Type Vigilance and the Truth About Transient (Techreport)

Contents

Author's address:

1 Common Definitions

1.1 Simple Definitions

$\boxed{\text{Simple language}}$

$$
\begin{array}{lcl}
e & ::= & x \mid n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \lambda(x{:}\tau).\,e \mid \langle e, e\rangle \mid \mathsf{app}\{\tau\}\,e\,e \mid \mathsf{fst}\{\tau\}\,e \mid \mathsf{snd}\{\tau\}\,e \mid \mathit{binop}\,e\,e \mid \mathsf{cast}\,\{\tau \Leftarrow \tau\}\,e \mid \mathsf{if}\ e\ \mathsf{then}\ e\ \mathsf{else}\ e \\
\tau & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid \tau{\times}\tau \mid \tau{\rightarrow}\tau \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
\Gamma & ::= & \cdot \mid \Gamma, (x{:}\tau) \\
n & ::= & \mathbb{N} \\
i & ::= & \mathbb{Z}
\end{array}
$$

$\boxed{\Gamma \vdash_{\mathsf{sim}} e : \tau}$ typing

$$\textsc{T-Var}\ \frac{(x_0{:}\tau_0) \in \Gamma_0}{\Gamma_0 \vdash_{\mathsf{sim}} x_0 : \tau_0} \qquad \textsc{T-Nat}\ \frac{}{\Gamma_0 \vdash_{\mathsf{sim}} n_0 : \mathsf{Nat}} \qquad \textsc{T-Int}\ \frac{}{\Gamma_0 \vdash_{\mathsf{sim}} i_0 : \mathsf{Int}} \qquad \textsc{T-True}\ \frac{}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{True} : \mathsf{Bool}} \qquad \textsc{T-False}\ \frac{}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{False} : \mathsf{Bool}}$$

$$\textsc{T-Lam}\ \frac{\Gamma_0,\ (x_0{:}\tau_0) \vdash_{\mathsf{sim}} e_0 : \tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} \lambda(x_0{:}\tau_0).\,e_0 : \tau_0 \rightarrow \tau_1} \qquad \textsc{T-Pair}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\mathsf{sim}} e_1 : \tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} \langle e_0, e_1\rangle : \tau_0{\times}\tau_1} \qquad \textsc{T-Cast}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0 \quad \tau_0 \sim \tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\ e_0 : \tau_1}$$

$$\textsc{T-App}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0 \rightarrow \tau_1 \quad \Gamma_0 \vdash_{\mathsf{sim}} e_1 : \tau_0}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{app}\{\tau_1\}\ e_0\ e_1 : \tau_1} \qquad \textsc{T-Fst}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0{\times}\tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{fst}\{\tau_0\}\ e_0 : \tau_0} \qquad \textsc{T-Snd}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0{\times}\tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{snd}\{\tau_1\}\ e_0 : \tau_1} \qquad \textsc{T-Binop}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\mathsf{sim}} e_1 : \tau_1 \quad \Delta(\mathit{binop}, \tau_0, \tau_1) = \tau_2}{\Gamma_0 \vdash_{\mathsf{sim}} \mathit{binop}\,e_0\ e_1 : \tau_2}$$

$$\textsc{T-If}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \mathsf{Bool} \quad \Gamma_0 \vdash_{\mathsf{sim}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\mathsf{sim}} e_2 : \tau_0}{\Gamma_0 \vdash_{\mathsf{sim}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_0} \qquad \textsc{T-Sub}\ \frac{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_0 \quad \tau_0 \leqslant: \tau_1}{\Gamma_0 \vdash_{\mathsf{sim}} e_0 : \tau_1}$$

$\boxed{\tau \sim \tau}$

$$\frac{}{* \sim \tau} \qquad \frac{}{\tau \sim *} \qquad \frac{}{\mathsf{Nat} \sim \mathsf{Nat}} \qquad \frac{}{\mathsf{Int} \sim \mathsf{Int}} \qquad \frac{\tau_0 \sim \tau_2 \qquad \tau_1 \sim \tau_3}{\tau_0{\times}\tau_1 \sim \tau_2{\times}\tau_3} \qquad \frac{\tau_2 \sim \tau_0 \qquad \tau_1 \sim \tau_3}{\tau_0 \rightarrow \tau_1 \sim \tau_2 \rightarrow \tau_3}$$

$\boxed{\tau \leqslant: \tau}$

$$\frac{}{\mathsf{Nat} \leqslant: \mathsf{Int}} \qquad \frac{\tau_0 \leqslant: \tau_2 \qquad \tau_1 \leqslant: \tau_3}{\tau_0{\times}\tau_1 \leqslant: \tau_2{\times}\tau_3} \qquad \frac{\tau_2 \leqslant: \tau_0 \qquad \tau_1 \leqslant: \tau_3}{\tau_0 \rightarrow \tau_1 \leqslant: \tau_2 \rightarrow \tau_3} \qquad \frac{}{\tau_0 \leqslant: \tau_0}$$

$$\boxed{\Delta : \mathit{binop} \times \tau \times \tau \longrightarrow \tau}$$

$$\begin{aligned}
\Delta(\mathsf{sum}, \mathsf{Nat}, \mathsf{Nat}) &= \mathsf{Nat} \\
\Delta(\mathsf{sum}, \mathsf{Int}, \mathsf{Int}) &= \mathsf{Int} \\
\Delta(\mathsf{quotient}, \mathsf{Nat}, \mathsf{Nat}) &= \mathsf{Nat} \\
\Delta(\mathsf{quotient}, \mathsf{Int}, \mathsf{Int}) &= \mathsf{Int}
\end{aligned}$$

1.2 Evaluation Language Definitions

$\boxed{\text{Evaluation Language}}$

$$
\begin{array}{lcl}
v & ::= & n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \langle v, v\rangle \mid w \\
w & ::= & \lambda(x\!:\!\tau).\, e \mid \mathsf{grd}\,\{\tau \Leftarrow \tau\}\, w \\
E & ::= & [\,] \mid \langle E, e\rangle \mid \langle v, E\rangle \mid \mathsf{fst}\{\tau\}\, E \mid \mathsf{snd}\{\tau\}\, E \mid \mathsf{app}\{\tau\}\, E\; e \mid \mathsf{app}\{\tau\}\, v\; E \mid E\; e \mid v\; E \mid \mathit{binop}\, E\, e \mid \mathit{binop}\, v\, E \\
 & \mid & \mathsf{cast}\,\{\tau \Leftarrow \tau'\}\, E \mid \mathsf{if}\; E\; \mathsf{then}\; e\; \mathsf{else}\; e \mid \mathsf{mon}\,\{\tau \Leftarrow \tau\}\, E \mid \mathsf{assert}\, \tau\, E \\
\mathsf{Err}^{\circ} & ::= & \mathsf{Wrong} \\
\mathsf{Err}^{\bullet} & ::= & \mathsf{DivErr} \mid \mathsf{TypeErr}(\tau,\, v) \\
\mathsf{Err} & ::= & \mathsf{Err}^{\circ} \mid \mathsf{Err}^{\bullet} \\
e & ::= & \mathsf{Err} \mid x \mid n \mid i \mid \lambda(x\!:\!\tau).\, e \mid \langle e, e\rangle \mid \mathsf{app}\{\tau\}\, e\; e \mid e\; e \mid \mathsf{fst}\{\tau\}\, e \mid \mathsf{snd}\{\tau\}\, e \mid \mathit{binop}\, e\, e \mid \mathsf{cast}\,\{\tau \Leftarrow \tau'\}\, e \\
 & \mid & \mathsf{if}\; e\; \mathsf{then}\; e\; \mathsf{else}\; e \mid \mathsf{mon}\,\{\tau \Leftarrow \tau\}\, e \mid \mathsf{grd}\,\{\tau \Leftarrow \tau\}\, e \mid \mathsf{assert}\, \tau\, e \\
K & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid {*}{\times}{*} \mid {*}{\to}{*} \mid * \\
\tau & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid \tau{\times}\tau \mid \tau{\to}\tau \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
n & ::= & \mathbb{N} \\
i & ::= & \mathbb{Z}
\end{array}
$$

$\boxed{\sim : K{\times}v \longrightarrow \mathbb{B}}$

$$
v_0 \sim K_0 = \begin{cases}
\mathsf{True} & \\
\quad \text{if } K_0 = \mathsf{Nat} \text{ and } v_0 \in \mathbb{N} & \\
\quad \text{or } K_0 = \mathsf{Int} \text{ and } v_0 \in \mathbb{Z} & \\
\quad \text{or } K_0 = \mathsf{Bool} \text{ and } v_0 \in \mathbb{B} & \\
\quad \text{or } K_0 = {*}{\times}{*} \text{ and } v_0 \in \langle v, v\rangle & \\
\quad \text{or } K_0 = {*}{\to}{*} \text{ and } v_0 \in w & \\
\quad \text{or } K_0 = * & \\
\mathsf{False} & \\
\quad \text{otherwise} &
\end{cases}
$$

$\boxed{\delta : \mathit{binop}{\times}v{\times}v \longrightarrow e}$

$$
\delta(\mathit{binop}, i_0, i_1) = \begin{cases}
i_0 + i_1 & \\
\quad \text{if } \mathit{binop} = \mathsf{sum}\{\tau\} & \\
\mathsf{DivErr} & \\
\quad \text{if } \mathit{binop} = \mathsf{quotient}\{\tau\} & \\
\quad \text{and } i_1 = 0 & \\
\lfloor i_0 / i_1 \rfloor & \\
\quad \text{if } \mathit{binop} = \mathsf{quotient}\{\tau\} & \\
\quad \text{and } i_1 \neq 0 &
\end{cases}
$$

$\boxed{\sim^{L}_{pos} \colon \tau \times v \longrightarrow \mathbb{B}}$

L	$v \sim^{L}_{bnd} \tau$	$v \sim^{L}_{mon} \tau$	$v \sim^{L}_{check} \tau$
N	$v \sim \lfloor \tau \rfloor$	$v \sim \lfloor \tau \rfloor$	True
T	$v \sim \lfloor \tau \rfloor$	True	$v \sim \lfloor \tau \rfloor$

1.3 Operational Semantics

$\boxed{\longrightarrow_L^*}$ reflexive-transitive closure of $\longrightarrow_L$

$\boxed{\longrightarrow_L}$ compatible closure of $\hookrightarrow_L$

$\boxed{e \rightarrowtail_L e}$

$$
\begin{array}{lll}
\mathsf{fst}\{\tau_0\}\, v_0 & \rightarrowtail_L & \mathsf{Wrong} \\
\quad \text{if } v_0 \neq \langle v_1, v_2 \rangle & & \\[1em]
\mathsf{fst}\{\tau_0\}\, \langle v_0, v_1 \rangle & \rightarrowtail_L & \mathsf{assert}\ \tau_0\ v_0 \\[1em]
\mathsf{snd}\{\tau_0\}\, v_0 & \rightarrowtail_L & \mathsf{Wrong} \\
\quad \text{if } v_0 \neq \langle v_1, v_2 \rangle & & \\[1em]
\mathsf{snd}\{\tau_0\}\, \langle v_0, v_1 \rangle & \rightarrowtail_L & \mathsf{assert}\ \tau_0\ v_1 \\[1em]
\mathit{binop}\, v_0\, v_1 & \rightarrowtail_L & \mathsf{Wrong} \\
\quad \text{if } \delta(\mathit{binop}, v_0, v_1) \text{ is undefined} & & \\[1em]
\mathit{binop}\, v_0\, v_1 & \rightarrowtail_L & \mathsf{assert}\ \tau_0\ \delta(\mathit{binop}, v_0, v_1) \\
\quad \text{if } \delta(\mathit{binop}, v_0, v_1) \text{ is defined} & & \\[1em]
\mathsf{app}\{\tau_0\}\, v_0\ v_1 & \rightarrowtail_L & \mathsf{assert}\ \tau_0\ (v_0\ v_1) \\[1em]
v_0\ v_1 & \rightarrowtail_L & \mathsf{Wrong} \\
\quad \text{if } v_0 \neq w_0 & & \\[1em]
(\lambda(x_0 : \tau_1).\, e_0)\ v_1 & \rightarrowtail_L & e_0[x_0 \leftarrow v_1] \\
\quad \text{if } v_1 \sim^L_{check} \tau_1 & & \\[1em]
(\lambda(x_0 : \tau_1).\, e_0)\ v_1 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_1,\ v_1) \\
\quad \text{if } \neg v_1 \sim^L_{check} \tau_1 & & \\[1em]
(\mathsf{grd}\,\{\tau_1 \Leftarrow \tau_2\}\, w_0)\ v_1 & \rightarrowtail_L & \mathsf{mon}\,\{\mathit{cod}(\tau_1) \Leftarrow \mathit{cod}(\tau_2)\}\,(w_0\ (\mathsf{mon}\,\{\mathit{dom}(\tau_2) \Leftarrow \mathit{dom}(\tau_1)\}\, v_1)) \\[1em]
\mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\, v_0 & \rightarrowtail_L & \mathsf{mon}\,\{\tau_1 \Leftarrow \tau_0\}\, v_0 \\
\quad \text{if } v_0 \sim^L_{bnd} \tau_1 & & \\
\quad \text{and } v_0 \sim^L_{bnd} \tau_0 & &
\end{array}
$$

$$\begin{array}{lll}
\mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\,v_0 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_1,\, v_0) \\
\quad \text{if } \neg v_0 \sim^L_{bnd} \tau_1 & & \\[1em]
\mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\,v_0 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_0,\, v_0) \\
\quad \text{if } \neg v_0 \sim^L_{bnd} \tau_0 & & \\[1em]
\mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,i_0 & \rightarrowtail_L & i_0 \\
\quad \text{if } i_0 \sim^L_{mon} \tau_1 \wedge i_0 \sim^L_{mon} \tau_2 & & \\[1em]
\mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,\langle v_0, v_1\rangle & \rightarrowtail_L & \langle \mathsf{mon}\,\{fst(\tau_1) \Leftarrow fst(\tau_2)\}\,v_0, \mathsf{mon}\,\{snd(\tau_1) \Leftarrow snd(\tau_2)\}\,v_1\rangle \\[1em]
\mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,w & \rightarrowtail_L & \mathsf{grd}\,\{\tau_1 \Leftarrow \tau_2\}\,w \\
\quad \text{if } w \sim^L_{mon} \tau_1 \wedge w \sim^L_{mon} \tau_2 & & \\[1em]
\mathsf{mon}\,\{\tau_0 \Leftarrow \tau_1\}\,v_0 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_0,\, v_0) \\
\quad \text{if } \neg v_0 \sim^L_{mon} \tau_0 & & \\[1em]
\mathsf{mon}\,\{\tau_0 \Leftarrow \tau_1\}\,v_0 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_1,\, v_0) \\
\quad \text{if } \neg v_0 \sim^L_{mon} \tau_1 & & \\[1em]
\mathsf{if}\ \mathsf{True}\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 & \rightarrowtail_L & e_1 \\[1em]
\mathsf{if}\ \mathsf{False}\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 & \rightarrowtail_L & e_2 \\[1em]
\mathsf{assert}\ \tau_0\, v_0 & \rightarrowtail_L & v_0 \\
\quad \text{if } v_0 \sim^L_{check} \tau_0 & & \\[1em]
\mathsf{assert}\ \tau_0\, v_0 & \rightarrowtail_L & \mathsf{TypeErr}(\tau_0,\, v_0) \\
\quad \text{if } \neg v_0 \sim^L_{check} \tau_0 & &
\end{array}$$

