN**:** Clearly because there is no operation belonging to $\{\}$.

Case H_OP**:** For some h', σ', op', $\boldsymbol{\beta'}^{J'}$, $\boldsymbol{K'}^{J'}$, A'', B'', p', k', e'', the following are given:

- $h = h' \uplus \{\mathsf{op}'\, \boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}\, p'\, k' \mapsto e'\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op}' : \forall \boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}.A'' \Rightarrow B''\}$,
- $\Gamma \vdash_{\sigma'} h' : A \Rightarrow^{\varepsilon} B$, and
- $\Gamma, \boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}, p' : A'', k' : B'' \rightarrow_{\varepsilon} B \vdash e : B \mid \varepsilon$.

If $\mathsf{op} = \mathsf{op}'$, then clearly.

If $\mathsf{op} \neq \mathsf{op}'$, then clearly by the induction hypothesis.

■

Lemma 3.17 (Independence of Evaluation Contexts)**.** *If $\Gamma \vdash E[e] : A \mid \varepsilon$, then there exist some A' and ε' such that*

- $\Gamma \vdash e : A' \mid \varepsilon'$, *and*

- $\Gamma, \Gamma' \vdash E[e'] : A \mid \varepsilon$ *holds for any* e' *and* Γ' *such that* $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$.

Proof. By induction on a derivation of $\Gamma \vdash E[e] : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_Let**:** If $E = \square$, then the required result is achieved immediately.

If $E \neq \square$, then we have

- $E = (\textbf{let}\, x = E' \,\textbf{in}\, e_2)$,
- $\Gamma \vdash E'[e] : B \mid \varepsilon$, and
- $\Gamma, x : B \vdash e_2 : A \mid \varepsilon$,

for some x, E', e_2, and B. By the induction hypothesis, there exist some A' and ε' such that

- $\Gamma \vdash e : A' \mid \varepsilon'$, and
- for any e' and Γ' such that $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$, typing judgment $\Gamma, \Gamma' \vdash E'[e'] : B \mid \varepsilon$ is derivable.

Let e' be an expression and Γ' be a typing context such that $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$. Without loss of generality, we can assume $x \notin \mathrm{dom}(\Gamma')$. The induction hypothesis result implies $\Gamma, \Gamma' \vdash E'[e'] : B \mid \varepsilon$. By Lemma 3.5 and T_Let, it suffices to show that $\vdash \Gamma, \Gamma'$, which is implied by Lemma 3.9.

Case T_Sub**:** For some A' and ε', given are the following:

- $\Gamma \vdash E[e] : A' \mid \varepsilon'$ and
- $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis, there exist some A'' and ε'' such that

- $\Gamma \vdash e : A'' \mid \varepsilon''$, and
- for any e' and Γ' such that $\Gamma, \Gamma' \vdash e' : A'' \mid \varepsilon''$, typing judgment $\Gamma, \Gamma' \vdash E[e'] : A' \mid \varepsilon'$ is derivable.

Let e' be an expression and Γ' be a typing context such that $\Gamma, \Gamma' \vdash e' : A'' \mid \varepsilon''$. Since only ST_Comp can derive $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\Gamma \vdash A' <: A$ and $\Gamma \vdash \varepsilon' \otimes \varepsilon$. We have $\Gamma, \Gamma' \vdash A' <: A$ and $\Gamma, \Gamma' \vdash \varepsilon' \otimes \varepsilon$ by Lemma 3.9, Lemma 3.5(2), and Lemma 3.5(3). Thus, because $\Gamma, \Gamma' \vdash E[e'] : A' \mid \varepsilon'$ by the induction hypothesis result, ST_Comp and T_Sub derive $\Gamma, \Gamma' \vdash E[e'] : A \mid \varepsilon$.

Case T_Handling**:** If $E = \square$, then the required result is achieved immediately.

If $E \neq \square$, then we have

- $E = \textbf{handle}_{l\, \boldsymbol{S}^N}\, E' \,\textbf{with}\, h$,
- $\Gamma \vdash E'[e] : A' \mid \varepsilon'$, and
- $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$,

for some l, $\boldsymbol{S}^N$, E', h, A', and ε'. By the induction hypothesis, there exist some A'' and ε'' such that

- $\Gamma \vdash e : A'' \mid \varepsilon''$, and
- for any e' and Γ' such that $\Gamma, \Gamma' \vdash e' : A'' \mid \varepsilon''$, typing judgment $\Gamma, \Gamma' \vdash E'[e'] : A' \mid \varepsilon'$ is derivable.

Because the premises of T_Handling other than the typing of handled expressions are independent of the handled expressions, the required result is achieved by Lemma 3.9, Lemma 3.5, and Lemma 3.2(2).

Case **others:** Clearly because $E = \square$.

■

Lemma 3.18 (Progress)**.** *If* $\emptyset \vdash e : A \mid \varepsilon$*, then one of the following holds:*

- e *is a value;*
- *There exists some expression* e' *such that* $e \longrightarrow e'$*; or*
- *There exist some* op*,* l*,* $\boldsymbol{S}^I$*,* $\boldsymbol{T}^J$*,* v*,* E*, and* n *such that* $e = E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v]$ *and* $n\mathrm{-free}(l\, \boldsymbol{S}^I, E)$*.*

Proof. By induction on a derivation of $\emptyset \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_Var**:** Cannot happen.

Case T_Abs**:** e is a value because of $e = \textbf{fun}\,(f_1, x_1, e_1)$ for some f_1, x_1, and e_1.

Case T_App**:** For some v_1, v_2, and B, the following are given:

- $e = v_1\, v_2$,
- $\emptyset \vdash v_1 : B \rightarrow_\varepsilon A \mid \mathbb{0}$, and
- $\emptyset \vdash v_2 : B \mid \mathbb{0}$.

By case analysis on the result of Lemma 3.15(1) on $\emptyset \vdash v_1 : B \rightarrow_\varepsilon A \mid \mathbb{0}$.

If $v_1 = \mathbf{fun}\,(f_1, x_1, e_1)$ for some f_1, x_1, and e_1, then R_App derives $e \longmapsto e_1[\mathbf{fun}\,(f_1, x_1, e_1)/f_1][v_2/x]$.

If $v_1 = \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J$ for some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, then the required result is implied by Lemma 3.14(4) and the fact that $e = \Box[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\, v_2]$.

Case **T_TAbs:** e is a value because of $e = \Lambda\alpha : K.e_1$ for some α, K, and e_1.

Case **T_TApp:** For some v, α, S, K, A_1, and ε_1, the following are given:

- $e = v\, S$,
- $A = A_1[S/\alpha]$,
- $\varepsilon = \varepsilon_1[S/\alpha]$,
- $\emptyset \vdash v : \forall\alpha : K.A_1{}^{\varepsilon_1} \mid \mathbb{0}$, and
- $\emptyset \vdash S : K$.

By Lemma 3.15(2), we have $v = \Lambda\alpha : K.e_1$ for some e_1. Thus, R_TApp derives $e \longmapsto e_1[S/\alpha]$.

Case **T_Let:** For some x, e_1, e_2, and B, given are the following:

- $e = (\mathbf{let}\, x = e_1\, \mathbf{in}\, e_2)$,
- $\emptyset \vdash e_1 : B \mid \varepsilon$, and
- $x : B \vdash e_2 : A \mid \varepsilon$.

By the induction hypothesis, we proceed by cases on the following conditions:

(1) e_1 is a value,

(2) There exists some e_1' such that $e_1 \longrightarrow e_1'$,

(3) There exist some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, and n such that $e_1 = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\, v]$ and $n\text{--free}(l\,\boldsymbol{S}^I, E)$.

Case **(1):** R_Let derives $e \longmapsto e_2[v_1/x]$ because e_1 is a value v_1.

Case **(2):** Since only E_Eval can derive $e_1 \longrightarrow e_1'$, we have

- $e_1 = E_1[e_{11}]$,
- $e_1' = E_1[e_{12}]$, and
- $e_{11} \longmapsto e_{12}$,

for some E_1, e_{11}, and e_{12}. Let $E = (\mathbf{let}\, x = E_1\, \mathbf{in}\, e_2)$. E_Eval derives $e \longrightarrow E[e_{12}]$ because of $e = E[e_{11}]$.

Case **(3):** Clearly because $e = (\mathbf{let}\, x = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\, v]\, \mathbf{in}\, e_2)$ and $n\text{--free}(l\,\boldsymbol{S}^I, \mathbf{let}\, x = E\, \mathbf{in}\, e_2)$.

Case **T_Sub:** Clearly by the induction hypothesis.

Case **T_Op:** e is a value because of $e = \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J$ for some op, l, $\boldsymbol{S}^I$, and $\boldsymbol{T}^J$.

Case **T_Handling:** For some e_1, h, l, $\boldsymbol{S}^N$, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, A_1, and ε_1, given are the following:

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e_1\, \mathbf{with}\, h$,
- $\emptyset \vdash e_1 : A_1 \mid \varepsilon_1$,
- $l :: \forall\boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A_1 \Rightarrow^\varepsilon A$, and
- $(l\,\boldsymbol{S}^N)^\uparrow \odot \varepsilon \sim \varepsilon_1$.

By the induction hypothesis, we proceed by cases on the following conditions:

(1) e_1 is a value,

(2) There exists some e_1' such that $e_1 \longrightarrow e_1'$,

(3) There exist some op', l', $\boldsymbol{S'}^{N'}$, $\boldsymbol{T}^J$, v, E, and n such that $e_1 = E[\mathsf{op}'_{l'\,\boldsymbol{S'}^{N'}}\,\boldsymbol{T}^J\, v]$ and $n\text{--free}(l'\,\boldsymbol{S'}^{N'}, E)$.

***Case* (1):** By Lemma 3.16(1), there exists some x and e_r such that $\mathbf{return}\, x \mapsto e_r \in h$. Thus, R_HANDLE1 derives $e \longmapsto e_r[v_1/x]$ because e_1 is a value v_1.

***Case* (2):** Since only E_EVAL can derive $e_1 \longrightarrow e_1'$, we have

- $e_1 = E_1[e_{11}]$,
- $e_1' = E_1[e_{12}]$, and
- $e_{11} \longmapsto e_{12}$,

for some E, e_{11}, and e_{12}. Let $E = \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E_1\, \mathbf{with}\, h$. E_EVAL derives $e \longrightarrow E[e_{12}]$ because of $e = E[e_{11}]$.

***Case* (3):** If $l\, \boldsymbol{S}^N \neq l'\, \boldsymbol{S'}^{N'}$, then $e = (\mathbf{handle}_{l\, \boldsymbol{S}^N}\, E\, \mathbf{with}\, h)[\mathsf{op}_{l'\, \boldsymbol{S'}^{N'}}\, \boldsymbol{T}^J\, v]$ and n−free$(l'\, \boldsymbol{S'}^{N'}, \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E\, \mathbf{with}\, h)$.

If $l\, \boldsymbol{S}^N = l'\, \boldsymbol{S'}^{N'}$, then by Lemma 3.17 and 3.14(4), we have

- $l' :: \forall \boldsymbol{\alpha'}^{N'} : \boldsymbol{K'}^{N'}.\sigma' \in \Xi$ and
- $\mathsf{op}' : \forall \boldsymbol{\beta'}^J : \boldsymbol{K_0'}^J.A' \Rightarrow B' \in \sigma'[\boldsymbol{S'}^{N'}/\boldsymbol{\alpha'}^{N'}]$,

for some $\boldsymbol{\alpha'}^{N'}$, $\boldsymbol{K'}^{N'}$, σ', $\boldsymbol{\beta'}^J$, A', and B'. Therefore, since $l\, \boldsymbol{S}^N = l'\, \boldsymbol{S'}^{N'}$, we have

- $\sigma = \sigma'$,
- $\boldsymbol{\alpha}^N = \boldsymbol{\alpha'}^{N'}$, and
- $\boldsymbol{K}^N = \boldsymbol{K_0}^{N'}$.

By $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A_1 \Rightarrow^\varepsilon A$ and $\mathsf{op}' : \forall \boldsymbol{\beta'}^J : \boldsymbol{K_0'}^J.A' \Rightarrow B' \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$ and Lemma 3.16(2), we have

$$\mathsf{op}'\, \boldsymbol{\beta'}^J : \boldsymbol{K_0'}^J\, p\, k \mapsto e' \in h$$

for some p, k, and e'. If $n = 0$, the evaluation of e proceeds by R_HANDLE2. Otherwise, there exists some m such that $n = m + 1$ and m−free$(l\, \boldsymbol{S}^N, \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E\, \mathbf{with}\, h)$.

■

Lemma 3.19 (Preservation in Reduction)**.** *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longmapsto e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. By induction on a derivation of $\Gamma \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

***Case* T_VAR:** There is no e' such that $e \longmapsto e'$.

***Case* T_ABS:** There is no e' such that $e \longmapsto e'$.

***Case* T_APP:** Since only R_APP can derive $e \longmapsto e'$, we have

- $e = (\mathbf{fun}\,(f_1, x_1, e_1))\, v_2$,
- $\emptyset \vdash \mathbf{fun}\,(f_1, x_1, e_1) : A_1 \rightarrow_\varepsilon A \mid \mathbb{0}$,
- $\emptyset \vdash v_2 : A_1 \mid \mathbb{0}$, and
- $e' = e_1[\mathbf{fun}\,(f_1, x_1, e_1)/f_1][v_2/x_1]$

for some f_1, x_1, e_1, v_2, and A_1. By Lemma 3.14(2), we have

- $f_1 : A_2 \rightarrow_{\varepsilon_2} B_2, x_1 : A_2 \vdash e_1 : B_2 \mid \varepsilon_2$ and
- $\emptyset \vdash A_2 \rightarrow_{\varepsilon_2} B_2 <: A_1 \rightarrow_\varepsilon A$.

for some A_2, ε_2, and B_2. Thus, T_ABS derives $\emptyset \vdash \mathbf{fun}\,(f_1, x_1, e_1) : A_2 \rightarrow_{\varepsilon_2} B_2 \mid \mathbb{0}$. By Lemma 3.13(1), we have

- $\emptyset \vdash A_1 <: A_2$,
- $\emptyset \vdash B_2 <: A$, and
- $\emptyset \vdash \varepsilon_2 \oslash \varepsilon$.

By Lemma 3.9 and Lemma 3.5(3), we have $f_1 : A_2 \rightarrow_{\varepsilon_2} B_2, x_1 : A_2 \vdash B_2 <: A$. Because Lemma 3.5(2), ST_COMP derives $f_1 : A_2 \rightarrow_{\varepsilon_2} B_2, x_1 : A_2 \vdash B_2 \mid \varepsilon_2 <: A \mid \varepsilon$. Therefore, T_SUB derives $f_1 : A_2 \rightarrow_{\varepsilon_2} B_2, x_1 : A_2 \vdash e_1 : A \mid \varepsilon$. Since T_SUB derives $\emptyset \vdash v_2 : A_2 \mid \mathbb{0}$, Lemma 3.7(5) makes $\emptyset \vdash e_1[\mathbf{fun}\,(f_1, x_1, e_1)/f_1][v_2/x_1] : A \mid \varepsilon$ hold as required.

Case T_TAbs: There is no e' such that $e \longmapsto e'$.

Case T_TApp: Since only R_TApp derives $e \longmapsto e'$, we have

- $e = (\Lambda\alpha : K.e_1)\, S$,
- $A = A_1[S/\alpha]$,
- $\varepsilon = \varepsilon_1[S/\alpha]$,
- $\emptyset \vdash \Lambda\alpha : K.e_1 : \forall\alpha : K.A_1{}^{\varepsilon_1} \mid \mathbb{0}$,
- $\emptyset \vdash S : K$, and
- $e' = e_1[S/\alpha]$

for some α, K, e_1, S, A_1, and ε_1. By Lemma 3.14(3), we have $\alpha : K \vdash e_1 : A_1 \mid \varepsilon_1$. Thus, Lemma 3.10(5) makes $\emptyset \vdash e_1[S/\alpha] : A_1[S/\alpha] \mid \varepsilon_1[S/\alpha]$ hold as required.

Case T_Let: Since only R_Let derives $e \longmapsto e'$, we have

- $e = (\mathbf{let}\, x = v\, \mathbf{in}\, e_1)$,
- $\emptyset \vdash v : B \mid \varepsilon$,
- $x : B \vdash e_1 : A \mid \varepsilon$, and
- $e' = e_1[v/x]$

for some x, v, e_1, and B. By Lemma 3.14(1) and Lemma 3.7(5), we have $\emptyset \vdash e_1[v/x] : A \mid \varepsilon$ as required.

Case T_Sub: For some A' and ε', we have

- $\emptyset \vdash e : A' \mid \varepsilon'$ and
- $\emptyset \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis, we have $\emptyset \vdash e' : A' \mid \varepsilon'$. Thus, T_Sub derives $\emptyset \vdash e' : A \mid \varepsilon$ as required.

Case T_Op: There is no e' such that $e \longmapsto e'$.

Case T_Handling: We proceed by cases on the derivation rule which derives $e \longmapsto e'$.

Case R_Handle1: We have

- $e = \mathbf{handle}_{l\, \boldsymbol{S}^I}\, v\, \mathbf{with}\, h$,
- $\mathbf{return}\, x \mapsto e_r \in h$,
- $\emptyset \vdash v : B \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h : B \Rightarrow^{\varepsilon} A$,
- $(l\, \boldsymbol{S}^I)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, and
- $e' = e_r[v/x]$

for some l, $\boldsymbol{S}^I$, $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, σ, v, h, B, and ε'. By $\emptyset \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h : B \Rightarrow^{\varepsilon} A$ and $\mathbf{return}\, x \mapsto e_r \in h$ and Lemma 3.16(1), we have

$$x : B \vdash e_r : A \mid \varepsilon.$$

By Lemma 3.14(1), we have $\emptyset \vdash v : B \mid \mathbb{0}$. Thus, Lemma 3.7(5) makes $\emptyset \vdash e_r[v/x] : A \mid \varepsilon$ hold as required.

Case R_Handle2: We have

- $e = \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[\mathsf{op}_{0\, l\, \boldsymbol{S}^N}\, \boldsymbol{T}^J\, v]\, \mathbf{with}\, h$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\mathsf{op}_0\, \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J\, p_0\, k_0 \mapsto e_0 \in h$,
- $0\text{−free}(l\, \boldsymbol{S}^N, E)$,
- $\emptyset \vdash E[\mathsf{op}_{0\, l\, \boldsymbol{S}^N}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$,
- $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, and

- $e' = e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\ E[z]\ \mathbf{with}\ h/k_0]$

for some l, $\boldsymbol{S}^N$, E, op_0, $\boldsymbol{T}^J$, v, h, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, σ, $\boldsymbol{\beta_0}^J$, $\boldsymbol{K_0}^J$, p_0, k_0, e_0, B, and ε'. By Lemma 3.17, there exist some B_1 and ε_1 such that

- $\emptyset \vdash \mathsf{op_0}_{l\,\boldsymbol{S}^N}\ \boldsymbol{T}^J\ v : B_1 \mid \varepsilon_1$, and
- for any e'' and Γ'', if $\Gamma'' \vdash e'' : B_1 \mid \varepsilon_1$, then $\Gamma'' \vdash E[e''] : B \mid \varepsilon'$.

By Lemma 3.14(5), we have $\emptyset \vdash \mathsf{op_0}_{l\,\boldsymbol{S}^N}\ \boldsymbol{T}^J : A_1 \to_{\varepsilon_1} B_1 \mid \mathbb{0}$ and $\emptyset \vdash v : A_1 \mid \mathbb{0}$ for some A_1. By Lemma 3.14(4) and 3.16(2), we have

- $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash \boldsymbol{T}^J : \boldsymbol{K_0}^J$,
- $\emptyset \vdash A_1 <: A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$,
- $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] <: B_1$, and
- $\emptyset \vdash (l\,\boldsymbol{S}^N)^{\uparrow} \otimes \varepsilon_1$.

for some A_0 and B_0. Thus, T_Sub with $\emptyset \vdash \mathbb{0} \otimes \mathbb{0}$ implied by Lemma 3.3 derives

$$\emptyset \vdash v : A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \mid \mathbb{0}.$$

By Lemma 3.11, we have $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] : \mathbf{Typ}$. Thus, C_Var derives $\vdash z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$. By $\emptyset \vdash \mathbb{0} : \mathbf{Eff}$, $\emptyset \vdash \varepsilon_1 : \mathbf{Eff}$ implied by Lemma 3.12, and $\mathbb{0} \odot \varepsilon_1 \sim \varepsilon_1$, we have $\emptyset \vdash \mathbb{0} \otimes \varepsilon_1$. Since T_Var and T_Sub derives $z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash z : B_1 \mid \varepsilon_1$, we have

$$z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\ E[z]\ \mathbf{with}\ h : A \mid \varepsilon$$

by the result of Lemma 3.17, Lemma 3.5, and T_Handling. Thus, T_Abs derives

$$\emptyset \vdash \lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\ E[z]\ \mathbf{with}\ h : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \to_{\varepsilon} A \mid \mathbb{0}.$$

Since

$$\boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J, p_0 : A_0, k_0 : B_0 \to_{\varepsilon} A \vdash e_0 : A \mid \varepsilon$$

by $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$ and $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$ and Lemma 3.16(2), Lemma 3.10(5) and Lemma 3.7(5) imply

$$\emptyset \vdash e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\ E[z]\ \mathbf{with}\ h/k_0] : A \mid \varepsilon$$

as required.

■

Lemma 3.20 (Preservation). *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longrightarrow e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. Since only E_Eval derives $e \longrightarrow e'$, we have

- $e = E[e_1]$,
- $e' = E[e_2]$, and
- $e_1 \longmapsto e_2$.

By Lemma 3.17, there exist some A' and ε' such that

- $\emptyset \vdash e_1 : A' \mid \varepsilon'$, and
- for any e_1' and Γ', if $\Gamma' \vdash e_1' : A' \mid \varepsilon'$, then $\Gamma' \vdash E[e_1'] : A \mid \varepsilon$.

By Lemma 3.19, we have $\emptyset \vdash e_2 : A' \mid \varepsilon'$. Thus, $\emptyset \vdash E[e_2] : A \mid \varepsilon$ holds as required. ■

Lemma 3.21. *If $n-\mathrm{free}(L, E)$, then $n = 0$.*

Proof. Straightforward by the induction on the derivation of $n-\mathrm{free}(L, E)$. ■

Lemma 3.22. *If $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\ \boldsymbol{T}^J\ v] : A \mid \varepsilon$ and $n-\mathrm{free}(l\,\boldsymbol{S}^I, E)$, then $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$.*

Proof. By induction on a derivation of $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\ \boldsymbol{T}^J\ v] : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.

Case T_App**:** For some B, we have

- $E = \square$,

- $\Gamma \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J : B \rightarrow_\varepsilon A \mid \mathbb{0}$, and
- $\Gamma \vdash v : B \mid \mathbb{0}$.

By Lemma 3.14(4), we have $\Gamma \vdash (l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon$. Thus, the required result is achieved.

Case **T_Let:** For some x, E_1, e, and B, we have

- $E = (\mathbf{let}\,x = E_1\,\mathbf{in}\,e)$,
- $\Gamma \vdash E_1[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v] : B \mid \varepsilon$, and
- $\Gamma, x : B \vdash e : A \mid \varepsilon$.

By n−free$(l\,\boldsymbol{S}^I, E_1)$ and the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon$ as required.

Case **T_Sub:** For some A' and ε', we have

- $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v] : A' \mid \varepsilon'$ and
- $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

Since only ST_Comp can derive $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\Gamma \vdash \varepsilon' \otimes \varepsilon$. By the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon'$. By the associativity of $\odot$, we have $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon$ as required.

Case **T_Handling:** For some l', $\boldsymbol{S'}^{I'}$, E_1, h, B, and ε', we have

- $E = \mathbf{handle}_{l'\,\boldsymbol{S'}^{I'}}\,E_1\,\mathbf{with}\,h$,
- $\Gamma \vdash E_1[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v] : B \mid \varepsilon'$, and
- $(l'\,\boldsymbol{S'}^{I'})^\uparrow \odot \varepsilon \sim \varepsilon'$.

By Lemma 3.21, we have $l\,\boldsymbol{S}^I \neq l'\,\boldsymbol{S'}^{I'}$ and 0−free$(l\,\boldsymbol{S}^I, E_1)$. By the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon'$. Thus, safety condition (2) makes $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon$ hold as required.

Case **others:** Cannot happen.

∎

Lemma 3.23 (Effect Safety). *If $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v] : A \mid \varepsilon$ and n−free$(l\,\boldsymbol{S}^I, E)$, then $\varepsilon \not\sim \mathbb{0}$.*

Proof. Assume that $\varepsilon \sim \mathbb{0}$. By Lemma 3.22, we have $(l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon$. Therefore, we have $(l\,\boldsymbol{S}^I)^\uparrow \odot \varepsilon' \sim \mathbb{0}$ for some ε'. However, this is contradictory with safety condition (1). ∎

Theorem 3.24 (Type and Effect Safety). *If $\emptyset \vdash e : A \mid \mathbb{0}$ and $e \longrightarrow^* e'$ and $e' \not\longrightarrow$, then e' is a value.*

Proof. By Lemma 3.20, $\emptyset \vdash e' : A \mid \mathbb{0}$ (it is easy to extend Lemma 3.20 to multi-step evaluation). By Lemma 3.23, $e' \neq E[\mathsf{op}_{l\,\boldsymbol{S}^N}\,\boldsymbol{T}^J\,v]$ for any E, l, $\boldsymbol{S}^N$, op, $\boldsymbol{T}^J$, and v such that n−free$(l\,\boldsymbol{S}^I, E)$ for some n. Thus, by Lemma 3.18, we have the fact that e' is a value. ∎

3.2 Properties with Shallow Handlers

This section assumes that the safety conditions in Definition 1.45 hold.

Lemma 3.25 (Weakening). *Suppose that $\vdash \Gamma_1, \Gamma_2$ and $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3) = \emptyset$.*

(1) If $\vdash \Gamma_1, \Gamma_3$, then $\vdash \Gamma_1, \Gamma_2, \Gamma_3$.

(2) If $\Gamma_1, \Gamma_3 \vdash S : K$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash S : K$.

(3) If $\Gamma_1, \Gamma_3 \vdash A <: B$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A <: B$.

(4) If $\Gamma_1, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$.

(5) If $\Gamma_1, \Gamma_3 \vdash e : A \mid \varepsilon$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e : A \mid \varepsilon$.

(6) If $\Gamma_1, \Gamma_3 \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$.

Proof.

(1)(2) Similarly to Lemma 3.5(1) and (2).

(3)(4) Similarly to Lemma 3.5(3) and (4).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_SHandling**:** For some N, e', A', ε', l, $\boldsymbol{S}^N$, $\boldsymbol{K}^N$, h, and σ, the following are given:

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e'\, \mathbf{with}\, h$,
- $\Gamma_1, \Gamma_3 \vdash e' : A' \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\Gamma_1, \Gamma_3 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\Gamma_1, \Gamma_3 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A'^{\varepsilon'} \Rightarrow^{\varepsilon} A$, and
- $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e' : A' \mid \varepsilon'$,
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$, and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A'^{\varepsilon'} \Rightarrow^{\varepsilon} A$.

Thus, T_SHandling derives

$$\Gamma_1, \Gamma_2, \Gamma_3 \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e\, \mathbf{with}\, h : A \mid \varepsilon.$$

Case SH_Return**:** Without loss of generality, we can choose x such that $x \notin \mathrm{dom}(\Gamma_2)$. For some e_r, the following are given:

- $h = \{\mathbf{return}\, x \mapsto e_r\}$,
- $\sigma = \{\}$,
- $\Gamma_1, \Gamma_3, x : A \vdash e_r : B \mid \varepsilon$, and
- $\Gamma_1, \Gamma_3 \vdash \varepsilon' : \mathbf{Eff}$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, \Gamma_3, x : A \vdash e_r : B \mid \varepsilon$. By Lemma 3.25(2), we have $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \varepsilon' : \mathbf{Eff}$. Thus, SH_Return derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\{\}} \{\mathbf{return}\, x \mapsto e_r\} : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$.