1.4 Store-Based Evaluation Language Definitions

$\boxed{\text{Store-Based Evaluation Language}}$

$$
\begin{array}{lcl}
v & ::= & \ell \mid n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \langle \ell, \ell \rangle \mid \lambda(x\!:\!\tau).\, e \\
\mathsf{Err}^{\circ} & ::= & \mathsf{Wrong} \\
\mathsf{Err}^{\bullet} & ::= & \mathsf{DivErr} \mid \mathsf{TypeErr}(\tau,\, v) \\
\mathsf{Err} & ::= & \mathsf{Err}^{\circ} \mid \mathsf{Err}^{\bullet} \\
e & ::= & \mathsf{Err} \mid x \mid \ell \mid v \mid \langle e, e \rangle \mid \mathsf{app}\{\tau\}\, e\; e \mid e\; e \mid \mathsf{fst}\{\tau\}\, e \mid \mathsf{snd}\{\tau\}\, e \mid \mathit{binop}\, e\, e \mid \mathsf{cast}\, \{\tau \Leftarrow \tau'\}\; e \\
 & \mid & \mathsf{if}\; e\; \mathsf{then}\; e\; \mathsf{else}\; e \mid \mathsf{mon}\, \{\tau \Leftarrow \tau\}\; e \mid \mathsf{assert}\, \tau\, e \\
K & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid * {\times} * \mid * {\rightarrow} * \mid * \\
\tau & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid \tau {\times} \tau \mid \tau {\rightarrow} \tau \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
\Sigma & \in & \mathbb{L} \mapsto \mathbb{V} \times \mathsf{option}(\mathbb{T} \times \mathbb{T}) \\
\ell & \in & \mathbb{L} \\
n & \in & \mathbb{N} \\
i & \in & \mathbb{Z} \\
E & ::= & [\,] \mid \langle E, e \rangle \mid \langle \ell, E \rangle \mid \mathsf{fst}\{\tau\}\, E \mid \mathsf{snd}\{\tau\}\, E \mid \mathsf{app}\{\tau\}\, E\; e \mid \mathsf{app}\{\tau\}\, \ell\; E \mid E\; e \mid \ell\; E \mid \mathit{binop}\, E\, e \mid \mathit{binop}\, \ell\, E \\
 & \mid & \mathsf{cast}\, \{\tau \Leftarrow \tau'\}\; E \mid \mathsf{if}\; E\; \mathsf{then}\; e\; \mathsf{else}\; e \mid \mathsf{mon}\, \{\tau \Leftarrow \tau\}\; E \mid \mathsf{assert}\, \tau\, E
\end{array}
$$

$\boxed{\sim : K \times \mathbb{V} \longrightarrow \mathbb{B}}$

$$
v_0 \sim K_0 = \begin{cases}
\mathsf{True} & \\
\quad \text{if } K_0 = \mathsf{Nat} \text{ and } v_0 \in \mathbb{N} & \\
\quad \text{or } K_0 = \mathsf{Int} \text{ and } v_0 \in \mathbb{Z} & \\
\quad \text{or } K_0 = \mathsf{Bool} \text{ and } v_0 \in \mathbb{B} & \\
\quad \text{or } K_0 = * \times * \text{ and } v_0 \in \langle \ell, \ell \rangle & \\
\quad \text{or } K_0 = * \rightarrow * \text{ and } v_0 \in \lambda(x\!:\!\tau).\, e & \\
\quad \text{or } K_0 = * & \\
\mathsf{False} & \\
\quad \text{otherwise} &
\end{cases}
$$

$\boxed{\delta : \mathit{binop} \times \mathbb{V} \times \mathbb{V} \longrightarrow \mathbb{E}}$

$$
\delta(\mathit{binop}, i_0, i_1) = \begin{cases}
i_0 + i_1 & \\
\quad \text{if } \mathit{binop} = \mathsf{sum}\{\tau\} & \\
\mathsf{DivErr} & \\
\quad \text{if } \mathit{binop} = \mathsf{quotient}\{\tau\} & \\
\quad \text{and } i_1 = 0 & \\
\lfloor i_0 / i_1 \rfloor & \\
\quad \text{if } \mathit{binop} = \mathsf{quotient}\{\tau\} & \\
\quad \text{and } i_1 \neq 0 &
\end{cases}
$$

$\boxed{\sim^{L}_{pos} : \mathbb{T} \times \mathbb{V} \longrightarrow \mathbb{B}}$

L	$v \sim^{L}_{bnd} \tau$	$v \sim^{L}_{mon} \tau$	$v \sim^{L}_{check} \tau$
N	$v \sim \lfloor \tau \rfloor$	$v \sim \lfloor \tau \rfloor$	True
T	$v \sim \lfloor \tau \rfloor$	True	$v \sim \lfloor \tau \rfloor$

$\boxed{\mathsf{pointsto}(\Sigma, \ell)}$

$$\mathsf{pointsto}(\Sigma, \ell) = \begin{cases} \mathit{fst}(\Sigma(\ell)) \\ \quad \text{if } \mathit{fst}(\Sigma(\ell)) \neq \ell' \\ \mathsf{pointsto}(\Sigma, \ell') \\ \quad \text{if } \mathit{fst}(\Sigma(\ell)) = \ell' \end{cases}$$

1.5 Store-Based Operational Semantics

$\boxed{\longrightarrow_L^*}$ reflexive-transitive closure of $\longrightarrow_L$

$\boxed{\longrightarrow_L}$ compatible closure of $\hookrightarrow_L$

$\boxed{\Sigma, e \hookrightarrow_L \Sigma, e}$

$\Sigma, v \quad \hookrightarrow_L \quad \Sigma[\ell \mapsto (v, \mathsf{none})], \ell$
where $loc \notin dom(\Sigma)$

$\Sigma, \mathsf{fst}\{\tau_0\}\ \ell_0 \quad \hookrightarrow_L \quad \Sigma, \mathsf{Wrong}$
if $\Sigma(\ell_0) \neq (\langle \ell_1, \ell_2 \rangle, \_)$

$\Sigma, \mathsf{fst}\{\tau_0\}\ \ell_0 \quad \hookrightarrow_L \quad \Sigma, \mathsf{assert}\ \tau_0\ \ell_0$
if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, \_)$

$\Sigma, \mathsf{snd}\{\tau_0\}\ \ell_0 \quad \hookrightarrow_L \quad \Sigma, \mathsf{Wrong}$
if $\Sigma(\ell_0) \neq (\langle \ell_1, \ell_2 \rangle, \_)$

$\Sigma, \mathsf{snd}\{\tau_0\}\ \ell_0 \quad \hookrightarrow_L \quad \Sigma, \mathsf{assert}\ \tau_0\ \ell_0$
if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, \_)$

$\Sigma, binop\ \ell_0\ \ell_1 \quad \hookrightarrow_L \quad \Sigma, \mathsf{Wrong}$
if $\delta(binop, \mathsf{pointsto}(\Sigma, \ell_0), \mathsf{pointsto}(\Sigma, \ell_1))$ is undefined

$\Sigma, binop\ \ell_0\ \ell_1 \quad \hookrightarrow_L \quad \Sigma, \mathsf{assert}\ \tau_0\ \delta(binop, \mathsf{pointsto}(\Sigma, \ell_0), \mathsf{pointsto}(\Sigma, \ell_1))$
if $\delta(binop, \mathsf{pointsto}(\Sigma, \ell_0), \mathsf{pointsto}(\Sigma, \ell_1))$ is defined

$\Sigma, \mathsf{app}\{\tau_0\}\ \ell_0\ \ell_1 \quad \hookrightarrow_L \quad \Sigma, \mathsf{assert}\ \tau_0\ (\ell_0\ \ell_1)$

$\Sigma, \ell_0\ \ell_1 \quad \hookrightarrow_L \quad \Sigma, \mathsf{Wrong}$
if $\Sigma(\ell_0) = (v, \_)$ and $v \notin \lambda(x\!:\!\tau).\, e \cup \ell$
or $\Sigma(\ell_0) = (\ell_0', \mathsf{none})$

$\Sigma, \ell_0\ \ell_1 \quad \hookrightarrow_L \quad \Sigma, e_0[x_0 \leftarrow \ell_1]$
if $\Sigma(\ell_0) = (\lambda(x_0\!:\!\tau_1).\, e_0, \_)$ and
$\mathsf{pointsto}(\Sigma, \ell_1) \sim^L_{check} \tau_1$

$\Sigma, \ell_0\ \ell_1 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_1,\ \ell_1)$
  if $\Sigma(\ell_0) = (\lambda(x_0 : \tau_1).\ e_0, \_)$ and
  $\neg\mathsf{pointsto}(\Sigma, \ell_1) \sim^L_{check} \tau_1$

$\Sigma, \ell_0\ \ell_1 \hookrightarrow_L \Sigma, \mathsf{mon}\,\{cod(\tau_1) \Leftarrow cod(\tau_2)\}\,(\ell_0\ (\mathsf{mon}\,\{dom(\tau_2) \Leftarrow dom(\tau_1)\}\,\ell_1))$
  if $\Sigma(\ell_0) = (\ell_2, \mathsf{some}(\tau_1, \tau_2))$

$\Sigma, \mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\,\ell_0 \hookrightarrow_L \Sigma, \mathsf{mon}\,\{\tau_1 \Leftarrow \tau_0\}\,\ell_0$
  if $\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{bnd} \tau_1$
  and $\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{bnd} \tau_0$

$\Sigma, \mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\,\ell_0 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_1,\ \ell_0)$
  if $\neg\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{bnd} \tau_1$

$\Sigma, \mathsf{cast}\,\{\tau_1 \Leftarrow \tau_0\}\,\ell_0 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_0,\ \ell_0)$
  if $\neg\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{bnd} \tau_0$

$\Sigma, \mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,\ell_0 \hookrightarrow_L \Sigma[\ell_1 \mapsto (\ell_0, \mathsf{some}(\tau_1, \tau_2))], \ell_1$
  if $\ell_1 \notin dom(\Sigma)$
  and $\mathsf{pointsto}(\Sigma, \ell_0) = v$ where $v = i$ or True or False
  and $v \sim^L_{mon} \tau_1 \wedge v \sim^L_{mon} \tau_2$

$\Sigma, \mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,\ell_0 \hookrightarrow_L \Sigma, \langle \mathsf{mon}\,\{fst(\tau_1) \Leftarrow fst(\tau_2)\}\,\ell_1, \mathsf{mon}\,\{snd(\tau_1) \Leftarrow snd(\tau_2)\}\,\ell_2 \rangle$
  if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, \_)$

$\Sigma, \mathsf{mon}\,\{\tau_1 \Leftarrow \tau_2\}\,\ell_0 \hookrightarrow_L \Sigma[\ell_1 \mapsto (\ell_0, \mathsf{some}(\tau_1, \tau_2))], \ell_1$
  if $\ell_1 \notin dom(\Sigma)$
  and $\mathsf{pointsto}(\Sigma, \ell_0) = v$ and $v = \lambda(x_0 : \tau_1).\ e_0$
  and $v \sim^L_{mon} \tau_1 \wedge v \sim^L_{mon} \tau_2$

$\Sigma, \mathsf{mon}\,\{\tau_0 \Leftarrow \tau_1\}\,\ell_0 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_1,\ \ell_0)$
  if $\neg\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{mon} \tau_1$

$\Sigma, \mathsf{mon}\,\{\tau_0 \Leftarrow \tau_1\}\,\ell_0 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_0,\ \ell_0)$
  if $\neg\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{mon} \tau_0$

$\Sigma, \mathsf{if}\ \ell_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \hookrightarrow_L \Sigma, e_1$
  if $\mathsf{pointsto}(\Sigma, \ell_0) = \mathsf{True}$

$\Sigma, \mathsf{if}\ \ell_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \hookrightarrow_L \Sigma, e_2$
  if $\mathsf{pointsto}(\Sigma, \ell_0) = \mathsf{False}$

$$\Sigma, \mathsf{if}\ \ell_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \hookrightarrow_L \Sigma, \mathsf{Wrong}$$
$$\quad \mathsf{if}\ \mathsf{pointsto}(\Sigma, \ell_0) \neq \ell\ \mathsf{or}\ \mathsf{True}\ \mathsf{or}\ \mathsf{False}$$

$$\Sigma, \mathsf{assert}\ \tau_0\ \ell_0 \hookrightarrow_L \Sigma, \ell_0$$
$$\quad \mathsf{if}\ \mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{check} \tau_0$$

$$\Sigma, \mathsf{assert}\ \tau_0\ \ell_0 \hookrightarrow_L \Sigma, \mathsf{TypeErr}(\tau_0,\ \ell_0)$$
$$\quad \mathsf{if}\ \neg\mathsf{pointsto}(\Sigma, \ell_0) \sim^L_{check} \tau_0$$

1.6 Operational Semantics Simulation Result

To compare the two semantics, we have to define a relation that compares values between the two languages. The store semantics will represent:

(1) Guards as a linked list of pairs of types, ending at a lambda with no types.
(2) Pairs as a pointer to the two subcomponents, with no types.
(3) Base values as a linked list of pairs of types, ending at a base value with no types.

We capture this in the following value equivalence:

$\boxed{(\Sigma, \ell) \equiv v}$

$$\frac{\mathsf{pointsto}(\Sigma, \ell) = v}{\ell \equiv v} \qquad \frac{\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_) \quad (\Sigma, \ell_1) \equiv v_1 \quad (\Sigma, \ell_2) \equiv v_2}{(\Sigma, \ell) \equiv \langle v_1, v_2 \rangle} \qquad \frac{\Sigma(\ell) = (\ell', \mathsf{some}(\tau', \tau)) \quad (\Sigma, \ell') \equiv v}{(\Sigma, \ell) \equiv \mathsf{grd}\,\{\tau' \Leftarrow \tau\}\ v} \qquad \frac{\Sigma(\ell) = (\lambda x : \tau.\, e, \_)}{(\Sigma, \ell) \equiv \lambda x : \tau.\, e}$$

Theorem 1.1 (Store and Non Store Operational Semantics Are Equivalent). *$e \longrightarrow_L^* e'$ and e' is irreducible iff $\forall \Sigma.\ \exists \Sigma', \ell.\ (\Sigma, e) \longrightarrow_L^* (\Sigma', \ell)$ and $(\Sigma', \ell) \equiv e'$*

2 Tag Typing

2.1 Definition

$\boxed{\text{Simple language}}$

$$
\begin{array}{lcl}
e & ::= & x \mid n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \lambda(x\!:\!K).\,e \mid \langle e, e\rangle \mid \mathsf{app}\{K\}\;e\;e \mid \mathsf{fst}\{K\}\;e \mid \mathsf{snd}\{K\}\;e \mid \mathit{binop}\,e\,e \mid \mathsf{cast}\;\{K \Leftarrow K\}\;e \mid \mathsf{if}\;e\;\mathsf{then}\;e\;\mathsf{else}\;e \\
K & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid {*}{\times}{*} \mid {*}{\to}{*} \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
\Gamma & ::= & \cdot \mid \Gamma, (x\!:\!K_0) \\
n & ::= & \mathbb{N} \\
i & ::= & \mathbb{Z}
\end{array}
$$

$\boxed{\Gamma \vdash_{\mathsf{tag}} e : \tau}$ typing

$$\textsc{T-Var}\quad \frac{(x_0\!:\!K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\mathsf{tag}} x_0 : K_0} \qquad \textsc{T-Nat}\quad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} n_0 : \mathsf{Nat}} \qquad \textsc{T-Int}\quad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} i_0 : \mathsf{Int}} \qquad \textsc{T-True}\quad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{True} : \mathsf{Bool}} \qquad \textsc{T-False}\quad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{False} : \mathsf{Bool}}$$

$$\textsc{T-Lam}\quad \frac{\Gamma_0,\ (x_0\!:\!K_0) \vdash_{\mathsf{tag}} e_0 : K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \lambda(x_0\!:\!K_0).\,e_0 : {*}{\to}{*}} \qquad \textsc{T-Pair}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \langle e_0, e_1\rangle : {*}{\times}{*}} \qquad \textsc{T-Cast}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad K_0 \sim K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{cast}\;\{K_1 \Leftarrow K_0\}\;e_0 : K_1}$$

$$\textsc{T-App}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : {*}{\to}{*} \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_0}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{app}\{K_1\}\;e_0\;e_1 : K_1} \qquad \textsc{T-Fst}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : {*}{\times}{*}}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{fst}\{K_0\}\;e_0 : K_0} \qquad \textsc{T-Snd}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : {*}{\times}{*}}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{snd}\{K_1\}\;e_0 : K_1} \qquad \textsc{T-Binop}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_1 \quad \Delta(\mathit{binop}, K_0, K_1) = K_2}{\Gamma_0 \vdash_{\mathsf{tag}} \mathit{binop}\,e_0\,e_1 : K_2}$$

$$\textsc{T-If}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : \mathsf{Bool} \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_2 : K_0}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{if}\;e_0\;\mathsf{then}\;e_1\;\mathsf{else}\;e_2 : K_0} \qquad \textsc{T-Sub}\quad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad K_0 \leqslant: K_1}{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_1}$$

2.2 Simple Typing Implies Tag Typing

$\boxed{e^+}$

$$
\begin{aligned}
i^+ &= i \\
b^+ &= b \\
\langle e_1, e_2\rangle^+ &= \langle e_1^+, e_2^+\rangle \\
(\lambda x : \tau.\, e)^+ &= \lambda x : \lfloor\tau\rfloor.\, e^+ \\
(\mathsf{app}\{\tau\}\, e_1\ e_2)^+ &= \mathsf{app}\{\lfloor\tau\rfloor\}\, e_1^+\ e_2^+ \\
(\mathsf{fst}\{\tau\}\, e)^+ &= \mathsf{fst}\{\lfloor\tau\rfloor\}\, e^+ \\
(\mathsf{snd}\{\tau\}\, e)^+ &= \mathsf{snd}\{\lfloor\tau\rfloor\}\, e^+ \\
(\mathit{binop}\, e_1\ e_2)^+ &= \mathit{binop}\, e_1^+\ e_2^+ \\
(\mathsf{cast}\ \{\tau' \Leftarrow \tau\}\ e)^+ &= \mathsf{cast}\ \{\lfloor\tau'\rfloor \Leftarrow \lfloor\tau\rfloor\}\ e^+ \\
(\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3)^+ &= \mathsf{if}\ e_1^+\ \mathsf{then}\ e_2^+\ \mathsf{else}\ e_3^+
\end{aligned}
$$

$\boxed{\Gamma^+}$

$$
\begin{aligned}
(\Gamma, x : \tau)^+ &= \Gamma^+, x : \lfloor\tau\rfloor \\
\cdot^+ &= \cdot
\end{aligned}
$$

Theorem 2.1 (Simple Typing Implies Tag Typing). *If* $\Gamma \vdash_{\mathsf{sim}} e : \tau$ *then* $\Gamma^+ \vdash_{\mathsf{tag}} e^+ : \lfloor\tau\rfloor$.

Proof. By induction over the typing derivation. The typing rules have a one to one correspondance, so each case follows by the induction hypothesis. □

3 Truer Transient Typing

3.1 Definition

$\boxed{\text{Simple language}}$

$$
\begin{array}{lcl}
e & ::= & x \mid n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \lambda(x\!:\!K).\,e \mid \langle e, e\rangle \mid \mathsf{app}\{K\}\,e\,e \mid \mathsf{fst}\{K\}\,e \mid \mathsf{snd}\{K\}\,e \mid \mathit{binop}\,e\,e \mid \mathsf{cast}\,\{K \Leftarrow K\}\,e \mid \mathsf{if}\ e\ \mathsf{then}\ e\ \mathsf{else}\ e \\
\tau & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid \tau\times\tau \mid * \rightarrow \tau \mid * \mid \bot \\
K & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid *\times* \mid * \rightarrow * \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
\Gamma & ::= & \cdot \mid \Gamma, (x\!:\!K_0) \\
n & ::= & \mathbb{N} \\
i & ::= & \mathbb{Z}
\end{array}
$$

$\boxed{\lfloor\tau\rfloor}$ tag of

$$
\begin{array}{lcl}
\lfloor \mathsf{Int} \rfloor & = & \mathsf{Int} \\
\lfloor \mathsf{Nat} \rfloor & = & \mathsf{Nat} \\
\lfloor \mathsf{Bool} \rfloor & = & \mathsf{Bool} \\
\lfloor \tau \times \tau' \rfloor & = & * \times * \\
\lfloor * \rightarrow \tau' \rfloor & = & * \rightarrow * \\
\lfloor * \rfloor & = & *
\end{array}
$$

$\boxed{\sqcup, \sqcap : \tau \times \tau \longrightarrow \tau}$

$$
\tau \sqcup \tau' = \begin{cases}
* & \text{if } \tau = * \text{ or } \tau' = * \text{ or } \lfloor\tau\rfloor \neq \lfloor\tau'\rfloor \text{ and } \tau \neq \bot \text{ and } \tau' \neq \bot \\
\tau & \text{if } \tau' = \bot \\
\tau' & \text{if } \tau = \bot \\
\mathsf{Int} & \text{if } \tau = \mathsf{Nat} \text{ and } \tau' = \mathsf{Int} \text{ or } \tau = \mathsf{Int} \text{ and } \tau' = \mathsf{Nat} \\
\tau & \text{if } \tau = \tau' \\
\tau_1 \sqcup \tau_1' \times \tau_2 \sqcup \tau_2' & \text{if } \tau = \tau_1 \times \tau_2 \text{ and } \tau' = \tau_1' \times \tau_2' \\
* \rightarrow (\tau_2 \sqcup \tau_2') & \text{if } \tau = * \rightarrow \tau_2 \text{ and } \tau' = * \rightarrow \tau_2'
\end{cases}
\qquad
\tau \sqcap \tau' = \begin{cases}
\bot & \text{if } \tau = \bot \text{ or } \tau' = \bot \text{ or } \lfloor\tau\rfloor \neq \lfloor\tau'\rfloor \text{ and } \tau \neq * \text{ and } \tau' \neq * \\
\tau & \text{if } \tau' = * \\
\tau' & \text{if } \tau = * \\
\mathsf{Nat} & \text{if } \tau = \mathsf{Nat} \text{ and } \tau' = \mathsf{Int} \text{ or } \tau = \mathsf{Int} \text{ and } \tau' = \mathsf{Nat} \\
\tau & \text{if } \tau = \tau' \\
\tau_1 \sqcap \tau_1' \times \tau_2 \sqcap \tau_2' & \text{if } \tau = \tau_1 \times \tau_2 \text{ and } \tau' = \tau_1' \times \tau_2' \\
* \rightarrow (\tau_2 \sqcap \tau_2') & \text{if } \tau = * \rightarrow \tau_2 \text{ and } \tau' = * \rightarrow \tau_2'
\end{cases}
$$

$\boxed{\Gamma \vdash_{\text{tru}} e : \tau}$ typing

$$\text{T-Var}\ \frac{(x_0{:}K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\text{tru}} x_0 : K_0} \qquad \text{T-Nat}\ \frac{}{\Gamma_0 \vdash_{\text{tru}} n_0 : \mathsf{Nat}} \qquad \text{T-Int}\ \frac{}{\Gamma_0 \vdash_{\text{tru}} i_0 : \mathsf{Int}} \qquad \text{T-True}\ \frac{}{\Gamma_0 \vdash_{\text{tru}} \mathsf{True} : \mathsf{Bool}} \qquad \text{T-False}\ \frac{}{\Gamma_0 \vdash_{\text{tru}} \mathsf{False} : \mathsf{Bool}}$$

$$\text{T-Lam}\ \frac{\Gamma_0,\ (x_0{:}K_0) \vdash_{\text{tru}} e_0 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \lambda(x_0{:}K_0).\, e_0 : * \!\to\! \tau_1} \qquad \text{T-Pair}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \langle e_0, e_1 \rangle : \tau_0 \!\times\! \tau_1} \qquad \text{T-Cast}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0}{\Gamma_0 \vdash_{\text{tru}} \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\ e_0 : K_1 \sqcap K_0 \sqcap \tau_0}$$

$$\text{T-App}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : * \!\to\! \tau_1 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0'}{\Gamma_0 \vdash_{\text{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 : K_1 \sqcap \tau_1} \qquad \text{T-AppBot}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \bot \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0'}{\Gamma_0 \vdash_{\text{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 : \bot} \qquad \text{T-Fst}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \!\times\! \tau_1}{\Gamma_0 \vdash_{\text{tru}} \mathsf{fst}\{K_0\}\ e_0 : K_0 \sqcap \tau_0} \qquad \text{T-FstBot}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \bot}{\Gamma_0 \vdash_{\text{tru}} \mathsf{fst}\{K_0\}\ e_0 : \bot}$$

$$\text{T-Snd}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \!\times\! \tau_1}{\Gamma_0 \vdash_{\text{tru}} \mathsf{snd}\{K_1\}\ e_0 : K_1 \sqcap \tau_1} \qquad \text{T-SndBot}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \bot}{\Gamma_0 \vdash_{\text{tru}} \mathsf{snd}\{K_1\}\ e_0 : \bot} \qquad \text{T-Binop}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \mathit{binop}\, e_0\, e_1 : \Delta(\mathit{binop}, \tau_0, \tau_1)}$$

$$\text{T-If}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \mathsf{Bool} \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_2 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_0 \sqcup \tau_1} \qquad \text{T-IfBot}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \bot \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_2 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \bot} \qquad \text{T-Sub}\ \frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \tau_0 \leqslant: \tau_1}{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_1}$$

$\boxed{\Delta : \mathit{binop} \times \tau \times \tau \longrightarrow \tau}$

$$\begin{aligned}
&\Delta(\mathsf{sum}, \mathsf{Nat}, \mathsf{Nat}) &&= \mathsf{Nat}\\
&\Delta(\mathsf{sum}, \mathsf{Int}, \mathsf{Int}) &&= \mathsf{Int}\\
&\Delta(\mathsf{quotient}, \mathsf{Nat}, \mathsf{Nat}) &&= \mathsf{Nat}\\
&\Delta(\mathsf{quotient}, \mathsf{Int}, \mathsf{Int}) &&= \mathsf{Int}\\
&\Delta(\mathit{binop}, \bot, \tau) &&= \bot \text{ if } \tau = \mathsf{Nat} \text{ or } \mathsf{Int} \text{ or } \bot\\
&\Delta(\mathit{binop}, \tau, \bot) &&= \bot \text{ if } \tau = \mathsf{Nat} \text{ or } \mathsf{Int} \text{ or } \bot
\end{aligned}$$

$\boxed{\tau \leq \tau}$

$$\frac{\tau_0 \leqslant: \tau_1}{\tau_0 \leq \tau_1} \qquad \frac{\tau_0 \leq \tau_2 \quad \tau_1 \leq \tau_3}{\tau_0 \!\times\! \tau_1 \leq \tau_2 \!\times\! \tau_3} \qquad \frac{\tau_0 \leq \tau_1}{* \!\to\! \tau_0 \leq * \!\to\! \tau_1} \qquad \frac{}{\bot \leq \tau} \qquad \frac{}{\tau \leq *}$$

3.2 Simple Typing Implies Truer Transient Typing

$\boxed{e^+}$

$$
\begin{aligned}
i^+ &= i \\
b^+ &= b \\
\langle e_1, e_2\rangle^+ &= \langle e_1^+, e_2^+\rangle \\
(\lambda x : \tau.\, e)^+ &= \lambda x : \lfloor \tau \rfloor.\, e^+ \\
(\mathsf{app}\{\tau\}\, e_1\, e_2)^+ &= \mathsf{app}\{\lfloor \tau \rfloor\}\, e_1^+\, e_2^+ \\
(\mathsf{fst}\{\tau\}\, e)^+ &= \mathsf{fst}\{\lfloor \tau \rfloor\}\, e^+ \\
(\mathsf{snd}\{\tau\}\, e)^+ &= \mathsf{snd}\{\lfloor \tau \rfloor\}\, e^+ \\
(\mathit{binop}\, e_1\, e_2)^+ &= \mathit{binop}\, e_1^+\, e_2^+ \\
(\mathsf{cast}\,\{\tau' \Leftarrow \tau\}\, e)^+ &= \mathsf{cast}\,\{\lfloor \tau' \rfloor \Leftarrow \lfloor \tau \rfloor\}\, e^+ \\
(\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3)^+ &= \mathsf{if}\ e_1^+\ \mathsf{then}\ e_2^+\ \mathsf{else}\ e_3^+
\end{aligned}
$$

$\boxed{\Gamma^+}$

$$
\begin{aligned}
(\Gamma, x : \tau)^+ &= \Gamma^+, x : \lfloor \tau \rfloor \\
\cdot^+ &= \cdot
\end{aligned}
$$

The following proofs will use the fact honest transient types with $\sqcup$ and $\sqcap$ form a lattice ordered by $\leq$.

Lemma 3.1 (Lattice join idempotent). $\tau \sqcup \tau = \tau$

Proof. By induction on the structure of τ, in each case following immediately from the definition of $\sqcup$. □

Lemma 3.2 (Lattice join absorption). $\tau_0 \sqcup (\tau_0 \sqcap \tau_1) = \tau_0$

Proof. By induction on the structure of τ_0; in each case by induction on the structure of τ_1, in each case following immediately from the definitions of $\sqcup$ and $\sqcap$ and the prior lemma. □

Lemma 3.3 (Lattice meet idempotent). $\tau \sqcap \tau = \tau$

Proof. By induction on the structure of τ, in each case following immediately from the definition of $\sqcap$. □

Lemma 3.4 (Lattice meet absorption). $\tau_0 \sqcap (\tau_0 \sqcup \tau_1) = \tau_0$

Proof. By induction on the structure of τ_0; in each case by induction on the structure of τ_1, in each case following immediately from the definitions of $\sqcup$ and $\sqcap$ and the prior lemma. □

Lemma 3.5 (Lattice ordering implies $\leq$). *If $\tau = \tau \sqcap \tau'$, then $\tau \leq \tau'$.*

Proof. We proceed by induction on the structure of the definition of $\tau \sqcap \tau'$:

$\bot$ Since $\tau = \tau \sqcap \tau$, $\tau = \bot$; it is immediate that $\tau_0 \leq \tau_1$.

τ This case occurs if $\tau' = *$; consequently it is immediate that $\tau \leq \tau'$.

τ' In this case, the hypothesis ensures that $\tau = \tau'$, so $\tau \leq \tau'$ by reflexivity.

Nat In this case, τ must be Nat and τ' must be Int. By definition, Nat $\leq$ Int.

τ In this case, $\tau = \tau'$; it is immediate that $\tau \leq \tau'$.

$\tau_1 \sqcap \tau_1' \times \tau_2 \sqcap \tau_2'$ In this case, by the hypothesis, $\tau_1 = \tau_1 \sqcap \tau_1'$ and $\tau_2 = \tau_2 \sqcap \tau_2'$, so by induction $\tau_1 \leq \tau_1'$ and $\tau_2 \leq \tau_2'$. Then it is immediate from the definition of the lattice ordering that $\tau_1 \times \tau_2 \leq \tau_1' \times \tau_2'$.

$* \to \tau_2 \sqcap \tau_2'$ In this case, $\tau_2 = \tau_2 \sqcap \tau_2'$ by the hypothesis, so $\tau_2 \leq \tau_2'$ by induction; hence it is immediate from the definition of the lattice ordering that $* \to tau_2 \leq * \to \tau_2'$.

□

Lemma 3.6 (Lattice ordering is implied by $\leq$). *If $\tau \leq \tau'$, then $\tau = (\tau \sqcap \tau')$.*

Proof. We proceed by induction on the structure of the definition of $\leq$, with the cases of $\leqslant:$ inlined:

Nat $\leqslant:$ Int This is immediate by the definition of $\sqcap$.

$\tau_0 \times \tau_1 \leqslant: \tau_2 \times \tau_3$ This is subsumed by the case $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ below.

$\tau_0 \to \tau_1 \leqslant: \tau_2 \to \tau_3$ Because we are considering the lattice of honest transient types, $\tau_0 = \tau_2 = *$, and this is subsumed by the case $* \to \tau_1 \leq * \to \tau_3$ below.

$\tau_0 \leq \tau_0$ This is immediate by the definition of $\sqcap$.

$\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ This rule requires that $\tau_0 \leq \tau_2$ and $\tau_1 \leq \tau_3$; hence, by induction $\tau_0 = \tau_0 \sqcap \tau_2$ and $\tau_1 = \tau_1 \sqcap \tau_3$. This is then immediate by the definition of $\sqcap$.

$* \to \tau_1 \leq * \to \tau_3$ This rule requires that $\tau_0 \leq \tau_1$, and so by induction $\tau_0 = \tau_0 \sqcap \tau_1$; this is then immediate by the definition of $\sqcap$.

$\bot \leq \tau$ This is immediate by the definition of $\sqcap$.

$\tau \leq *$ This is immediate by the definition of $\sqcap$.

□

Theorem 3.7 (Simple Typing Implies Truer Transient Typing).
If $\Gamma \vdash_{\mathsf{sim}} e : \tau$ then $\Gamma^+ \vdash_{\mathsf{tru}} e^+ : \tau'$ where $\tau' \leq \lfloor \tau \rfloor$.

Proof. Proceed by induction on the simple typing derivation:

T-Var By the definition of lowering, if $x : \tau \in \Gamma$, then $x : \lfloor \tau \rfloor \in \Gamma^+$, so T-Var applies and $\lfloor \tau \rfloor$ is precisely the τ' such that $\Gamma^+ \vdash e^+ : \tau'$ and $\tau' \leq \lfloor \tau \rfloor$.