Case SH_Op**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ and p and k such that:

- $\{\boldsymbol{\beta}^J\} \cap \mathrm{dom}(\Gamma_2) = \emptyset$,
- $p \notin \mathrm{dom}(\Gamma_2)$, and
- $k \notin \mathrm{dom}(\Gamma_2)$.

For some h', σ', op, A', B', and e, the following are given:

- $h = h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B'\}$,
- $\Gamma_1, \Gamma_3 \vdash_{\sigma'} h' : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$, and
- $\Gamma_1, \Gamma_3, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} B \vdash e : B \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma'} h' : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$ and
- $\Gamma_1, \Gamma_2, \Gamma_3, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} B \vdash e : B \mid \varepsilon$.

Thus, SH_Op derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_{\sigma} h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\} : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$.

Case **others:** Similarly to Lemma 3.5(5) and (6).

∎

Lemma 3.26 (Substitution of values)**.** *Suppose that* $\Gamma_1 \vdash v : A \mid \mathbb{0}$.

(1) If $\vdash \Gamma_1, x : A, \Gamma_2$, *then* $\vdash \Gamma_1, \Gamma_2$.

(2) If $\Gamma_1, x : A, \Gamma_2 \vdash S : K$, *then* $\Gamma_1, \Gamma_2 \vdash S : K$.

(3) If $\Gamma_1, x : A, \Gamma_2 \vdash B <: C$, *then* $\Gamma_1, \Gamma_2 \vdash B <: C$.

(4) If $\Gamma_1, x : A, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$, *then* $\Gamma_1, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$.

(5) If $\Gamma_1, x : A, \Gamma_2 \vdash e : B \mid \varepsilon$, *then* $\Gamma_1, \Gamma_2 \vdash e[v/x] : B \mid \varepsilon$.

(6) If $\Gamma_1, x : A, \Gamma_2 \vdash_{\sigma} h : B^{\varepsilon'} \Rightarrow^{\varepsilon} C$, *then* $\Gamma_1, \Gamma_2 \vdash_{\sigma} h[v/x] : B^{\varepsilon'} \Rightarrow^{\varepsilon} C$.

Proof.(1)(2) Similarly to Lemma 3.7(1) and (2).

(3)(4) Similarly to Lemma 3.7(3) and (4).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_SHandling**:** For some N, e', A', ε', l, $\boldsymbol{S}^N$, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, h, and σ, the following are given:

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e' \,\mathbf{with}\, h$,
- $\Gamma_1, x : A, \Gamma_2 \vdash e' : A' \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\Gamma_1, x : A, \Gamma_2 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\Gamma_1, x : A, \Gamma_2 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A'^{\varepsilon'} \Rightarrow^{\varepsilon} B$, and
- $(l\,\boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2 \vdash e'[v/x] : A' \mid \varepsilon'$,
- $\Gamma_1, \Gamma_2 \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$, and
- $\Gamma_1, \Gamma_2 \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h[v/x] : A'^{\varepsilon'} \Rightarrow^{\varepsilon} A$.

Thus, T_SHandling derives

$$\Gamma_1, \Gamma_2 \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\, e'[v/x] \,\mathbf{with}\, h[v/x] : B \mid \varepsilon.$$

Case SH_Return**:** Without loss of generality, we can choose y such that $y \neq x$ and $y \notin \mathrm{FV}(v)$. For some e_r, the following are given:

- $h = \{\mathbf{return}\, y \mapsto e_r\}$,
- $\sigma = \{\}$,
- $\Gamma_1, x : A, \Gamma_2, y : B \vdash e_r : C \mid \varepsilon$, and
- $\Gamma_1, x : A, \Gamma_2 \vdash \varepsilon' : \mathbf{Eff}$.

By the induction hypothesis, we have $\Gamma_1, \Gamma_2, y : B \vdash e_r[v/x] : C \mid \varepsilon$. By Lemma 3.26(2), we have $\Gamma_1, \Gamma_2 \vdash \varepsilon' : \mathbf{Eff}$. Thus, SH_Return derives

$$\Gamma_1, \Gamma_2 \vdash_{\{\}} \{\mathbf{return}\, y \mapsto e_r[v/x]\} : B^{\varepsilon'} \Rightarrow^{\varepsilon} C.$$

Case SH_Op**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ and p and k such that:

- $p \neq x$,
- $k \neq x$,
- $p \notin \mathrm{FV}(v)$,
- $k \notin \mathrm{FV}(k)$, and
- $\{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(v) = \emptyset$.

For some h', σ', op, A', B', and e, the following are given:

- $h = h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B'\}$,
- $\Gamma_1, x : A, \Gamma_2 \vdash_{\sigma'} h' : B^{\varepsilon'} \Rightarrow^{\varepsilon} C$, and
- $\Gamma_1, x : A, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} C \vdash e : C \mid \varepsilon$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2 \vdash_{\sigma'} h'[v/x] : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$ and
- $\Gamma_1, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} B \vdash e[v/x] : B \mid \varepsilon$.

Thus, SH_Op derives

$$\Gamma_1, \Gamma_2 \vdash_{\sigma} h'[v/x] \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\, p\, k \mapsto e[v/x]\} : B^{\varepsilon'} \Rightarrow^{\varepsilon} C$$

.

Case **others:** Similarly to Lemma 3.7(5) and (6).

■

Lemma 3.27 (Substitution of Typelikes)**.** *Suppose that* $\Gamma_1 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$.

(1) If $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$*, then* $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

(2) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash T : K$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : K$*.*

(3) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A <: B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(4) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(5) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e : A \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(6) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S}/\boldsymbol{\alpha}]} h[\boldsymbol{S}/\boldsymbol{\alpha}] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

Proof.(1)(2) Similarly to Lemma 3.10(1) and (2).

(3)(4) Similarly to Lemma 3.10(3) and (4).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_SHandling**:** For some N, e', A', ε', l, $\boldsymbol{S_0}^N$, $\boldsymbol{\alpha_0}^N$, $\boldsymbol{K_0}^N$, h, and σ, the following are given:

- $e = \mathbf{handle}_{l\,\boldsymbol{S_0}^N}\, e'\, \mathbf{with}\, h$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e' : A' \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha_0}^N : \boldsymbol{K_0}^N.\sigma \in \Xi$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \boldsymbol{S_0}^N : \boldsymbol{K_0}^N$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_{\sigma[\boldsymbol{S_0}^N/\boldsymbol{\alpha_0}^N]} h : A'^{\varepsilon'} \Rightarrow^{\varepsilon} A$, and
- $(l\,\boldsymbol{S_0}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By the induction hypothesis, case (2), and that a typelike substitution is homomorphism for $\odot$ and $\sim$, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N : \boldsymbol{K_0}^N$,
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S_0}^N/\boldsymbol{\alpha_0}^N][\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and
- $(l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N)^{\uparrow} \odot \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \sim \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Now, because we can assume that

- $\{\boldsymbol{\alpha}^I\} \cap \{\boldsymbol{\alpha_0}^N\} = \emptyset$ and
- $\{\boldsymbol{\alpha_0}^N\} \cap \mathrm{FTV}(\boldsymbol{S}^I) = \emptyset$

without loss of generality, we have

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N/\boldsymbol{\alpha_0}^N]} h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Thus, T_SHandling derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \mathbf{handle}_{l\,\boldsymbol{S_0}[\boldsymbol{S^I}/\boldsymbol{\alpha^I}]^N}\, e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\, \mathbf{with}\, h[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Case SH_Return**:** For some x and e_r, the following are given:

- $h = \{\mathbf{return}\, y \mapsto e_r\}$,
- $\sigma = \{\}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, x : A \vdash e_r : B \mid \varepsilon$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash \varepsilon' : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], x : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e_r[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

By Lemma 3.27(2), we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Eff}$.

Thus, SH_Return derives

$$\Gamma_1, \Gamma_2 \vdash_{\{\}} \{\mathbf{return}\, x \mapsto e_r[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\} : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

Case SH_Op**:** Without loss of generality, we can choose $\boldsymbol{\beta}^J$ such that:

- $\{\boldsymbol{\beta}^J\} \cap \{\boldsymbol{\alpha}^I\} = \emptyset$ and
- $\{\boldsymbol{\beta}^J\} \cap \mathrm{FTV}(\boldsymbol{S}^I) = \emptyset$.

For some h', σ', op, A', B', and e, the following are given:

- $h = h' \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\ p\ k \mapsto e\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J . A' \Rightarrow B'\}$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_{\sigma'} h' : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$, and
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \rightarrow_{\varepsilon'} B \vdash e : B \mid \varepsilon$.

By the induction hypothesis and Definition 1.10, we have

- $\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] = \sigma'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \uplus \{\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K}^J . A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow B'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\}$,
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$, and
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I], k : B'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \rightarrow_{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, SH_Op derives

$$\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \uplus \{\mathsf{op}\, \boldsymbol{\beta}^J : \boldsymbol{K}^J\ p\ k \mapsto e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]\} : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]^{\varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I].$$

***Case* others:** Similarly to Lemma 3.10(5) and (6).

■

Lemma 3.28 (Well-formedness of contexts in typing judgments)**.**

- *If* $\Gamma \vdash e : A \mid \varepsilon$*, then* $\vdash \Gamma$.
- *If* $\Gamma \vdash_{\sigma} h : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$*, then* $\vdash \Gamma$.

Proof. Straightforward by mutual induction on the derivations. ■

Lemma 3.29 (Well-kinded of Typing)**.**

- *If* $\Gamma \vdash e : A \mid \varepsilon$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.
- *If* $\Gamma \vdash_{\sigma} h : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon' : \mathbf{Eff}$ *and* $\Gamma \vdash B : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Proof. By mutual induction on derivations of the judgments. We proceed by cases on the typing rule applied lastly to the derivation.

***Case* T_SHandling:** For some A', ε', σ, N, $\boldsymbol{\alpha}^N$, and $\boldsymbol{S}^N$, we have

$$\Gamma \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A'^{\varepsilon'} \Rightarrow^{\varepsilon} A.$$

By the induction hypothesis, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

***Case* SH_Return:** For some x and e_r, we have

- $\Gamma, x : A \vdash e_r : B \mid \varepsilon$ and
- $\Gamma \vdash \varepsilon' : \mathbf{Eff}$.

By the induction hypothesis, we have

- $\Gamma, x : A \vdash B : \mathbf{Typ}$ and
- $\Gamma, x : A \vdash \varepsilon : \mathbf{Eff}$.

By Lemma 3.2(2), we have

- $\Delta(\Gamma) \vdash B : \mathbf{Typ}$ and
- $\Delta(\Gamma) \vdash \varepsilon : \mathbf{Eff}$.

By Lemma 3.6, we have

- $\Gamma \vdash B : \mathbf{Typ}$ and
- $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Now, we have $\vdash \Gamma, x : A$ by Lemma 3.28. Since only C_Var can derive $\vdash \Gamma, x : A$, we have $\Gamma \vdash A : \mathbf{Typ}$.

***Case* SH_Op:** For some h' and σ', we have $\Gamma \vdash_{\sigma'} h' : A^{\varepsilon'} \Rightarrow^{\varepsilon} B$. By the induction hypothesis, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon' : \mathbf{Eff}$ and $\Gamma \vdash B : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

***Case* others:** Similarly to Lemma 3.12(1) and (2). ■

Lemma 3.30 (Inversion)**.**

(1) If $\Gamma \vdash v : A \mid \varepsilon$, then $\Gamma \vdash v : A \mid \mathbb{0}$.

(2) If $\Gamma \vdash \mathbf{fun}\,(f, x, e) : A_1 \to_{\varepsilon_1} B_1 \mid \varepsilon$, then $\Gamma, f : A_2 \to_{\varepsilon_2} B_2, x : A_2 \vdash e : B_2 \mid \varepsilon_2$ for some A_2, ε_2, and B_2 such that $\Gamma \vdash A_2 \to_{\varepsilon_2} B_2 <: A_1 \to_{\varepsilon_1} B_1$.

(3) If $\Gamma \vdash \Lambda\alpha : K.e : \forall\alpha : K.{A_1}^{\varepsilon_1} \mid \varepsilon$, then $\Gamma, \alpha : K \vdash e : A_1 \mid \varepsilon_1$.

(4) If $\Gamma \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J : A_1 \to_{\varepsilon_1} B_1 \mid \varepsilon$, then the following hold:

- $l :: \forall\boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$,
- $\Gamma \vdash A_1 <: A[\boldsymbol{T}^J/\boldsymbol{\beta}^J]$,
- $\Gamma \vdash B[\boldsymbol{T}^J/\boldsymbol{\beta}^J] <: B_1$, *and*
- $\Gamma \vdash (l\,\boldsymbol{S}^I)^\uparrow \otimes \varepsilon_1$

for some $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, σ, $\boldsymbol{\beta}^J$, $\boldsymbol{K'}^J$, A, and B.

(5) If $\Gamma \vdash v_1\,v_2 : B \mid \varepsilon$, then there exists some type A such that $\Gamma \vdash v_1 : A \to_\varepsilon B \mid \mathbb{0}$ and $\Gamma \vdash v_2 : A \mid \mathbb{0}$.

Proof. Similarly to Lemma 3.14; Lemmas 3.28 and 3.29 are used instead of Lemmas 3.9 and 3.12, respectively. ■

Lemma 3.31 (Canonical Form)**.**

(1) If $\emptyset \vdash v : A \to_\varepsilon B \mid \varepsilon'$, then either of the following holds:

- $v = \mathbf{fun}\,(f, x, e)$ *for some f, x, and e, or*
- $v = \mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J$ *for some op, l, $\boldsymbol{S}^I$, and $\boldsymbol{T}^J$.*

(2) If $\emptyset \vdash v : \forall\alpha : K.A^\varepsilon \mid \varepsilon'$, then $v = \Lambda\alpha : K.e$ for some e.

Proof. Similarly to Lemma 3.15. ■

Lemma 3.32 (Inversion of Handler Typing)**.**

(1) If $\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$, then there exist some x and e_r such that $\mathbf{return}\,x \mapsto e_r \in h$ and $\Gamma, x : A \vdash e_r : B \mid \varepsilon$.

(2) If $\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$ and $\mathsf{op} : \forall\boldsymbol{\beta}^J : \boldsymbol{K}^J.A' \Rightarrow B' \in \sigma$, then

- $\mathsf{op}\,\boldsymbol{\beta}^J : \boldsymbol{K}^J\,p\,k \mapsto e \in h$ *and*
- $\Gamma, \boldsymbol{\beta}^J : \boldsymbol{K}^J, p : A', k : B' \to_{\varepsilon'} B \vdash e : B \mid \varepsilon$

for some p, k, and e.

Proof. (1) By induction on a derivation of $\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$. We proceed by cases on the typing rule applied lastly to this derivation.

***Case* H_Return:** Clearly.

***Case* H_Op:** Clearly by the induction hypothesis.

(2) By induction on a derivation of $\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$. We proceed by cases on the typing rule applied lastly to this derivation.

***Case* H_Return:** Clearly because there is no operation belonging to $\{\}$.

***Case* H_Op:** For some h', σ', op', $\boldsymbol{\beta'}^{J'}$, $\boldsymbol{K'}^{J'}$, A'', B'', p', k', e'', and ε', the following are given:

- $h = h' \uplus \{\mathsf{op}'\,\boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}\,p'\,k' \mapsto e'\}$,
- $\sigma = \sigma' \uplus \{\mathsf{op}' : \forall\boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}.A'' \Rightarrow B''\}$,
- $\Gamma \vdash_{\sigma'} h' : A^{\varepsilon'} \Rightarrow^\varepsilon B$, and

- $\Gamma, \boldsymbol{\beta'}^{J'} : \boldsymbol{K'}^{J'}, p' : A'', k' : B'' \rightarrow_{\varepsilon'} B \vdash e : B \mid \varepsilon$.

If $\mathsf{op} = \mathsf{op}'$, then clearly.

If $\mathsf{op} \neq \mathsf{op}'$, then clearly by the induction hypothesis.

■

Lemma 3.33 (Independence of Evaluation Contexts). *If $\Gamma \vdash E[e] : A \mid \varepsilon$, then there exist some A' and ε' such that*

- *$\Gamma \vdash e : A' \mid \varepsilon'$, and*
- *$\Gamma, \Gamma' \vdash E[e'] : A \mid \varepsilon$ holds for any e' and Γ' such that $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$.*

Proof. Similarly to Lemma 3.17; Lemmas 3.25 and 3.28 are used instead of Lemmas 3.5 and 3.9, respectively. ■

Lemma 3.34 (Progress). *If $\emptyset \vdash e : A \mid \varepsilon$, then one of the following holds:*

- *e is a value;*
- *There exists some expression e' such that $e \longrightarrow e'$; or*
- *There exist some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, and n such that $e = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v]$ and n–free$(l\,\boldsymbol{S}^I, E)$.*

Proof. Similarly to Lemma 3.18; Lemmas 3.30, 3.31, and 3.32 are used instead of Lemmas 3.14, 3.15, and 3.16, respectively. ■

Lemma 3.35 (Preservation in Reduction). *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longmapsto e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. By induction on a derivation of $\Gamma \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_SHandling**:** We proceed by cases on the derivation rule which derives $e \longmapsto e'$.

Case R_Handle1**:** We have

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^I}\, v\, \mathbf{with}\, h$,
- $\mathbf{return}\, x \mapsto e_r \in h$,
- $\emptyset \vdash v : B \mid \varepsilon'$,
- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h : B^{\varepsilon'} \Rightarrow^{\varepsilon} A$,
- $(l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, and
- $e' = e_r[v/x]$

for some l, $\boldsymbol{S}^I$, $\boldsymbol{\alpha}^I$, $\boldsymbol{K}^I$, σ, v, h, B, and ε'. By $\emptyset \vdash_{\sigma[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} h : B^{\varepsilon'} \Rightarrow^{\varepsilon} A$ and $\mathbf{return}\, x \mapsto e_r \in h$ and Lemma 3.32(1), we have

$$x : B \vdash e_r : A \mid \varepsilon.$$

By Lemma 3.30(1), we have $\emptyset \vdash v : B \mid \mathbb{0}$. Thus, Lemma 3.26(5) makes $\emptyset \vdash e_r[v/x] : A \mid \varepsilon$ hold as required.

Case R_Handle2**:** We have

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[\mathsf{op}_{0\,l\,\boldsymbol{S}^N}\, \boldsymbol{T}^J\, v]\, \mathbf{with}\, h$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\mathsf{op}_0\, \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J\, p_0\, k_0 \mapsto e_0 \in h$,
- 0–free$(l\,\boldsymbol{S}^N, E)$,
- $\emptyset \vdash E[\mathsf{op}_{0\,l\,\boldsymbol{S}^N}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B^{\varepsilon'} \Rightarrow^{\varepsilon} A$,
- $(l\,\boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, and
- $e' = e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.E[z]/k_0]$

for some l, $\boldsymbol{S}^N$, E, op_0, $\boldsymbol{T}^J$, v, h, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, σ, $\boldsymbol{\beta_0}^J$, $\boldsymbol{K_0}^J$, p_0, k_0, e_0, B, and ε'. By Lemma 3.33, there exist some B_1 and ε_1 such that

- $\emptyset \vdash \mathsf{op}_{0\,l\,\boldsymbol{S}^N}\, \boldsymbol{T}^J\, v : B_1 \mid \varepsilon_1$, and

- for any e'' and Γ'', if $\Gamma'' \vdash e'' : B_1 \mid \varepsilon_1$, then $\Gamma'' \vdash E[e''] : B \mid \varepsilon'$.

By Lemma 3.30(5), we have $\emptyset \vdash \mathsf{op}_{0\, l\, \boldsymbol{S}^N}\, \boldsymbol{T}^J : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \mathbb{0}$ and $\emptyset \vdash v : A_1 \mid \mathbb{0}$ for some A_1. By Lemma 3.30(4) and 3.32(2), we have

- $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J . A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N / \boldsymbol{\alpha}^N]$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash \boldsymbol{T}^J : \boldsymbol{K_0}^J$,
- $\emptyset \vdash A_1 <: A_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J]$,
- $\emptyset \vdash B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] <: B_1$, and
- $\emptyset \vdash (l\, \boldsymbol{S}^N)^{\uparrow} \otimes \varepsilon_1$.

for some A_0 and B_0. Thus, T_Sub with $\emptyset \vdash \mathbb{0} \otimes \mathbb{0}$ implied by Lemma 3.3 derives

$$\emptyset \vdash v : A_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] \mid \mathbb{0}.$$

By Lemma 3.11, we have $\emptyset \vdash B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] : \mathbf{Typ}$. Thus, C_Var derives $\vdash z : B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J]$. By $\emptyset \vdash \mathbb{0} : \mathbf{Eff}$, $\emptyset \vdash \varepsilon_1 : \mathbf{Eff}$ implied by Lemma 3.12, and $\mathbb{0} \odot \varepsilon_1 \sim \varepsilon_1$, we have $\emptyset \vdash \mathbb{0} \otimes \varepsilon_1$. Since T_Var and T_Sub derives $z : B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] \vdash z : B_1 \mid \varepsilon_1$, we have

$$z : B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] \vdash E[z] : B \mid \varepsilon'$$

by the result of Lemma 3.33. Thus, T_Abs derives

$$\emptyset \vdash \lambda z. E[z] : B_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J] \rightarrow_{\varepsilon'} B \mid \mathbb{0}.$$

Since

$$\boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J, p_0 : A_0, k_0 : B_0 \rightarrow_{\varepsilon'} B \vdash e_0 : A \mid \varepsilon$$

by $\emptyset \vdash_{\sigma[\boldsymbol{S}^N / \boldsymbol{\alpha}^N]} h : B^{\varepsilon'} \Rightarrow^{\varepsilon} A$ and $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J . A_0 \Rightarrow B_0 \in \sigma$ and Lemma 3.32(2), Lemma 3.27(5) and Lemma 3.26(5) imply

$$\emptyset \vdash e_0[\boldsymbol{T}^J / \boldsymbol{\beta_0}^J][v / p_0][\lambda z. E[z] / k_0] : A \mid \varepsilon$$

as required.

***Case* others:** Similarly to Lemma 3.19; Lemmas 3.30, 3.25, 3.28, 3.29, 3.26, and 3.27 are used instead of Lemmas 3.14, 3.5, 3.9, 3.12, 3.7, and 3.10 respectively.

■

Lemma 3.36 (Preservation)**.** *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longrightarrow e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. Similarly to Lemma 3.20; Lemmas 3.33 and 3.35 are used instead of Lemmas 3.17 and 3.19. ■

Lemma 3.37. *If $\Gamma \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and $n-\mathrm{free}(l\, \boldsymbol{S}^I, E)$, then $\Gamma \vdash (l\, \boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$.*

Proof. Similarly to Lemma 3.22; Lemma 3.30 is used instead of Lemma 3.14. ■

Lemma 3.38 (Effect Safety)**.** *If $\Gamma \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and $n-\mathrm{free}(l\, \boldsymbol{S}^I, E)$, then $\varepsilon \nsim \mathbb{0}$.*

Proof. Similarly to Lemma 3.23; Lemma 3.37 is used instead of Lemma3.22. ■

Theorem 3.39 (Type and Effect Safety)**.** *If $\emptyset \vdash e : A \mid \mathbb{0}$ and $e \longrightarrow^* e'$ and $e' \not\longrightarrow$, then e' is a value.*

Proof. Similarly to Theorem 3.24; Lemmas 3.36, 3.38, and 3.34 are used instead of Lemmas 3.20, 3.23, and 3.18, respectively. ■

3.3 Properties with Lift Coercions

This section assumes that the safety conditions in Definition 1.45 and the safety condition for lift coercions in Definition 1.46 hold.

Lemma 3.40 (Weakening)**.** *Suppose that $\vdash \Gamma_1, \Gamma_2$ and $\mathrm{dom}(\Gamma_2) \cap \mathrm{dom}(\Gamma_3) = \emptyset$.*

(1) If $\vdash \Gamma_1, \Gamma_3$, then $\vdash \Gamma_1, \Gamma_2, \Gamma_3$.

(2) If $\Gamma_1, \Gamma_3 \vdash S : K$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash S : K$.

(3) If $\Gamma_1, \Gamma_3 \vdash A <: B$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A <: B$*.*

(4) If $\Gamma_1, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*.*

(5) If $\Gamma_1, \Gamma_3 \vdash e : A \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e : A \mid \varepsilon$*.*

(6) If $\Gamma_1, \Gamma_3 \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash_\sigma h : A \Rightarrow^\varepsilon B$*.*

Proof.(1)(2) Similarly to Lemma 3.5(1) and (2).

(3)(4) Similarly to Lemma 3.5(3) and (4).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_Lift**:** For some e', L, and ε', the following are given:

- $e = [e']_L$,
- $\Gamma_1, \Gamma_3 \vdash e' : A \mid \varepsilon'$,
- $\Gamma_1, \Gamma_3 \vdash L : \mathbf{Lab}$, and
- $(L)^\uparrow \odot \varepsilon' \sim \varepsilon$.

By the induction hypothesis and case (2), we have

- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash e' : A \mid \varepsilon'$ and
- $\Gamma_1, \Gamma_2, \Gamma_3 \vdash L : \mathbf{Lab}$.

Thus, T_Lift derives $\Gamma_1, \Gamma_2, \Gamma_3 \vdash [e']_L : A \mid \varepsilon$.

Case **others:** Similarly to Lemma 3.5(5) and (6).

■

Lemma 3.41 (Substitution of values)**.** *Suppose that* $\Gamma_1 \vdash v : A \mid \mathbb{0}$*.*

(1) If $\vdash \Gamma_1, x : A, \Gamma_2$*, then* $\vdash \Gamma_1, \Gamma_2$*.*

(2) If $\Gamma_1, x : A, \Gamma_2 \vdash S : K$*, then* $\Gamma_1, \Gamma_2 \vdash S : K$*.*

(3) If $\Gamma_1, x : A, \Gamma_2 \vdash B <: C$*, then* $\Gamma_1, \Gamma_2 \vdash B <: C$*.*

(4) If $\Gamma_1, x : A, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$*.*

(5) If $\Gamma_1, x : A, \Gamma_2 \vdash e : B \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2 \vdash e[v/x] : B \mid \varepsilon$*.*

(6) If $\Gamma_1, x : A, \Gamma_2 \vdash_\sigma h : B^{\varepsilon'} \Rightarrow^\varepsilon C$*, then* $\Gamma_1, \Gamma_2 \vdash_\sigma h[v/x] : B^{\varepsilon'} \Rightarrow^\varepsilon C$*.*

Proof.(1)(2) Similarly to Lemma 3.7(1) and (2).

(3)(4) Similarly to Lemma 3.7(3) and 3.7(4).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_Lift**:** For some e', ε', and L, the following are given:

- $e = [e']_L$,
- $\Gamma_1, x : A, \Gamma_2 \vdash e' : B \mid \varepsilon'$,
- $\Gamma_1, x : A, \Gamma_2 \vdash L : \mathbf{Lab}$, and
- $(L)^\uparrow \odot \varepsilon' \sim \varepsilon$.

By the induction hypothesis and case 3.41(2), we have

- $\Gamma_1, \Gamma_2 \vdash e'[v/x] : B \mid \varepsilon'$ and
- $\Gamma_1, \Gamma_2 \vdash L : \mathbf{Lab}$.

Thus, T_Lift derives $\Gamma_1, \Gamma_2 \vdash [e'[v/x]]_L : A \mid \varepsilon$.

Case **others:** Similarly to Lemma 3.7(5) and (6).

■

Lemma 3.42 (Substitution of Typelikes)**.** *Suppose that* $\Gamma_1 \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$*.*

(1) If $\vdash \Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2$*, then* $\vdash \Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$*.*

(2) If $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash T : K$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash T[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : K$*.*

(3) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A <: B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

(4) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash A_1 \mid \varepsilon_1 <: A_2 \mid \varepsilon_2$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash A_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_1[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] <: A_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

(5) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e : A \mid \varepsilon$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

(6) *If* $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash_{\sigma[\boldsymbol{S}/\boldsymbol{\alpha}]} h[\boldsymbol{S}/\boldsymbol{\alpha}] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \Rightarrow^{\varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Proof.(1)(2) Similarly to Lemma 3.10(1) and (2).

(3)(4) Similarly to Lemma 3.10(3) and 3.10(3).