T-Nat, T-Int, T-True, T-False For each base type literal, a corresponding rule exists in the honest transient type system, which ascribes the same time (which is also equal to, and hence below in the lattice, the original simple type).

T-Lam Consider arbitrary $\Gamma_0, x_0, \tau_0, e_0, \tau_1$, such that $\Gamma_0 \vdash \lambda(x_0 : \tau_0).\, e_0 : \tau_0 \to \tau_1$. Then by induction we know that $(\Gamma_0, (x_0) : \tau_0)^+ \vdash e_0^+ : \tau_1'$, for some $\tau_1' \leq \lfloor \tau_1 \rfloor$. Note that $(\Gamma_0, (x_0 : \tau_0))^+ = \Gamma_0^+, x_0 : \lfloor \tau_0 \rfloor$ by definition, and similarly that $(\lambda x_0 : \tau_0.\, e_0)^+ = \lambda(x_0 : \lfloor \tau_0 \rfloor).\, e_0^+$ by definition. Then T-Lam applies s.t. $\Gamma_0^+ \vdash \lambda(x_0 : K_0).\, e_0^+ : * \to \tau_1'$. Note that $\lfloor \tau_0 \to \tau_1 \rfloor = * \to * \leq * \to *$ by the definition of lattice ordering, completing the proof.

T-Pair Consider arbitrary $\Gamma_0, e_0, e_1, \tau_0, \tau_1$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Pair if $e = \langle e_0, e_1 \rangle$ and $\tau = \tau_0 \times \tau_1$. Then by induction, there exist some τ_0' and τ_1', s.t. $\Gamma_0^+ \vdash e_0^+ : \tau_0'$, $\Gamma_0^+ : e_1^+ : \tau_1'$, $\tau_0' \leq \lfloor \tau_0 \rfloor$, and $\tau_1' \leq \lfloor \tau_1 \rfloor$. Then instantiate

$\tau' = \tau_0 \times \tau_1$; it is clear that the honest transient typing rule T-Pair applies, since $(\langle e_0, e_1 \rangle)^+ = \langle e_0^+, e_1^+ \rangle$, and it is immediate by the definition of $\leq$ that $\tau' \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$.

T-Cast Consider arbitrary Γ_0, e_0, τ_0, τ_1, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Cast if $e = \mathsf{cast}\,\{\tau_0 \Leftarrow \tau_1\}\, e_0$ and $\tau = \tau_1$. Then by induction, $\Gamma_0^+ \vdash e_0^+ : \tau_0'$ for some τ_0' s.t. $\tau_0' \leq \lfloor \tau_0 \rfloor$. Instantiate τ' by $\lfloor \tau_1 \rfloor \sqcap \lfloor \tau_0 \rfloor \sqcap \tau_0'$; then it is clear that the honest transient typing rule T-Cast applies, since by definition $e^+ = \mathsf{cast}\,\{\lfloor \tau_0 \rfloor \Leftarrow \lfloor \tau_1 \rfloor\}\, e_0^+$. It remains to be shown that $\lfloor \tau_1 \rfloor \sqcap \lfloor \tau_0 \rfloor \sqcap \tau_0' \leq \lfloor \tau_1 \rfloor$; this follows immediately from the properties of the lattice meet operation.

T-App Consider arbitrary Γ_0, e_0, τ_0, τ_1 s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-App if $e = \mathsf{app}\{\tau_1\}\, e_0\, e_1$ and $\tau = \tau_1$. Then by induction, $\Gamma_0^+ \vdash e_0^+ : \tau_l$ for some $\tau_l \leq \lfloor \tau_0 \to \tau_1 \rfloor = * \to *$, and $\Gamma_0^+ \vdash e_1^+ : \tau_0'$ for some $\tau_0' \leq \lfloor \tau_0 \rfloor$. By inspection of $\leq$, note that τ_l must be either $\bot$ or $* \to \tau_l'$ for some τ_l'. Note that $e^+ = \mathsf{app}\{\lfloor \tau_1 \rfloor\}\, e_0^+\, e_1^+$, and so in the former case T-AppBot syntactically applies and in the latter T-App; consider each case:

$\tau_l = \bot$**:** Instantiate $\tau' = \bot$; then it is clear that $\Gamma_0' \vdash e' : \tau'$ by T-AppBot. Then $\bot \leq \lfloor \tau_1 \rfloor$ is immediate by the definition of lattice ordering.

$\tau_l = * \to \tau_l'$**:** Instantiate $\tau' = \lfloor \tau_1 \rfloor \sqcap \tau_l'$; then it is clear that $\Gamma_0' \vdash e' : \tau'$ by T-App, so what remains to be shown is that $\lfloor \tau_1 \rfloor \sqcap \tau_l \leq \lfloor \tau_1 \rfloor$; this is immediate by the definition of meet on a lattice.

T-Fst Consider arbitrary Γ_0, e_0, τ_0, τ_1, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Fst with premise $\Gamma_0 \vdash e_0 : \tau_0 \times \tau_1$ if $e = \mathsf{fst}\{\tau_0\}\, e_0$ and $\tau = \tau_0$. Then, by induction, $\Gamma_0' \vdash e : \tau_p'$ s.t. $\tau_p' \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$. By inspection on $\leq$, note that τ_p' must be either $\bot$ or ${\tau_p}'_0 \times {\tau_p}'_1$ for some τ_{p_0} and τ_{p_1}. Since $e^+ = \mathsf{fst}\{\lfloor \tau_0 \rfloor\}\, e_0^+$, the rule T-FstBot applies in the former case, and similarly T-Fst applies in the latter. Consider each of these cases:

$\tau_p' = \bot$**:** Instantiate $\tau' = \bot$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-FstBot, and $\bot \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of lattice ordering.

$\tau_p' = {\tau_p}'_0 \times {\tau_p}'_1$**:** Instantiate τ' with $\lfloor \tau_0 \rfloor \sqcap {\tau_p}'_0$. Then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-Fst, and $\tau' \leq \lfloor \tau_0 \rfloor$ by the the definition of meet on a lattice.

T-Snd Consider arbitrary Γ_0, e_0, τ_0, τ_1, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Snd with premise $\Gamma_0 \vdash e_0 : \tau_0 \times \tau_1$ if $e = \mathsf{snd}\{\tau_1\}\, e_0$ and $\tau = \tau_1$. Then, by induction, $\Gamma_0' \vdash e : \tau_p'$ s.t. $\tau_p' \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$. By inspection on $\leq$, note that τ_p' must be either $\bot$ or ${\tau_p}'_0 \times {\tau_p}'_1$ for some τ_{p_0} and τ_{p_1}. Since $e^+ = \mathsf{snd}\{\lfloor \tau_1 \rfloor\}\, e_0^+$, the rule T-SndBot applies in the former case, and similarly T-Snd applies in the latter. Consider each of these cases:

$\tau_p' = \bot$**:** Instantiate $\tau' = \bot$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-SndBot, and $\bot \leq \lfloor \tau_1 \rfloor$ follows immediately from the definition of lattice ordering.

$\tau_p' = {\tau_p}'_0 \times {\tau_p}'_1$**:** Instantiate τ' with $\lfloor \tau_1 \rfloor \sqcap {\tau_p}'_1$. Then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-Snd, and $\tau' \leq \lfloor \tau_1 \rfloor$ by the the definition of meet on a lattice.

T-Binop Consider arbitrary Γ_0, *binop*, e_0, e_1, τ_0, τ_1, and τ_2, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Binop with premise $\Delta(\mathit{binop}, \tau_0, \tau_1) = \tau_2$ if $e = \mathit{binop}\, e_0\, e_1$ and $\tau = \tau_2$. By induction, note that $\Gamma_0^+ \vdash e_0^+ : \tau_0'$ for some $\tau_0' \leq \lfloor \tau_0 \rfloor$, and $\Gamma_0^+ \vdash e_1^+ : \tau_1'$ for some $\tau_1' \leq \lfloor \tau_1 \rfloor$. Note that for the simple typing $\Delta(\mathit{binop}, \tau_0, \tau_1)$ to be defined, τ_0 and τ_1 must each be either Nat or Int; consequently, by inspection of the lattice order, τ_0' and τ_1' must each be Nat, Int, or $\bot$. Then by inspection, in any such case, $\Delta(\mathit{binop}, \tau_0', \tau_1')$ is defined and $\leq \Delta(\mathit{binop}, \tau_0, \tau_1) = \tau_2$. Then instantiate τ' with $\lfloor \tau_2 \rfloor \sqcap \Delta(\mathit{binop}, \tau_0', \tau_1')$; since $e^+ = \mathit{binop}\, e_0\, e_1$, the rule S-Binop applies, and by the definition of meet on a lattice, $\lfloor \tau_2 \rfloor \leq \tau'$.

T-If Consider arbitrary Γ_0, e_0, e_1, e_2, τ_0, s.t. $\Gamma_0 \vdash \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 : \tau_0$ by the T-If simple typing rule. Let $e = \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3$ and $\tau = \tau_0$. Then by induction, there exist some $\tau_b' \leq \lfloor \mathsf{Bool} \rfloor = \mathsf{Bool}$, $\tau_0' \leq \lfloor \tau_0 \rfloor$, and $\tau_1' \leq \lfloor \tau_0 \rfloor$, s.t. $\Gamma_0^+ \vdash e_0^+ : \tau_b'$, $\Gamma_0^+ \vdash e_1^+ : \tau_0'$, and $\Gamma_0^+ :\vdash e_2^+ : \tau_1'$. Notice that τ_b' may be only $\bot$ or Bool, by the definition of lattice ordering. Since $e^+ = \text{if } e_0^+ \text{ then } e_1^+ \text{ else } e_2^+$, in the former case the rule T-IfBot applies; in the latter the rule T-If applies. Consider each of these cases:

$\tau_b' = \bot$: By T-IfBot, $\Gamma_0^+ \vdash e^+ : \bot$, so instantiate $\tau' = \bot$. Notice then that $\bot \leq \lfloor\tau\rfloor$ by lattice ordering, so the proof is completed.

$\tau_b' = \mathsf{Bool}$: By T-If, $\Gamma_0^+ \vdash e^+ : \tau_0' \sqcup \tau_1'$. Instantiate τ' by $\tau_0' \sqcup \tau_1'$; then we must show that $\tau' \leq \lfloor\tau\rfloor$. Since $\tau_0' \leq \lfloor\tau_0\rfloor$ and $\tau_1' \leq \lfloor\tau_0\rfloor$, $\lfloor\tau_0\rfloor$ is an upper bound of τ_0' and τ_1'. By the definition of join on a lattice, $\tau_0' \sqcup \tau_1'$ is less-than-or-equal-to any other upper bound of τ_0 and τ_1, so this is shown.

T-Sub Consider arbitrary Γ_0, e_0, τ_1, τ_0, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Sub with premise $\tau_0 \lessdot: \tau_1$ if $e = e_0$ and $\tau = \tau_1$. By induction, $\Gamma_0 \vdash e^+ : \tau_0'$ for some $\tau_0' \leq \lfloor\tau_0\rfloor$. Then instantiate $\tau' = \tau_0'$. It is immediate that $\Gamma_0 \vdash e^+ : \tau'$; it remains to be shown that $\tau' \leq \lfloor\tau_1\rfloor$. Since $\tau_0 \lessdot: \tau_1$, $\tau_0 \leq \tau_1$. By Lemma 3.8, $\lfloor\tau_0\rfloor \leq \lfloor\tau_1\rfloor$. Then by Lemma 3.9, $\tau' = \tau_0' \leq \lfloor\tau_0\rfloor \leq \lfloor\tau_1\rfloor$ so $\tau' \leq \lfloor\tau_1\rfloor$.

□

Lemma 3.8 (Lattice ordering is preserved by tag-of). *If $\tau_0 \leq \tau_1$, then $\lfloor\tau_0\rfloor \leq \lfloor\tau_1\rfloor$.*

Proof. By cases on the structure of the definition of $\leq$; in each case the lemma is immediate. □

Lemma 3.9 (Lattice ordering is transitive). *If $\tau \leq \tau'$ and $\tau' \leq \tau''$, then $\tau \leq \tau''$.*

Proof. By induction on the structure of the definition of $\tau \leq \tau'$ (generalized with respect to τ''), with the cases of $\lessdot:$ inlined:

$\mathsf{Nat} \lessdot: \mathsf{Int}$: Since by assumption $\mathsf{Int} \leq \tau''$, it is clear by inspection that τ'' must be either Int or $*$; in either case $\mathsf{Nat} \lessdot: \tau''$ is immediate.

$\tau_0 \times \tau_1 \lessdot: \tau_2 \times \tau_3$: This is subsumed by the case $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ below.

$\tau_0 \to \tau_1 \lessdot: \tau_2 \to \tau_3$: Because we are considering the lattice of honest transient types, $\tau_0 = \tau_2 = *$, and this is subsumed by the case $* \to \tau_1 \leq * \to \tau_3$ below.

$\tau \leq \tau$: Since by assumption $\tau' \leq \tau''$, $\tau = \tau' \leq \tau_2$.

$\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$: Since by assumption $\tau' = \tau_2 \times \tau_3 \leq \tau''$, it is clear that τ'' must be either $*$ or $\tau_0'' \times \tau_1''$ for some τ_0'', τ_1'' s.t. $\tau_2 \leq \tau_0''$ and $\tau_3 \leq \tau_1''$. If τ'' is $*$, the lemma follows immediately. Otherwise, note that this rule requires that $\tau_0 \leq \tau_2$ and $\tau_1 \leq \tau_3$; hence, by induction, $\tau_0 \leq \tau_0''$ and $\tau_1 \leq \tau_1''$, and therefore $\tau \leq \tau''$.

$* \to \tau_1 \leq * \to \tau_3$: Since by assumption $\tau' = * \to \tau_3 \leq \tau''$, it is clear that τ'' must be either $*$ or $* \to \tau_1''$ for some τ_1'' s.t. $\tau_3 \leq \tau_1''$. If τ'' is $*$, the lemma follows immediately. Otherwise, note that this rule requires that $\tau_1 \leq \tau_3$; hence, by induction, $\tau_1 \leq \tau_1''$, and therefore $\tau \leq \tau'$".

$\bot \leq \tau$ $\tau = \bot \leq \tau''$ is immediate by the definition of lattice ordering.

$\tau \leq *$ Since by assumption $\tau' = * \leq \tau''$, τ'' must be $*$, and so the lemma follows immediately.

□

3.3 Tag Typing Implies Truer Transient Typing

THEOREM 3.10 (TAG TYPING IMPLIES TRUER TRANSIENT TYPING). *If* $\Gamma \vdash_{\mathsf{tag}} e : K$ *then* $\exists \tau \leq K$ *such that* $\Gamma \vdash_{\mathsf{tru}} e : \tau$.

PROOF. By induction over the tag typing derivation.

$$\frac{(x_0 : K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\mathsf{tag}} x_0 : K_0}\ \textsc{T-Var} \qquad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} n_0 : \mathsf{Nat}}\ \textsc{T-Nat} \qquad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} i_0 : \mathsf{Int}}\ \textsc{T-Int} \qquad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{True} : \mathsf{Bool}}\ \textsc{T-True} \qquad \frac{}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{False} : \mathsf{Bool}}\ \textsc{T-False}$$

These cases are immediate by applying the corresponding truer typing rule and from premises.

$$\frac{\Gamma_0,\ (x_0 : K_0) \vdash_{\mathsf{tag}} e_0 : K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \lambda(x_0 : K_0).\ e_0 : * \to *}\ \textsc{T-Lam} \qquad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \langle e_0, e_1 \rangle : * \times *}\ \textsc{T-Pair}$$

$$\frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : \mathsf{Bool} \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_2 : K_0}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : K_0}\ \textsc{T-If} \qquad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad K_0 \leqslant: K_1}{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_1}\ \textsc{T-Sub}$$

These cases follows by the induction hypothesis and the corresponding rule.

$$\frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : * \to * \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_0}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{app}\{K_1\}\ e_0\ e_1 : K_1}\ \textsc{T-App} \qquad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : * \times *}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{fst}\{K_0\}\ e_0 : K_0}\ \textsc{T-Fst} \qquad \frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : * \times *}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{snd}\{K_1\}\ e_0 : K_1}\ \textsc{T-Snd}$$

These cases follow by induction and their corresponding typing rule, with the caveat that if the truer type of the premise is $\bot$, the corresponding bot rule must be used.

$$\frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad K_0 \sim K_1}{\Gamma_0 \vdash_{\mathsf{tag}} \mathsf{cast}\ \{K_1 \Leftarrow K_0\}\ e_0 : K_1}\ \textsc{T-Cast}$$

This case follows by induction and applying the bnd rule in truer, noting truer doesn't require any relationships between the type of what's underneath and the tags on the bnds.

$$\frac{\Gamma_0 \vdash_{\mathsf{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\mathsf{tag}} e_1 : K_1 \quad \Delta(\mathit{binop}, K_0, K_1) = K_2}{\Gamma_0 \vdash_{\mathsf{tag}} \mathit{binop}\, e_0\, e_1 : K_2}\ \textsc{T-Binop}$$

This case follows by induction, noting that if either of the truer types corresponding to K_0 or K_1 are $\bot$, then the result type is $\bot$. If the truer types are different, ie one is Nat and the other Int, we apply subsumption to get both at Int, and then can apply the binop rule. Otherwise, we directly apply the binop rule.

□

4 Vigilance

4.1 Vigilance Logical Relation

$$[\![\Gamma \vdash_t e : \tau]\!]^L_V \triangleq \forall (k, \Psi, \Sigma, \gamma) \in \mathcal{G}^L[\![\Gamma]\!] \text{ where } \Sigma : (k, \Psi).\ (k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^L[\![\tau]\!]$$

$$\begin{aligned}\mathcal{G}^L[\![\Gamma, x : \tau]\!] \triangleq \{(k, \Psi, \Sigma, \gamma[x \mapsto \ell]) \mid{}& (k, \Psi, \Sigma, \gamma) \in \mathcal{G}^L[\![\Gamma]\!] \\ &\wedge \ell \in dom(\Psi) \wedge \ell \notin dom(\gamma) \\ &\wedge (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\tau]\!]_k\}\end{aligned}$$

$$\mathcal{G}^L[\![\bullet]\!] \triangleq \{(k, \Psi, \Sigma, \emptyset)\}$$

$$\begin{aligned}\vdash \Sigma \triangleq\ \forall \ell \in dom(\Sigma).\ & \Sigma(\ell) = ((\ell', \mathsf{some}(\tau', \tau)) \wedge \tau' \sim \mathsf{pointsto}(\Sigma, \ell) \wedge \tau \sim \mathsf{pointsto}(\Sigma, \ell) \\ &\qquad\qquad \wedge \neg \ast\!\!\times\!\!\ast \sim \mathsf{pointsto}(\Sigma, \ell)) \\ &\vee \Sigma(\ell) = (v, \mathsf{none}) \text{ where } v \notin \mathbb{L}\end{aligned}$$

$$\begin{aligned}\Sigma : (k, \Psi) \triangleq\ & dom(\Sigma) = dom(\Psi) \wedge\ \vdash \Sigma\ \wedge\ \forall j < k, \ell \in dom(\Sigma).((j, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![\Psi(\ell)]\!] \\ &\wedge (\Sigma(\ell) = (\ell', \mathsf{some}(\tau, \tau')) \Rightarrow \Psi(\ell) = [\tau, \tau', \Psi(\ell')] \wedge \Psi(\ell') = [\tau'', \ldots] \wedge \tau'' <: \tau') \\ &\wedge (\Sigma(\ell) = (v, \mathsf{none}) \wedge v \notin \mathbb{L} \Rightarrow \exists \tau.\Psi(\ell) = [\tau]))\end{aligned}$$

This is an unfolded version of the definition in the paper. We break up the definition there for ease of explanation, and unfold here for ease of use.

$$(j, \Psi) \sqsupseteq (k, \Psi) \triangleq j \leq k \wedge \forall \ell \in dom(\Psi).\ \Psi'(\ell) = \Psi(\ell)$$

$$\begin{aligned}\mathcal{EH}^L[\![\overline{\tau}]\!] \triangleq \{(k, \Psi, \Sigma, e) \mid{}& \forall j \leq k.\ \forall \Sigma' \supseteq \Sigma, e'.\ (\Sigma, e) \longrightarrow^j_L (\Sigma', e') \wedge \mathtt{irred}(e') \\ &\Rightarrow (e' = \mathsf{Err}^\bullet \vee (\exists (k - j, \Psi') \sqsupseteq (k, \Psi).\ \Sigma' : (k - j, \Psi') \wedge (k - j, \Psi', \Sigma', e') \in \mathcal{VH}^L[\![\overline{\tau}]\!]))\}\end{aligned}$$

$$\mathcal{VH}^L[\![\mathsf{Int}, \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\mathsf{Int}, \tau_2, \ldots \tau_n].\ (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\tau]\!]\}$$

$$\mathcal{VH}^L[\![\mathsf{Nat}, \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\mathsf{Nat}, \tau_2, \ldots \tau_n].\ (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\tau]\!]\}$$

$$\mathcal{VH}^L[\![\mathsf{Bool}, \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\mathsf{Bool}, \tau_2, \ldots \tau_n].\ (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\tau]\!]\}$$

$$\mathcal{VH}^L[\![\tau_1' \times \tau_1'', \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_) \\ \land (k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^L[\![\tau_1', \mathit{fst}(\tau_2), \ldots \mathit{fst}(\tau_n)]\!] \\ \land (k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^L[\![\tau_1'', \mathit{snd}(\tau_2), \ldots \mathit{snd}(\tau_n)]\!]\}$$

$$\mathcal{VH}^L[\![\tau_1' \to \tau_1'', \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi), \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi'). \\ \forall \tau_0 \text{ where } \mathit{cod}(\tau_1'') \leqslant: \tau_0 . \forall \ell_v \text{ where } (j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^L[\![\tau_1']\!]. \\ (j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{EH}^L[\![[\tau_0, \mathit{cod}(\tau_2), \ldots \mathit{cod}(\tau_n)]]\!]\}$$

$$\mathcal{VH}^L[\![*, \tau_2, \ldots \tau_n]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![\mathsf{Int}, \tau_2, \ldots \tau_n]\!] \\ (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![\mathsf{Bool}, \tau_2, \ldots \tau_n]\!] \\ \lor (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![* \times *, \tau_2, \ldots, \tau_n]\!] \\ \lor (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![* \to *, \tau_2, \ldots, \tau_n]\!]\}$$

$$\mathcal{E}^L[\![\tau]\!] \triangleq \{(k, \Psi, \Sigma, e) \mid \forall j \leq k. \forall \Sigma' \supseteq \Sigma, e'. (\Sigma, e) \longrightarrow_L^j (\Sigma', e') \land \mathtt{irred}(e') \\ \Rightarrow (e' = \mathsf{Err}^\bullet \lor (\exists (k-j, \Psi') \sqsupseteq (k, \Psi). \Sigma' : (k-j, \Psi') \land (k-j, \Psi', \Sigma', e') \in \mathcal{V}^L[\![\tau]\!]))\}$$

$$\mathcal{V}^L[\![\mathsf{Int}]\!] \triangleq \{(k, \Psi, \Sigma, \ell \mid \mathsf{pointsto}(\Sigma, \ell) \in \mathbb{Z}\}$$

$$\mathcal{V}^L[\![\mathsf{Nat}]\!] \triangleq \{(k, \Psi, \Sigma, \ell \mid \mathsf{pointsto}(\Sigma, \ell) \in \mathbb{N}\}$$

$$\mathcal{V}^L[\![\mathsf{Bool}]\!] \triangleq \{(k, \Psi, \Sigma, \ell \mid \mathsf{pointsto}(\Sigma, \ell) \in \mathbb{B}\}$$

$$\mathcal{V}^L[\![\tau_1 \times \tau_2]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_) \land\ (k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^L[\![\tau_1]\!] \land\ (k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^L[\![\tau_2]\!]\}$$

$$\mathcal{V}^L[\![\tau_1 \to \tau_2]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi). \forall \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi'). \\ \forall \ell_v \text{ where } (j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^L[\![\tau_1]\!]. \forall \tau_o. \text{ where } \tau_2 \leqslant: \tau_0 \\ (j, \Psi', \Sigma', \mathsf{app}\{\tau_o\}\ \ell\ \ell_v) \in \mathcal{E}^L[\![\tau_0]\!]\}$$

$$
\begin{aligned}
\mathcal{V}^L[\![*]\!] \triangleq \{(k, \Psi, \Sigma, \ell) \mid\ & (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\mathsf{Int}]\!] \\
& (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\mathsf{Bool}]\!] \\
& \vee (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![* \times *]\!] \\
& \vee (k-1, \Psi \ell \Sigma, \ell) \in \mathcal{V}^L[\![* \to *]\!]\}
\end{aligned}
$$

4.2 Vigilance Theorem

$\Gamma \vdash_t e : \tau$ then $[\![\Gamma \vdash_t e : \sigma]\!]^L_V$

4.3 Vigilance Fundamental Property for Natural with Simple Typing

In this subsection, we use $\Gamma \vdash e : \tau$ to mean $\Gamma \vdash_{\mathsf{sim}} e : \tau$.

4.3.1 Lemmas Used Without Mention

Lemma 4.1 (Stepping to Error Implies Expression Relation). *If* $(\Sigma, e) \longrightarrow^{j}_{N} (\Sigma', \mathsf{Err}^{\bullet})$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{E}^{N}[\![\tau]\!]$

Proof. If $k < j$, then we're done because the condition in the expression relation is vacuously true. Otherwise, we can use j as our steps, Σ' as our ending value log, and $\mathsf{Err}^{\bullet}$ as our irreducible expression, and we satisfy the condition in the expression relation. □

Lemma 4.2 (Stepping to Error Implies Expression History Relation). *If* $(\Sigma, e) \longrightarrow^{j}_{N} (\Sigma', \mathsf{Err}^{\bullet})$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^{N}[\![\overline{\tau}]\!]$

Proof. Similar to the previous proof. □

Lemma 4.3 (Anti-Reduction - Head Expansion - Expression Relation Commutes With Steps). *If* $(k, \Psi', \Sigma', e') \in \mathcal{E}^{N}[\![\tau]\!]$ *and* $(\Sigma, e) \longrightarrow^{j}_{N} (\Sigma', e')$ *and* $\Sigma' : (k, \Psi')$ *then* $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^{N}[\![\tau]\!]$

Proof. Unfolding the expression relation in our hypothesis, there exists (Σ'', e''), j' such that $(\Sigma', e') \longrightarrow^{j'}_{N} (\Sigma'', e'')$ and (Σ''', e'') is irreducible.
Either $e'' = \mathsf{Err}^{\bullet}$, in which case $(\Sigma, e) \longrightarrow^{j+j'}_{N} (\Sigma'', \mathsf{Err}^{\bullet})$, so we're done.
Otherwise, there is a $(k - j', \Psi'') \sqsupseteq (k, \Psi')$ such that $\Sigma'' : (k - j', \Psi'')$, and $(k - j', \Psi'', \Sigma'', e'') \in \mathcal{V}^{N}[\![\tau]\!]$.
Using this information, we can show $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^{N}[\![\tau]\!]$ by noting $(\Sigma, e) \longrightarrow^{j+j'}_{N} (\Sigma'', e'')$. □

Lemma 4.4 (Anti-Reduction - Head Expansion - Expression History Commutes With Steps). *If* $(k, \Psi', \Sigma', e') \in \mathcal{EH}^{N}[\![\overline{\tau}]\!]$ *and* $(\Sigma, e) \longrightarrow^{j}_{N} (\Sigma', e')$ *and* $\Sigma' : (k, \Psi')$ *then* $(k + j, \Psi, \Sigma, e) \in \mathcal{EH}^{N}[\![\overline{\tau}]\!]$

Proof. Similar to the previous proof. □

Lemma 4.5 (The Operational Semantics Preserves Well Formed Value Logs). *If* $\vdash \Sigma$ *and* $(\Sigma, e) \longrightarrow^{*}_{N} (\Sigma', e')$ *then* $\vdash \Sigma'$.

Proof. The proof is immediate by inspection of the Operational Semantics. □

Lemma 4.6 (Not Enough Steps Implies Any Expression Relation). *If* $(\Sigma, e) \longrightarrow^{k}_{N} (\Sigma', e')$ *and* (Σ', e') *is not irreducible, then* $\forall j \leq k.\ (j, \Psi, \Sigma, e) \in \mathcal{E}^{N}[\![\tau]\!]$ *and* $(j, \Psi, \Sigma, e) \in \mathcal{EH}^{N}[\![\tau]\!]$.

Proof. Both conclusions are immediate, since the implications in the relations are vacuously true. □

Lemma 4.7 (The Operational Semantics Only Grows Stores). *If* $(\Sigma, e) \longrightarrow^{*}_{N} (\Sigma', e')$ *then* $\Sigma' \supseteq \Sigma$.

Proof. This is a corollary of Lemma 4.8. □

4.3.2 Lemmas Used With Mention

Lemma 4.8 (The Operational Semantics Produces Value Log Extensions). *If* $(\Sigma, e) \longrightarrow^{*}_{N} (\Sigma', e')$, *then* $\exists \overline{\ell} \subseteq dom(\Sigma')$ *such that* $\overline{\ell \notin dom(\Sigma)}$ *and* $\Sigma' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.

Proof. By inspection of the Operational Semantics, no steps modify the value stored in the value log, meaning $\Sigma' \supseteq \Sigma$.
And also by the inspection of the Operational Semantics, there is exactly one rule to allocate new entries in the value log, meaning $\Sigma' \setminus \Sigma$ is a suitable choice for $\overline{[\ell \mapsto (v, \_)]}$. □

Lemma 4.9 (Steps are Preserved in Future Value Logs). *If $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$ and $\overline{\ell \notin dom(\Sigma')}$ then $(\Sigma\overline{[\ell \mapsto (v, \_)]}, e) \longrightarrow_N^j (\Sigma'\overline{[\ell \mapsto (v, \_)]}, e')$.*

Proof. Since all of the added locations are not in Σ', and therefore also not in Σ, no rule that will lookup a label in the derivation tree for $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$ will find a different value or type.
The only remaining notable reduction steps are those that allocate a new label and value entry, but since $\overline{\ell \notin dom(\Sigma')}$, we can allocate the same entry unchanged. □

Lemma 4.10 (Subtyping Preserves Logical Relations). *$\forall \Sigma, k, \Psi, \tau, \tau'$. where $\Sigma : (k, \Psi)$ and $\tau \leqslant: \tau'$.*

(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau']\!]$
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau]\!]$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau']\!]$
(3) If $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N[\![\tau, \overline{\tau}]\!]$ then $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N[\![\tau', \overline{\tau}]\!]$
(4) If $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\tau, \overline{\tau}]\!]$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\tau', \overline{\tau}]\!]$

Proof. Proceed by mutual induction on k and τ:

- $k = 0$: Both 1 and 3 are immediate if $e \neq \ell$.
  If $e = \ell$ then 1 and 3 follow immediately from 2 and 4.
  2 and 4 follow identically in the $k = 0$ case as they do in the $k > 0$ case, but the function case is vacuously true.
- $k > 0$:
  (1) Unfolding our hypothesis, there is some $(\Sigma', e'), j$ such that $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$.
  If $e' = \mathsf{Err}^\bullet$ then we're done.
  Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi')$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N[\![\tau]\!]$.
  We now have two obligations:
    a) $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N[\![\tau']\!]$.
    b) $\Sigma' : (k - j, \Psi')$.
  For a) by IH 2) (not necessarily smaller by type or index), we have $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N[\![\tau']\!]$, which is what we wanted to show.

  For b), this is immediate from the premise.
  (2) Case split on $\tau \leqslant: \tau'$:
    i) $\tau \leqslant: \tau$: immediate.
    ii) $\mathsf{Nat} \leqslant: \mathsf{Int}$: immediate because $\mathbb{N} \subseteq \mathbb{Z}$.
    iii) $\tau_1 \times \tau_2 \leqslant: \tau_1' \times \tau_2'$, with $\tau_1 \leqslant: \tau_1'$ and $\tau_2 \leqslant: \tau_2'$:
    We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau']\!]$.
    Unfolding our hypothesis, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
    We want to show $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N[\![\tau_1']\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N[\![\tau_2']\!]$.
    We can apply IH 2) (smaller by type) to both of these judgements to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N[\![\tau_1']\!]$ and

$(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N[\![\tau_2']\!]$.