(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case T_Lift**:** For some e', ε', and L, the following are given:

- $e = [e']_L$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash e' : A \mid \varepsilon'$,
- $\Gamma_1, \boldsymbol{\alpha}^I : \boldsymbol{K}^I, \Gamma_2 \vdash L : \mathbf{Lab}$, and
- $(L)^\uparrow \odot \varepsilon' \sim \varepsilon$.

By the induction hypothesis, case 3.42(2), and the fact that a typelike substitution is homomorphism for $\odot$ and $\sim$, we have

- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash e'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$,
- $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash L[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] : \mathbf{Lab}$, and
- $(L)^\uparrow[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \odot \varepsilon'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \sim \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Thus, T_Lift derives $\Gamma_1, \Gamma_2[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \vdash [e'[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]]_{L[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]} : A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I] \mid \varepsilon[\boldsymbol{S}^I/\boldsymbol{\alpha}^I]$.

Case **others:** Similarly to Lemma 3.10(5) and (6).

■

Lemma 3.43 (Well-formedness of contexts in typing judgments)**.**

- *If* $\Gamma \vdash e : A \mid \varepsilon$*, then* $\vdash \Gamma$.
- *If* $\Gamma \vdash_\sigma h : A \Rightarrow^\varepsilon B$*, then* $\vdash \Gamma$.

Proof. Straightforward by mutual induction on the derivations. ■

Lemma 3.44 (Well-kinded of Typing)**.**

- *If* $\Gamma \vdash e : A \mid \varepsilon$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.
- *If* $\Gamma \vdash_\sigma h : A^{\varepsilon'} \Rightarrow^\varepsilon B$*, then* $\Gamma \vdash A : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon' : \mathbf{Eff}$ *and* $\Gamma \vdash B : \mathbf{Typ}$ *and* $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Proof. By mutual induction on derivations of the judgments. We proceed by cases on the typing rule applied lastly to the derivation.

Case T_Lift**:** For some e', ε', and L, the following are given:

- $e = [e']_L$,
- $\Gamma \vdash e' : A \mid \varepsilon'$,
- $\Gamma \vdash L : \mathbf{Lab}$, and
- $(L)^\uparrow \odot \varepsilon' \sim \varepsilon$.

By the induction hypothesis, we have $\Gamma \vdash A : \mathbf{Typ}$ and $\Gamma \vdash \varepsilon' : \mathbf{Eff}$. $(-)^\uparrow$, $\odot$, and $\sim$ preserve well-formedness, we have $\Gamma \vdash \varepsilon : \mathbf{Eff}$.

Case **others:** Similarly to Lemma 3.12(1) and (2).

■

Lemma 3.45 (Inversion)**.**

(1) *If* $\Gamma \vdash v : A \mid \varepsilon$*, then* $\Gamma \vdash v : A \mid \mathbb{0}$.

(2) *If* $\Gamma \vdash \mathbf{fun}\,(g, x, e) : A_1 \to_{\varepsilon_1} B_1 \mid \varepsilon$*, then* $\Gamma, g : A_2 \to_{\varepsilon_2} B_2, x : A_2 \vdash e : B_2 \mid \varepsilon_2$ *for some* A_2*,* ε_2*, and* B_2 *such that* $\Gamma \vdash A_2 \to_{\varepsilon_2} B_2 <: A_1 \to_{\varepsilon_1} B_1$.

(3) If $\Gamma \vdash \Lambda\alpha : K.e : \forall\alpha : K.{A_1}^{\varepsilon_1} \mid \varepsilon$*, then* $\Gamma, \alpha : K \vdash e : A_1 \mid \varepsilon_1$*.*

(4) If $\Gamma \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J : A_1 \to_{\varepsilon_1} B_1 \mid \varepsilon$*, then the following hold:*

- $l :: \forall \boldsymbol{\alpha}^I : \boldsymbol{K}^I.\sigma \in \Xi$,
- $\mathsf{op} : \forall \boldsymbol{\beta}^J : \boldsymbol{K'}^J.A \Rightarrow B \in \sigma$,
- $\vdash \Gamma$,
- $\Gamma \vdash \boldsymbol{S}^I : \boldsymbol{K}^I$,
- $\Gamma \vdash \boldsymbol{T}^J : \boldsymbol{K'}^J$,
- $\Gamma \vdash A_1 <: A[\boldsymbol{S}^I/\boldsymbol{\alpha}^I][\boldsymbol{T}^J/\boldsymbol{\beta}^J]$,
- $\Gamma \vdash B[\boldsymbol{S}^I/\boldsymbol{\alpha}^I][\boldsymbol{T}^J/\boldsymbol{\beta}^J] <: B_1$*, and*
- $\Gamma \vdash (l\, \boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon_1$

for some $\boldsymbol{\alpha}^I$*,* $\boldsymbol{K}^I$*,* σ*,* $\boldsymbol{\beta}^J$*,* $\boldsymbol{K'}^J$*,* A*, and* B*.*

(5) If $\Gamma \vdash v_1\, v_2 : B \mid \varepsilon$*, then there exists some type* A *such that* $\Gamma \vdash v_1 : A \to_{\varepsilon} B \mid \mathbb{0}$ *and* $\Gamma \vdash v_2 : A \mid \mathbb{0}$*.*

Proof. Similarly to Lemma 3.14; Lemmas 3.43 and 3.44 are used instead of Lemmas 3.9 and 3.12, respectively. ■

Lemma 3.46 (Canonical Form)**.**

(1) If $\emptyset \vdash v : A \to_{\varepsilon} B \mid \varepsilon'$*, then either of the following holds:*

- $v = \mathbf{fun}\,(g, x, e)$ *for some* g*,* x*, and* e*, or*
- $v = \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J$ *for some* op*,* l*,* $\boldsymbol{S}^I$*, and* $\boldsymbol{T}^J$*.*

(2) If $\emptyset \vdash v : \forall\alpha : K.A^{\varepsilon} \mid \varepsilon'$*, then* $v = \Lambda\alpha : K.e$ *for some* e*.*

Proof. Similarly to Lemma 3.15. ■

Lemma 3.47 (Independence of Evaluation Contexts)**.** *If* $\Gamma \vdash E[e] : A \mid \varepsilon$*, then there exist some* A' *and* ε' *such that*

- $\Gamma \vdash e : A' \mid \varepsilon'$*, and*
- $\Gamma, \Gamma' \vdash E[e'] : A \mid \varepsilon$ *holds for any* e' *and* Γ' *such that* $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$*.*

Proof. By induction on a derivation of $\Gamma \vdash E[e] : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case **T_Lift:** If $E = \Box$, then the required result is achieved immediately.

If $E \neq \Box$, then we have

- $E = [E']_L$,
- $\Gamma \vdash E'[e] : A \mid \varepsilon'$,
- $\Gamma \vdash L : \mathbf{Lab}$, and
- $(L)^{\uparrow} \odot \varepsilon' \sim \varepsilon$,

for some E', L, and ε'. By the induction hypothesis, there exist some A' and ε'' such that

- $\Gamma \vdash e : A' \mid \varepsilon''$ and
- for any e' and Γ' such that $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon''$, typing judgment $\Gamma, \Gamma' \vdash E'[e'] : A \mid \varepsilon'$ is derivable.

Let e' be an expression and Γ' be a typing context such that $\Gamma, \Gamma' \vdash e' : A' \mid \varepsilon'$. The induction hypothesis gives us $\Gamma, \Gamma' \vdash E'[e'] : A \mid \varepsilon'$. By Lemma 3.40(2), we have $\Gamma, \Gamma' \vdash L : \mathbf{Lab}$. Thus, T_Lift derives $\Gamma, \Gamma' \vdash [E'[e']]_L : A \mid \varepsilon$ as required.

Case **others:** Similarly to Lemma 3.17; Lemmas 3.40 and 3.43 are used instead of Lemmas 3.5 and 3.9, respectively. ■

Lemma 3.48 (Progress)**.** *If* $\emptyset \vdash e : A \mid \varepsilon$*, then one of the following holds:*

- *e is a value;*
- *There exists some expression* e' *such that* $e \longrightarrow e'$*; or*

- *There exist some* op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, *and* n *such that* $e = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v]$ *and* n*−*$\mathrm{free}(l\,\boldsymbol{S}^I, E)$.

Proof. By induction on a derivation of $\emptyset \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_LIFT**:** For some e_1, L and ε_1, the following are given:

- $e = [e_1]_L$,
- $\emptyset \vdash e_1 : A \mid \varepsilon_1$, and
- $\emptyset \vdash (L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon$.

By the induction hypothesis, we proceed by cases on the following conditions:

(1) e_1 is a value,

(2) There exists some e_1' such that $e_1 \longrightarrow e_1'$,

(3) There exist some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, and n such that $e_1 = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\,\boldsymbol{T}^J\,v]$ and $n-\mathrm{free}(l\,\boldsymbol{S}^I, E)$.

Case **(1):** R_LIFT derives $[e_1]_L \longmapsto e_1$ because e_1 is a value.

Case **(2):** Since only E_EVAL can derive $e_1 \longrightarrow e_1'$, we have

- $e_1 = E_1[e_{11}]$,
- $e_1' = E_1[e_{12}]$, and
- $e_{11} \longmapsto e_{12}$,

for some E_1, e_{11}, and e_{12}. Let $E = ([E_1]_L)$. E_EVAL derives $e \longrightarrow E[e_{12}]$ because of $e = E[e_{11}]$.

Case **(3):** If $L \neq l\,\boldsymbol{S}^I$, then we have $n-\mathrm{free}(l\,\boldsymbol{S}^I, [E]_L)$.
If $L = l\,\boldsymbol{S}^I$, then we have $n+1-\mathrm{free}(l\,\boldsymbol{S}^I, [E]_L)$.

Case **others:** Similarly to Lemma 3.18; Lemmas 3.45 and 3.46 are used instead of Lemmas 3.14 and 3.15, respectively.

■

Lemma 3.49 (Preservation in Reduction)**.** *If* $\emptyset \vdash e : A \mid \varepsilon$ *and* $e \longmapsto e'$*, then* $\emptyset \vdash e' : A \mid \varepsilon$.

Proof. By induction on a derivation of $\Gamma \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

Case T_LIFT**:** Since only R_LIFT derives $e \longmapsto e'$, we have

- $e = [v]_L$,
- $\emptyset \vdash v : A \mid \varepsilon_1$,
- $\emptyset \vdash L : \mathbf{Lab}$,
- $(L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon$, and
- $e' = v$.

for some v, L, and ε_1. By Lemma 3.45(1), we have $\emptyset \vdash v : A \mid \mathbb{0}$. By $\mathbb{0} \odot \varepsilon \sim \varepsilon$, we have $\emptyset \vdash v : A \mid \varepsilon$ as required.

Case **others:** Similarly to Lemma 3.19; Lemmas 3.45, 3.40, 3.43, 3.44, 3.41, and 3.42 are used instead of Lemmas 3.14, 3.5, 3.9, 3.12, 3.7, and 3.10 respectively.

■

Lemma 3.50 (Preservation)**.** *If* $\emptyset \vdash e : A \mid \varepsilon$ *and* $e \longrightarrow e'$*, then* $\emptyset \vdash e' : A \mid \varepsilon$.

Proof. Similarly to Lemma 3.20; Lemmas 3.47 and 3.49 are used instead of Lemmas 3.17 and 3.19. ■

Definition 3.51 (Label Inclusion)**.**
Label Inclusion $\boxed{L \otimes^n \varepsilon}$

$$\frac{}{L \otimes^0 \varepsilon}\ \textrm{LI\_EMPTY} \qquad \frac{L \otimes^n \varepsilon_1 \quad (L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2}{L \otimes^{n+1} \varepsilon_2}\ \textrm{LI\_HANDLING}$$

$$\frac{L \otimes^n \varepsilon_1 \quad (L')^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2 \quad L \neq L'}{L \otimes^n \varepsilon_2}\ \textrm{LI\_NOHANDLING}$$

Lemma 3.52. *If $L \circledcirc^n \varepsilon_1$ and $\varepsilon_1 \odot \varepsilon_2 \sim \varepsilon_3$, then $L \circledcirc^n \varepsilon_3$.*

Proof. By induction on a derivation of $L \circledcirc^n \varepsilon_1$. We proceed by case analysis on the rule applied lastly to this derivation.
Case LI_EMPTY**:** We have $n = 0$. LI_EMPTY derives $L \circledcirc^0 \varepsilon_3$ as required.

Case LI_HANDLING**:** We have

- $n = n' + 1$,
- $L \circledcirc^{n'} \varepsilon_4$, and
- $(L)^\uparrow \odot \varepsilon_4 \sim \varepsilon_1$,

for some n' and ε_4. By the induction hypothesis, we have $L \circledcirc^{n'} \varepsilon_5$ such that $\varepsilon_4 \odot \varepsilon_2 \sim \varepsilon_5$. Thus, LI_HANDLING derives $L \circledcirc^{n'+1} \varepsilon_3$ as required.

Case LI_NOHANDLING**:** We have

- $L \circledcirc^n \varepsilon_4$,
- $(L')^\uparrow \odot \varepsilon_4 \sim \varepsilon_1$, and
- $L \neq L'$,

for some L' and ε_4. By the induction hypothesis, we have $L \circledcirc^n \varepsilon_5$ such that $\varepsilon_4 \odot \varepsilon_2 \sim \varepsilon_5$. Thus, LI_NOHANDLING derives $L \circledcirc^n \varepsilon_3$ as required.

∎

Lemma 3.53. *If $L \circledcirc^{n+1} \varepsilon_2$ and $(L)^\uparrow \odot \varepsilon_1 \sim \varepsilon_2$, then $L \circledcirc^n \varepsilon_1$.*

Proof. By induction on a derivation of $L \circledcirc^{n+1} \varepsilon_2$. We proceed by case analysis on the rule lastly applied to this derivation.
Case LI_EMPTY**:** Cannot happen.

Case LI_HANDLING**:** We have

- $L \circledcirc^n \varepsilon'_1$ and
- $(L)^\uparrow \odot \varepsilon'_1 \sim \varepsilon_2$

for some ε'_1. By safety condition (3), we have $\varepsilon_1 \sim \varepsilon'_1$. By Lemma 3.52 and $\varepsilon'_1 \odot \mathbb{0} \sim \varepsilon_1$, we have $L \circledcirc^n \varepsilon_1$ as required.

Case LI_NOHANDLING**:** We have

- $L \circledcirc^{n+1} \varepsilon_3$,
- $(L')^\uparrow \odot \varepsilon_3 \sim \varepsilon_2$, and
- $L \neq L'$,

for some L' and ε_3. By safety condition (2) and $L \neq L'$, we have $(L)^\uparrow \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4. By safety condition (3), we have $\varepsilon_1 \sim (L')^\uparrow \odot \varepsilon_4$. By the induction hypothesis, we have $L \circledcirc^n \varepsilon_4$. Thus, LI_NOHANDLING derives $L \circledcirc^n \varepsilon_1$ as required.

∎

Lemma 3.54. *If $L \circledcirc^n \varepsilon_2$ and $(L')^\uparrow \odot \varepsilon_1 \sim \varepsilon_2$ and $L \neq L'$, then $L \circledcirc^n \varepsilon_1$.*

Proof. By induction on a derivation of $L \circledcirc^n \varepsilon_2$. We proceed by case analysis on the rule lastly applied to this derivation.
Case LI_EMPTY**:** We have $n = 0$. LI_EMPTY derives $L \circledcirc^0 \varepsilon_1$ as required.

Case LI_HANDLING**:** We have

- $n = n' + 1$,
- $L \circledcirc^{n'} \varepsilon_3$, and
- $(L)^\uparrow \odot \varepsilon_3 \sim \varepsilon_2$,

for some n' and ε_3. By safety condition (2) and $L \neq L'$, we have $(L')^\uparrow \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4. By safety condition (3), we have $\varepsilon_1 \sim (L)^\uparrow \odot \varepsilon_4$. By the induction hypothesis, we have $L \circledcirc^{n'} \varepsilon_4$. Thus, LI_HANDLING derives $L \circledcirc^{n'+1} \varepsilon_1$ as required.

Case LI_NoHandling**:** We have

- $L \otimes^n \varepsilon_3$,
- $(L'')^\uparrow \odot \varepsilon_3 \sim \varepsilon_2$, and
- $L \neq L''$,

for some L'' and ε_3.

If $L' = L''$, then we have $\varepsilon_1 \sim \varepsilon_3$ by safety condition (3). Thus, Lemma 3.52 gives us $L \otimes^n \varepsilon_1$ as required.

If $L' \neq L''$, then we have $(L')^\uparrow \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4 by safety condition (2) and $L' \neq L''$. By safety condition (3), we have $\varepsilon_1 \sim (L'')^\uparrow \odot \varepsilon_4$. By the induction hypothesis, we have $L \otimes^n \varepsilon_4$. Thus, LI_NoHandling derives $L \otimes^n \varepsilon_1$ as required.

∎

Lemma 3.55. *If* $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ *and* $n-\mathrm{free}(l\, \boldsymbol{S}^I, E)$*, then* $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$*.*

Proof. By induction on a derivation of $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.

Case T_App**:** For some B, we have

- $E = \Box$,
- $\emptyset \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J : B \to_\varepsilon A \mid \mathbb{0}$, and
- $\emptyset \vdash v : B \mid \mathbb{0}$.

By Lemma 3.45(4), we have $\emptyset \vdash (l\, \boldsymbol{S}^I)^\uparrow \otimes \varepsilon$. Thus, the required result is achieved.

Case T_Let**:** For some x, E_1, e, and B, we have

- $E = (\mathbf{let}\, x = E_1\, \mathbf{in}\, e)$,
- $\emptyset \vdash E_1[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon$,
- $n-\mathrm{free}(l\, \boldsymbol{S}^I, E_1)$, and
- $x : B \vdash e : A \mid \varepsilon$.

By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$ as required.

Case T_Sub**:** For some A'' and ε'', we have

- $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A' \mid \varepsilon'$ and
- $\emptyset \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon'$. Since only ST_Comp can derive $\emptyset \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\emptyset \vdash \varepsilon' \otimes \varepsilon$. Thus, Lemma 3.52 derives $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$ as required.

Case T_Lift**:** For some L, ε', and E', we have

- $E = [E']_L$,
- $\emptyset \vdash E'[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon'$,
- $\emptyset \vdash L : \mathbf{Lab}$, and
- $(L)^\uparrow \odot \varepsilon' \sim \varepsilon$.

If $l\, \boldsymbol{S}^I \neq L$, then $n-\mathrm{free}(l\, \boldsymbol{S}^I, E')$. By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon'$. LI_NoHandling derives $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$ as required.

If $l\, \boldsymbol{S}^I = L$, then there exists some m such that $n = m + 1$ and $m-\mathrm{free}(l\, \boldsymbol{S}^I, E')$. By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{m+1} \varepsilon'$. LI_Handling derives $l\, \boldsymbol{S}^I \otimes^{m+2} \varepsilon$ as required.

Case T_Handling**:** For some l', $\boldsymbol{S'}^{I'}$, E_1, h, B, and ε', we have

- $E = \mathbf{handle}_{l'\, \boldsymbol{S'}^{I'}}\, E_1\, \mathbf{with}\, h$,
- $\emptyset \vdash E_1[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$, and
- $(l'\, \boldsymbol{S'}^{I'})^\uparrow \odot \varepsilon \sim \varepsilon'$.

If $l\, \boldsymbol{S}^I \neq l'\, \boldsymbol{S'}^{I'}$, then $n-\mathrm{free}(l\, \boldsymbol{S}^I, E_1)$. By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon'$. By Lemma 3.54, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$.

If $l\, \boldsymbol{S}^I = l'\, \boldsymbol{S'}^{I'}$, then $n + 1-\mathrm{free}(l\, \boldsymbol{S}^I, E_1)$. By the induction hypothesis, we have $l\, \boldsymbol{S}^I \otimes^{n+2} \varepsilon'$. By Lemma 3.53, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \varepsilon$.

***Case* others:** Cannot happen. ■

Lemma 3.56 (No Inclusion by Empty Effect)**.** *If $L \otimes^n \varepsilon$ and $\varepsilon \sim \mathbb{0}$, then $n = 0$.*

Proof. By induction on the derivation of $L \otimes^n \varepsilon$. We proceed by case analysis on the rule applied lastly to this derivation.
Case LI_EMPTY**:** Clearly.

Case LI_HANDLING**:** This case cannot happen. If this case happens, we have $(L)^{\uparrow} \odot \varepsilon' \sim \varepsilon$ for some m and ε'. Thus, we have $(L)^{\uparrow} \odot \varepsilon' \sim \mathbb{0}$ by $\varepsilon \sim \mathbb{0}$. However, it is contradictory with safety condition (1).

Case LI_NOHANDLING**:** This case cannot happen. If this case happens, we have $(L')^{\uparrow} \odot \varepsilon' \sim \varepsilon$ for some L' and ε'. Thus, we have $(L')^{\uparrow} \odot \varepsilon' \sim \mathbb{0}$ by $\varepsilon \sim \mathbb{0}$. However, it is contradictory with safety condition (1). ■

Lemma 3.57 (Effect Safety)**.** *If $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and n−free$(l\, \boldsymbol{S}^I, E)$, then $\varepsilon \nsim \mathbb{0}$.*

Proof. Assume that $\varepsilon \sim \mathbb{0}$. By Lemma 3.55 and Lemma 3.52, we have $l\, \boldsymbol{S}^I \otimes^{n+1} \mathbb{0}$. However, it is contradictory with Lemma 3.56. ■

Theorem 3.58 (Type and Effect Safety)**.** *If $\emptyset \vdash e : A \mid \mathbb{0}$ and $e \longrightarrow^* e'$ and $e' \not\longrightarrow$, then e' is a value.*

Proof. Similarly to Theorem 3.24; Lemmas 3.50 , 3.57, and 3.48 are used instead of Lemmas 3.20 , 3.23, and 3.18, respectively. ■

3.4 Properties with Type-Erasure Semantics

This section assumes that the safety conditions in Definition 1.45 and the safety condition for type-erasure semantics in Definition 1.47 hold, and that the semantics adapts R_HANDLE2'instead of R_HANDLE2.

Remark 3.59. *The change of semantics only affects Lemma 3.18, Lemma 3.19, Lemma 3.20, Lemma 3.22, Lemma 3.23, and Theorem 3.24. Therefore, we can use other lemmas in this type-erasure setting.*

Lemma 3.60 (Progress)**.** *If $\emptyset \vdash e : A \mid \varepsilon$, then one of the following holds:*

- *e is a value;*
- *There exists some e' such that $e \longrightarrow e'$; or*
- *There exist some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, and n such that $eE[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v]$ and n−free(l, E).*

Proof. By induction on a derivation of $\emptyset \vdash e : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.
Case T_HANDLING**:** For some l, $\boldsymbol{S}^N$, h, e_1, A_1, ε_1, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, σ, given are the following:

- $e = \mathbf{handle}_{l\, \boldsymbol{S}^N}\, e_1\, \mathbf{with}\, h$,
- $\emptyset \vdash e_1 : A_1 \mid \varepsilon_1$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A_1 \Rightarrow^{\varepsilon} A$, and
- $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon_1$.

By the induction hypothesis, we proceed by case analysis on the following conditions:

(1) e_1 is a value,
(2) There exists some e_1' such that $e_1 \longrightarrow e_1'$, and
(3) There exist some op', l', $\boldsymbol{S'}^{N'}$, $\boldsymbol{T}^J$, v, E, and n such that $e_1 = E[\mathsf{op}'_{l'\, \boldsymbol{S'}^{N'}}\, \boldsymbol{T}^J\, v]$ and n−free(l', E).

***Case* (1):** By Lemma 3.16(1), there exists some x and e_r such that $\mathbf{return}\, x \mapsto e_r \in h$. Thus, R_HANDLE1 derives $e \longmapsto e_r[v_1/x]$ because e_1 is a value v_1.

***Case* (2):** Since only E_EVAL can derive $e_1 \longrightarrow e_1'$, we have

- $e_1 = E_1[e_{11}]$,
- $e_1' = E_1[e_{12}]$, and

- $e_{11} \longmapsto e_{12}$,

for some E_1, e_{11}, and e_{12}. Let $E = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E_1\, \mathbf{with}\, h$, E_Eval derives $e \longrightarrow E[e_{12}]$ because $e = E[e_{11}]$.

***Case* (3):** If $l \neq l'$, then $e = (\mathbf{handle}_{l\,\boldsymbol{S}^N}\, E\, \mathbf{with}\, h)[\mathsf{op}'_{l'\,\boldsymbol{S'}^{N'}}\, \boldsymbol{T}^J\, v]$ and n−free$(l', \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E\, \mathbf{with}\, h)$. If $l = l'$, then by Lemma 3.17 and 3.14(4), we have

- $l' :: \forall \boldsymbol{\alpha'}^{N'} : \boldsymbol{K'}^{N'}.\sigma' \in \Xi$ and
- $\mathsf{op}' : \forall \boldsymbol{\beta'}^J : \boldsymbol{K'_0}^J.A' \Rightarrow B' \in \sigma'[\boldsymbol{S'}^{N'}/\boldsymbol{\alpha'}^{N'}]$,

for some $\boldsymbol{\alpha'}^{N'}$, $\boldsymbol{K'}^{N'}$, σ', $\boldsymbol{\beta'}^J$, A', and B'. Therefore, since $l = l'$, we have

- $\boldsymbol{\alpha}^N = \boldsymbol{\alpha'}^{N'}$,
- $\boldsymbol{K}^N = \boldsymbol{K'}^{N'}$, and
- $\sigma = \sigma'$.

By $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : A_1 \Rightarrow^{\varepsilon} A$, $\mathsf{op}' : \forall \boldsymbol{\beta'}^J : \boldsymbol{K'_0}^J.A'' \Rightarrow B'' \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$ for some A'' and B'', and Lemma 3.16(2), we have

$$\mathsf{op}'\, \boldsymbol{\beta'}^J : \boldsymbol{K'_0}^J\, p\, k \mapsto e' \in h$$

for some p, k, and e'. If $n = 0$, the evaluation of e proceeds by R_Handle2'. Otherwise, there exists some m such that $n = m + 1$ and m−free$(l, \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E\, \mathbf{with}\, h)$.

***Case* others:** Similarly to Lemma 3.18.

■

Lemma 3.61. *If n−free(l, E), then $n = \mathsf{0}$.*

Proof. Straightforward by the induction on the derivation of n−free(l, E). ■

Lemma 3.62. *If $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and n−free(l, E), then $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$.*

Proof. By induction on a derivation of $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.

***Case* T_App:** For some B, we have

- $E = \Box$,
- $\Gamma \vdash \mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J : B \rightarrow_{\varepsilon} A \mid \mathbb{0}$, and
- $\Gamma \vdash v : B \mid \mathbb{0}$.

By Lemma 3.14(4), we have $\Gamma \vdash (l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$. Thus, the required result is achieved.

***Case* T_Let:** For some x, E_1, e, and B, we have

- $E = (\mathbf{let}\, x = E_1\, \mathbf{in}\, e)$,
- $\Gamma \vdash E_1[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon$, and
- $\Gamma, x : B \vdash e : A \mid \varepsilon$.

By the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$ as required.

***Case* T_Sub:** For some A' and ε', we have

- $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A' \mid \varepsilon'$ and
- $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

Since only ST_Comp can derive $\Gamma \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\Gamma \vdash \varepsilon' \otimes \varepsilon$. By the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon'$. By the associativity of $\odot$, we have $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$ as required.

***Case* T_Handling:** For some l', $\boldsymbol{S'}^{I'}$, E_1, h, B, and ε', we have

- $E = \mathbf{handle}_{l'\,\boldsymbol{S'}^{I'}}\, E_1\, \mathbf{with}\, h$,
- $\Gamma \vdash E_1[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$, and
- $(l'\,\boldsymbol{S'}^{I'})^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

By Lemma 3.61, we have $l \neq l'$ and 0−free(l, E_1). By the induction hypothesis, we have $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon'$. Thus, safety condition (2) makes $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$ hold as required.

***Case* others:** Cannot happen.

∎

Lemma 3.63 (Preservation in Reduction). *If* $\emptyset \vdash e : A \mid \varepsilon$ *and* $e \longmapsto e'$, *then* $\emptyset \vdash e' : A \mid \varepsilon$.