This is sufficient to show $(k, \Psi, \Sigma, \Sigma(\ell)) \in \mathcal{V}^N[\![\tau']\!]$.

iv) $\tau_1 \to \tau_2 \leqslant: \tau_1' \to \tau_2'$, with $\tau_1' \leqslant: \tau_1$ and $\tau_2 \leqslant: \tau_2'$:
We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau']\!]$.
Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1']\!]$.
Let $\tau_0 \geqslant: \tau_2'$.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
From IH 2) (smaller by type) applied to the facts that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1']\!]$ and that $\tau_1' \leqslant: \tau_1$ gives us $(j + 1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
Then, we can apply our hypothesis about $\Sigma(\ell)$ (noting that $\tau_0 \geqslant: \tau_2' \geqslant: \tau_2$) to get $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$, which is what we wanted to prove.

(3) Unfolding our hypothesis, we get that there are some $(\Sigma', e'), j$ such that $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$ and (Σ', e') are irreducible.
If $e' = \mathsf{Err}^\bullet$, then we're done.
Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{VH}^N[\![\tau, \overline{\tau}]\!]$, which means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.
Then by IH 4) (not necessarily smaller by type or index) with $\tau \leqslant: \tau'$, we get $(k - j, \Psi', \Sigma', \ell) \in \mathcal{VH}^N[\![\tau', \overline{\tau}]\!]$, which is what we wanted to show.

(4) We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\tau', \overline{\tau}]\!]$.
We case split on $\tau \leqslant: \tau'$:

i) $\tau = \tau'$: immediate by premise.

ii) $\mathsf{Nat} \leqslant: \mathsf{Int}$:
by our premise, we already get that $\forall \tau_o \in \overline{\tau}, (k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau_o]\!]$.
Therefore, it suffices to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\mathsf{Int}]\!]$ given $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\mathsf{Nat}]\!]$ which is immediate since $\mathbb{N} \subset \mathbb{Z}$.

iii) $\tau_1 \times \tau_2 \leqslant: \tau_1' \times \tau_2$ with $\tau_1 \leqslant: \tau_1'$ and $\tau_2 \leqslant: \tau_2'$:
by our premise, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$ and $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^N[\![\tau_1, fst(\overline{\tau})]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^N[\![\tau_2, snd(\overline{\tau})]\!]$.
We can apply IH 4) (smaller by type) to both to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^N[\![\tau_1', fst(\overline{\tau})]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^N[\![\tau_2', snd(\overline{\tau})]\!]$, which is what we wanted to show.

iv) $\tau_1 \to \tau_2 \leqslant: \tau_1' \to \tau_2'$ with $\tau_1' \leqslant: \tau_1$ and $\tau_2 \leqslant: \tau_2'$:
unfolding what we want to show, let $\Sigma' \supseteq \Sigma$, $(j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1']\!]$.
Let $\tau_0 \leqslant: \tau_2'$.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{EH}^N[\![\tau_0, cod(\overline{\tau})]\!]$.

By IH 2) (smaller by type), we get that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
We can then apply the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\tau, \overline{\tau}]\!]$ to get $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{EH}^N[\![\tau_0, cod(\overline{\tau})]\!]$, which is what we wanted to show.

□

Lemma 4.11 (RV-Monotonicity). *If* $\Sigma : (k, \Psi)$ *and* $0 \leq j \leq k$ *and* $\Sigma' \supseteq \Sigma$ *and* $(k-j, \Psi') \sqsupseteq (k, \Psi)$ *and* $\Sigma' : (k-j, \Psi')$ *and* $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\overline{\tau}]\!]$ *then* $(k-j, \Psi', \Sigma', \ell) \in \mathcal{VH}^N[\![\overline{\tau}]\!]$

Proof. We want to show $(k-j, \Psi', \Sigma', \ell)\mathcal{VH}^N[\![\overline{\tau}]\!]$.
Let τ be the head of $\overline{\tau}$ so that $\overline{\tau} = [\tau, \ldots]$.
We proceed by induction over k and τ:

- $k = 0$: The function and dynamic cases are vacuously true, and the rest follow as in the other case.
- $k > 0$:
  i) $\tau = \mathsf{Int}$: immediate because $\Sigma(\ell) = \Sigma'(\ell)$.
  ii) $\tau = \mathsf{Nat}$: same as previous case.
  iii) $\tau = \mathsf{Bool}$: same as previous case.
  iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
  We want to show $(k-j, \Psi', \Sigma', \ell_1) \in \mathcal{VH}^L[\![\tau_1, \overline{fst(\tau)}]\!]$ and $(k-j, \Psi', \Sigma', \ell_2) \in \mathcal{VH}^L[\![\tau_2, \overline{snd(\tau)}]\!]$.
  We have $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^L[\![\tau_1, \overline{fst(\tau)}]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^L[\![\tau_2, \overline{snd(\tau)}]\!]$.
  Both follow by IH (smaller by type).
  v) $\tau = \tau_1 \rightarrow \tau_2$:
  Let $(j', Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ such that $\Sigma'' : (j', \Psi')$.
  Let $\ell_v \in dom(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
  Let $\tau_0 \geqslant: \tau_2$.
  We want to show $(j', \Psi'', \Sigma'', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
  Since $(j', \Psi'') \sqsupseteq (k, \Psi)$ and $\Sigma'' \supseteq \Sigma$, we can apply our premise to finish the case.
  vi) $\tau = *$: note by downward closure, $\Sigma' : (k-j-1, \Psi')$.
  Then we want to show $(k-j-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![\mathsf{Int}]\!]$ or $(k-j-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![* \times *]\!]$ or $(k-j-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![* \rightarrow *]\!]$.
  We know $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\mathsf{Int}]\!]$ or $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![* \times *]\!]$ or $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![* \rightarrow *]\!]$.
  The case follows by the IH (smaller by index).

□

Lemma 4.12 (Extensions Preserve Value Log Typing). *If* $\Sigma : (k, \Psi)$ *and* $0 \leq j \leq k$ *and* $\Sigma' \supseteq \Sigma$ *and* $(k-j, \Psi') \sqsupseteq (k, \Psi)$ *and* $\Sigma' : (k-j, \Psi')$ *and* $\overline{\ell \notin dom(\Sigma')}$ *and* $\Sigma\overline{[\ell \mapsto (v, \_)]} : (k, \Psi\overline{[\ell \mapsto \overline{\tau}]})$ *then* $\Sigma'\overline{[\ell \mapsto (v, \_)]} : (k-j, \Psi'\overline{[\ell \mapsto \overline{\tau}]})$.

Proof. Note that all of the conditions in $\Sigma'\overline{[\ell \mapsto (v, \_)]} : (k-j, \Psi'\overline{[\ell \mapsto \overline{\tau}]})$ besides those concerning the history relation are immediate from the hypotheses.

Let $\Sigma'' = \Sigma'\overline{[\ell \mapsto (v, \_)]}$ and let $\Psi'' = \Psi'\overline{[\ell \mapsto \overline{\tau}]}$.
We want to show $\forall j' < k-j$, and $\forall \ell \in dom(\Sigma'')$, $(j', \Psi'', \Sigma'', \ell) \in \mathcal{VH}^N[\![\Psi''(\ell)]\!]$.
Note by downward closure, $\Sigma'' : (j', \Psi'')$. If $\ell \in dom(\Sigma')$, then we can apply Lemma 4.11 with the fact that $(j', \Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$.
If $\ell \notin dom(\Sigma')$, then $\ell \in \overline{\ell}$.
Then we can apply Lemma 4.11 with the fact that $(j', \Psi'') \sqsupseteq (k, \Psi\overline{[\ell \mapsto \overline{\tau}]})$ and $\Sigma'' \supseteq \Sigma\overline{[\ell \mapsto (v, \_)]}$ to get $(j', \Psi'', \Sigma'', \ell) \in \mathcal{VH}^N[\![\Psi''(\ell)]\!]$, which is what we wanted to show. □

Lemma 4.13 (Later Than Preserved By Lower Steps). *If* $(j, \Psi') \sqsupseteq (k, \Psi)$ *and* $j' \leq j$ *then* $(j - j', \Psi') \sqsupseteq (k - j', \Psi)$.

Proof. Unfolding the world extension definition, we need to show $j - j' \leq k - j'$ and $\forall \ell \in dom(\Psi), \Psi'(\ell) = \Psi(\ell)$. For the first condition, since $j \leq k$ and $j' \leq j$, $j - j' \leq k - j'$.
For the second condition, we can unfold the hypothesis to get the statement we need. □

Lemma 4.14 (RE-Monotonicity). *If* $\Sigma : (k, \Psi)$ *and* $0 \leq j \leq k$ *and* $\Sigma' \supseteq \Sigma$ *and* $(k - j, \Psi') \sqsupseteq (k, \Psi)$ *and* $\Sigma' : (k - j, \Psi')$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N[\![\overline{\tau}]\!]$ *then* $(k - j, \Psi', \Sigma', e) \in \mathcal{EH}^N[\![\overline{\tau}]\!]$.

Proof. Unfolding the relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \longrightarrow^{j'}_N (\Sigma'', e')$.
If $e' = \mathrm{Err}^\bullet$ then we're done.
Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{VH}^N[\![\overline{\tau}]\!]$.

By Lemma 4.8, $\Sigma'' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.
By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi\overline{[\ell \mapsto \overline{\tau}]}$.
We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma\overline{[\ell' \mapsto (v', \_)]}$.
And from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi\overline{[\ell' \mapsto \overline{\tau'}]}$.
By alpha renaming, we can assume that $\overline{\ell' \notin dom(\Sigma'')}$.
Then by Lemma 4.9, we get that $(\Sigma', e) \longrightarrow^{j'}_N (\Sigma''\overline{[\ell' \mapsto (v', \_)]}, e')$.

Now, unfolding the expression relation in what we want to show, we have two obligations:

a) $\Sigma''\overline{[\ell' \mapsto (v', \_)]} : (k - j - j', \Psi''\overline{[\ell' \mapsto \overline{\tau'}]})$.
b) $(k - j - j', \Psi''\overline{[\ell' \mapsto \overline{\tau'}]}, \Sigma''\overline{[\ell' \mapsto (v', \_)]}, e') \in \mathcal{VH}^N[\![\overline{\tau}]\!]$.

For a) we can apply Lemma 4.12. We have a number of obligations:

i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
ii) $\Sigma'' \supseteq \Sigma$: immediate.
iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 4.13.
iv) $\Sigma'' : (k - j - j', \Psi'')$i: immediate by downward closure.
v) $\overline{\ell' \notin dom(\Sigma'')}$: assumed above by alpha renaming.
vi) $\Sigma\overline{[\ell' \mapsto (v', \_)]} : (k - j, \Psi\overline{[\ell' \mapsto \overline{\tau'}]})$: this is exactly $\Sigma' : (k - j, \Psi')$.

For b), we can apply Lemma 4.11 with the fact proven in a). □

Lemma 4.15 (E-V-Monotonicity). *If* $\Sigma : (k, \Psi)$ *and* $0 \leq j \leq k$ *and* $\Sigma' \supseteq \Sigma$ *and* $(k - j, \Psi') \sqsupseteq (k, \Psi)$ *and* $\Sigma' : (k - j, \Psi')$ *then*

(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$ *then* $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^N[\![\tau]\!]$
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau]\!]$ *then* $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![\tau]\!]$

Proof. Proceed by simultaneous induction on k and τ:

- $k = 0$: 1) follows immediately from 2).
  Proceeds similarly to the other case, but function and dynamic cases are vacuously true.
- $k > 0$:

1) Unfolding the expression relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \longrightarrow_N^{j'} (\Sigma'', e')$.
   If $e' = \mathsf{Err}^{\bullet}$ then we're done.
   Otherwise, there is some $(k-j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k-j', \Psi'')$ and $(k-j', \Psi'', \Sigma'', e') \in \mathcal{V}^N[\![\tau]\!]$.

   By Lemma 4.8, $\Sigma'' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.
   By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi\overline{[\ell \mapsto \tau]}$.
   We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma\overline{[\ell' \mapsto (v', \_)]}$, and from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi\overline{[\ell' \mapsto \tau']}$.
   By alpha renaming, we can assume that $\overline{\ell' \notin dom(\Sigma'')}$.
   Then by Lemma 4.9, we get that $(\Sigma', e) \longrightarrow_N^{j'} (\Sigma''\overline{[\ell' \mapsto (v', \_)]}, e')$.

   Now, unfolding the expression relation in what we want to show, we have two obligations:
   a) $\Sigma''\overline{[\ell' \mapsto (v', \_)]} : (k - j - j', \Psi''\overline{[\ell' \mapsto \tau']})$.
   b) $(k - j - j', \Psi''\overline{[\ell' \mapsto \tau']}, \Sigma''\overline{[\ell' \mapsto (v', \_)]}, e') \in \mathcal{V}^N[\![\tau]\!]$.

   For a) we can apply Lemma 4.12. We have a number of obligations:
   i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
   ii) $\Sigma'' \supseteq \Sigma$: immediate.
   iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 4.13.
   iv) $\Sigma'' : (k - j - j', \Psi'')$i: immediate by downward closure.
   v) $\overline{\ell' \notin dom(\Sigma'')}$: assumed above by alpha renaming.
   vi) $\Sigma\overline{[\ell' \mapsto (v', \_)]} : (k - j, \Psi\overline{[\ell' \mapsto \tau']})$: this is exactly $\Sigma' : (k - j, \Psi')$.

   For b), we can apply the IH 2) (not necessarily smaller by type or index) with the fact proven in a).
2) We want to show that $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![\tau]\!]$.
   We case split on τ:
   i) $\tau = \mathsf{Nat}$: then $\Sigma(\ell) = (n, \_)$ where $n \in \mathbb{N}$, so the case is immediate.

   ii) $\tau = tint$: same as above.

   iii) $\tau = \mathsf{Bool}$: same as above.

   iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
   Unfolding our hypothesis gives us $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$.
   Applying IH 2) (smaller by type) to both gives us $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$, which is sufficient to complete the case.
   v) $\tau = \tau_1 \to \tau_2$: Let $\Sigma'' \supseteq \Sigma'$ and $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (j', \Psi'')$.
   Let $\ell_v \in dom(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
   Let $\tau_0 \geqslant: \tau_2$.
   We want to show $(j', \Psi'', \Sigma'', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
   Since $\supseteq$ and $\sqsupseteq$ are both transitive, we have $\Sigma'' \supseteq \Sigma$, and $(j', \Psi'') \sqsupseteq (k, \Psi)$.
   Therefore we can apply the hypothesis to complete the case.

vi) $\tau = *$: we want to show $(k-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![\mathsf{Int}]\!]$ or $\mathcal{V}^N[\![\mathsf{Bool}]\!]$ or $\mathcal{V}^N[\![* \times *]\!]$ or $\mathcal{V}^N[\![* \to *]\!]$.
This follows from IH 2) (smaller by index).

□

Lemma 4.16 (Check is a No Op in Natural). *(1)* $(k+1, \Psi, \Sigma, \mathsf{assert}\ \tau_0\ e) \in \mathcal{E}^N[\![\tau]\!]$ *iff* $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$.
(2) $(k+1, \Psi, \Sigma, \mathsf{assert}\ \tau_0\ e) \in \mathcal{EH}^V[\![\overline{\tau}]\!]$ *iff* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^V[\![\overline{\tau}]\!]$.

Proof. By the operational semantics, $(\Sigma, \mathsf{assert}\ \tau_0\ e) \longrightarrow_N (\Sigma, e)$, so the statement is immediate. □

Lemma 4.17 (App Annotations Don't Matter in Natural). *(1)* $(k{+}1, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\ e_1\ e_2) \in \mathcal{E}^N[\![\tau]\!]$ *iff* $(k, \Psi, \Sigma, e_1\ e_2) \in \mathcal{E}^N[\![\tau]\!]$.
(2) $(k+1, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\ e_1\ e_2) \in \mathcal{EH}^V[\![\overline{\tau}]\!]$ *iff* $(k, \Psi, \Sigma, e_1\ e_2) \in \mathcal{EH}^V[\![\overline{\tau}]\!]$.