Proof. By induction on a derivation of $\Gamma \vdash e : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.

***Case* T_Handling:** We proceed by case analysis on the derivation rule that derives $e \longmapsto e'$.

***Case* R_Handle1:** Similarly to Lemma 3.19.

***Case* R_Handle2':** For some l, $\boldsymbol{S}^N$, E, op_0, $\boldsymbol{S'}^N$, $\boldsymbol{T}^J$, v, h, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, σ, $\boldsymbol{\beta_0}^J$, $\boldsymbol{K_0}^J$, A_0, B_0, p_0, k_0, e_0, B, and ε', we have

- $e = \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[\mathsf{op}_{0\,l\,\boldsymbol{S'}^N}\, \boldsymbol{T}^J\, v]\,\mathbf{with}\, h$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\mathsf{op}_0\, \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J\; p_0\; k_0 \mapsto e_0 \in h$,
- $0\mathrm{-free}(l, E)$,
- $\emptyset \vdash E[\mathsf{op}_{0\,l\,\boldsymbol{S'}^N}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$,
- $(l\,\boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, and
- $e' = e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[z]\,\mathbf{with}\, h/k_0]$.

By Lemma 3.62, we have $(l\,\boldsymbol{S'}^N)^{\uparrow} \otimes \varepsilon'$. Thus, we get $\boldsymbol{S'}^N = \boldsymbol{S}^N$ by $(l\,\boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$ and safety condition (4). By Lemma 3.17, there exist some B_1 and ε_1 such that

- $\emptyset \vdash \mathsf{op}_{0\,l\,\boldsymbol{S}^N}\, \boldsymbol{T}^J\, v : B_1 \mid \varepsilon_1$, and
- for any e' and Γ', if $\Gamma' \vdash e' : B_1 \mid \varepsilon_1$, then $\Gamma' \vdash E[e'] : B \mid \varepsilon'$.

By Lemma 3.14(5), we have $\emptyset \vdash \mathsf{op}_{0\,l\,\boldsymbol{S}^N}\, \boldsymbol{T}^J : A_1 \to_{\varepsilon_1} B_1 \mid \mathbb{0}$ and $\emptyset \vdash v : A_1 \mid \mathbb{0}$ for some A_1. By Lemma 3.14(4) and 3.16(2), we have

- $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash \boldsymbol{T}^J : \boldsymbol{K_0}^J$,
- $\emptyset \vdash A_1 <: A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$,
- $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] <: B_1$, and
- $\emptyset \vdash (l\,\boldsymbol{S}^N)^{\uparrow} \otimes \varepsilon_1$,

for some A_0 and B_0. Thus, T_Sub with $\emptyset \vdash \mathbb{0} \otimes \mathbb{0}$ implied by Lemma 3.3 derives

$$\emptyset \vdash v : A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \mid \mathbb{0}.$$

By Lemma 3.11, we have $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] : \mathbf{Typ}$. Thus, C_Var derives $\vdash z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$. By $\emptyset \vdash \mathbb{0} : \mathbf{Eff}$, $\emptyset \vdash \varepsilon_1 : \mathbf{Eff}$ implied by Lemma 3.12, and $\mathbb{0} \odot \varepsilon_1 \sim \varepsilon_1$, we have $\emptyset \vdash \mathbb{0} \otimes \varepsilon_1$. Since T_Var and T_Sub derives $z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash z : B_1 \mid \varepsilon_1$, we have

$$z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash \mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[z]\,\mathbf{with}\, h : A \mid \varepsilon$$

by the result of Lemma 3.17, Lemma 3.5, and T_Handling. Thus, T_Abs derives

$$\emptyset \vdash \lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[z]\,\mathbf{with}\, h : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \to_{\varepsilon} A \mid \mathbb{0}.$$

Since

$$\boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J, p_0 : A_0, k_0 : B_0 \to_{\varepsilon} A \vdash e_0 : A \mid \varepsilon$$

by $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$ and $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$ and Lemma 3.16(2), Lemma 3.10(5) and Lemma 3.7(5) imply

$$\emptyset \vdash e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\,\boldsymbol{S}^N}\, E[z]\,\mathbf{with}\, h/k_0] : A \mid \varepsilon$$

as required.

***Case* others:** Similarly to Lemma 3.19.

∎

Lemma 3.64 (Preservation). *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longrightarrow e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. Similarly to Lemma 3.20; Lemma 3.63 is used instead of Lemma 3.19. ∎

Lemma 3.65 (Effect Safety). *If $\Gamma \vdash E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and $n-\mathrm{free}(l, E)$, then $\varepsilon \not\sim \mathbb{0}$.*

Proof. Similarly to Lemma 3.23; Lemma 3.62 is used instead of Lemma 3.22. ∎

Theorem 3.66 (Type and Effect Safety). *If $\emptyset \vdash e : A \mid \mathbb{0}$ and $e \longrightarrow^* e'$ and $e' \not\longrightarrow$, then e' is a value.*

Proof. Similarly to Theorem 3.24; Lemmas 3.64 , 3.65, and 3.60 are used instead of Lemmas 3.20 , 3.23, and 3.18, respectively. ∎

3.5 Properties with Lift Coercions and Type-Erasure Semantics

This section assumes that the safety conditions in Definition 1.45 and the safety conditions for type-erasure semantics and lift coercions in Definition 1.47 and 1.46 hold, and that the semantics adapts R_Handle2' instead of R_Handle2.

Lemma 3.67 (Progress). *If $\emptyset \vdash e : A \mid \varepsilon$, then one of the following holds:*

- *e is a value;*
- *There exists some expression e' such that $e \longrightarrow e'$; or*
- *There exist some op, l, $\boldsymbol{S}^I$, $\boldsymbol{T}^J$, v, E, and n such that $e = E[\mathsf{op}_{l\,\boldsymbol{S}^I}\, \boldsymbol{T}^J\, v]$ and $n-\mathrm{free}(l, E)$.*

Proof. Similarly to Lemma 3.48. ∎

Definition 3.68 (Label Inclusion with Type-Erasure).
Label Inclusion with Type-Erasure $\boxed{l \otimes^{\mathcal{P}} \varepsilon}$ *where* $\mathcal{P} ::= \bullet \mid \boldsymbol{S}^I \blacktriangleright \mathcal{P}$

$$\frac{}{l \otimes^{\bullet} \varepsilon}\ \text{LITE\_Empty} \qquad \frac{l \otimes^{\mathcal{P}} \varepsilon_1 \quad (l\, \boldsymbol{S_0}^{I_0})^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2}{l \otimes^{\boldsymbol{S_0}^{I_0} \blacktriangleright \mathcal{P}} \varepsilon_2}\ \text{LITE\_Handling}$$

$$\frac{l \otimes^{\mathcal{P}} \varepsilon_1 \quad (L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2 \quad \forall \boldsymbol{S_0}^{I_0}.(L \neq l\, \boldsymbol{S_0}^{I_0})}{l \otimes^{\mathcal{P}} \varepsilon_2}\ \text{LITE\_NoHandling}$$

If $n = 0$, then $\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n} \blacktriangleright \mathcal{P}$ means $\mathcal{P}$.

Lemma 3.69. *If $l \otimes^{\mathcal{P}} \varepsilon_1$ and $\varepsilon_1 \odot \varepsilon_2 \sim \varepsilon_3$, then $l \otimes^{\mathcal{P}} \varepsilon_3$.*

Proof. By induction on a derivation of $l \otimes^{\mathcal{P}} \varepsilon_1$. We proceed by case analysis on the rule applied lastly to this derivation.
Case LITE_Empty**:** We have $\mathcal{P} = \bullet$. LITE_Empty derives $l \otimes^{\bullet} \varepsilon_2$ as required.

Case LITE_Handling**:** We have

- $\mathcal{P} = \boldsymbol{S}^I \blacktriangleright \mathcal{P}'$,
- $l \otimes^{\mathcal{P}'} \varepsilon_4$, and
- $(l\, \boldsymbol{S}^I)^{\uparrow} \odot \varepsilon_4 \sim \varepsilon_1$,

for some $\mathcal{P}'$, ε_4, and $\boldsymbol{S}^I$. By the induction hypothesis, we have $l \otimes^{\mathcal{P}'} \varepsilon_5$ such that $\varepsilon_4 \odot \varepsilon_2 \sim \varepsilon_5$. Thus, LITE_Handling derives $l \otimes^{\boldsymbol{S}^I \blacktriangleright \mathcal{P}'} \varepsilon_2$ as required.

Case LITE_NoHandling**:** We have

- $l \otimes^{\mathcal{P}} \varepsilon_4$,
- $(L)^{\uparrow} \odot \varepsilon_4 \sim \varepsilon_1$, and
- $\forall \boldsymbol{S}^I.(L \neq l\, \boldsymbol{S}^I)$,

for some L and ε_4. By the induction hypothesis, we have $l \otimes^{\mathcal{P}} \varepsilon_5$ such that $\varepsilon_4 \odot \varepsilon_2 \sim \varepsilon_5$. Thus, LITE_NoHandling derives $l \otimes^{\mathcal{P}} \varepsilon_3$ as required. ∎

Lemma 3.70. *If $l \otimes^{\boldsymbol{S}^I \blacktriangleright \mathcal{P}} \varepsilon_2$ and $(l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2$, then $l \otimes^{\mathcal{P}} \varepsilon_1$.*

Proof. By induction on a derivation of $l \otimes^{\boldsymbol{S}^I \blacktriangleright \mathcal{P}} \varepsilon_2$. We proceed by case analysis on the rule lastly applied to this derivation.
Case LITE_EMPTY**:** Cannot happen.

Case LITE_HANDLING**:** We have

- $l \otimes^{\mathcal{P}} \varepsilon'_1$ and
- $(l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon'_1 \sim \varepsilon_2$

for some ε'_1. By safety condition (3), we have $\varepsilon_1 \sim \varepsilon'_1$. By Lemma 3.69 and $\varepsilon'_1 \odot \mathbb{0} \sim \varepsilon_1$, we have $l \otimes^{\mathcal{P}} \varepsilon_1$ as required.

Case LITE_NOHANDLING**:** We have

- $l \otimes^{\boldsymbol{S}^I \blacktriangleright \mathcal{P}} \varepsilon_3$,
- $(L)^{\uparrow} \odot \varepsilon_3 \sim \varepsilon_2$, and
- $\forall \boldsymbol{S_0}^{I_0}.(L \neq l\,\boldsymbol{S_0}^{I_0})$,

for some L and ε_3. By safety condition (2) and $L \neq l\,\boldsymbol{S}^I$, we have $(l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4. By safety condition (3), we have $\varepsilon_1 \sim (L)^{\uparrow} \odot \varepsilon_4$. By the induction hypothesis, we have $l \otimes^{\mathcal{P}} \varepsilon_4$. Thus, LITE_NOHANDLING derives $l \otimes^{\mathcal{P}} \varepsilon_1$ as required.

∎

Lemma 3.71. *If $l \otimes^{\mathcal{P}} \varepsilon_2$ and $(L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon_2$ and $\forall \boldsymbol{S}^I.(L \neq l\,\boldsymbol{S}^I)$, then $l \otimes^{\mathcal{P}} \varepsilon_1$.*

Proof. By induction on a derivation of $l \otimes^{\mathcal{P}} \varepsilon_2$. We proceed by case analysis on the rule lastly applied to this derivation.
Case LITE_EMPTY**:** We have $\mathcal{P} = \bullet$. LITE_EMPTY derives $l \otimes^{\bullet} \varepsilon_1$ as required.

Case LITE_HANDLING**:** We have

- $\mathcal{P} = \boldsymbol{S}^I \blacktriangleright \mathcal{P}'$,
- $l \otimes^{\mathcal{P}'} \varepsilon_3$, and
- $(l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon_3 \sim \varepsilon_2$,

for some $\mathcal{P}'$, ε_3, and $\boldsymbol{S}^I$. By safety condition (2) and $L \neq l\,\boldsymbol{S}^I$, we have $(L)^{\uparrow} \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4. By safety condition (3), we have $\varepsilon_1 \sim (l\,\boldsymbol{S}^I)^{\uparrow} \odot \varepsilon_4$. By the induction hypothesis, we have $l \otimes^{\mathcal{P}'} \varepsilon_4$. Thus, LITE_HANDLING derives $l \otimes^{\boldsymbol{S}^I \blacktriangleright \mathcal{P}'} \varepsilon_1$ as required.

Case LITE_NOHANDLING**:** We have

- $l \otimes^{\mathcal{P}} \varepsilon_3$,
- $(L')^{\uparrow} \odot \varepsilon_3 \sim \varepsilon_2$, and
- $\forall \boldsymbol{S}^I.(L' \neq l\,\boldsymbol{S}^I)$,

for some L' and ε_3.

If $L = L'$, then we have $\varepsilon_1 \sim \varepsilon_3$ by safety condition (3). Thus, Lemma 3.69 gives us $l \otimes^{\mathcal{P}} \varepsilon_1$ as required.

If $L \neq L'$, then we have $(L)^{\uparrow} \odot \varepsilon_4 \sim \varepsilon_3$ for some ε_4 by safety condition (2) and $L \neq L'$. By safety condition (3), we have $\varepsilon_1 \sim (L')^{\uparrow} \odot \varepsilon_4$. By the induction hypothesis, we have $l \otimes^{\mathcal{P}'} \varepsilon_4$. Thus, LITE_NOHANDLING derives $l \otimes^{\mathcal{P}} \varepsilon_1$ as required.

∎

Lemma 3.72. *If $l \otimes^{\boldsymbol{S_0}^{I_0} \blacktriangleright \mathcal{P}} \varepsilon$ and $(l\,\boldsymbol{S}^I)^{\uparrow} \otimes \varepsilon$, then $\boldsymbol{S}^I = \boldsymbol{S_0}^{I_0}$.*

Proof. By induction on a derivation of $l \otimes^{\boldsymbol{S_0}^{I_0} \blacktriangleright \mathcal{P}} \varepsilon$. We proceed by case analysis on the rule lastly applied to this derivation.
Case LITE_EMPTY**:** Cannot happen.

Case LITE_HANDLING**:** We have

- $l \otimes^{\mathcal{P}} \varepsilon_1$ and
- $(l\,\boldsymbol{S_0}^{I_0})^{\uparrow} \odot \varepsilon_1 \sim \varepsilon$

for some ε_1. By safety condition (4), we have $\boldsymbol{S}^I = \boldsymbol{S_0}^{I_0}$ as required.

Case LITE_NoHandling**:** We have

- $l \olessthan^{\boldsymbol{S_0}^{I_0} \blacktriangleright \mathcal{P}} \varepsilon_1$,
- $(L)^{\uparrow} \odot \varepsilon_1 \sim \varepsilon$, and
- $\forall \boldsymbol{S'}^{I'}.(L \neq \boldsymbol{S'}^{I'})$

for some L and ε_1. By safety condition (2) and $L \neq l\, \boldsymbol{S}^I$, we have $(l\, \boldsymbol{S}^I)^{\uparrow} \olessthan \varepsilon_1$. Thus, by the induction hypothesis, we have $\boldsymbol{S}^I = \boldsymbol{S_0}^{I_0}$ as required.

■

Lemma 3.73. *If* $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ *and* $n\text{−free}(l, E)$*, then* $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$*.*

Proof. By induction on a derivation of $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$. We proceed by case analysis on the typing rule applied lastly to this derivation.

Case T_App**:** For some B, we have

- $E = \Box$,
- $\emptyset \vdash \mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J : B \rightarrow_{\varepsilon} A \mid \mathbb{0}$, and
- $\emptyset \vdash v : B \mid \mathbb{0}$.

By Lemma 3.45(4), we have $\emptyset \vdash (l\, \boldsymbol{S}^I)^{\uparrow} \olessthan \varepsilon$. Thus, LITE_Empty and LITE_Handling derive $l \olessthan^{\boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$.

Case T_Let**:** For some x, E_1, e, and B, we have

- $E = (\mathbf{let}\ x = E_1\ \mathbf{in}\ e)$,
- $\emptyset \vdash E_1[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon$,
- $n\text{−free}(l, E_1)$, and
- $x : B \vdash e : A \mid \varepsilon$.

By the induction hypothesis, we have $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$ as required.

Case T_Sub**:** For some A' and ε', we have

- $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A' \mid \varepsilon'$ and
- $\emptyset \vdash A' \mid \varepsilon' <: A \mid \varepsilon$.

By the induction hypothesis, we have $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon'$. Since only ST_Comp can derives $\emptyset \vdash A' \mid \varepsilon' <: A \mid \varepsilon$, we have $\emptyset \vdash \varepsilon' \olessthan \varepsilon$. Thus, Lemma 3.69 derives $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$ as required.

Case T_Lift**:** For some L, ε', and E', we have

- $E = [E']_L$,
- $\emptyset \vdash E'[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon'$,
- $\emptyset \vdash L : \mathbf{Lab}$, and
- $(L)^{\uparrow} \odot \varepsilon' \sim \varepsilon$.

If $L \neq l\, \boldsymbol{S'}^{I'}$ for any $\boldsymbol{S'}^{I'}$, then we have $n\text{−free}(l, E')$. By the induction hypothesis, we have $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon'$. Thus, LITE_NoHandling derives $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$ as required.

If $L = l\, \boldsymbol{S'}^{I'}$ for some $\boldsymbol{S'}^{I'}$, then there exists some m such that $n = m+1$ and $m\text{−free}(l, E')$. By the induction hypothesis, we have $l \olessthan^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_m}^{I_m}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon'$. Thus, LITE_Handling derives $l \olessthan^{\boldsymbol{S'}^{I'}, \boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_m}^{I_m}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$ as required.

Case T_Handling**:** For some l', $\boldsymbol{S'}^{I'}$, E_1, h, B, and ε', we have

- $E = \mathbf{handle}_{l'\, \boldsymbol{S'}^{I'}}\ E_1\ \mathbf{with}\ h$,
- $\emptyset \vdash E_1[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$, and
- $(l'\, \boldsymbol{S'}^{I'})^{\uparrow} \odot \varepsilon \sim \varepsilon'$.

If $l \neq l'$, then $n-\mathrm{free}(l, E_1)$. By the induction hypothesis, we have $l \otimes^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright\bullet} \varepsilon'$. By Lemma 3.71, we have $l \otimes^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright\bullet} \varepsilon$ as required.

If $l = l'$, then $n+1-\mathrm{free}(l, E_1)$. By the induction hypothesis, we have $l \otimes^{\boldsymbol{S_0}^{I_0}, \boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright\bullet} \varepsilon'$. By Lemma 3.72, we have $\boldsymbol{S_0}^{I_0} = \boldsymbol{S'}^{I'}$. By Lemma 3.70, we have $l \otimes^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright\bullet} \varepsilon$ as required.

***Case* others:** Cannot happen.

■

Lemma 3.74 (Preservation in Reduction). *If* $\emptyset \vdash e : A \mid \varepsilon$ *and* $e \longmapsto e'$, *then* $\emptyset \vdash e' : A \mid \varepsilon$.

Proof. By induction on a derivation of $\Gamma \vdash e : A \mid \varepsilon$. We proceed by cases on the typing rule applied lastly to this derivation.

***Case* T_Handling:** We proceed by cases on the derivation rule which derives $e \longmapsto e'$.

***Case* R_Handle1:** Similarly to Lemma 3.49.

***Case* R_Handle2:** We have

- $e = \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[\mathsf{op}_{0\, l\, \boldsymbol{S'}^N}\, \boldsymbol{T}^J\, v]\, \mathbf{with}\, h$,
- $l :: \forall \boldsymbol{\alpha}^N : \boldsymbol{K}^N.\sigma \in \Xi$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\mathsf{op}_0\, \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J\, p_0\, k_0 \mapsto e_0 \in h$,
- $\emptyset \vdash E[\mathsf{op}_{0\, l\, \boldsymbol{S'}^N}\, \boldsymbol{T}^J\, v] : B \mid \varepsilon'$,
- $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$,
- $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$,
- $0-\mathrm{free}(l, E)$, and
- $e' = e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[z]\, \mathbf{with}\, h/k_0]$

for some l, $\boldsymbol{S}^N$, E, op_0, $\boldsymbol{S'}^N$, $\boldsymbol{T}^J$, v, h, $\boldsymbol{\alpha}^N$, $\boldsymbol{K}^N$, σ, $\boldsymbol{\beta_0}^J$, $\boldsymbol{K_0}^J$, p_0, k_0, e_0, B, and ε'. By Lemma 3.73, we have $l \otimes^{\boldsymbol{S'}^N \blacktriangleright\bullet} \varepsilon'$. By Lemma 3.72 and $(l\, \boldsymbol{S}^N)^{\uparrow} \odot \varepsilon \sim \varepsilon'$, we have $\boldsymbol{S}^N = \boldsymbol{S'}^N$. By Lemma 3.47, there exist some B_1 and ε_1 such that

- $\emptyset \vdash \mathsf{op}_{0\, l\, \boldsymbol{S}^N}\, \boldsymbol{T}^J\, v : B_1 \mid \varepsilon_1$, and
- for any e'' and Γ'', if $\Gamma'' \vdash e'' : B_1 \mid \varepsilon_1$, then $\Gamma'' \vdash E[e''] : B \mid \varepsilon'$.

By Lemma 3.45(5), we have $\emptyset \vdash \mathsf{op}_{0\, l\, \boldsymbol{S}^N}\, \boldsymbol{T}^J : A_1 \rightarrow_{\varepsilon_1} B_1 \mid \mathbb{0}$ and $\emptyset \vdash v : A_1 \mid \mathbb{0}$ for some A_1. By Lemma 3.45(4) and 3.16(2), we have

- $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$,
- $\emptyset \vdash \boldsymbol{S}^N : \boldsymbol{K}^N$,
- $\emptyset \vdash \boldsymbol{T}^J : \boldsymbol{K_0}^J$,
- $\emptyset \vdash A_1 <: A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$,
- $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] <: B_1$, and
- $\emptyset \vdash (l\, \boldsymbol{S}^N)^{\uparrow} \otimes \varepsilon_1$,

for some A_0 and B_0. Thus, T_Sub with $\emptyset \vdash \mathbb{0} \otimes \mathbb{0}$ implied by Lemma 3.3 derives

$$\emptyset \vdash v : A_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \mid \mathbb{0}.$$

By Lemma 3.11, we have $\emptyset \vdash B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] : \mathbf{Typ}$. Thus, C_Var derives $\vdash z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J]$. By $\emptyset \vdash \mathbb{0} : \mathbf{Eff}$, $\emptyset \vdash \varepsilon_1 : \mathbf{Eff}$ implied by Lemma 3.12, and $\mathbb{0} \odot \varepsilon_1 \sim \varepsilon_1$, we have $\emptyset \vdash \mathbb{0} \otimes \varepsilon_1$. Since T_Var and T_Sub derives $z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash z : B_1 \mid \varepsilon_1$, we have

$$z : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \vdash \mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[z]\, \mathbf{with}\, h : A \mid \varepsilon$$

by the result of Lemma 3.17, Lemma 3.5, and T_Handling. Thus, T_Abs derives

$$\emptyset \vdash \lambda z.\mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[z]\, \mathbf{with}\, h : B_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J] \rightarrow_{\varepsilon} A \mid \mathbb{0}.$$

Since

$$\boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J, p_0 : A_0, k_0 : B_0 \rightarrow_{\varepsilon} A \vdash e_0 : A \mid \varepsilon$$

by $\emptyset \vdash_{\sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]} h : B \Rightarrow^{\varepsilon} A$ and $\mathsf{op}_0 : \forall \boldsymbol{\beta_0}^J : \boldsymbol{K_0}^J.A_0 \Rightarrow B_0 \in \sigma[\boldsymbol{S}^N/\boldsymbol{\alpha}^N]$ and Lemma 3.16(2), Lemma 3.10(5) and Lemma 3.7(5) imply

$$\emptyset \vdash e_0[\boldsymbol{T}^J/\boldsymbol{\beta_0}^J][v/p_0][\lambda z.\mathbf{handle}_{l\, \boldsymbol{S}^N}\, E[z]\, \mathbf{with}\, h/k_0] : A \mid \varepsilon$$

as required.

***Case* others:** Similarly to Lemma 3.49.

■

Lemma 3.75 (Preservation)**.** *If $\emptyset \vdash e : A \mid \varepsilon$ and $e \longrightarrow e'$, then $\emptyset \vdash e' : A \mid \varepsilon$.*

Proof. Similarly to Lemma 3.50; Lemma 3.74 is used instead of Lemma 3.49. ■

Lemma 3.76 (No Inclusion by Empty Effect)**.** *If $l \circledless^{\mathcal{P}} \varepsilon$ and $\varepsilon \sim \mathbb{0}$, then $\mathcal{P} = \bullet$.*

Proof. By induction on the derivation of $l \circledless^{\mathcal{P}} \varepsilon$. We proceed by case analysis on the rule applied lastly to this derivation.
***Case* LITE_EMPTY:** Clearly.

***Case* LITE_HANDLING:** This case cannot happen. If this case happens, we have $(l\, \boldsymbol{S_0}^{I_0})^{\uparrow} \odot \varepsilon' \sim \varepsilon$ for some ε' and $\boldsymbol{S_0}^{I_0}$. Thus, we have $(l\, \boldsymbol{S_0}^{I_0})^{\uparrow} \odot \varepsilon' \sim \mathbb{0}$ by $\varepsilon \sim \mathbb{0}$. However, it is contradictory with safety condition (1).

***Case* LITE_NOHANDLING:** This case cannot happen. If this case happens, we have $(L)^{\uparrow} \odot \varepsilon' \sim \varepsilon$ for some L and ε'. Thus, we have $(L)^{\uparrow} \odot \varepsilon' \sim \mathbb{0}$ by $\varepsilon \sim \mathbb{0}$. However, it is contradictory with safety condition (1).

■

Lemma 3.77 (Effect Safety)**.** *If $\emptyset \vdash E[\mathsf{op}_{l\, \boldsymbol{S}^I}\, \boldsymbol{T}^J\, v] : A \mid \varepsilon$ and $n-$free$(l\, \boldsymbol{S}^I, E)$, then $\varepsilon \nsim \mathbb{0}$.*

Proof. Assume that $\varepsilon \sim \mathbb{0}$. By Lemma 3.73 and Lemma 3.69, we have $l \circledless^{\boldsymbol{S_1}^{I_1} \blacktriangleright \cdots \blacktriangleright \boldsymbol{S_n}^{I_n}, \boldsymbol{S}^I \blacktriangleright \bullet} \varepsilon$. However, it is contradictory with Lemma 3.76. ■

Theorem 3.78 (Type and Effect Safety)**.** *If $\emptyset \vdash e : A \mid \mathbb{0}$ and $e \longrightarrow^{*} e'$ and $e' \not\longrightarrow$, then e' is a value.*

Proof. Similarly to Theorem 3.58; Lemmas 3.75 , 3.77, and 3.67 are used instead of Lemmas 3.50 , 3.57, and 3.48, respectively. ■

3.6 Safety Conditions about Instances

Lemma 3.79. *In Example 1.23, we write a and b to denote $\{\}$ or ρ or $\{L\}$. If $a_1 \sqcup \cdots \sqcup a_m \sim_{\mathrm{Set}} b_1 \sqcup \cdots \sqcup b_n$, then*

- *for any $i \in \{1, \ldots, m\}$, $a_i = \{\}$ or there exists some j such that $a_i = b_j$, and*
- *for any $j \in \{1, \ldots, n\}$, $b_j = \{\}$ or there exists some i such that $a_i = b_j$.*

Proof. By induction on the derivation of $a_1 \sqcup \cdots \sqcup a_m \sim_{\mathrm{Set}} b_1 \sqcup \cdots \sqcup b_n$. ■

Theorem 3.80. *Example 1.23 meets safety conditions.*

Proof.
(1) Clearly by Lemma 3.79.

(2) Clearly by Lemma 3.79.

■

Lemma 3.81. *In Example 1.24, we write a and b to denote $\{\}$ or ρ or $\{L\}$. If $a_1 \sqcup \cdots \sqcup a_m \sim_{\mathrm{MSet}} b_1 \sqcup \cdots \sqcup b_n$, then*

- *for any a such that $a \neq \{\}$, the number of a_i such that $a_i = a$ is equal to the number of b_j such that $b_i = a$.*

Proof. By induction on the derivation of $a_1 \sqcup \cdots \sqcup a_m \sim_{\mathrm{MSet}} b_1 \sqcup \cdots \sqcup b_n$. ■

Theorem 3.82. *Example 1.24 meets safety conditions (for lift coercions).*

Proof.
(1) Clearly by Lemma 3.81.