Proof. By the operational semantics, $(\Sigma, \mathsf{app}\{\tau_0\}\ e_1\ e_2) \longrightarrow_N (\Sigma, \mathsf{assert}\ \tau_0\ e_1\ e_2)$.
We can apply Lemma 4.16 to complete the proof. □

Lemma 4.18 (Pairs of Semantically Well Typed Terms are Semantically Well Typed). *If* $(k, \Psi, \Sigma, e_1) \in \mathcal{E}^N[\![\tau_1]\!]$ *and* $(k, \Psi, \Sigma, e_2) \in \mathcal{E}^N[\![\tau_2]\!]$ *then* $(k, \Psi, \Sigma, \langle e_1, e_2\rangle) \in \mathcal{E}^N[\![\tau_1 \times \tau_2]\!]$.

Proof. Unfolding the expression relation in our hypothesis about e_1, we get that there are $(\Sigma, e_1'), j$ such that $(\Sigma, e_1) \longrightarrow_N^j (\Sigma, e_1')$ and (Σ', e_1') is irreducible.
If $e_1' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi)$ and $(k-j, \Psi', \Sigma', e_1') \in \mathcal{V}^N[\![\tau_1]\!]$.
This means $e_1' = \ell_1$ for some $\ell_1 \in dom(\Sigma')$.

With this and by the OS, we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_N^j (\Sigma', \langle loc_1, e_2\rangle)$.

We can apply Lemma 4.15 to our hypothesis about e_2 to get $(k-j, \Psi', \Sigma', e_2) \in \mathcal{E}^N[\![\tau_2]\!]$.
Unfolding the expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \longrightarrow_N^{j'} (\Sigma', e_2')$ and (Σ'', e_2') is irreducible.
If $e_2' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j', \Psi'', \Sigma'', e_2') \in \mathcal{V}^N[\![\tau_2]\!]$, which means $e_2' = \ell_2$ for some $\ell_2 \in dom(\Sigma'')$.

Putting everything together we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_N^{j'} (\Sigma'', \langle \ell_1, \ell_2\rangle)$, with $\Sigma'' : (k-j-j', \Psi'')$.
Note by OS, $(\Sigma'', \langle \ell_1, \ell_2\rangle) \longrightarrow_N (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)])$ where $\ell' \notin dom(\Sigma'')$.

We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)] : (k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
Note the only interesting part of this statement is that $\forall k' < k-j-j'-1.\ (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \ell') \in \mathcal{VH}^N[\![\Psi''(\ell_1) \times \Psi''(\ell_2)]\!]$.
This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in \mathcal{VH}^N[\![\Psi''(\ell_1)]\!]$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^N[\![\Psi''(\ell_2)]\!]$.

We know that $(k - j, \Psi', \Sigma', \ell_1') \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$, and Lemma 4.15 with downward closure and the store typing judgement above.
From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)], \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle], \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$.
This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)], \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N[\![\tau_1 \times \tau_2]\!]$, which is what we wanted to prove. □

Lemma 4.19 (Pairs of History Related Terms are History Related). *If* $(k, \Psi, \Sigma, e_1) \in \mathcal{EH}^N[\![fst(\overline{\tau})]\!]$ *and* $(k, \Psi, \Sigma, e_2) \in \mathcal{EH}^N[\![snd(\overline{\tau})]\!]$ *then* $(k, \Psi, \Sigma, \langle e_1, e_2 \rangle) \in \mathcal{EH}^N[\![\overline{\tau}]\!]$.

Proof. Unfolding the erroring expression relation in our hypothesis about e_1, we get that there are $(\Sigma, e_1'), j$ such that $(\Sigma, e_1) \longrightarrow_N^j (\Sigma, e_1')$ and (Σ', e_1') is irreducible.
If $e_1' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi)$ and $(k - j, \Psi', \Sigma', e_1') \in \mathcal{VH}^N[\![fst(\overline{\tau})]\!]$.
This means $e_1' = \ell_1$ for some $\ell_1 \in dom(\Sigma')$.

With this and by the OS, we get $(\Sigma, \langle e_1, e_2 \rangle) \longrightarrow_N^j (\Sigma', \langle loc_1, e_2 \rangle)$.

We can apply Lemma 4.14 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{EH}^N[\![snd(\overline{\tau})]\!]$.
Unfolding the erroring expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \longrightarrow_N^{j'} (\Sigma', e_2')$ and (Σ'', e_2') is irreducible.
If $e_2' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in \mathcal{VH}^N[\![snd(\overline{\tau})]\!]$, which means $e_2' = \ell_2$ for some $\ell_2 \in dom(\Sigma'')$.

Putting everything together we get $(\Sigma, \langle e_1, e_2 \rangle) \longrightarrow_N^{j'} (\Sigma'', \langle \ell_1, \ell_2 \rangle)$, with $\Sigma'' : (k - j - j', \Psi'')$.
Note by OS, $(\Sigma'', \langle \ell_1, \ell_2 \rangle) \longrightarrow_N (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)])$ where $\ell' \notin dom(\Sigma'')$.

We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)] : (k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
Note the only interesting part of this statement is that $\forall k' < k - j - j' - 1. \ (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)], \ell') \in \mathcal{VH}^N[\![\Psi''(\ell_1) \times \Psi''(\ell_2)]\!]$.
This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in \mathcal{VH}^N[\![\Psi''(\ell_1)]\!]$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^N[\![\Psi''(\ell_2)]\!]$.

We know that $(k - j, \Psi', \Sigma', \ell_1') \in \mathcal{VH}^N[\![fst(\overline{\tau})]\!]$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^N[\![snd(\overline{\tau})]\!]$, and Lemma 4.11 with downward closure and the store typing judgement above.
From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)], \ell_1) \in \mathcal{VH}^N[\![fst(\overline{\tau})]\!]$ and $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle], \ell_2) \in \mathcal{VH}^N[\![snd(\overline{\tau})]\!]$.
This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, \_)], \langle \ell_1, \ell_2 \rangle) \in \mathcal{VH}^N[\![\overline{\tau}]\!]$, which is what we wanted to prove. □

Lemma 4.20 (Applications of Semantically Well Typed Terms are Semantically Well Typed). *If* $(k, \Psi, \Sigma, e_f) \in \mathcal{E}^N[\![\tau \to \tau']\!]$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$ *then* $\forall \tau_0 \geqslant: \tau'$, $(k, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \in \mathcal{E}^N[\![\tau_0]\!]$.

Proof. Unfolding the expression relation in our hypothesis about e_f, we get that there are $(\Sigma', e'_f), j$ such that $(\Sigma, e_f) \longrightarrow^j_N (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.
If $e'_f = \mathsf{Err}^\bullet$, then we're done because the entire application steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e'_f) \in \mathcal{V}^N[\![\tau \to \tau']\!]$.
This means $e'_f = \ell_f$ for some $\ell_f \in dom(\Sigma')$.

Using this, we know from the OS that $(\Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \longrightarrow^j_N (\Sigma', \mathsf{app}\{\tau_0\}\ \ell_f\ e)$.

We can apply Lemma 4.15 with $\Sigma' : (k-j, \Psi')$ to our hypothesis about e to get $(k-j, \Psi', \Sigma', e) \in \mathcal{E}^N[\![\tau]\!]$.
Unfolding the expression relation, we get that there are $(\Sigma'', e'), j'$ such that $(\Sigma', e) \longrightarrow^{j'}_N (\Sigma'', e')$ where (Σ'', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ than we're done, because the whole application errors.
Otherwise, there exists $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j'.\Psi'', \Sigma'', e') \in \mathcal{V}^N[\![\tau]\!]$.
This means $e' = \ell$ for some $\ell \in dom(\Sigma'')$.

Putting what we have together, by the OS, $(\Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \longrightarrow^{j+j'}_N (\Sigma'', (\mathsf{app}\{\tau_0\}\ \ell_f\ \ell))$.
We have $(k-j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^N[\![\tau \to \tau']\!]$ and $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ and $\Sigma'' : (k-j-j', \Psi'')$ and $\tau_0 \geqslant: \tau'$.
We can combine these to get $(k-j-j', \Psi'', \Sigma'', \mathsf{app}\{\tau_0\}\ \ell_f\ \ell) \in \mathcal{E}^N[\![\tau_0]\!]$.
This is sufficient to complete the proof. □

Corollary 4.21. *If* $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^N[\![*]\!]$ *and* $\Sigma(\ell) = w$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![*]\!]$ *then* $(k-1, \Psi, \Sigma, \mathsf{app}\{*\}\ w\ e) \in \mathcal{E}^N[\![*]\!]$.

Lemma 4.22 (Applications of History Related Terms are History Related). *If* $(k, \Psi, \Sigma, e_f) \in \mathcal{EH}^N[\![\tau, \overline{\tau}]\!]$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![dom(\tau)]\!]$ *then* $\forall \tau_0 \geqslant: cod(tau)$, $(k, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \in \mathcal{EH}^N[\![\tau_0, cod(\overline{\tau})]\!]$.

Proof. Unfolding the erroring expression relation in our hypothesis about e_f, we get that there are $(\Sigma', e'_f), j$ such that $(\Sigma, e_f) \longrightarrow^j_N (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.
If $e'_f = \mathsf{Err}^\bullet$, then we're done because the entire application steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e'_f) \in \mathcal{VH}^N[\![\tau, \overline{\tau}]\!]$.
This means $e'_f = \ell_f$ for some $\ell_f \in dom(\Sigma')$.

Using this, we know from the OS that $(\Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \longrightarrow^j_N (\Sigma', \mathsf{app}\{\tau_0\}\ \ell_f\ e)$.

We can apply Lemma 4.15 with $\Sigma' : (k-j, \Psi')$ to our hypothesis about e to get $(k-j, \Psi', \Sigma', e) \in \mathcal{E}^N[\![dom(\tau)]\!]$.
Unfolding the expression relation, we get that there are $(\Sigma'', e'), j'$ such that $(\Sigma', e) \longrightarrow^{j'}_N (\Sigma'', e')$ where (Σ'', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ than we're done, because the whole application errors.
Otherwise, there exists $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j'.\Psi'', \Sigma'', e') \in \mathcal{V}^N[\![\tau]\!]$.
This means $e' = \ell$ for some $\ell \in dom(\Sigma'')$.

Putting what we have together, by the OS, $(\Sigma, \mathsf{app}\{\tau_0\}\, e_f\ e) \longrightarrow_N^{j+j'} (\Sigma'', (\mathsf{app}\{\tau_0\}\, \ell_f\ \ell))$.
We have $(k-j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^N[\![\tau \to \tau']\!]$ and $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ and $\Sigma'' : (k-j-j', \Psi'')$ and $\tau_0 \geqslant: \tau'$.
We can combine these to get $(k-j-j', \Psi'', \Sigma'', \mathsf{app}\{\tau_0\}\, \ell_f\ \ell) \in \mathcal{EH}^N[\![\tau_0, cod(\overline{\tau})]\!]$.
This is sufficient to complete the proof. ◻

Corollary 4.23. *If $(k, \Psi, \Sigma, e_f) \in \mathcal{EH}^N[\![*, \overline{\tau}]\!]$ and $(k-1, \Psi, \Sigma, e) \in \mathcal{E}^N[\![*]\!]$ then $(k-1, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\, e_f\ e) \in \mathcal{EH}^N[\![*, cod(\overline{\tau})]\!]$.*

Lemma 4.24 (Expression Relation implies Expression History Relation).
(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$ then $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N[\![\tau]\!]$.
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau]\!]$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\tau]\!]$.

Proof. Proceed by induction on k and τ:

- $k = 0$: 1) is immediate from 2).
  - $\tau = \mathsf{Int}$: immediate.
  - $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
    The case follows from the IH on ℓ_1 and ℓ_2.
  - $\tau = \tau_1 \to \tau_2$: vacuously true.
  - $\tau = *$: vacuously true.
- $k > 0$: 1) is immediate from 2).
  - $\tau = \mathsf{Int}$: immediate.
  - $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
    The case follows from the IH on ℓ_1 and ℓ_2.
  - $\tau = \tau_1 \to \tau_2$: Follows from 1) from the IH (smaller by index).
  - $\tau = *$: Follows from 2) from the IH (smaller by index), using $* \times *$, $* \to *$, or Int.

◻

Lemma 4.25 (Monitor Compatibility). *If $\Sigma : (k, \Psi)$, then*

(1) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau]\!]$ and $\Sigma(\ell') = (\ell, \mathsf{some}(\tau', \tau))$, then $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![\tau']\!]$
(2) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N[\![\tau]\!]$ then $(k, \Psi, \Sigma, \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, e) \in \mathcal{E}^N[\![\tau']\!]$.
(3) If $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N[\![\Psi(\ell)]\!]$ and $\Psi(\ell) = [\tau_s, \ldots]$ and $\tau \geqslant: \tau_s$ and $\Sigma' = \Sigma[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))]$ and $\Psi' = [\ell' \mapsto \tau', \tau, \Psi(\ell)]\Psi$ and $\ell' \notin dom(\Sigma)$ and $\vdash \Sigma'$ then $(k, \Psi', \Sigma', \ell') \in \mathcal{VH}^N[\![\tau', \tau, \Psi(\ell)]\!]$
(4) If $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N[\![\overline{\tau}]\!]$ and $\overline{\tau} = [\tau, \ldots]$ then $(k, \Psi, \Sigma, \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, e) \in \mathcal{EH}^N[\![\tau', \tau, \overline{\tau}]\!]$

Proof. Proceed by simultaneous induction on k and τ.

- $k = 0$: 2) and 4) follow from 1) and 3) respectively.
  The proofs follow similarly to the other case, but any function or dynamic cases are vacuously true.
- $k > 0$:

1) Unfolding the relation in the statement we want to prove, note from our hypothesis about Σ, we get that $\vdash \Sigma$.

Proceed by case analysis on τ':

a) $\tau' = \mathsf{Nat}$: Since $\vdash \Sigma$, we have $\mathsf{pointsto}(\Sigma, \ell') \sim \mathsf{Nat}$.
Therefore, we have $\mathsf{pointsto}(\Sigma, \ell') \in \mathbb{N}$, which is sufficient to complete the case.

b) $\tau' = \mathsf{Int}$: same reasoning as Nat.

c) $\tau' = \mathsf{Bool}$: same reasoning as Nat.

d) $\tau' = \tau'_1 \times \tau'_2$: By the fact that $\vdash \Sigma$, this case is a contradiction.

e) $\tau' = \tau'_1 \to \tau'_2$: Unfolding the value relation, let $\Sigma' \supseteq \Sigma$, and $(j, \Psi') \sqsupseteq (k, \Psi)$, such that $\Sigma' : (j, \Psi')$.
Let ℓ_v such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![dom(\tau')]\!]$.
Let $\tau_0 \leqslant: cod(\tau')$.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\, \ell'\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
Note by the operational semantics, $(\Sigma', \mathsf{app}\{\tau_0\}\, \ell'\ \ell_v) \longrightarrow^2_N$ $(\Sigma', \mathsf{assert}\ \tau_0\ (\mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))))$.
Note by downward closure we have $\Sigma' : (j-2, \Psi')$.
Therefore it suffices to show $(j{-}2, \Psi', \Sigma', \mathsf{assert}\ \tau_0\ (\mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v)))) \in \mathcal{E}^N[\![\tau_0]\!]$.

Note that $\tau_0 \geqslant: cod(\tau')$.
By Lemma 4.10, it suffices to show $(j{-}2, \Psi', \Sigma', \mathsf{assert}\ \tau_0\ (\mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v)))) \in \mathcal{E}^N[\![cod(\tau')]\!]$.

By Lemma 4.16, it suffices to show $(j{-}3, \Psi', \Sigma', \mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau')\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))) \in \mathcal{E}^N[\![cod(\tau')]\!]$.

By IH 2) (smaller by type), it suffices to show $(j-3, \Psi', \Sigma', \ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v)) \in \mathcal{E}^N[\![cod(\tau')]\!]$.

By Lemma 4.17, it suffices to show $(j-2, \Psi', \Sigma', \mathsf{app}\{cod(\tau')\}\, \ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v)) \in \mathcal{E}^N[\![cod(\tau')]\!]$.

We now have two cases:

i) $\tau = *$:
Then by Lemma 4.21 it suffices to show $(j{-}1, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![*]\!]$ and $(j{-}1, \Psi', \Sigma', \mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v) \in \mathcal{E}^N[\![dom(\tau')]\!]$.
Both follow by Lemma 4.15, and IH 2) (smaller by index) in the second case.

ii) $\tau = \tau_1 \to \tau_2$:
Then by Lemma 4.20 it suffices to show $(j{-}2, \Psi', \Sigma', \ell) \in \mathcal{V}^N[\![\tau]\!]$ and $(j{-}2, \Psi', \Sigma', \mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v) \in \mathcal{E}^N[\![dom(\tau')]\!]$.
Both follow by Lemma 4.15, and IH 2) (smaller by index) in the second case.

f) $\tau' = *$: Unfolding the relation in what we want to show, we want to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![\mathsf{Int}]\!]$ or $\mathcal{V}^N[\![\mathsf{Bool}]\!]$ or $\mathcal{V}^N[\![* \times *]\!]$ or $\mathcal{V}^N[\![* \to *]\!]$.
In each case, we can apply IH 1) (smaller by index) to complete the case.