(2) Clearly by Lemma 3.81.

(3) Clearly by Lemma 3.81.

■

Lemma 3.83. *In Example 1.25, we write a and b to denote $\langle\rangle$ or ρ. If $\langle L_1 \mid \langle \cdots \langle L_m \mid a \rangle \cdots \rangle\rangle \sim_{\mathrm{SimpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b \rangle \cdots \rangle\rangle$, then*

- *$a = b$,*

- *for any $i \in \{1, \ldots, m\}$, there exists some j such that $L_i = L'_j$, and*

- *for any $j \in \{1, \ldots, n\}$, there exists some i such that $L_i = L'_j$.*

Proof. By induction on the derivation of $\langle L_1 \mid \langle \cdots \langle L_m \mid a \rangle \cdots \rangle\rangle \sim_{\text{SimpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b \rangle \cdots \rangle\rangle$. ■

Theorem 3.84. *Example 1.25 meets safety conditions.*

Proof.

(1) Clearly by Lemma 3.83.

(2) Clearly by Lemma 3.83.

■

Lemma 3.85. *In Example 1.26, we write a and b to denote $\langle\rangle$ or ρ. If $\langle L_1 \mid \langle \cdots \langle L_m \mid a \rangle \cdots \rangle\rangle \sim_{\text{ScpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b \rangle \cdots \rangle\rangle$, then*

- *$a = b$ and*

- *for any L, the number of L_i such that $L_i = L$ is equal to the number of L'_j such that $L'_j = L$.*

Proof. By induction on the derivation of $\langle L_1 \mid \langle \cdots \langle L_m \mid a \rangle \cdots \rangle\rangle \sim_{\text{ScpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b \rangle \cdots \rangle\rangle$. ■

Theorem 3.86. *Example 1.26 meets safety conditions (for lift coercions).*

Proof.

(1) Clearly by Lemma 3.85.

(2) Clearly by Lemma 3.85.

(3) Clearly by Lemma 3.85. ■

Lemma 3.87. *In Example 1.27, we write a and b to denote $\{\}$ or ρ or $\{L\}$. If $a_1 \sqcup \cdots \sqcup a_m \sim_{\text{ESet}} b_1 \sqcup \cdots \sqcup b_n$, then*

- *for any $i \in \{1, \ldots, m\}$, $a_i = \{\}$ or there exists some j such that $a_i = b_j$ or label names of them are the same, and*

- *for any $j \in \{1, \ldots, n\}$, $b_j = \{\}$ or there exists some i such that $a_i = b_j$ or label names of them are the same.*

Proof. By induction on the derivation of $a_1 \sqcup \cdots \sqcup a_m \sim_{\text{ESet}} b_1 \sqcup \cdots \sqcup b_n$. ■

Lemma 3.88. *In Example 1.27, we define the function FO as follows:*

$$FO(l, \{\}) = \bot \qquad FO(l, \{\iota\}) = \bot \qquad FO(l, \rho) = \bot \qquad FO(l, \{l\, \boldsymbol{S}^I\}) = \boldsymbol{S}^I \qquad FO(l, \{l'\, \boldsymbol{S}^I\}) = \bot \quad (\textit{where } l \neq l')$$

$$FO(l, \varepsilon_1 \sqcup \varepsilon_2) = \begin{cases} FO(l, \varepsilon_2) & (\textit{if } FO(l, \varepsilon_1) = \bot) \\ FO(l, \varepsilon_1) & (\textit{otherwise}) \end{cases}$$

If $\varepsilon_1 \sim_{\text{ESet}} \varepsilon_2$, then for any l, $FO(l, \varepsilon_1) = FO(l, \varepsilon_2)$.

Proof. By induction on the derivation of $\varepsilon_1 \sim_{\text{ESet}} \varepsilon_2$. ■

Theorem 3.89. *Example 1.27 meets safety conditions.*

Proof.

(1) Clearly by Lemma 3.87.

(2) Clearly by Lemma 3.87 and 3.88.

(4) Clearly by Lemma 3.88.

■

Lemma 3.90. *In Example 1.28, we write a and b to denote $\{\}$ or ρ or $\{L\}$. If $a_1 \sqcup \cdots \sqcup a_m \sim_{\text{EMSet}} b_1 \sqcup \cdots \sqcup b_n$, then*

- *for any a such that $a \neq \{\}$, the number of a_i such that $a_i = a$ is equal to the number of b_j such that $b_i = a$.*

Proof. By induction on the derivation of $a_1 \sqcup \cdots \sqcup a_m \sim_{\text{EMSet}} b_1 \sqcup \cdots \sqcup b_n$. ∎

Lemma 3.91. *In Example 1.28, we define the function FO as follows:*

$$FO(l, \{\}) = \bot \quad FO(l, \{\iota\}) = \bot \quad FO(l, \rho) = \bot \quad FO(l, \{l\, \boldsymbol{S}^I\}) = \boldsymbol{S}^I \quad FO(l, \{l'\, \boldsymbol{S}^I\}) = \bot \quad (\textit{where } l \neq l')$$

$$FO(l, \varepsilon_1 \sqcup \varepsilon_2) = \begin{cases} FO(l, \varepsilon_2) & (\textit{if } FO(l, \varepsilon_1) = \bot) \\ FO(l, \varepsilon_1) & (\textit{otherwise}) \end{cases}$$

If $\varepsilon_1 \sim_{\text{EMSet}} \varepsilon_2$, then for any l, $FO(l, \varepsilon_1) = FO(l, \varepsilon_2)$.

Proof. By induction on the derivation of $\varepsilon_1 \sim_{\text{EMSet}} \varepsilon_2$. ∎

Theorem 3.92. *Example 1.28 meets safety conditions.*

Proof.
(1) Clearly by Lemma 3.90.

(2) Clearly by Lemma 3.90.

(4) Clearly by Lemma 3.91.

∎

Lemma 3.93. *In Example 1.29, we write a and b to denote $\langle\rangle$ or ρ. If $\langle L_1 \mid \langle \cdots \langle L_m \mid a\rangle \cdots \rangle\rangle \sim_{\text{ESimpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b\rangle \cdots \rangle\rangle$, then*
- *$a = b$,*
- *for any $i \in \{1, \ldots, m\}$, there exists some j such that $L_i = L'_j$ or label names of them are the same, and*
- *for any $j \in \{1, \ldots, n\}$, there exists some i such that $L_i = L'_j$ or label names of them are the same.*

Proof. By induction on the derivation of $\langle L_1 \mid \langle \cdots \langle L_m \mid a\rangle \cdots \rangle\rangle \sim_{\text{ESimpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b\rangle \cdots \rangle\rangle$. ∎

Lemma 3.94. *In Example 1.29, we define the function FO as follows:*

$$FO(l, \langle\rangle) = \bot \quad FO(l, \rho) = \bot \quad FO(l, \langle l\, \boldsymbol{S}^I \mid \varepsilon\rangle) = \boldsymbol{S}^I \quad FO(l, \langle l'\, \boldsymbol{S}^I \mid \varepsilon\rangle) = FO(l, \varepsilon) \quad (\textit{where } l \neq l')$$
$$FO(l, \langle \iota \mid \varepsilon\rangle) = FO(l, \varepsilon)$$

If $\varepsilon_1 \sim_{\text{ESimpR}} \varepsilon_2$, then for any l, $FO(l, \varepsilon_1) = FO(l, \varepsilon_2)$.

Proof. By induction on the derivation of $\varepsilon_1 \sim_{\text{ESimpR}} \varepsilon_2$. ∎

Theorem 3.95. *Example 1.29 meets safety conditions (for type-erasure).*

Proof.
(1) Clearly by Lemma 3.93.

(2) Clearly by Lemma 3.93 and 3.94.

(4) Clearly by Lemma 3.94.

∎

Lemma 3.96. *In Example 1.30, we write a and b to denote $\langle\rangle$ or ρ. If $\langle L_1 \mid \langle \cdots \langle L_m \mid a\rangle \cdots \rangle\rangle \sim_{\text{EScpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b\rangle \cdots \rangle\rangle$, then*
- *$a = b$ and*
- *for any L, the number of L_i such that $L_i = L$ is equal to the number of L'_j such that $L'_j = L$.*

Proof. By induction on the derivation of $\langle L_1 \mid \langle \cdots \langle L_m \mid a\rangle \cdots \rangle\rangle \sim_{\text{EScpR}} \langle L'_1 \mid \langle \cdots \langle L'_m \mid b\rangle \cdots \rangle\rangle$. ∎

Lemma 3.97. *In Example 1.30, we define the function FO as follows:*

$$FO(l, \langle\rangle) = \bot \quad FO(l, \rho) = \bot \quad FO(l, \langle l\, \boldsymbol{S}^I \mid \varepsilon\rangle) = \boldsymbol{S}^I \quad FO(l, \langle l'\, \boldsymbol{S}^I \mid \varepsilon\rangle) = FO(l, \varepsilon) \quad (\textit{where } l \neq l')$$
$$FO(l, \langle \iota \mid \varepsilon\rangle) = FO(l, \varepsilon)$$

If $\varepsilon_1 \sim_{\text{EScpR}} \varepsilon_2$, then for any l, $FO(l, \varepsilon_1) = FO(l, \varepsilon_2)$.

Proof. By induction on the derivation of $\varepsilon_1 \sim_{\mathrm{EScpR}} \varepsilon_2$. ∎

Theorem 3.98. *Example 1.30 meets safety conditions (for lift coercions and type-erasure).*

Proof.

(1) Clearly by Lemma 3.96.

(2) Clearly by Lemma 3.96.

(3) Clearly by Lemma 3.96.

(4) Clearly by Lemma 3.97.

∎

Theorem 3.99 (Unsafe Effect Algebras with Lift Coercions)**.** *The effect algebras* $\mathrm{EA}_{\mathrm{Set}}$ *and* $\mathrm{EA}_{\mathrm{SimpR}}$ *do not meet safety condition (3). Furthermore, there exists an expression such that it is well typed under* $\mathrm{EA}_{\mathrm{Set}}$ *and* $\mathrm{EA}_{\mathrm{SimpR}}$*, but its evaluation gets stuck.*

Proof. We consider only $\mathrm{EA}_{\mathrm{Set}}$ here; a similar discussion can be applied to $\mathrm{EA}_{\mathrm{SimpR}}$. Recall that the operation $\odot$ in $\mathrm{EA}_{\mathrm{Set}}$ is implemented by the set union, so it meets idempotence: $\{L\} \cup \{L\} \sim \{L\}$. Furthermore, we can use the empty set as the identity element, so $\{L\} \cup \{L\} \sim \{L\} \cup \{\}$. If safety condition (3) was met, $\{L\} \sim \{\}$ (where $\{L\}$, $\{\}$, and 0 are taken as ε_1, ε_2, and n, respectively, in Definition 1.46). However, the equivalence does not hold.

As a program that is typeable under $\mathrm{EA}_{\mathrm{Set}}$, consider $\mathbf{handle}_{\mathsf{Exc}}\,[\mathsf{raise}_{\mathsf{Exc}}\,\mathsf{Unit}\,()]_{\mathsf{Exc}}\,\mathbf{with}\,h$ where $\mathsf{Exc} :: \{\mathsf{raise} : \forall \alpha : \mathbf{Typ}.\mathsf{Unit} \Rightarrow \alpha\}$. This program can be typechecked under an appropriate assumption as illustrated by the following typing derivation:

$$\dfrac{\cdots \quad \{\mathsf{Exc}\} \cup \{\} \sim \{\mathsf{Exc}\} \quad \dfrac{\emptyset \vdash \mathsf{raise}_{\mathsf{Exc}}\,\mathsf{Unit}\,() : A \mid \{\mathsf{Exc}\} \quad \{\mathsf{Exc}\} \cup \{\mathsf{Exc}\} \sim \{\mathsf{Exc}\}}{\emptyset \vdash [\mathsf{raise}_{\mathsf{Exc}}\,\mathsf{Unit}\,()]_{\mathsf{Exc}} : A \mid \{\mathsf{Exc}\}}\ \textsc{T\_Lift}}{\emptyset \vdash \mathbf{handle}_{\mathsf{Exc}}\,[\mathsf{raise}_{\mathsf{Exc}}\,\mathsf{Unit}\,()]_{\mathsf{Exc}}\,\mathbf{with}\,h : B \mid \{\}}\ \textsc{T\_Handling}$$

However, the call to raise is not handled because it needs to be handled by the *second* closest effect handler. ∎

Theorem 3.100 (Unsafe Effect Algebras in Type-Erasure Semantics)**.** *The effect algebras* $\mathrm{EA}_{\mathrm{Set}}$*,* $\mathrm{EA}_{\mathrm{MSet}}$*,* $\mathrm{EA}_{\mathrm{SimpR}}$*, and* $\mathrm{EA}_{\mathrm{ScpR}}$ *do not meet safety condition (4). Furthermore, there exists an expression that is well typed under these algebras and gets stuck.*

Proof. Here we focus on the effect algebra $\mathrm{EA}_{\mathrm{Set}}$, but a similar discussions can be applied to the other algebras. Recall that $\odot$ in $\mathrm{EA}_{\mathrm{Set}}$ is implemented by the union operation for sets, and therefore it is commutative (i.e., it allows exchanging labels in a set no matter what label names and what type arguments are in the labels). Hence, for example, $\{l\,\mathsf{Int}\} \cup \{l\,\mathsf{Bool}\} \sim_{\mathrm{Set}} \{l\,\mathsf{Bool}\} \cup \{l\,\mathsf{Int}\}$ for a label name l taking one type parameter. It means that $\mathrm{EA}_{\mathrm{Set}}$ violates safety condition (4).

To give a program that is typeable under $\mathrm{EA}_{\mathrm{Set}}$ but unsafe in the type-erasure semantics, consider the following which uses an effect label $\mathsf{Writer} :: \forall \alpha : \mathbf{Typ}.\{\mathsf{tell} : \alpha \Rightarrow \mathsf{Unit}\}$:

$\mathbf{handle}_{\mathsf{Writer\,Int}}\ \mathbf{handle}_{\mathsf{Writer\,Bool}}$
$\quad\quad \mathsf{tell}_{\mathsf{Writer\,Int}}\ 42$
$\quad \mathbf{with}\,\{\,\mathbf{return}\,x \mapsto 0\} \uplus \{\mathsf{tell}\ p\ k \mapsto \mathbf{if}\,p\,\mathbf{then}\,0\,\mathbf{else}\,42\}$
$\mathbf{with}\,\{\,\mathbf{return}\,x \mapsto x\} \uplus \{\mathsf{tell}\ p\ k \mapsto p\}$

This program is well typed because

- the operation call $\mathsf{tell}_{\mathsf{Writer\,Int}}\ 42$ can have effect $\{\mathsf{Writer\,Bool}\} \cup \{\mathsf{Writer\,Int}\}$ via subeffecting $\{\mathsf{Writer\,Int}\} \oslash \{\mathsf{Writer\,Bool}\} \cup \{\mathsf{Writer\,Int}\}$ (which holds because $\mathsf{Writer\,Int}$ and $\mathsf{Writer\,Bool}$ are exchangeable),
- the inner handling expression is well typed and its effect is $\{\mathsf{Writer\,Int}\}$, and
- the outer one is well typed and its effect is $\{\}$.

Note that this typing rests on the fact that the inner handler assumes that the argument variable p of its tell clause will be replaced by Boolean values as indicated by the type argument Bool to Writer. However, this program reaches the stuck state: because the operation call is handled by the innermost handler for the label *name* Writer, the inner handler is chosen and then the Boolean parameter p of the tell clause in it will be replaced by integer 42. ∎

4 Comparison of Instances and Previous Work

4.1 Comparison to [Pretnar(2015)]

We define the targets of comparison: one is an instance of λ_{EA} (Example 1.23), and another is a minor changed language of [Pretnar(2015)].

Definition 4.1 (Minor Changed Version of [Pretnar(2015)]). *Change list:*

- *removing Boolean and if expressions,*
- *removing handlers from values and handler types from types,*
- *adding well-formedness of contexts and type, and*
- *adding well-formedness of dirt to the return rule.*

The syntax of a minor changed version of [Pretnar(2015)] is as follows.

$$
\begin{array}{rcll}
A, B & ::= & A \to \underline{C} & \textit{(value types)} \\
\underline{C}, \underline{D} & ::= & A!\Delta & \textit{(computation types)} \\
\Delta & ::= & \{\mathsf{op}_1, \ldots, \mathsf{op}_n\} & \textit{(dirt)} \\
v & ::= & x \mid \mathbf{fun}\, x \mapsto c & \textit{(values)} \\
c & ::= & \mathbf{return}\, v \mid \mathsf{op}(v; y.c) \mid \mathbf{do}\, x \leftarrow c_1 \,\mathbf{in}\, c_2 \mid v_1\, v_2 \mid & \textit{(computation)} \\
 & & \mathbf{with}\, h\, \mathbf{handle}\, c & \\
h & ::= & \mathbf{handler}\, \{\mathbf{return}\, x \mapsto c_r, \mathsf{op}_1(x_1; k_1) \mapsto c_1, \ldots, \mathsf{op}_n(x_n; k_n) \mapsto c_n\} & \textit{(handlers)} \\
\Sigma & ::= & \{\mathsf{op}_1 : A_1 \to B_1, \ldots, \mathsf{op}_n : A_n \to B_n\} & \textit{(signature)} \\
\Gamma & ::= & \emptyset \mid \Gamma, x : A & \textit{(typing contexts)}
\end{array}
$$

Well-formedness rules consist of the following.

Contexts Well-formedness $\boxed{\vdash \Gamma}$

$$
\frac{}{\vdash \emptyset}\ \textsc{Cp\_Empty} \qquad \frac{x \notin \mathrm{dom}(\Gamma) \quad \Gamma \vdash A}{\vdash \Gamma, x : A}\ \textsc{Cp\_Var}
$$

Kinding $\boxed{\Gamma \vdash A}$

$$
\frac{\Gamma \vdash A \quad \Delta \subseteq \mathrm{dom}(\Sigma) \quad \Gamma \vdash B}{\Gamma \vdash A \to B!\Delta}\ \textsc{Kp\_Fun}
$$

Typing $\boxed{\Gamma \vdash v : A}$ $\boxed{\Gamma \vdash c : \underline{C}}$

$$
\frac{\vdash \Gamma \quad x : A \in \Gamma}{\Gamma \vdash x : A}\ \textsc{Tp\_Var} \qquad \frac{\Gamma, x : A \vdash c : \underline{C}}{\Gamma \vdash \mathbf{fun}\, x \mapsto c : A \to \underline{C}}\ \textsc{Tp\_Abs} \qquad \frac{\Gamma \vdash v : A \quad \Delta \subseteq \mathrm{dom}(\Sigma)}{\Gamma \vdash \mathbf{return}\, v : A!\Delta}\ \textsc{Tp\_Return}
$$

$$
\frac{\Gamma \vdash v_1 : A \to \underline{C} \quad \Gamma \vdash v_2 : A}{\Gamma \vdash v_1\, v_2 : \underline{C}}\ \textsc{Tp\_App}
$$

$$
\frac{\mathsf{op} : A \to B \in \Sigma \quad \Gamma \vdash v : A \quad \Gamma, y : B \vdash c : A_0!\Delta \quad \mathsf{op} \in \Delta}{\Gamma \vdash \mathsf{op}(v; y.c) : A_0!\Delta}\ \textsc{Tp\_OpApp}
$$

$$
\frac{\Gamma \vdash c_1 : A!\Delta \quad \Gamma, x : A \vdash c_2 : B!\Delta}{\Gamma \vdash \mathbf{do}\, x \leftarrow c_1 \,\mathbf{in}\, c_2 : B!\Delta}\ \textsc{Tp\_Do} \qquad \frac{\Gamma \vdash c : \underline{C} \quad \Gamma \vdash h : \underline{C} \Rightarrow \underline{D}}{\Gamma \vdash \mathbf{with}\, h\, \mathbf{handle}\, c : \underline{D}}\ \textsc{Tp\_Handle}
$$

Handler Typing $\boxed{\Gamma \vdash h : \underline{C} \Rightarrow \underline{D}}$

$$
\frac{\begin{array}{c}\Gamma, x : A \vdash c_r : B!\Delta' \quad \Delta \setminus \{\mathsf{op}_1, \ldots, \mathsf{op}_n\} \subseteq \Delta' \\ \forall i \in \{1, \ldots, n\}.(\mathsf{op}_i : A_i \to B_i \in \Sigma \quad \Gamma, x_i : A_i, k_i : B_i \to B!\Delta' \vdash c_i : B!\Delta')\end{array}}{\Gamma \vdash \mathbf{handler}\, \{\mathbf{return}\, x \mapsto c_r, \mathsf{op}_1(x_1; k_1) \mapsto c_1, \ldots, \mathsf{op}_n(x_n; k_n) \mapsto c_n\} : A!\Delta \Rightarrow B!\Delta'}\ \textsc{Hp\_Handler}
$$

Definition 4.2 (Translation from Pretnars to An Instance). *We assume that*[1]*:*

- *there exists a unique partition of* Σ*,*
- *any dirt is a disjoint union of the partition results of* Σ*, and*
- *target operations of any handlers must be one of the partition results of* Σ*.*

We write $\mathtt{S2s}(\Sigma)$ *to denote the set of the partition results of* Σ*. We write* $\mathtt{d2l}$ *to denote the function that assigns unique label* l *such that* $l : \mathbf{Lab} \in \Sigma_{\mathrm{lab}}$ *to* $s \in \mathtt{S2s}(\Sigma)$*.*

We define $\mathtt{d2l}(\Delta)$ *as the labels whose label is* $\mathtt{d2l}(s)$ *where* $\mathrm{dom}(s) \subseteq \Delta$ *and* $s \in \mathtt{S2s}(\Sigma)$*. We define* $\mathtt{d2l}(h)$ *as* $\mathtt{d2l}(s)$ *where* $h = \mathbf{handler}\,\{\mathbf{return}\,x \mapsto c_r, \mathsf{op}_1(x_1;k_1) \mapsto c_1, \ldots, \mathsf{op}_n(x_n;k_n) \mapsto c_n\}$ *and* $s = \{\mathsf{op}_1 : A_1 \to B_1, \ldots, \mathsf{op}_n : A_n \to B_n\}$*.*

We define $\mathtt{P2I}$ *as follows.*

Types

$$\mathtt{P2I}(A \to B!\Delta) = \mathtt{P2I}(A) \to_{\mathtt{P2I}(\Delta)} \mathtt{P2I}(B)$$

Dirts

$$\mathtt{P2I}(\emptyset) \;=\; \{\} \qquad \mathtt{P2I}(\Delta \uplus \mathrm{dom}(s)) \;=\; \mathtt{P2I}(\Delta) \cup \{\mathtt{d2l}(s)\} \quad (\textit{if } s \in \mathtt{S2s}(\Sigma))$$

Values

$$\mathtt{P2I}(x) \;=\; x \qquad \mathtt{P2I}(\mathbf{fun}\,x \mapsto c) \;=\; \mathbf{fun}(f, x, \mathtt{P2I}(c)) \quad (\textit{where } f \textit{ is fresh})$$

Computations

$$\begin{aligned}
\mathtt{P2I}(\mathbf{return}\,v) &= \mathtt{P2I}(v)\\
\mathtt{P2I}(v_1\,v_2) &= \mathtt{P2I}(v_1)\,\mathtt{P2I}(v_2)\\
\mathtt{P2I}(\mathbf{do}\,x \leftarrow c_1\,\mathbf{in}\,c_2) &= \mathbf{let}\,x = \mathtt{P2I}(c_1)\,\mathbf{in}\,\mathtt{P2I}(c_2)\\
\mathtt{P2I}(\mathsf{op}(v; y.c)) &= \mathbf{let}\,y = \mathsf{op}_{\mathtt{d2l}(s)}\,\mathtt{P2I}(v)\,\mathbf{in}\,\mathtt{P2I}(c) \quad (\textit{where } \mathsf{op} \in \mathrm{dom}(s))\\
\mathtt{P2I}(\mathbf{with}\,h\,\mathbf{handle}\,c) &= \mathbf{handle}_{\mathtt{d2l}(h)}\,\mathtt{P2I}(c)\,\mathbf{with}\,\mathtt{P2I}(h)
\end{aligned}$$

Handlers

$$\begin{aligned}
\mathtt{P2I}(h) \;=\;& \{\mathbf{return}\,x \mapsto \mathtt{P2I}(c_r)\} \uplus \{\mathsf{op}_1\,x_1\,k_1 \mapsto \mathtt{P2I}(c_1)\} \uplus \cdots \uplus \{\mathsf{op}_n\,x_n\,k_n \mapsto \mathtt{P2I}(c_n)\}\\
& (\textit{where } h = \mathbf{handler}\,\{\mathbf{return}\,x \mapsto c_r, \mathsf{op}_1(x_1;k_1) \mapsto c_1, \ldots, \mathsf{op}_n(x_n;k_n) \mapsto c_n\})
\end{aligned}$$

Effect contexts

$$\begin{aligned}
\mathtt{P2I}(\Sigma) \;=\;& \textstyle\bigcup_{s \in \mathtt{S2s}(\Sigma)} \{\mathtt{d2l}(s) :: \{\mathsf{op}_1 : \mathtt{P2I}(A_1) \Rightarrow \mathtt{P2I}(B_1), \ldots, \mathsf{op}_n : \mathtt{P2I}(A_n) \Rightarrow \mathtt{P2I}(B_n)\}\}\\
& (\textit{where } s = \{\mathsf{op}_1 : A_1 \to B_1, \ldots, \mathsf{op}_n : A_n \to B_n\})
\end{aligned}$$

Typing Contexts

$$\mathtt{P2I}(\emptyset) \;=\; \emptyset \qquad \mathtt{P2I}(\Gamma, x : A) \;=\; \mathtt{P2I}(\Gamma), x : \mathtt{P2I}(A)$$

Lemma 4.3. $\mathrm{dom}(\Gamma) = \mathrm{dom}(\mathtt{P2I}(\Gamma))$.

Proof. Clearly by definition of $\mathtt{P2I}$. ■

Lemma 4.4. *If* $\Delta \subseteq \mathrm{dom}(\Sigma)$*, then* $\Gamma \vdash \mathtt{P2I}(\Delta) : \mathbf{Eff}$ *for any* Γ *such that* $\vdash \Gamma$*.*

Proof. By induction on the size of Γ.

If $\Delta = \emptyset$, then clearly because $\mathtt{P2I}(\emptyset) = \{\}$.

If $\Delta = \Delta' \uplus \mathrm{dom}(s)$ for some Δ' and $s \in \mathtt{S2s}(\Sigma)$, then $\mathtt{P2I}(\Delta) = \mathtt{P2I}(\Delta') \mathbin{\underline{\cup}} \{\mathtt{d2l}(s)\}$ where $\mathtt{d2l}(s) : \mathbf{Lab} \in \Sigma_{\mathrm{lab}}$. Let Γ be a typing context such that $\vdash \Gamma$. By the induction hypothesis, we have $\Gamma \vdash \mathtt{P2I}(\Delta') : \mathbf{Eff}$. Thus, K_Cons derives $\Gamma \vdash \mathtt{P2I}(\Delta') \mathbin{\underline{\cup}} \{\mathtt{d2l}(s)\} : \mathbf{Eff}$ because we have $\Gamma \vdash \{\mathtt{d2l}(s)\} : \mathbf{Eff}$. ■

Lemma 4.5. *If* $\vdash \Gamma$ *and* $x : A \in \Gamma$*, then* $x : \mathtt{P2I}(A) \in \mathtt{P2I}(\Gamma)$*.*

[1] These assumptions arise from our formalization of labels and operations. They are easily removed if we omit labels.

Proof. By structual induction on Γ.

If $\Gamma = \emptyset$, then $x : A \in \Gamma$ cannot happen.

If $\Gamma = \Gamma', y : B$ for some y, B, and Γ', then we have $\texttt{P2I}(\Gamma) = \texttt{P2I}(\Gamma'), y : \texttt{P2I}(B)$. In this case, if $x = y$, then we have $A = B$ and $y : \texttt{P2I}(B) \in \texttt{P2I}(\Gamma)$ as required. If $x \neq y$, then we have $x : A \in \Gamma'$. By the induction hypothesis, we have $x : \texttt{P2I}(A) \in \texttt{P2I}(\Gamma')$. Thus, we have $x : \texttt{P2I}(A) \in \texttt{P2I}(\Gamma)$ as required. ∎

Theorem 4.6.

(1) If $\vdash \Gamma$, *then* $\vdash \texttt{P2I}(\Gamma)$.

(2) If $\Gamma \vdash A$, *then* $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(A) : \textbf{Typ}$.

(3) If $\Gamma \vdash v : A$, *then* $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v) : \texttt{P2I}(A) \mid \{\}$.

(4) If $\Gamma \vdash c : A!\Delta$, *then* $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(c) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$.

(5) If $\Gamma \vdash h : A!\Delta \Rightarrow B!\Delta'$, *then* $\texttt{P2I}(\Delta) \mathbin{\underline{\cup}} \varepsilon \sim_{\mathrm{Set}} \texttt{d2l}(h) \mathbin{\underline{\cup}} \texttt{P2I}(\Delta')$ *for some* ε *and there exists some* σ *such that* $\texttt{P2I}(\Gamma) \vdash_\sigma \texttt{P2I}(h) : \texttt{P2I}(A) \Rightarrow^{\texttt{P2I}(\Delta')} \texttt{P2I}(B)$ *and* $\texttt{d2l}(h) :: \sigma \in \texttt{P2I}(\Sigma)$.

Proof.(1)(2) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case Cp_Empty**:** Clearly.

Case Cp_Var**:** We have

- $\Gamma = \Gamma', x : A$,
- $x \notin \mathrm{dom}(\Gamma')$, and
- $\Gamma' \vdash A$,

for some x, A, and Γ'. By the induction hypothesis and Lemma 4.3, we have $x \notin \mathrm{dom}(\texttt{P2I}(\Gamma'))$ and $\texttt{P2I}(\Gamma') \vdash \texttt{P2I}(A) : \textbf{Typ}$. Thus, C_Var derives $\vdash \texttt{P2I}(\Gamma'), x : \texttt{P2I}(A)$ as required.

Case Kp_Fun**:** We have

- $A = A_1 \to B_1!\Delta$,
- $\Gamma \vdash A_1$,
- $\Delta \subseteq \mathrm{dom}(\Sigma)$, and
- $\Gamma \vdash B_1$,

for some A_1, B_1, and Δ. By the induction hypothesis and Lemma 4.4, we have

- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(A_1) : \textbf{Typ}$,
- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(\Delta) : \textbf{Eff}$, and
- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(B_1) : \textbf{Typ}$.

Thus, K_Fun derives

$$\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(A_1) \to_{\texttt{P2I}(\Delta)} \texttt{P2I}(B_1) : \textbf{Typ}$$

as required.

(3)(4)(5) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivation.

Case Tp_Var**:** We have

- $v = x$,
- $\vdash \Gamma$, and
- $x : A \in \Gamma$,

for some x. By Lemma 4.5 and Theorem 4.6(1), we have

- $\vdash \texttt{P2I}(\Gamma)$ and
- $x : \texttt{P2I}(A) \in \texttt{P2I}(\Gamma)$.

Thus, T_Var derives

$$\texttt{P2I}(\Gamma) \vdash x : \texttt{P2I}(A) \mid \{\}$$

as required.

Case Tp_Abs**:** We have

- $v = \textbf{fun}\, x \mapsto c$ and

– $\Gamma, x : A \vdash c : B!\Delta$

for some x, c, A, B, and Δ. By the induction hypothesis, we have

$$\texttt{P2I}(\Gamma), x : \texttt{P2I}(A) \vdash \texttt{P2I}(c) : \texttt{P2I}(B) \mid \texttt{P2I}(\Delta).$$

Without loss of generality, we can choose f such that

– $f \notin FV(\texttt{P2I}(c))$,
– $f \neq x$,
– $f \notin \mathrm{dom}(\Gamma)$, and
– $\texttt{P2I}(\mathbf{fun}\, x \mapsto c) = \mathbf{fun}(f, x, \texttt{P2I}(c))$.

By Lemma 3.12 and Lemma 3.2(2) and Lemma 3.6, we have

– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(B) : \mathbf{Typ}$ and
– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(\Delta) : \mathbf{Eff}$.

By Lemma 3.9, we have $\vdash \texttt{P2I}(\Gamma), x : \texttt{P2I}(A)$. Since only C_Var can derive $\vdash \texttt{P2I}(\Gamma), x : \texttt{P2I}(A)$, we have $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(A) : \mathbf{Typ}$. Thus, C_Var derives

$$\vdash \texttt{P2I}(\Gamma), f : \texttt{P2I}(A) \rightarrow_{\texttt{P2I}(\Delta)} \texttt{P2I}(B).$$

Thus, Lemma 3.5 and T_Abs derives

$$\texttt{P2I}(\Gamma) \vdash \mathbf{fun}(f, x, \texttt{P2I}(c)) : \texttt{P2I}(A) \rightarrow_{\texttt{P2I}(\Delta)} \texttt{P2I}(B) \mid \{\}$$

as required.

Case Tp_Return**:** We have

– $c = \mathbf{return}\, v$,
– $\Gamma \vdash v : A$, and
– $\Delta \subseteq \mathrm{dom}(\Sigma)$,

for some v. By the induction hypothesis and Lemma 4.4, we have

– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v) : \texttt{P2I}(A) \mid \{\}$ and
– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(\Delta) : \mathbf{Eff}$.

Thus, T_Sub derives

$$\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$$

as required.

Case Tp_App**:** We have

– $c = v_1\, v_2$,
– $\Gamma \vdash v_1 : B \rightarrow A!\Delta$, and
– $\Gamma \vdash v_2 : B$,

for some v_1, v_2, and B. By the induction hypothesis, we have

– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v_1) : \texttt{P2I}(B) \rightarrow_{\texttt{P2I}(\Delta)} \texttt{P2I}(A) \mid \{\}$ and
– $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v_2) : \texttt{P2I}(B) \mid \{\}$.

Thus, T_App derives

$$\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v_1)\, \texttt{P2I}(v_2) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$$

as required.

Case Tp_OpApp**:** We have

– $c = \mathsf{op}(v; y.c')$,
– $\mathsf{op} : A' \rightarrow B' \in \Sigma$,
– $\Gamma \vdash v : A'$,
– $\Gamma, y : B' \vdash c' : A!\Delta$, and
– $\mathsf{op} \in \Delta$,

for some op, v, y, c', A', B', and Δ. By $\mathsf{op} \in \Delta$, there uniquely exists some s such that

– $s \in \texttt{S2s}(\Sigma)$,
– $\mathsf{op} : A' \rightarrow B' \in s$,
– $\mathrm{dom}(s) \subseteq \Delta$.

Thus, we have
- $l :: \sigma \in \texttt{P2I}(\Sigma)$,
- $\mathsf{op} : \texttt{P2I}(A') \Rightarrow \texttt{P2I}(B') \in \sigma$, and
- $\{\texttt{d2l}(s)\} \mathbin{\underline{\cup}} \varepsilon \sim_{\mathrm{Set}} \texttt{P2I}(\Delta)$,

for some σ. By the induction hypothesis, we have
- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(v) : \texttt{P2I}(A') \mid \{\}$ and
- $\texttt{P2I}(\Gamma), y : \texttt{P2I}(B') \vdash \texttt{P2I}(c') : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$.

Thus, T_Op and T_App derives

$$\texttt{P2I}(\Gamma) \vdash \mathsf{op}_{\texttt{d2l}(s)}\, \texttt{P2I}(v) : \texttt{P2I}(B') \mid \{\texttt{d2l}(s)\}.$$

Thus, T_Sub and T_Let derives

$$\texttt{P2I}(\Gamma) \vdash \mathbf{let}\, y = \mathsf{op}_{\texttt{d2l}(s)}\, \texttt{P2I}(v)\, \mathbf{in}\, \texttt{P2I}(c') : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$$

as required.

Case Tp_Do**:** We have
- $c = \mathbf{do}\, x \leftarrow c_1\, \mathbf{in}\, c_2$,
- $\Gamma \vdash c_1 : B!\Delta$, and
- $\Gamma, x : B \vdash c_2 : A!\Delta$,

for some x, c_1, c_2, and Δ. By the induction hypothesis, we have
- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(c_1) : \texttt{P2I}(B) \mid \texttt{P2I}(\Delta)$ and
- $\texttt{P2I}(\Gamma), x : \texttt{P2I}(B) \vdash \texttt{P2I}(c_2) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$.

Thus, T_Let derives

$$\texttt{P2I}(\Gamma) \vdash \mathbf{let}\, x = \texttt{P2I}(c_1)\, \mathbf{in}\, \texttt{P2I}(c_2) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$$

as required.

Case Tp_Handle**:** We have
- $c = \mathbf{with}\, h\, \mathbf{handle}\, c'$,
- $\Gamma \vdash c' : A'!\Delta'$, and
- $\Gamma \vdash h : A'!\Delta' \Rightarrow A!\Delta$.

for some c', h, A', and Δ'. By the induction hypothesis, we have
- $\texttt{P2I}(\Gamma) \vdash \texttt{P2I}(c') : \texttt{P2I}(A') \mid \texttt{P2I}(\Delta')$,
- $\texttt{d2l}(h) :: \sigma \in \texttt{P2I}(\Sigma)$,
- $\texttt{P2I}(\Gamma) \vdash_\sigma \texttt{P2I}(h) : \texttt{P2I}(A') \Rightarrow^{\texttt{P2I}(\Delta)} \texttt{P2I}(A)$, and
- $\texttt{P2I}(\Delta') \mathbin{\underline{\cup}} \varepsilon \sim_{\mathrm{Set}} \texttt{d2l}(h) \mathbin{\underline{\cup}} \texttt{P2I}(\Delta)$,

for some ε and σ. Thus, T_Sub and T_Handling derive

$$\texttt{P2I}(\Gamma) \vdash \mathbf{handle}_{\texttt{P2I}(h)}\, \texttt{P2I}(c')\, \mathbf{with}\, \texttt{P2I}(h) : \texttt{P2I}(A) \mid \texttt{P2I}(\Delta)$$

as required.

Case Hp_Handler**:** We have
- $h = \mathbf{handler}\, \{\mathbf{return}\, x \mapsto c_r, \mathsf{op}_1(x_1; k_1) \mapsto c_1, \ldots, \mathsf{op}_n(x_n; k_n) \mapsto c_n\}$,
- $\Gamma, x : A \vdash c_r : B!\Delta'$,
- $\mathsf{op}_i : A_i \to B_i \in \Sigma$ for any $i \in \{1, \ldots, n\}$,
- $\Gamma, x_i : A_i, k_i : B_i \to B!\Delta' \vdash c_i : B!\Delta'$ for any $i \in \{1, \ldots, n\}$, and
- $\Delta \setminus \{\mathsf{op}_1, \ldots, \mathsf{op}_n\} \subseteq \Delta'$,

for some n, x, c_r, op_i, x_i, k_i, c_i, A_i, and B_i, where $i \in \{1, \ldots, n\}$.

By the assumptions, we have
- $s \in \texttt{S2s}(\Sigma)$ and
- $\texttt{d2l}(h) = \texttt{d2l}(s)$

where $s = \{\mathtt{op}_1 : A_1 \to B_1, \ldots, \mathtt{op}_n : A_n \to B_n\}$. Thus, we have $\mathtt{d2l}(h) :: \sigma \in \mathtt{P2I}(\Sigma)$ where $\sigma = \{\mathtt{op}_1 : \mathtt{P2I}(A_1) \Rightarrow \mathtt{P2I}(B_1), \ldots, \mathtt{op}_n : \mathtt{P2I}(A_n) \Rightarrow \mathtt{P2I}(B_n)\}$.

By $\Delta \setminus \{\mathtt{op}_1, \ldots, \mathtt{op}_n\} \subseteq \Delta'$, we have $\Delta \subseteq \mathrm{dom}(s) \cup \Delta'$. By the assumptions, we have either $\mathrm{dom}(s) \subseteq \Delta'$ or $\mathtt{op}_i \notin \Delta'$ for any i. In any case, we have $\mathtt{P2I}(\Delta) \sqcup \varepsilon \sim_{\mathrm{Set}} \mathtt{d2l}(h) \sqcup \mathtt{P2I}(\Delta')$ for some ε.

By the induction hypothesis, we have

- $\mathtt{P2I}(\Gamma), x : \mathtt{P2I}(A) \vdash \mathtt{P2I}(c_r) : \mathtt{P2I}(B) \mid \mathtt{P2I}(\Delta')$ and
- $\mathtt{P2I}(\Gamma), x_i : \mathtt{P2I}(A_i), k_i : \mathtt{P2I}(B_i) \to_{\mathtt{P2I}(\Delta')} \mathtt{P2I}(B) \vdash \mathtt{P2I}(c_i) : \mathtt{P2I}(B) \mid \mathtt{P2I}(\Delta')$ for any $i \in \{1, \ldots, n\}$.

Therefore, H_Return and H_Op derive $\mathtt{P2I}(\Gamma) \vdash_\sigma \mathtt{P2I}(h) : \mathtt{P2I}(A) \Rightarrow^{\mathtt{P2I}(\Delta')} \mathtt{P2I}(B)$.

Thus, the required result is achieved.

■

4.2 Comparison to [Hillerström et al.(2017)]

We give the targets of comparison: one is an instance of λ_{EA} (Example 1.25), and another is a minorly changed language of [Hillerström et al.(2017)].

Definition 4.7 (Minor Changed Version of [Hillerström et al.(2017)]). *Change list:*

- *removing variants and records,*
- *removing presence and handler types,*
- *removing computation kinds, and*
- *adding well-formedness rules of contexts.*

The syntax of a minor changed version of [Hillerström et al.(2017)] is as follows.

$$
\begin{array}{rcll}
V, W & ::= & x \mid \lambda x^A.M \mid \Lambda \alpha^K.M & \textit{(values)} \\
M, N & ::= & V\,W \mid V\,T \mid \mathbf{return}\,M \mid \mathbf{let}\,x \leftarrow M\,\mathbf{in}\,N & \textit{(computations)} \\
 & & \mid (\mathbf{do}\,l\,V)^E \mid \mathbf{handle}\,M\,\mathbf{with}\,H & \\
H & ::= & \{\mathbf{return}\,x \mapsto M\} \mid H \uplus \{l\,p\,r \mapsto M\} & \textit{(handlers)} \\
A, B & ::= & A \to C \mid \forall \alpha^K.C \mid \alpha & \textit{(value types)} \\
C, D & ::= & A!E & \textit{(computations types)} \\
E & ::= & \{R\} & \textit{(effect types)} \\
R & ::= & l : P; R \mid \rho \mid \cdot & \textit{(row types)} \\
P & ::= & \mathsf{Pre}(A \to B) \mid \mathsf{Abs} & \textit{(presence types)} \\
T & ::= & A \mid C \mid E \mid R & \textit{(types)} \\
K & ::= & \mathsf{Type} \mid \mathsf{Row}_{\mathcal{L}} \mid \mathsf{Effect} & \textit{(kinds)} \\
\mathcal{L} & ::= & \emptyset \mid \{l\} \uplus \mathcal{L} & \textit{(label sets)} \\
\Gamma & ::= & \cdot \mid \Gamma, x : A & \textit{(type environments)} \\
\Delta & ::= & \cdot \mid \Delta, \alpha : K & \textit{(kind environments)}
\end{array}
$$

Well-formedness, kinding, and typing rules consist of the following.

Kinding Contexts Well-formedness $\boxed{\vdash \Delta}$

$$
\frac{}{\vdash \cdot}\ \textsc{KCh\_Empty} \qquad \frac{\vdash \Delta \quad \alpha \notin \mathrm{dom}(\Delta)}{\vdash \Delta, \alpha : K}\ \textsc{KCh\_TVar}
$$

Contexts Well-formedness $\boxed{\vdash \Delta; \Gamma}$

$$
\frac{\vdash \Delta}{\Delta \vdash \cdot}\ \textsc{Ch\_Empty} \qquad \frac{\Delta \vdash \Gamma \quad x \notin \mathrm{dom}(\Gamma) \quad \Delta \vdash A : \mathsf{Type}}{\Delta \vdash \Gamma, x : A}\ \textsc{Ch\_Var}
$$

Kinding $\boxed{\Delta \vdash T : K}$

$$\frac{\vdash \Delta, \alpha : K}{\Delta, \alpha : K \vdash \alpha : K}\ \textsc{Kh\_Var} \qquad \frac{\Delta \vdash A : \mathsf{Type} \quad \Delta \vdash B : \mathsf{Type} \quad \Delta \vdash E : \mathsf{Effect}}{\Delta \vdash A \to B!E : \mathsf{Type}}\ \textsc{Kh\_Fun}$$

$$\frac{\Delta, \alpha : K \vdash A : \mathsf{Type} \quad \Delta, \alpha : K \vdash E : \mathsf{Effect}}{\Delta \vdash \forall \alpha^K . A!E : \mathsf{Type}}\ \textsc{Kh\_Forall} \qquad \frac{\Delta \vdash R : \mathsf{Row}_\emptyset}{\Delta \vdash \{R\} : \mathsf{Effect}}\ \textsc{Kh\_Effect}$$

$$\frac{\forall i \in \{1, \dots, n\}.(P_i = \mathsf{Abs} \text{ or } (P_i = \mathsf{Pre}(A_i \to B_i) \text{ and } \Delta \vdash A_i : \mathsf{Type} \text{ and } \Delta \vdash B_i : \mathsf{Type})) \quad \vdash \Delta}{\Delta \vdash l_1 : P_1; \cdots ; l_n : P_n; \cdot : \mathsf{Row}_\emptyset}\ \textsc{Kh\_CloseRow}$$

$$\frac{\begin{array}{c}\forall i \in \{1, \dots, n\}.(P_i = \mathsf{Abs} \text{ or } (P_i = \mathsf{Pre}(A_i \to B_i) \text{ and } \Delta \vdash A_i : \mathsf{Type} \text{ and } \Delta \vdash B_i : \mathsf{Type})) \\ \Delta \vdash \rho : \mathsf{Row}_{\mathcal{L}} \quad \mathcal{L} = \{l_1, \dots, l_n\}\end{array}}{\Delta \vdash l_1 : P_1; \cdots ; l_n : P_n; \rho : \mathsf{Row}_\emptyset}\ \textsc{Kh\_OpenRow}$$

Typing $\boxed{\Delta; \Gamma \vdash V : A}$ $\boxed{\Delta; \Gamma \vdash M : C}$

$$\frac{\Delta \vdash \Gamma \quad x : A \in \Gamma}{\Delta; \Gamma \vdash x : A}\ \textsc{Th\_Var} \qquad \frac{\Delta; \Gamma, x : A \vdash M : C}{\Delta; \Gamma \vdash \lambda x^A . M : A \to C}\ \textsc{Th\_Lam}$$

$$\frac{\Delta, \alpha : K; \Gamma \vdash M : C \quad \Delta \vdash \Gamma}{\Delta; \Gamma \vdash \Lambda \alpha^K . M : \forall \alpha^K . C}\ \textsc{Th\_PolyLam} \qquad \frac{\Delta; \Gamma \vdash V : A \to C \quad \Delta; \Gamma \vdash W : A}{\Delta; \Gamma \vdash V\, W : C}\ \textsc{Th\_App}$$

$$\frac{\Delta; \Gamma \vdash V : \forall \alpha^K . C \quad \Delta \vdash T : K}{\Delta; \Gamma \vdash V\, T : C[T/\alpha]}\ \textsc{Th\_PolyApp} \qquad \frac{\Delta; \Gamma \vdash V : A \quad \Delta \vdash E : \mathsf{Effect}}{\Delta; \Gamma \vdash \mathbf{return}\, V : A!E}\ \textsc{Th\_Return}$$

$$\frac{\Delta; \Gamma \vdash M : A!E \quad \Delta; \Gamma, x : A \vdash N : B!E}{\Delta; \Gamma \vdash \mathbf{let}\, x \leftarrow M\, \mathbf{in}\, N : B!E}\ \textsc{Th\_Let}$$

$$\frac{\Delta; \Gamma \vdash V : A \quad E = \{l : \mathsf{Pre}(A \to B); R\} \quad \Delta \vdash E : \mathsf{Effect}}{\Delta; \Gamma \vdash (\mathbf{do}\, l\, V)^E : B!E}\ \textsc{Th\_Do}$$

$$\frac{\Delta; \Gamma \vdash M : C \quad \Delta; \Gamma \vdash H : C \Rightarrow D}{\Delta; \Gamma \vdash \mathbf{handle}\, M\, \mathbf{with}\, H : D}\ \textsc{Th\_Handle}$$

Handler Typing $\boxed{\Delta; \Gamma \vdash H : C \Rightarrow D}$

$$\frac{\begin{array}{c} C = A!\{l_1 : \mathsf{Pre}(A_1 \to B_1); \cdots ; l_n : \mathsf{Pre}(A_n \to B_n); R\} \\ D = B!\{l_1 : P_1; \cdots ; l_n : P_n; R\} \quad H = \{\mathbf{return}\, x \mapsto M\} \uplus \{l_1\, y_1\, r_1 \mapsto N_1\} \uplus \cdots \uplus \{l_n\, y_n\, r_n \mapsto N_n\} \\ \Delta; \Gamma, x : A \vdash M : D \quad \forall i \in \{1, \dots, n\}.(\Delta; \Gamma, y_i : A_i, r_i : B_i \to D \vdash N_i : D) \end{array}}{\Delta; \Gamma \vdash H : C \Rightarrow D}\ \textsc{Hh\_Handler}$$

Definition 4.8 (Translation from Hillerström's to An Instance)**.** *We assume that:*

- *there exists a unique set that has any label (we call it* $\mathbb{L}$*),*
- *there exists a unique partition of* $\mathbb{L}$*,*
- *for any row, a set of presence labels in that row is a disjoint union of the partition result of* $\mathbb{L}$*,*
- *for any handler, target labels of that handler is one of the partition result of* $\mathbb{L}$*, and*
- *a unique closed type can be attached to* l *as presence.*

We write $\mathtt{L2S}(\mathbb{L})$ *to denote the set of partition results of* $\mathbb{L}$*,* $\mathtt{r2l}$ *to denote the function that assigns a unique label* l *such that* $l : \mathbf{Lab} \in \Sigma_{\text{eff}}$ *to* $\mathcal{L} \in \mathtt{L2S}(\mathbb{L})$*. We write* $\mathtt{r2l}(H)$ *to denote* l *such that* $\mathtt{r2l}(\{l_1, \dots, l_n\}) = l$ *where* $H = \{\mathbf{return}\, x \mapsto M\} \uplus \{l_1\, p_1\, r_1 \mapsto N_1\} \uplus \cdots \{l_n\, p_n\, r_n \mapsto N_n\}$*. We define* $\mathtt{l2T}$ *as the function that takes a label* l *and returns the type that corresponds to the unique presence type of* l*. We define* $\mathtt{l2Op}$ *as the function that takes a label* l *and returns a unique operation name. We also assume that*

$$\mathtt{r2l}(\{l_1, \dots, l_n\}) :: \{\mathtt{l2Op}(l_1) : \mathtt{l2T}(l_1), \dots, \mathtt{l2Op}(l_n) : \mathtt{l2T}(l_n)\} \in \Sigma.$$

We define $\mathtt{H2I}$ *as follows.*

Kinds

$$\mathtt{H2I}(\mathsf{Type}) = \mathbf{Typ} \qquad \mathtt{H2I}(\mathsf{Row}_{\mathcal{L}}) = \mathtt{H2I}(\mathsf{Effect}) = \mathbf{Eff}$$

Types

$$\mathtt{H2I}(A \to B!E) = \mathtt{H2I}(A) \to_{\mathtt{H2I}(E)} \mathtt{H2I}(B) \qquad \mathtt{H2I}(\forall\alpha^K.A!E) = \forall\alpha : \mathtt{H2I}(K).\mathtt{H2I}(A)^{\mathtt{H2I}(E)}$$

Effects

$$\begin{aligned}
\mathtt{H2I}(\{R\}) &= \mathtt{H2I}(R) \\
\mathtt{H2I}(l_1 : P_1; \cdots ; l_n : P_n; \cdot) &= \langle l'_1 \mid \langle \cdots \mid \langle l'_m \mid \langle\rangle\rangle\rangle\rangle \quad (\textit{where } l'_i = \mathtt{r2l}(\mathcal{L}_i) \textit{ and } \mathcal{L}_1 \uplus \cdots \mathcal{L}_m = \{l_j \mid P_j \neq \mathsf{Abs}\}) \\
\mathtt{H2I}(l_1 : P_1; \cdots ; l_n : P_n; \rho) &= \langle l'_1 \mid \langle \cdots \mid \langle l'_m \mid \rho\rangle\rangle\rangle \quad (\textit{where } l'_i = \mathtt{r2l}(\mathcal{L}_i) \textit{ and } \mathcal{L}_1 \uplus \cdots \mathcal{L}_m = \{l_j \mid P_j \neq \mathsf{Abs}\})
\end{aligned}$$

Values

$$\begin{aligned}
\mathtt{H2I}(x) &= x \qquad \mathtt{H2I}(\Lambda\alpha^K.M) = \Lambda\alpha : \mathtt{H2I}(K).\mathtt{H2I}(M) \\
\mathtt{H2I}(\lambda x^A.M) &= \mathbf{fun}(z, x, \mathtt{H2I}(M)) \quad (\textit{where } z \textit{ is fresh})
\end{aligned}$$

Computations

$$\begin{aligned}
\mathtt{H2I}(V\,W) &= \mathtt{H2I}(V)\,\mathtt{H2I}(W) \qquad \mathtt{H2I}(V\,T) = \mathtt{H2I}(V)\,\mathtt{H2I}(T) \\
\mathtt{H2I}(\mathbf{return}\,M) &= \mathtt{H2I}(M) \qquad \mathtt{H2I}(\mathbf{let}\,x \leftarrow M\,\mathbf{in}\,N) = \mathbf{let}\,x = \mathtt{H2I}(M)\,\mathbf{in}\,\mathtt{H2I}(N) \\
\mathtt{H2I}((\mathbf{do}\,l\,V)^E) &= \mathtt{l2Op}(l)_{\mathtt{r2l}(\mathcal{L})}\,\mathtt{H2I}(V) \quad (\textit{where } l \in \mathcal{L} \in \mathtt{L2S}(\mathbb{L})) \\
\mathtt{H2I}(\mathbf{handle}\,M\,\mathbf{with}\,H) &= \mathbf{handle}_{\mathtt{r2l}(H)}\,\mathtt{H2I}(M)\,\mathbf{with}\,\mathtt{H2I}(H)
\end{aligned}$$

Handlers

$$\begin{aligned}
\mathtt{H2I}(\{\mathbf{return}\,x \mapsto M\}) &= \{\mathbf{return}\,x \mapsto \mathtt{H2I}(M)\} \\
\mathtt{H2I}(\{l\,p\,r \mapsto M\} \uplus H) &= \mathtt{H2I}(H) \uplus \{\mathtt{l2Op}(l)\,p\,r \mapsto \mathtt{H2I}(M)\}
\end{aligned}$$

Contexts

$$\begin{aligned}
\mathtt{H2I}(\cdot) &= \emptyset \qquad \mathtt{H2I}(\Gamma, x : A) = \mathtt{H2I}(\Gamma), x : \mathtt{H2I}(A) \\
\mathtt{H2I}(\Delta, \alpha : K) &= \mathtt{H2I}(\Delta), \alpha : \mathtt{H2I}(K)
\end{aligned}$$

Lemma 4.9.

(1) If $\vdash \Gamma_1, \alpha : K, x : A, \Gamma_3$ *and* $\vdash \Gamma_1, x : A$*, then* $\vdash \Gamma_1, x : A, \alpha : K, \Gamma_3$.

(2) If $\Gamma_1, \alpha : K, x : A, \Gamma_3 \vdash S : K'$ *and* $\vdash \Gamma_1, x : A$*, then* $\Gamma_1, x : A, \alpha : K, \Gamma_3 \vdash S : K'$.

(3) If $\Gamma_1, \alpha : K, x : A, \Gamma_3 \vdash B <: C$ *and* $\vdash \Gamma_1, x : A$*, then* $\Gamma_1, x : A, \alpha : K, \Gamma_3 \vdash B <: C$.

(4) If $\Gamma_1, \alpha : K, x : A, \Gamma_3 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$ *and* $\vdash \Gamma_1, x : A$*, then* $\Gamma_1, x : A, \alpha : K, \Gamma_3 \vdash B_1 \mid \varepsilon_1 <: B_2 \mid \varepsilon_2$.

(5) If $\Gamma_1, \alpha : K, x : A, \Gamma_3 \vdash e : B \mid \varepsilon$ *and* $\vdash \Gamma_1, x : A$*, then* $\Gamma_1, x : A, \alpha : K, \Gamma_3 \vdash e : B \mid \varepsilon$.

(6) If $\Gamma_1, \alpha : K, x : A, \Gamma_3 \vdash_\sigma h : B \Rightarrow^\varepsilon C$ *and* $\vdash \Gamma_1, x : A$*, then* $\Gamma_1, x : A, \alpha : K, \Gamma_3 \vdash_\sigma h : B \Rightarrow^\varepsilon C$.

Proof. Straightforward by mutual induction on the derivations. ■

Theorem 4.10.

(1) If $\vdash \Delta$*, then* $\vdash \mathtt{H2I}(\Delta)$.

(2) If $\Delta \vdash T : K$*, then* $\mathtt{H2I}(\Delta) \vdash \mathtt{H2I}(T) : \mathtt{H2I}(K)$.

(3) If $\Delta \vdash \Gamma$*, then* $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma)$.

(4) If $\Delta; \Gamma \vdash V : A$*, then* $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V) : \mathtt{H2I}(A) \mid \emptyset_E$.

(5) If $\Delta; \Gamma \vdash M : A!E$*, then* $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(M) : \mathtt{H2I}(A) \mid \mathtt{H2I}(E)$.

(6) *If* $\Delta; \Gamma \vdash \{\mathbf{return}\, x \mapsto M\} \uplus \{l_1\, p_1\, r_1 \mapsto N_1\} \uplus \cdots \uplus \{l_n\, p_n\, r_n \mapsto N_n\} : A!E \Rightarrow B!E'$, *then*

- $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash_\sigma \texttt{H2I}(H) : \texttt{H2I}(A) \Rightarrow^{\texttt{H2I}(E')} \texttt{H2I}(B)$,
- $\texttt{r2l}(H) : \sigma \in \Sigma$, *and*
- $\langle \texttt{r2l}(H) \mid \texttt{H2I}(E') \rangle \sim_{\mathrm{SimpR}} \texttt{H2I}(E)$,

where $\sigma = \{\texttt{l2Op}(l_1) : \texttt{l2T}(l_1), \ldots, \texttt{l2Op}(l_n) : \texttt{l2T}(l_n)\}$.

Proof.

(1) Straightforward by induction on the derivation.

(2) By induction on a derivation of the judgment. We proceed by case analysis on the rule applied lastly to the derivation.

Case Kh_Var**:** We have
- $T = \alpha$,
- $\vdash \Delta', \alpha : K$, and
- $\Delta = \Delta', \alpha : K$,

for some Δ'.

By definition of H2I, we have $\alpha : \texttt{H2I}(K) \in \texttt{H2I}(\Delta', \alpha : K)$. By case (1), K_Var derives $\texttt{H2I}(\Gamma) \vdash \alpha : \texttt{H2I}(K)$

Case Kh_Fun**:** We have
- $T = A \to B!E$,
- $K = \mathsf{Type}$,
- $\Delta \vdash A : \mathsf{Type}$,
- $\Delta \vdash B : \mathsf{Type}$,
- $\Delta \vdash E : \mathsf{Effect}$,

for some A, B, and E. By the induction hypothesis, we have
- $\texttt{H2I}(\Delta) \vdash \texttt{H2I}(A) : \mathbf{Typ}$,
- $\texttt{H2I}(\Delta) \vdash \texttt{H2I}(B) : \mathbf{Typ}$, and
- $\texttt{H2I}(\Delta) \vdash \texttt{H2I}(E) : \mathbf{Eff}$.

Thus, K_Fun derives

$$\texttt{H2I}(\Delta) \vdash \texttt{H2I}(A) \to_{\texttt{H2I}(E)} \texttt{H2I}(B) : \mathbf{Typ}$$

as required.

Case Kh_Forall**:** We have
- $T = \forall \alpha^{K'}.A!E$,
- $K = \mathsf{Type}$,
- $\Delta, \alpha : K' \vdash A : \mathsf{Type}$, and
- $\Delta, \alpha : K' \vdash E : \mathsf{Effect}$,

for some α, K', A, and E. By the induction hypothesis, we have
- $\texttt{H2I}(\Delta, \alpha : K') \vdash \texttt{H2I}(A) : \mathbf{Typ}$ and
- $\texttt{H2I}(\Delta, \alpha : K') \vdash \texttt{H2I}(E) : \mathbf{Eff}$.

Thus, K_Poly derives

$$\texttt{H2I}(\Delta) \vdash \forall \alpha : \texttt{H2I}(K').\texttt{H2I}(A)^{\texttt{H2I}(E)} : \mathbf{Typ}$$

as required.

Case Kh_Effect**:** Clearly by the induction hypothesis.

Case Kh_CloseRow**:** Clearly by the assumptions and K_Cons.

Case Kh_OpenRow**:** Clearly by the assumptions and K_Cons.

(3) By induction on a derivation of the judgment. We proceed by case analysis on the rule applied lastly to the derivation.

Case Ch_Empty**:** Clearly because case (1).

Case Ch_Var**:** We have

- $\Gamma = \Gamma', x : A$,
- $\Delta \vdash \Gamma'$,
- $x \notin \mathrm{dom}(\Gamma')$, and
- $\Delta \vdash A : \mathsf{Type}$,

for some Γ', x, and A. By the induction hypothesis and case (2), we have
- $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma')$ and
- $\mathtt{H2I}(\Delta) \vdash \mathtt{H2I}(A) : \mathbf{Typ}$.

By $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma')$ and Lemma 3.5(2), we have $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma') \vdash \mathtt{H2I}(A) : \mathbf{Typ}$. By definition of $\mathtt{H2I}$, we have $x \notin \mathrm{dom}(\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma'))$. Thus, C_Var derives

$$\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma'), x : \mathtt{H2I}(A)$$

as required.

(4)(5)(6) By mutual induction on derivations of the judgments. We proceed by case analysis on the rule applied lastly to the derivations.

Case Th_Var**:** We have
- $V = x$,
- $\Delta \vdash \Gamma$, and
- $x : A \in \Gamma$,

for some x. By Theorem (3), we have $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma)$. By definition of $\mathtt{H2I}$, we have $x : \mathtt{H2I}(A) \in \mathtt{H2I}(\Gamma)$. Thus, T_Var derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash x : \mathtt{H2I}(A) \mid \emptyset_E$$

as required.

Case Th_Lam**:** We have
- $V = \lambda x^{A_0}.M$,
- $A = A_0 \to A_1!E$, and
- $\Delta; \Gamma, x : A_0 \vdash M : A_1!E$,

for some x, A_0, A_1, E, and M. By the induction hypothesis, we have

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma), x : \mathtt{H2I}(A_0) \vdash \mathtt{H2I}(M) : \mathtt{H2I}(A_1) \mid \mathtt{H2I}(E).$$

Without loss of generality, we can choose z such that
- $z \notin FV(\mathtt{H2I}(M))$,
- $z \neq x$,
- $z \notin \mathrm{dom}(\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma))$, and
- $\mathtt{H2I}(\lambda x^{A_0}.M) = \mathbf{fun}(z, x, \mathtt{H2I}(M))$.

By Lemma 3.12 and Lemma 3.2(2) and Lemma 3.6, we have
- $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(A_1) : \mathbf{Typ}$ and
- $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(E) : \mathbf{Eff}$.

By Lemma 3.9, we have $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma), x : \mathtt{H2I}(A_0)$. Since only C_Var can derive this judgment, we have $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(A_0) : \mathbf{Typ}$. Thus, C_Var derives

$$\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma), z : \mathtt{H2I}(A_0) \to_{\mathtt{H2I}(E)} \mathtt{H2I}(A_1).$$

Thus, Lemma 3.5(5) and T_Abs derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathbf{fun}(z, x, \mathtt{H2I}(M)) : \mathtt{H2I}(A_0) \to_{\mathtt{H2I}(E)} \mathtt{H2I}(A_1) \mid \emptyset_E$$

as required.

Case Th_PolyLam**:** We have
- $V = \Lambda\alpha^K.M$,
- $A = \forall\alpha^K.B!E$,
- $\Delta, \alpha : K; \Gamma \vdash M : B!E$, and

– $\Delta \vdash \Gamma$,

for some α, K, M, B, and E. By the induction hypothesis and case (3), we have

– $\vdash \mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma)$ and
– $\mathtt{H2I}(\Delta), \alpha : \mathtt{H2I}(K), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(M) : \mathtt{H2I}(B) \mid \mathtt{H2I}(E)$.

By applying Lemma 4.9 repeatedly, we have

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma), \alpha : \mathtt{H2I}(K) \vdash \mathtt{H2I}(M) : \mathtt{H2I}(B) \mid \mathtt{H2I}(E).$$

Thus, T_TAbs derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \Lambda\alpha : \mathtt{H2I}(K).\mathtt{H2I}(M) : \forall\alpha : \mathtt{H2I}(K).\mathtt{H2I}(B)^{\mathtt{H2I}(E)} \mid \emptyset_E$$

as required.

Case Th_App**:** We have

– $M = V\, W$,
– $\Delta; \Gamma \vdash V : B \to A!E$, and
– $\Delta; \Gamma \vdash W : B$,

for some V, W, and B. By the induction hypothesis, we have

– $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V) : \mathtt{H2I}(B) \to_{\mathtt{H2I}(E)} \mathtt{H2I}(A) \mid \emptyset_E$ and
– $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(W) : \mathtt{H2I}(B) \mid \emptyset_E$.

Thus, T_App derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V)\, \mathtt{H2I}(W) : \mathtt{H2I}(A) \mid \mathtt{H2I}(E)$$

as required.

Case Th_PolyApp**:** We have

– $M = V\, T$,
– $A = (B!E)[T/\alpha]$,
– $\Delta; \Gamma \vdash V : \forall\alpha^K.B!E$, and
– $\Delta \vdash T : K$

for some V, T, α, K, B, and E. By the induction hypothesis and case (2), we have

– $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V) : \forall\alpha : \mathtt{H2I}(K).\mathtt{H2I}(B)^{\mathtt{H2I}(E)} \mid \emptyset_E$ and
– $\mathtt{H2I}(\Delta) \vdash \mathtt{H2I}(T) : \mathtt{H2I}(K)$.

By Lemma 3.9 and Lemma 3.5(2), we have $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(T) : \mathtt{H2I}(K)$. Thus, T_TApp derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V)\, \mathtt{H2I}(T) : \mathtt{H2I}(B[T/\alpha]) \mid \mathtt{H2I}(E[T/\alpha])$$

as required.

Case Th_Return**:** We have

– $M = \mathbf{return}\, V$,
– $\Delta; \Gamma \vdash V : A$, and
– $\Delta \vdash E : \mathsf{Effect}$,

for some V. By the induction hypothesis and case (2), we have

– $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V) : \mathtt{H2I}(A) \mid \emptyset_E$ and
– $\mathtt{H2I}(\Delta) \vdash \mathtt{H2I}(E) : \mathbf{Eff}$.

Thus, T_Sub derives

$$\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(V) : \mathtt{H2I}(A) \mid \mathtt{H2I}(E)$$

as required.

Case Th_Let**:** We have

– $M = \mathbf{let}\, x \leftarrow M_0 \,\mathbf{in}\, M_1$,
– $\Delta; \Gamma \vdash M_0 : B!E$, and
– $\Delta; \Gamma, x : B \vdash M_1 : A!E$,

for some x, M_0, M_1, and B. By the induction hypothesis, we have

– $\mathtt{H2I}(\Delta), \mathtt{H2I}(\Gamma) \vdash \mathtt{H2I}(M_0) : \mathtt{H2I}(B) \mid \mathtt{H2I}(E)$ and

– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma), x : \texttt{H2I}(B) \vdash \texttt{H2I}(M_1) : \texttt{H2I}(A) \mid \texttt{H2I}(E)$.

Thus, T_Let derives

$$\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash \textbf{let}\, x = \texttt{H2I}(M_0)\, \textbf{in}\, \texttt{H2I}(M_1) : \texttt{H2I}(A) \mid \texttt{H2I}(E)$$

as required.

Case Th_Do**:** We have

– $M = (\textbf{do}\, l\, V)^E$,
– $\Delta; \Gamma \vdash V : B$,
– $\{l : \mathsf{Pre}(B \to A); R\}$, and
– $\Delta \vdash E : \mathsf{Effect}$,

for some l, V, E, B, and R. By the induction hypothesis and case (2), we have

– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash \texttt{H2I}(V) : \texttt{H2I}(B) \mid \emptyset_E$ and
– $\texttt{H2I}(\Delta) \vdash \texttt{H2I}(E) : \textbf{Eff}$.

There uniquely exists some $\mathcal{L}$ such that

– $\mathcal{L} \in \texttt{L2S}(\mathbb{L})$,
– $l \in \mathcal{L}$, and
– $\mathcal{L} \subseteq \mathrm{dom}(E)$.

Thus, we have

– $\texttt{r2l}(\mathcal{L}) :: \sigma \in \Sigma$,
– $\texttt{l2Op}(l) : \texttt{l2T}(l) \in \sigma$, and
– $\langle \texttt{r2l}(\mathcal{L}) \mid \varepsilon \rangle \sim_{\mathrm{SimpR}} \texttt{H2I}(E)$,

for some σ and ε. Because Lemma 3.9 gives us $\vdash \texttt{H2I}(\Delta), \texttt{H2I}(\Gamma)$, T_Op and T_App and T_Sub derive

$$\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash \texttt{l2Op}(l)_{\texttt{r2l}(\mathcal{L})}\, \texttt{H2I}(V) : \texttt{H2I}(B) \mid \texttt{H2I}(E)$$

as required.

Case Th_Handle**:** We have

– $M = \textbf{handle}\, N\, \textbf{with}\, H$,
– $\Delta; \Gamma \vdash N : B!E'$, and
– $\Delta; \Gamma \vdash H : B!E' \Rightarrow A!E$,

for some N, H, B, and E'. By the induction hypothesis, we have

– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash \texttt{H2I}(N) : \texttt{H2I}(B) \mid \texttt{H2I}(E')$,
– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash_\sigma \texttt{H2I}(H) : \texttt{H2I}(B) \Rightarrow^{\texttt{H2I}(E)} \texttt{H2I}(A)$,
– $\texttt{r2l}(H) :: \sigma \in \Sigma$, and
– $\langle \texttt{r2l}(H) \mid \texttt{H2I}(E) \rangle \sim_{\mathrm{SimpR}} \texttt{H2I}(E')$.

for some σ. Thus, T_Handling derives

$$\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash \textbf{handle}_{\texttt{r2l}(H)}\, \texttt{H2I}(N)\, \textbf{with}\, \texttt{H2I}(H) : \texttt{H2I}(A) \mid \texttt{H2I}(E)$$

as required.

Case Hh_Handler**:** We have

– $\Delta; \Gamma, x : A \vdash M : B!E'$,
– $\Delta; \Gamma, y_i : A_i, r_i : B_i \to B!E' \vdash N_i : B!E'$ for any $i \in \{1, \ldots, n\}$,
– $E = \{l_1 : \mathsf{Pre}(A_1 \to B_1); \cdots; l_n : \mathsf{Pre}(A_n \to B_n); R\}$, and
– $E' = \{l_1 : P_1; \cdots; l_n : P_n; R\}$,

for some A_i, B_i, and P_i, where $i \in \{1, \ldots, n\}$. By the assumptions, we have

– $\{l_1, \ldots, l_n\} \in \texttt{L2S}(\mathbb{L})$,
– $\texttt{l2T}(l_i) = \texttt{H2I}(A_i) \Rightarrow \texttt{H2I}(B_i)$ for any $i \in \{1, \ldots, n\}$,
– $\texttt{r2l}(\{l_1, \ldots, l_n\}) :: \{\texttt{l2Op}(l_1) : \texttt{l2T}(l_1), \ldots, \texttt{l2Op}(l_n) : \texttt{l2T}(l_n)\} \in \Sigma$, and
– $\forall i \in \{1, \ldots, n\}.(P_i = \mathsf{Abs})$ or $\forall i \in \{1, \ldots, n\}.(P_i = \mathsf{Pre}(A_i \to B_i))$.

Thus, we have $\langle \texttt{r2l}(H) \mid \texttt{H2I}(E') \rangle \sim_{\mathrm{SimpR}} \texttt{H2I}(E)$.

By the induction hypothesis, we have

– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma), x : \texttt{H2I}(A) \vdash \texttt{H2I}(M) : \texttt{H2I}(B) \mid \texttt{H2I}(E')$ and
– $\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma), y_i : \texttt{H2I}(A_i), r_i : \texttt{H2I}(B_i) \rightarrow_{\texttt{H2I}(E')} \texttt{H2I}(B) \vdash \texttt{H2I}(N_i) : \texttt{H2I}(B) \mid \texttt{H2I}(E')$ for any $i \in \{1, \ldots, n\}$.

Therefore, H_Return and H_Op derive

$$\texttt{H2I}(\Delta), \texttt{H2I}(\Gamma) \vdash_{\{\texttt{12Op}(l_1):\texttt{12T}(l_1), \ldots, \texttt{12Op}(l_n):\texttt{12T}(l_n)\}} \texttt{H2I}(H) : \texttt{H2I}(A) \Rightarrow^{\texttt{H2I}(E')} \texttt{H2I}(B).$$

Thus, the required result is achieved.

■

4.3 Comparison to [Leijen(2017)]

We give the targets of comparison: one is an instance of λ_{EA} (Example 1.26), and another is a minorly changed language of [Leijen(2017)].

Definition 4.11 (Minor Changed Version of [Leijen(2017)])**.** *Change list:*

- *changing implicit polymorphism to explicit polymorphism,*
- *removing constants from values,*
- *removing the assumption that the initial environment has effect declarations, and adding such declarations to Σ,*
- *adding type variables to contexts, and*
- *adding well-formedness of contexts.*

The syntax of a minor changed version of [Leijen(2017)] is as follows.

$$\begin{array}{rcll}
e & ::= & v \mid e(e) \mid e(\tau^k) \mid \mathsf{val}\, x = e_1; e_2 \mid \mathsf{handle}\{h\}(e) & \textit{(expressions)} \\
v & ::= & x \mid op \mid \lambda x.e \mid \Lambda\alpha^k.e & \textit{(values)} \\
h & ::= & \mathsf{return}\, x \rightarrow e \mid op(x) \rightarrow e; h & \textit{(clauses)} \\
\tau^k & ::= & \alpha^k \mid c^{(k_1,\ldots,k_n)\rightarrow k}\langle \tau_1^{k_1}, \ldots, \tau_n^{k_n}\rangle & \textit{(types)} \\
k & ::= & * \mid \mathsf{e} \mid \mathsf{k} \mid (k_1, \ldots, k_n) \rightarrow k & \textit{(kinds)} \\
\sigma & ::= & \forall\alpha^k.\sigma \mid \tau^* & \textit{(type scheme)} \\
\Gamma & ::= & \emptyset \mid \Gamma, x : \sigma \mid \Gamma, \alpha^k & \textit{(typing contexts)} \\
\Sigma & ::= & \emptyset \mid \Sigma, l : \{op_1, \ldots, op_n\} & \textit{(signature environment)} \\
- \rightarrow -- & :: & (*, \mathsf{e}, *) \rightarrow * & \textit{(functions)} \\
\langle\rangle & :: & \mathsf{e} & \textit{(empty effect)} \\
\langle - \mid - \rangle & :: & (\mathsf{k}, \mathsf{e}) \rightarrow \mathsf{e} & \textit{(effect extension)} \\
l & := & c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}\langle \tau_1^{k_1}, \ldots, \tau_n^{k_n}\rangle & \textit{(effect labels)}
\end{array}$$

Well-formedness rules, free type variable, and typing rules consist of the following.

Contexts Well-formedness $\boxed{\vdash \Gamma}$

$$\frac{}{\vdash \emptyset}\ \textsc{Cl\_Empty} \qquad \frac{\vdash \Gamma \quad x \notin \mathrm{dom}(\Gamma) \quad \mathsf{ftv}(\tau^*) \setminus \{\overline{\alpha^{k'}}\} \subseteq \Gamma \quad \forall \overline{k}.(\{\overline{\alpha^k}\} \cap \Gamma = \emptyset)}{\vdash \Gamma, x : \forall \overline{\alpha^{k'}}.\tau^*}\ \textsc{Cl\_Var}$$

$$\frac{\vdash \Gamma \quad \forall k.(\alpha^k \notin \Gamma)}{\vdash \Gamma, \alpha^{k'}}\ \textsc{Cl\_TVar}$$

Free Type Variable $\boxed{\mathsf{ftv}(\tau^k)}$ $\boxed{\mathsf{ftv}(\sigma)}$

$$\mathsf{ftv}(\alpha^k) = \{\alpha^k\} \quad \mathsf{ftv}(\tau_1^* \rightarrow \tau_2^{\mathsf{e}}\, \tau_3^*) = \mathsf{ftv}(\tau_1^*) \cup \mathsf{ftv}(\tau_2^{\mathsf{e}}) \cup \mathsf{ftv}(\tau_3^*) \quad \mathsf{ftv}(\langle\rangle) = \emptyset \quad \mathsf{ftv}(\langle \tau_1^{\mathsf{k}} \mid \tau_2^{\mathsf{e}}\rangle) = \mathsf{ftv}(\tau_1^{\mathsf{k}}) \cup \mathsf{ftv}(\tau_2^{\mathsf{e}})$$

$$\mathsf{ftv}(c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}\langle \tau_1^{k_1}, \ldots, \tau_n^{k_n}\rangle) = \bigcup_{i \in \{1,\ldots,n\}} \mathsf{ftv}(\tau_i^{k_i}) \qquad \mathsf{ftv}(\forall\alpha^k.\sigma) = \mathsf{ftv}(\sigma) \setminus \{\alpha^k\}$$

Typing $\boxed{\Gamma \vdash e : \sigma \mid \epsilon}$

$$\frac{\vdash \Gamma \quad \Gamma(x) = \sigma \quad \mathsf{ftv}(\epsilon) \subseteq \Gamma}{\Gamma \vdash x : \sigma \mid \epsilon} \text{ TL\_VAR} \qquad \frac{\Gamma, x : \tau_1 \vdash e : \tau_2 \mid \epsilon' \quad \mathsf{ftv}(\epsilon) \subseteq \Gamma}{\Gamma \vdash \lambda x.e : \tau_1 \rightarrow \epsilon' \, \tau_2 \mid \epsilon} \text{ TL\_LAM}$$

$$\frac{\Gamma \vdash e_1 : \sigma_1 \mid \epsilon \quad \Gamma, x : \sigma \vdash e_2 : \tau \mid \epsilon}{\Gamma \vdash \mathsf{val}\, x = e_1; e_2 : \tau \mid \epsilon} \text{ TL\_LET} \qquad \frac{\Gamma \vdash e_1 : \tau_2 \rightarrow \epsilon \, \tau \mid \epsilon \quad \Gamma \vdash e_2 : \tau_2 \mid \epsilon}{\Gamma \vdash e_1(e_2) : \tau \mid \epsilon} \text{ TL\_APP}$$

$$\frac{\Gamma, \overline{\alpha^k} \vdash e : \tau \mid \langle\rangle \quad \mathsf{ftv}(\epsilon) \subseteq \Gamma}{\Gamma \vdash \Lambda \overline{\alpha^k}.e : \forall \overline{\alpha^k}.\tau \mid \epsilon} \text{ TL\_TABS} \qquad \frac{\Gamma \vdash e : \forall \overline{\alpha^k}.\tau \mid \epsilon \quad \mathsf{ftv}(\overline{\tau_0^k}) \subseteq \Gamma}{\Gamma \vdash e(\overline{\tau_0^k}) : \tau[\overline{\alpha^k \mapsto \tau_0^k}] \mid \epsilon} \text{ TL\_TAPP}$$

$$\frac{\begin{array}{c}\Gamma \vdash e : \tau \mid \langle l \mid \epsilon \rangle \quad \Gamma, x : \tau \vdash e_r : \tau_r \mid \epsilon \\ \Sigma(l) = \{op_1, \ldots, op_n\} \quad \Gamma \vdash op_i : \tau_i \rightarrow \langle l \mid \langle\rangle\rangle \, \tau'_i \mid \langle\rangle \\ \Gamma, x_i : \tau_i, resume : \tau'_i \rightarrow \epsilon \, \tau_r \vdash e_i : \tau_r \mid \epsilon\end{array}}{\Gamma \vdash \mathsf{handle}\{op_1(x_1) \rightarrow e_1; \cdots; op_n(x_n) \rightarrow e_n; \mathsf{return}\, x \rightarrow e_r\}(e) : \tau_r \mid \epsilon} \text{ TL\_HANDLE}$$

$$\frac{\Gamma \vdash e : \tau_1 \rightarrow \langle l_1, \ldots, l_n \mid \langle\rangle\rangle \, \tau_2 \mid \epsilon \quad \mathsf{ftv}(\epsilon') \subseteq \Gamma}{\Gamma \vdash e : \tau_1 \rightarrow \langle l_1, \ldots, l_n \mid \epsilon' \rangle \, \tau_2 \mid \epsilon} \text{ TL\_OPEN}$$

Definition 4.12 (Translation from Leijen's to An Instance)**.** *We assume that there is no constants other than* $- \rightarrow --$, $\langle\rangle$, $\langle - \mid - \rangle$ *and* $c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}$. *We define* `c2l` *as the injective function that assigns a label name* l *such that* l *: (belonging to an instance) to* $c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}$ *(belonging to Leijen's). We define* `L2I`, `h2l`, *and* `c2l` *as follows. We require* `c2l` *to be injective.*

Kinds

$$\mathtt{L2I}(*) = \textbf{Typ} \qquad \mathtt{L2I}(\mathsf{e}) = \textbf{Eff} \qquad \mathtt{L2I}(\mathsf{k}) = \textbf{Lab}$$

Types

$$\begin{array}{rcl} \mathtt{L2I}(\alpha^k) &=& \alpha \\ \mathtt{L2I}(\langle\rangle) &=& \langle\rangle \\ \mathtt{L2I}(\forall \alpha^k.\sigma) &=& \forall \alpha : \mathtt{L2I}(k).\mathtt{L2I}(\sigma)^{\langle\rangle} \\ \mathtt{L2I}(c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}\langle \tau_1^{k_1}, \ldots, \tau_n^{k_n}\rangle) &=& \mathtt{c2l}(c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}})\, \mathtt{L2I}(\tau_1^{k_1}) \cdots \mathtt{L2I}(\tau_n^{k_n}) \\ \mathtt{c2l}(c^{(k_1,\ldots,k_n)\rightarrow \mathsf{k}}) &=& l \quad (\textit{where } l : \mathtt{L2I}(k_1) \times \ldots \times \mathtt{L2I}(k_n) \rightarrow \textbf{Lab} \in \Sigma_{\text{eff}}) \end{array}$$

$$\begin{array}{rcl} \mathtt{L2I}(\tau_1^* \rightarrow \tau_2^{\mathsf{e}}\, \tau_3^*) &=& \mathtt{L2I}(\tau_1^*) \rightarrow_{\mathtt{L2I}(\tau_2^{\mathsf{e}})} \mathtt{L2I}(\tau_3^*) \\ \mathtt{L2I}(\langle l \mid \epsilon \rangle) &=& \langle \mathtt{L2I}(l) \mid \mathtt{L2I}(\epsilon) \rangle \end{array}$$

Expressions

$$\begin{array}{rcl} \mathtt{L2I}(x) &=& x \\ \mathtt{L2I}(op) &=& \mathsf{op}_{\mathtt{c2l}(c)\, \overline{\mathtt{L2I}(\tau^k)}} \quad (\textit{where } op \in \Sigma(c\langle \overline{\tau^k} \rangle)) \\ \mathtt{L2I}(\lambda x.e) &=& \textbf{fun}(z, x, \mathtt{L2I}(e)) \quad (\textit{where } z \textit{ is fresh}) \\ \mathtt{L2I}(\Lambda \alpha^k.e) &=& \Lambda \alpha : \mathtt{L2I}(k).\mathtt{L2I}(e) \\ \mathtt{L2I}(e_1(e_2)) &=& \textbf{let}\, x = \mathtt{L2I}(e_1)\, \textbf{in}\, \textbf{let}\, y = \mathtt{L2I}(e_2)\, \textbf{in}\, x\, y \\ \mathtt{L2I}(e(\tau^k)) &=& \textbf{let}\, x = \mathtt{L2I}(e)\, \textbf{in}\, x\, \mathtt{L2I}(\tau^k) \\ \mathtt{L2I}(\mathsf{val}\, x = e_1; e_2) &=& \textbf{let}\, x = \mathtt{L2I}(e_1)\, \textbf{in}\, \mathtt{L2I}(e_2) \\ \mathtt{L2I}(\mathsf{handle}\{h\}(e)) &=& \textbf{handle}_{\mathtt{h2l}(h)}\, \mathtt{L2I}(e)\, \textbf{with}\, \mathtt{L2I}(h) \end{array}$$

Handlers

$$\begin{array}{rcl} \mathtt{L2I}(\mathsf{return}\, x \rightarrow e) &=& \{\textbf{return}\, x \mapsto \mathtt{L2I}(e)\} \\ \mathtt{L2I}(op(x) \rightarrow e; h) &=& \mathtt{L2I}(h) \uplus \{\mathsf{op}\, x\, resume \mapsto \mathtt{L2I}(e)\} \end{array}$$

Contexts

$$\mathtt{L2I}(\emptyset) = \emptyset \qquad \mathtt{L2I}(\Gamma, x : \sigma) = \mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\sigma) \qquad \mathtt{L2I}(\Gamma, \alpha^k) = \mathtt{L2I}(\Gamma), \alpha : \mathtt{L2I}(k)$$

Effect Contexts

$$\begin{array}{rcl} \mathtt{L2I}(\emptyset) &=& \emptyset \\ \mathtt{L2I}(\Sigma, c\langle \overline{\tau^k} \rangle : \{op_1, \ldots, op_n\}) &=& \mathtt{L2I}(\Sigma), \mathtt{c2l}(c) :: \forall \overline{\alpha : \mathtt{L2I}(k)}.\sigma \\ && (\textit{where } \Gamma_0 \ni op_i : \tau_i \rightarrow \langle c\langle \overline{\tau^k} \rangle \mid \langle\rangle \rangle \, \tau'_i \\ && \textit{and } \sigma[\overline{\mathtt{L2I}(\tau^k)/\alpha}] = \{\mathsf{op}_1 : \tau_1 \Rightarrow {\tau_1}', \ldots, \mathsf{op}_n : \tau_n \Rightarrow {\tau_n}'\}) \end{array}$$

Translation from Handlers to Labels

$$\mathtt{h2l}(op_1(x_1) \to e_1; \cdots; op_n(x_n) \to e_n; \mathsf{return}\, x \to e) = \begin{cases} l & (\textit{if } l = \mathtt{c2l}(c) \textit{ and } \{op_1, \ldots, op_n\} = \Sigma(c\langle \cdots \rangle)) \\ \textit{undefined} & (\textit{otherwise}) \end{cases}$$

Lemma 4.13. *If* $x \notin \mathrm{dom}(\Gamma)$*, then* $x \notin \mathrm{dom}(\mathtt{L2I}(\Gamma))$*.*

Proof. Straightforward by structual induction on Γ and the definition of $\mathtt{L2I}$. ■

Lemma 4.14. $\alpha^k \in \Gamma$ *iff* $\alpha : \mathtt{L2I}(k) \in \mathtt{L2I}(\Gamma)$*.*

Proof. Straightforward by structual induction on Γ and the definition of $\mathtt{L2I}$. ■

Lemma 4.15. *If* $x : \sigma \in \Gamma$*, then* $x : \mathtt{L2I}(\sigma) \in \mathtt{L2I}(\Gamma)$*.*

Proof. Straightforward by structual induction on Γ and the definition of $\mathtt{L2I}$. ■

Lemma 4.16. *If* $\Gamma \vdash e : \sigma \mid \epsilon$*, then* $\vdash \Gamma$*.*

Proof. Straightforward by induction on a derivation of $\Gamma \vdash e : \sigma \mid \epsilon$. ■

Theorem 4.17.

(1) If $\mathsf{ftv}(\tau^k) \subseteq \Gamma$ *and* $\vdash \mathtt{L2I}(\Gamma)$*, then* $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau^k) : \mathtt{L2I}(k)$*.*

(2) If $\vdash \Gamma$*, then* $\vdash \mathtt{L2I}(\Gamma)$*.*

(3) If $\Gamma \vdash e : \sigma \mid \epsilon$*, then* $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e) : \mathtt{L2I}(\sigma) \mid \mathtt{L2I}(\epsilon)$*.*

Proof.

(1) By structual induction on τ^k.

Case $\tau^k = \alpha^k$**:** We have $\alpha^k \in \Gamma$. By Lemma 4.14, we have $\alpha : \mathtt{L2I}(k) \in \mathtt{L2I}(\Gamma)$. Thus, K_VAR derives

$$\mathtt{L2I}(\Gamma) \vdash \alpha : \mathtt{L2I}(k)$$

as required.

Case $\tau^k = \tau_1^* \to \tau_2^{\mathsf{e}}\, \tau_3^*$**:** We have

- $k = *$,
- $\mathsf{ftv}(\tau_1^*) \subseteq \Gamma$,
- $\mathsf{ftv}(\tau_2^{\mathsf{e}}) \subseteq \Gamma$, and
- $\mathsf{ftv}(\tau_3^*) \subseteq \Gamma$.

By the induction hypothesis, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1^*) : \mathbf{Typ}$,
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_2^{\mathsf{e}}) : \mathbf{Eff}$, and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_3^*) : \mathbf{Typ}$.

Thus, K_FUN derives

$$\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1^*) \to_{\mathtt{L2I}(\tau_2^{\mathsf{e}})} \mathtt{L2I}(\tau_3^*) : \mathbf{Typ}$$

as required.

Case $\tau^k = \langle\rangle$**:** We have $k = \mathsf{e}$. Thus, by $\vdash \mathtt{L2I}(\Gamma)$, we have

$$\mathtt{L2I}(\Gamma) \vdash \langle\rangle : \mathbf{Eff}$$

as required.

Case $\tau^k = \langle \tau_1^{\mathsf{k}} \mid \tau_2^{\mathsf{e}} \rangle$**:** We have

- $k = \mathsf{e}$,
- $\mathsf{ftv}(\tau_1^{\mathsf{k}}) \subseteq \Gamma$, and
- $\mathsf{ftv}(\tau_2^{\mathsf{e}}) \subseteq \Gamma$.

By the induction hypothesis, we have

- $\texttt{L2I}(\Gamma) \vdash \texttt{L2I}(\tau_1^{\mathsf{k}}) : \textbf{Lab}$ and
- $\texttt{L2I}(\Gamma) \vdash \texttt{L2I}(\tau_2^{\mathsf{e}}) : \textbf{Eff}$.

Thus, K_Cons derives

$$\texttt{L2I}(\Gamma) \vdash \langle \texttt{L2I}(\tau_1^{\mathsf{k}}) \mid \texttt{L2I}(\tau_2^{\mathsf{e}}) \rangle : \textbf{Eff}$$

as required.

Case $\tau^k = c^{(k_1,\ldots,k_n)\to\mathsf{k}}\langle \tau_1^{k_1}, \ldots, \tau_n^{k_n} \rangle$**:** We have $\mathsf{ftv}(\tau_i^i) \subseteq \Gamma$ for any $i \in \{1, \ldots, n\}$. By the induction hypothesis and definition of $\texttt{c2l}$, we have

- $\texttt{L2I}(\Gamma) \vdash \texttt{L2I}(\tau_i^{k_i}) : \texttt{L2I}(k_i)$ for any $i \in \{1, \ldots, n\}$,
- $\texttt{c2l}(c^{(k_1,\ldots,k_n)\to\mathsf{k}}) : \texttt{L2I}(k_1) \times \ldots \times \texttt{L2I}(k_n) \to \textbf{Lab} \in \Sigma_{\mathrm{eff}}$.

Thus, K_Cons derives

$$\texttt{L2I}(\Gamma) \vdash \texttt{c2l}(c^{(k_1,\ldots,k_n)\to\mathsf{k}})\, \texttt{L2I}(\tau_1^{k_1}) \cdots \texttt{L2I}(\tau_n^{k_n}) : \textbf{Lab}$$

as required.

(2) By induction on a derivation of the judgment. We proceed by case analysis on the rule applied lastly to the derivation.

Case Cl_Empty**:** Clearly by C_Empty and the definition of $\texttt{L2I}$.

Case Cl_Var**:** We have

- $\Gamma = \Gamma', x : \forall \overline{\alpha^{k'}}.\tau^*$,
- $\vdash \Gamma'$,
- $x \notin \mathrm{dom}(\Gamma')$,
- $\mathsf{ftv}(\tau^*) \setminus \{\overline{\alpha^{k'}}\} \subseteq \Gamma'$, and
- $\forall \overline{k}.(\{\overline{\alpha^k}\} \cap \Gamma' = \emptyset)$,

for some Γ', x, $\overline{\alpha^{k'}}$, and τ^*. By the induction hypothesis and Lemma 4.13, we have

- $\vdash \texttt{L2I}(\Gamma')$ and
- $x \notin \mathrm{dom}(\texttt{L2I}(\Gamma'))$.

By Lemma 4.14 and C_TVar, we have

$$\vdash \texttt{L2I}(\Gamma'), \overline{\alpha} : \overline{\texttt{L2I}(k')}.$$

By $\mathsf{ftv}(\tau^*) \subseteq \Gamma', \overline{\alpha^{k'}}$ and case (1), we have

$$\texttt{L2I}(\Gamma'), \overline{\alpha} : \overline{\texttt{L2I}(k')} \vdash \texttt{L2I}(\tau^*) : \textbf{Typ}.$$

Thus, K_Poly and C_Var derives

$$\vdash \texttt{L2I}(\Gamma'), x : \texttt{L2I}(\forall \overline{\alpha^{k'}}.\tau^*)$$

as required.

Case Cl_TVar**:** We have

- $\Gamma = \Gamma', \alpha^k$,
- $\vdash \Gamma'$, and
- $\forall k.(\alpha^k \notin \Gamma')$,

for some Γ' and α^k. By the induction hypothesis and Lemma 4.14, we have

- $\vdash \texttt{L2I}(\Gamma')$ and
- $\alpha \notin \mathrm{dom}(\texttt{L2I}(\Gamma'))$.

Thus, C_TVar derives

$$\vdash \texttt{L2I}(\Gamma'), \alpha : \texttt{L2I}(k)$$

as required.

(3) By induction on a derivation of the judgement. We proceed by case analysis on the rule applied lastly to the derivation.

Case Tl_Var**:** We have

- $e = x$,
- $\Gamma(x) = \sigma$, and
- $\mathsf{ftv}(\epsilon) \subseteq \Gamma$

for some x. By Lemma 4.16 and case (2), we have $\vdash \mathtt{L2I}(\Gamma)$. By case (1), we have

$$\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\epsilon) : \mathbf{Eff}.$$

By Lemma 4.15, we have $x : \mathtt{L2I}(\sigma) \in \mathtt{L2I}(\Gamma)$. Thus, T_Var derives

$$\mathtt{L2I}(\Gamma) \vdash x : \mathtt{L2I}(\sigma) \mid \langle\rangle.$$

By Lemma 3.12, we have $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\sigma) : \mathbf{Typ}$. Thus, ST_Refl and Lemma 3.3(1) and T_Sub derive

$$\mathtt{L2I}(\Gamma) \vdash x : \mathtt{L2I}(\sigma) \mid \mathtt{L2I}(\epsilon)$$

as required.

Case Tl_Lam**:** We have

- $e = \lambda x.e'$,
- $\sigma = \tau_1 \rightarrow \epsilon' \, \tau_2$,
- $\Gamma, x : \tau_1 \vdash e' : \tau_2 \mid \epsilon'$, and
- $\mathsf{ftv}(\epsilon) \subseteq \Gamma$

for some x, e', τ_1, ϵ', and τ_2. By Lemma 4.16, we have $\vdash \Gamma, x : \tau_1$. Since only Cl_Var can derive $\vdash \Gamma, x : \tau_1$, we have $\vdash \Gamma$. By case (2), we have $\vdash \mathtt{L2I}(\Gamma)$. By the induction hypothesis and case (1), we have

- $\mathtt{L2I}(\Gamma, x : \tau_1) \vdash \mathtt{L2I}(e') : \mathtt{L2I}(\tau_2) \mid \mathtt{L2I}(\epsilon')$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\epsilon) : \mathbf{Eff}$.

Without loss of generality, we can choose z such that

- $z \notin FV(\mathtt{L2I}(e'))$,
- $z \neq x$,
- $z \notin \mathrm{dom}(\mathtt{L2I}(\Gamma))$, and
- $\mathtt{L2I}(\lambda x.e') = \mathbf{fun}(z, x, \mathtt{L2I}(e'))$.

By Lemma 3.12 and Lemma 3.2(2) and Lemma 3.6, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_2) : \mathbf{Typ}$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\epsilon') : \mathbf{Eff}$.

By Lemma 3.9, we have $\vdash \mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_1)$. Since only C_Var can derive $\vdash \mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_1)$, we have $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1) : \mathbf{Typ}$. Thus, K_Fun derives

$$\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1) \rightarrow_{\mathtt{L2I}(\epsilon')} \mathtt{L2I}(\tau_2) : \mathbf{Typ}.$$

Thus, C_Var derives

$$\vdash \mathtt{L2I}(\Gamma), z : \mathtt{L2I}(\tau_1) \rightarrow_{\mathtt{L2I}(\epsilon')} \mathtt{L2I}(\tau_2).$$

Thus, Lemma 3.5 and T_Abs derives

$$\mathtt{L2I}(\Gamma) \vdash \mathbf{fun}(z, x, \mathtt{L2I}(e')) : \mathtt{L2I}(\tau_1) \rightarrow_{\mathtt{L2I}(\epsilon')} \mathtt{L2I}(\tau_2) \mid \langle\rangle.$$

Thus, ST_Refl and T_Sub derive

$$\mathtt{L2I}(\Gamma) \vdash \mathbf{fun}(z, x, \mathtt{L2I}(e')) : \mathtt{L2I}(\tau_1) \rightarrow_{\mathtt{L2I}(\epsilon')} \mathtt{L2I}(\tau_2) \mid \mathtt{L2I}(\epsilon).$$

as required.

Case Tl_Let**:** We have

- $e = \mathsf{val}\, x = e_1; e_2$,
- $\sigma = \tau$,
- $\Gamma \vdash e_1 : \sigma' \mid \epsilon$, and
- $\Gamma, x : \sigma' \vdash e_2 : \tau \mid \epsilon$,

for some x, e_1, e_2, τ, and σ'. By the induction hypothesis, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e_1) : \mathtt{L2I}(\sigma') \mid \mathtt{L2I}(\epsilon)$ and
- $\mathtt{L2I}(\Gamma, x : \sigma') \vdash \mathtt{L2I}(e_2) : \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon)$.

By definition of L2I and T_LET, we have

$$\mathtt{L2I}(\Gamma) \vdash \textbf{let}\, x = \mathtt{L2I}(e_1)\, \textbf{in}\, \mathtt{L2I}(e_2) : \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon)$$

as required.

Case TL_APP**:** We have

- $e = e_1(e_2)$,
- $\sigma = \tau$,
- $\Gamma \vdash e_1 : \tau_2 \to \epsilon\, \tau \mid \epsilon$, and
- $\Gamma \vdash e_2 : \tau_2 \mid \epsilon$,

for some e_1, e_2, τ, and τ_2. By the induction hypothesis, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e_1) : \mathtt{L2I}(\tau_2) \to_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon)$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e_2) : \mathtt{L2I}(\tau_2) \mid \mathtt{L2I}(\epsilon)$.

Without loss of generality, we can choose x and y such that

- $x \neq y$,
- $x \notin \mathrm{dom}(\mathtt{L2I}(\Gamma))$, and
- $y \notin \mathrm{dom}(\mathtt{L2I}(\Gamma))$.

Because Lemma 3.12(1) and C_VAR give us

- $\vdash \mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_2) \to_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau)$ and
- $\vdash \mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_2) \to_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau), y : \mathtt{L2I}(\tau_2)$.

Thus, T_VAR and T_APP derive

$$\mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_2) \to_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau), y : \mathtt{L2I}(\tau_2) \vdash x\, y : \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon).$$

By Lemma 3.5(5), T_LET derive

$$\mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau_2) \to_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau) \vdash \textbf{let}\, y = \mathtt{L2I}(e_2)\, \textbf{in}\, x\, y : \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon).$$

Thus, T_LET derives

$$\mathtt{L2I}(\Gamma) \vdash \textbf{let}\, x = e_1\, \textbf{in}\, \textbf{let}\, y = \mathtt{L2I}(e_2)\, \textbf{in}\, x\, y : \mathtt{L2I}(\tau) \mid \mathtt{L2I}(\epsilon)$$

as required.

Case TL_TABS**:** We have

- $e = \Lambda \overline{\alpha^k}.e'$,
- $\sigma = \forall \overline{\alpha^k}.\tau$,
- $\Gamma, \overline{\alpha^k} \vdash e' : \tau \mid \langle\rangle$, and
- $\mathsf{ftv}(\epsilon) \subseteq \Gamma$,

for some $\overline{\alpha^k}$, e', and τ. By Lemma 4.16, we have $\vdash \Gamma, \overline{\alpha^k}$. Since only CL_TVAR derive $\vdash \Gamma, \overline{\alpha^k}$, we have $\vdash \Gamma$. By case (2), we have $\vdash \mathtt{L2I}(\Gamma)$. By the induction hypothesis and case (1), we have

- $\mathtt{L2I}(\Gamma), \overline{\alpha} : \overline{\mathtt{L2I}(k)} \vdash \mathtt{L2I}(e') : \mathtt{L2I}(\tau) \mid \langle\rangle$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\epsilon) : \textbf{Eff}$.

By applying T_TABS repeatedly, we have

$$\mathtt{L2I}(\Gamma) \vdash \Lambda \overline{\alpha} : \overline{\mathtt{L2I}(k)}.\mathtt{L2I}(e') : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \mid \langle\rangle.$$

Thus, T_SUB derives

$$\mathtt{L2I}(\Gamma) \vdash \Lambda \overline{\alpha} : \overline{\mathtt{L2I}(k)}.\mathtt{L2I}(e') : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \mid \mathtt{L2I}(\epsilon).$$

as required.

Case TL_TAPP**:** We have

- $e = e'(\overline{\tau_0^k})$,
- $\sigma = \tau[\overline{\alpha^k} \mapsto \overline{\tau_0^k}]$,

- $\Gamma \vdash e' : \forall \overline{\alpha^k}.\tau \mid \epsilon$, and
- $\mathsf{ftv}(\overline{\tau_0^k}) \subseteq \Gamma$,

for some e', $\overline{\tau_0^k}$, and $\overline{\alpha^k}$. By Lemma 4.16, we have $\vdash \Gamma$. By case (2), we have $\vdash \mathtt{L2I}(\Gamma)$. By the induction hypothesis and case (1), we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e') : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \mid \mathtt{L2I}(\epsilon)$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\overline{\tau_0^k}) : \overline{\mathtt{L2I}(k)}$.

Without loss of generality, we can choose x such that $x \notin \mathrm{dom}(\mathtt{L2I}(\Gamma))$. By Lemma 3.12(1) and C_Var, we have

$$\vdash \mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle}.$$

By Lemma 3.5(2), we have

$$\mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \vdash \overline{\tau_0^k} : \overline{\mathtt{L2I}(k)}.$$

Thus, T_Var and applying T_TApp repeatedly derive

$$\mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \vdash x\,\mathtt{L2I}(\overline{\tau_0^k}) : \mathtt{L2I}(\tau)[\overline{\tau_0^k}/\overline{\alpha^k}] \mid \langle\rangle.$$

Because Lemma 3.12(1) and Lemma 3.5(2) give us

- $\mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \vdash \mathtt{L2I}(\tau)[\overline{\tau_0^k}/\overline{\alpha^k}] : \mathbf{Eff}$ and
- $\mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \vdash \mathtt{L2I}(\epsilon) : \mathbf{Eff}$,

ST_Refl and T_Sub derives

$$\mathtt{L2I}(\Gamma), x : \forall \alpha_0 : \mathtt{L2I}(k_0).(\cdots(\forall \alpha_n : \mathtt{L2I}(k_n).\mathtt{L2I}(\tau)^{\langle\rangle})\cdots)^{\langle\rangle} \vdash x\,\mathtt{L2I}(\overline{\tau_0^k}) : \mathtt{L2I}(\tau)[\overline{\tau_0^k}/\overline{\alpha^k}] \mid \mathtt{L2I}(\epsilon).$$

Thus, T_Let derives

$$\mathtt{L2I}(\Gamma) \vdash \mathbf{let}\, x = \mathtt{L2I}(e')\,\mathbf{in}\, x\,\mathtt{L2I}(\overline{\tau_0^k}) : \mathtt{L2I}(\tau)[\overline{\tau_0^k}/\overline{\alpha^k}] \mid \mathtt{L2I}(\epsilon)$$

as required.

Case Tl_Handle**:** We have

- $h = op_1(x_1) \rightarrow e_1; \cdots; op_n(x_n) \rightarrow e_n; \mathsf{return}\, x \rightarrow e_r$,
- $e = \mathsf{handle}\{h\}(e')$,
- $\sigma = \tau_r$,
- $\Gamma \vdash e' : \tau \mid \langle c\langle\overline{\tau^k}\rangle \mid \epsilon\rangle$,
- $\Gamma, x : \tau \vdash e_r : \tau_r \mid \epsilon$,
- $\Sigma(c\langle\overline{\tau^k}\rangle) = \{op_1, \ldots, op_n\}$,
- $\Gamma \vdash op_i : \tau_i \rightarrow \langle c\langle\overline{\tau^k}\rangle \mid \langle\rangle\rangle\, \tau'_i \mid \langle\rangle$, and
- $\Gamma, \mathit{resume} : \tau'_i \rightarrow \epsilon\, \tau_r, x_i : \tau_i \vdash e_i : \tau_r \mid \epsilon$ for any $i \in \{1, \ldots, n\}$,

for some h, e', x, e_r, op_i, x_i, e_i, τ_i, τ'_i, and $c\langle\overline{\tau^k}\rangle$ where $i \in \{1, \ldots, n\}$. By the induction hypothesis and definition of L2I, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e') : \mathtt{L2I}(\tau) \mid \langle \mathtt{c2l}(c)\, \overline{\mathtt{L2I}(\tau^k)} \mid \mathtt{L2I}(\epsilon)\rangle$,
- $\mathtt{L2I}(\Gamma), x : \mathtt{L2I}(\tau) \vdash \mathtt{L2I}(e_r) : \mathtt{L2I}(\tau_r) \mid \mathtt{L2I}(\epsilon)$,
- $\mathtt{L2I}(\Gamma), x_i : \mathtt{L2I}(\tau_i), \mathit{resume} : \mathtt{L2I}(\tau'_i) \rightarrow_{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau_r) \vdash \mathtt{L2I}(e_i) : \mathtt{L2I}(\tau_r) \mid \mathtt{L2I}(\epsilon)$ for any $i \in \{1, \ldots, n\}$,
- $\mathtt{c2l}(c) :: \forall \overline{\alpha : \mathtt{L2I}(k)}.\sigma \in \mathtt{L2I}(\Sigma)$, and
- $\sigma[\overline{\mathtt{L2I}(\tau^k)/\alpha}] = \{\mathsf{op}_1 : \tau_1 \Rightarrow \tau'_1, \ldots, \mathsf{op}_n : \tau_n \Rightarrow \tau'_n\}$.

Because H_Return and H_Op derive

$$\mathtt{L2I}(\Gamma) \vdash_{\sigma[\overline{\mathtt{L2I}(\tau^k)/\alpha}]} \mathtt{L2I}(h) : \mathtt{L2I}(\tau) \Rightarrow^{\mathtt{L2I}(\epsilon)} \mathtt{L2I}(\tau_r),$$

T_Handling derives

$$\mathtt{L2I}(\Gamma) \vdash \mathbf{handle}_{\mathtt{h2l}(h)}\, \mathtt{L2I}(e')\,\mathbf{with}\,\mathtt{L2I}(h) : \mathtt{L2I}(\tau_r) \mid \mathtt{L2I}(\epsilon)$$

as required.

Case TL_OPEN**:** We have

- $\sigma = \tau_1 \to \langle l_1, \ldots, l_n \mid \epsilon' \rangle\, \tau_2$,
- $\Gamma \vdash e : \tau_1 \to \langle l_1, \ldots, l_n \mid \langle\rangle \rangle\, \tau_2 \mid \epsilon$, and
- $\mathsf{ftv}(\epsilon') \subseteq \Gamma$,

for some τ_1, τ_2, $l_1, \ldots, l_n$, and ϵ'. By Lemma 4.16, we have $\vdash \Gamma$. By case (2), we have $\vdash \mathtt{L2I}(\Gamma)$. By the induction hypothesis and case (1), we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e) : \mathtt{L2I}(\tau_1) \to_{\langle \mathtt{L2I}(l_1), \ldots, \mathtt{L2I}(l_n) \mid \langle\rangle\rangle} \mathtt{L2I}(\tau_2) \mid \mathtt{L2I}(\epsilon)$ and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\epsilon') : \mathbf{Eff}$.

By Lemma 3.12(1), we have

$$\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1) \to_{\langle \mathtt{L2I}(l_1), \ldots, \mathtt{L2I}(l_n) \mid \langle\rangle\rangle} \mathtt{L2I}(\tau_2) : \mathbf{Typ}.$$

Since only K_FUN can derive this judgment, we have

- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_1) : \mathbf{Typ}$,
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\langle \mathtt{L2I}(l_1), \ldots, \mathtt{L2I}(l_n) \mid \langle\rangle\rangle) : \mathbf{Eff}$, and
- $\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(\tau_2) : \mathbf{Typ}$.

Thus, by ST_REFL and ST_FUN and Lemma 3.3(1) and T_SUB, we have

$$\mathtt{L2I}(\Gamma) \vdash \mathtt{L2I}(e) : \mathtt{L2I}(\tau_1) \to_{\langle \mathtt{L2I}(l_1), \ldots, \mathtt{L2I}(l_n) \mid \epsilon' \rangle} \mathtt{L2I}(\tau_2) \mid \mathtt{L2I}(\epsilon)$$

as required.

■

References

[Hillerström et al.(2017)] Daniel Hillerström, Sam Lindley, Robert Atkey, and K. C. Sivaramakrishnan. 2017. Continuation Passing Style for Effect Handlers. In *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK (LIPIcs, Vol. 84)*, Dale Miller (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:19. https://doi.org/10.4230/LIPIcs.FSCD.2017.18

[Leijen(2017)] Daan Leijen. 2017. Type directed compilation of row-typed algebraic effects. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, Giuseppe Castagna and Andrew D. Gordon (Eds.). ACM, 486–499. https://doi.org/10.1145/3009837.3009872

[Pretnar(2015)] Matija Pretnar. 2015. An Introduction to Algebraic Effects and Handlers. Invited tutorial paper. In *The 31st Conference on the Mathematical Foundations of Programming Semantics, MFPS 2015, Nijmegen, The Netherlands, June 22-25, 2015 (Electronic Notes in Theoretical Computer Science, Vol. 319)*, Dan R. Ghica (Ed.). Elsevier, 19–35. https://doi.org/10.1016/j.entcs.2015.12.003