2) Unfolding the expression relation in our hypothesis, we have that there are (e', Σ'), j such that $(e, \Sigma) \longrightarrow_N^j (e', \Sigma')$ with (e', Σ') irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done, because the monitor will step to an error as well.
Otherwise, there is $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N[\![\tau]\!]$.
This means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.

If $\neg\mathsf{pointsto}(\Sigma', \ell) \sim \tau'$, then $(\Sigma, \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, e) \longrightarrow_N^j (\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma', \mathsf{TypeErr}(\tau', \ell))$, so we're done.
Otherwise, we have $\mathsf{pointsto}(\Sigma', \ell) \sim \tau'$, and since $\mathsf{pointsto}(\Sigma', \ell) \sim \tau$, we also have $\tau \sim \tau'$.

We have 5 cases:

(a) $\tau' = \mathsf{Nat}$:
Then $(\Sigma', \mathsf{mon}\,\{\mathsf{Nat} \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\mathsf{Nat}, \tau))], \ell')$.
It suffices to show $(k - j - 1, \Psi'[\ell' \mapsto \mathsf{Nat}, \tau, \Psi(\ell)], \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\mathsf{Nat}, \tau))], \ell) \in \mathcal{V}^N[\![\mathsf{Nat}]\!]$, and that $\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\mathsf{Nat}, \tau))] : (k - j - 1, \Psi'[\ell' \mapsto \mathsf{Nat}, \tau, \Psi(\ell)])$.
The first follows from downward closure, and the fact that $\Sigma'(\ell) \sim \mathsf{Nat}$ means $\Sigma'(\ell) = n$.
The second follows from IH 3) (smaller by index).

(b) $\tau' = \mathsf{Int}$: Essentially the same as Nat.

(c) $\tau' = \mathsf{Bool}$: Essentially the same as Nat.

(d) $\tau' = \tau'_1 \times \tau'_2$:
By the fact that $fst(\Sigma'(\ell)) \sim \tau'_1 \times \tau'_2$, we have that $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
Then by the OS we have that $(\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma', \langle \mathsf{mon}\,\{\tau'_1 \Leftarrow fst(\tau)\}\, \ell_1, \mathsf{mon}\,\{\tau'_2 \Leftarrow snd(\tau)\}\, \ell_2 \rangle)$.
By downward closure, we get $\Sigma' : (k - j - 1, \Psi')$.
By Lemma 4.18, it suffices to show $(k - j - 1, \Psi', \Sigma', \mathsf{mon}\,\{\tau'_1 \Leftarrow fst(\tau)\}\, \ell_1) \in \mathcal{E}^N[\![\tau'_1]\!]$ and $(k - j - 1, \Psi', \Sigma', \mathsf{mon}\,\{\tau'_2 \Leftarrow snd(\tau)\}\, \ell_2) \in \mathcal{E}^N[\![\tau'_2]\!]$.

If $\tau = \tau_1 \times \tau_2$, then we have $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$, and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$.
Then we just need to apply IH 2) (smaller by type) and Lemma 4.15.

If $\tau = *$, then we have $(k - j, \Psi', \Sigma', \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N[\![*]\!]$.
This means $(k - j - 1, \Psi', \Sigma', \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N[\![* \times *]\!]$.
Therefore $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![*]\!]$, and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N[\![*]\!]$.
Then we just need to apply IH 2) (smaller by index).

(e) $\tau' = \tau'_1 \to \tau'_2$:
By the fact that $\tau \sim \tau'$, and by the OS, we have $(\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))])$ for $\ell' \notin dom(\Sigma')$.
Let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))]$, and $\Psi'' = \Psi'[\ell' \mapsto [\tau', \tau, \Psi'(\ell)]$.
We want to show $\Sigma'' : (k - j - 2, \Psi'')$.

To start, the condition on entries in the value log is immediate.
Otherwise the only interesting case is the value history relation.
Let $k' < k - j - 2$.
Then by downward closure, we get $\Sigma' : (k', \Psi')$.
By IH 3) (smaller by index), we get $(k', \Psi'', \Sigma'', \ell') \in \mathcal{VH}^N[\![\tau', \tau, \Psi(\ell)]\!]$, which is sufficient.
Then we just need to apply IH 1) (smaller by index).

(f) $\tau' = *$: case spit on the shape of $\mathsf{pointsto}(\Sigma', \ell)$:
   i) $\mathsf{pointsto}(\Sigma', \ell) = i$: the proof follows identically to the Nat case.
   ii) $\mathsf{pointsto}(\Sigma', \ell) = b$: the proof follows identically to the Bool case.
   iii) $\mathsf{pointsto}(\Sigma', \ell) = \lambda x : \_ . e$: then by the operational semantics, $(\Sigma', \mathsf{mon}\,\{* \Leftarrow \tau\}\,\ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(*, \tau))], \ell')$.
   Therefore we want to show:
   - $\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(*, \tau))] : (k - j - 2, \Psi'[\ell' \mapsto [*, \tau, \Psi'(\ell)]])$
   - $(k - j - 2, \Psi'[\ell' \mapsto [*, \tau, \Psi'(\ell)]], \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(*, \tau))], \ell') \in \mathcal{V}^N[\![*]\!]$

   The first condition follows from applications of IH 3) (smaller by index).
   The second condition follows from an application of IH 1) (smaller by index).
   iv) $\mathsf{pointsto}(\Sigma', \ell) = \langle \ell_1, \ell_2 \rangle$:
   By the operational semantics, either:
   - $(\Sigma', \mathsf{mon}\,\{* \Leftarrow \tau\}\,\ell) \longrightarrow_N (\Sigma', \langle \mathsf{mon}\,\{* \Leftarrow \mathit{fst}(\tau)\}\,\ell_1, \mathsf{mon}\,\{* \Leftarrow \mathit{snd}(\tau)\}\,\ell_2 \rangle)$ or
   - $(\Sigma', \mathsf{mon}\,\{* \Leftarrow \tau\}\,\ell) \longrightarrow_N (\Sigma', \mathsf{TypeErr}(\tau, \ell))$

   In the case it errors, we're done.
   Otherwise, it suffices to show $(k - j - 1, \Psi', \Sigma', \langle \mathsf{mon}\,\{* \Leftarrow \mathit{fst}(\tau)\}\,\ell_1, \mathsf{mon}\,\{* \Leftarrow \mathit{snd}(\tau)\}\,\ell_2 \rangle) \in \mathcal{E}^N[\![*]\!]$.
   By Lemma 4.18, it suffices to show:
   - $(k - j - 1, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow \mathit{fst}(\tau)\}\,\ell_1) \in \mathcal{E}^N[\![*]\!]$
   - $(k - j - 1, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow \mathit{snd}(\tau)\}\,\ell_2) \in \mathcal{E}^N[\![*]\!]$

   We can unfold our hypothesis that $(k, \Psi, \Sigma, \ell)\mathcal{V}^N[\![\tau]\!]$ to get $(k, \Psi, \Sigma, \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N[\![\tau]\!]$.
   We now have two cases depending on whether $\tau = *$ or $\tau_1 \times \tau_2$:
   - If $\tau = *$, then $(k - 1, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N[\![*]\!]$ and $(k - 1, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N[\![*]\!]$.
     By Lemma 4.15, $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![*]\!]$ and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N[\![*]\!]$.
     Then we can apply IH 2) (smaller by index) to get what we need.
   - If $\tau = \tau_1 \times \tau_2$, then $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$.
     By Lemma 4.15, $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N[\![\tau_2]\!]$.
     Then we can apply IH 2) (smaller by index) to get what we need.

3) We proceed by case analysis on τ':
   (a) $\tau' = \mathsf{Nat}$: Since we already know $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^V[\![N]\!]\Psi(\ell)$, it suffices to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![\tau']\!]$ and $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![\tau]\!]$.
   This is immediate from $\vdash \Sigma'$, which implies $\tau' \sim \mathsf{pointsto}(\Sigma', \ell')$ and $\tau \sim \mathsf{pointsto}(\Sigma', \ell')$.
   (b) $\tau' = \mathsf{Int}$: same as the Nat case.
   (c) $\tau' = \mathsf{Bool}$: same as the Nat case.
   (d) $\tau' = \tau_1' \times \tau_2'$: this case is a contradiction by the fact that $\vdash \Sigma$.

(e) $\tau' = \tau'_1 \to \tau'_2$: Unfolding the relation in what we want to prove, let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
Let τ_0 such that $cod(\tau') \leqslant: \tau_0$.
Let ℓ_v such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![dom(\tau')]\!]$.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell'\ \ell_v) \in \mathcal{EH}^N[\![\tau_0, cod(tau), cod(\Psi'(\ell))]\!]$.
We know by the OS that $(\Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \longrightarrow_N (\Sigma', \mathsf{assert}\ \tau_0\ (\ell\ \ell_v)) \longrightarrow_N$
$(\Sigma', \mathsf{assert}\ \tau_0\ (\mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))))$.
Note by downward closure, $\Sigma' : (j-2, \Psi')$.

By Lemma 4.10, it suffices to show $(j-2, \Psi', \Sigma', \mathsf{assert}\ \tau_0\ (\mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))))$ $\in \mathcal{EH}^N[\![cod(\tau'), cod(\tau), cod(\Psi'(\ell))]\!]$

By Lemma 4.16, it suffices to show $(j-1, \Psi', \Sigma', \mathsf{mon}\ \{cod(\tau') \Leftarrow cod(\tau)\}\ (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))) \in \mathcal{EH}^N[\![cod(\tau'), cod(\tau), cod(\Psi'(\ell))]\!]$.

By IH 4) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', (\ell\ (\mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v))) \in \mathcal{EH}^N[\![cod(\Psi'(\ell))]\!]$.

We now have two cases:

i) $\tau = *$: By Lemma 4.23, it suffices to show $(j, \Psi', \Sigma', \ell) \in \mathcal{EH}^N[\![\Psi'(\ell)]\!]$ and $(j-1, \Psi', \Sigma', \mathsf{mon}\ \{* \Leftarrow dom(\tau')\}\ \ell_v) \in \mathcal{E}^N[\![*]\!]$ (since $\Psi'(\ell) = [\tau, \ldots]$).

The first follows from the fact that $(j, \Psi', \Sigma', \ell) \in \mathcal{VH}^N[\![\Psi'(\ell)]\!]$ by Lemma 4.11.

For the second, by IH 2) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{E}^N[\![dom(\tau')]\!]$.
This follows by Lemma 4.15 applied to the fact that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![dom(\tau')]\!]$.

ii) $\tau = \tau_1 \to \tau_2$:
By Lemma 4.22, it suffices to show $(j-1, \Psi', \Sigma', \ell) \in \mathcal{EH}^N[\![\Psi'(\ell)]\!]$ and $(j-1, \Psi', \Sigma', \mathsf{mon}\ \{dom(\tau) \Leftarrow dom(\tau')\}\ \ell_v) \in \mathcal{E}^N[\![dom(\tau)]\!]$ (since $\Psi'(\ell) = [\tau, \ldots]$).

The first follows from the fact that $(j-1, \Psi', \Sigma', \ell) \in \mathcal{VH}^N[\![\Psi'(\ell)]\!]$ by Lemma 4.11.

For the second, by IH 2) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{E}^N[\![dom(\tau')]\!]$.
This follows by Lemma 4.15 applied to the fact that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![dom(\tau')]\!]$.

(f) $\tau' = *$: unfolding the relation in what we want to show, the proof follows by IH 3) (smaller by index).

4) Unfolding the expression relation in our hypothesis, we have that there are (e', Σ'), j such that $(e, \Sigma) \longrightarrow^j_N (e', \Sigma')$ with (e', Σ') irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done, because the monitor will step to an error as well.
Otherwise, there is $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{VH}^N[\![\overline{\tau}]\!]$.
This means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$, and $\Psi'(\ell) = \overline{\tau}$.

If $\neg\mathsf{pointsto}(\Sigma', \ell) \sim \tau'$, then $(\Sigma, \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, e) \longrightarrow_N^j (\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma', \mathsf{TypeErr}(\tau', \ell))$, so we're done.

Otherwise, we have $\mathsf{pointsto}(\Sigma', \ell) \sim \tau'$, and since $\mathsf{pointsto}(\Sigma', \ell) \sim \tau$, we also have $\tau \sim \tau'$.

We want to show $(k - j, \Psi', \Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \in \mathcal{EH}^N[\![\tau', \tau, \Psi'(\ell)]\!]$.

We have three cases:

a) $\mathsf{pointsto}(\Sigma', \ell) = i$: By OS, $(\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))], \ell')$.
   Let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))]$ and $\Psi'' = Psi'[\ell' \mapsto \tau', \tau, \Psi(\ell)]$.
   Unfolding the relation in what we want to show, it suffices to show $\forall \tau_z \in \Psi''(\ell), (k-j-1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N[\![\tau_z]\!]$ and $\Sigma'' : (k - j - 1, \Psi'')$.

   For the second, we can apply IH 3) (smaller by index).
   For the first, by downward closure, by Lemma 4.11, $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{VH}^N[\![\Psi'(\ell)]\!]$.
   Then we already know $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N[\![\tau_z]\!]$ when $\tau_z \in \Psi'(\ell)$.
   So it suffices to show $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N[\![\tau']\!]$.

   If $\tau' = \mathsf{Int}$, then we're done.
   Otherwise, $\tau' = *$, in which case we need to show $(k - j - 2, \Psi'', \Sigma'', \ell') \in \mathcal{V}^N[\![\mathsf{Int}]\!]$, which is also immediate.

b) $\mathsf{pointsto}(\Sigma', \ell) = b$: essentially the same as the previous case.

c) $\Sigma'(\ell) = \langle \ell_1, \ell_2 \rangle$:
   By OS, $(\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma', \langle \mathsf{mon}\,\{fst(\tau') \Leftarrow fst(\tau)\}\, \ell_1, \mathsf{mon}\,\{snd(\tau') \Leftarrow snd(\tau)\}\, \ell_2 \rangle)$.
   Note by downward closue, $\Sigma' : (k - j - 2, \Psi')$.
   By Lemma 4.19, it suffices to show $(k-j-2, \Psi', \Sigma', \mathsf{mon}\,\{fst(\tau') \Leftarrow fst(\tau)\}\, \ell_1) \in \mathcal{EH}^N[\![fst(\tau'), fst(\tau), fst(\Psi'(\ell))]\!]$ and $(k - j - 2, \Psi', \Sigma', \mathsf{mon}\,\{snd(\tau') \Leftarrow snd(\tau)\}\, \ell_1) \in \mathcal{EH}^N[\![snd(\tau'), snd(\tau), snd(\Psi'(\ell))]\!]$.
   Both of these follow by unfolding the relation in the hypothesis about ℓ, applying Lemma 4.14, and applying IH 4) (smaller by index).

d) $\mathsf{pointsto}(\Sigma', \ell) = \lambda x : \_.\, e$:
   By OS, $(\Sigma', \mathsf{mon}\,\{\tau' \Leftarrow \tau\}\, \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))], \ell')$, where $\ell' \notin dom(\Sigma')$.
   Then let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(\tau', \tau))]$ and let $\Psi'' = \Psi'[\ell' \mapsto \tau', \tau, \Psi'(\ell)]$.
   By IH 3) (smaller by index) we get $(k - j - 2, \Psi'', \Sigma'', \ell') \in \mathcal{VH}^N[\![\tau', \tau, \Psi'(\ell)]\!]$, so all that's left is to show is $\Sigma'' : (k - j - 2, \Psi'')$.

   Let $k' < k - j - 2$.
   Note by downward closure, $\Sigma' : (k', \Psi')$, so $\forall \ell'' \in dom(\Sigma')$, by Lemma 4.11, $(k', \Psi'', \Sigma'', \ell'') \in \mathcal{VH}^N[\![\Psi''(\ell'')]\!]$ (note $\Psi'(\ell'') = \Psi''(\ell'')$).
   So the final condition is $(k', \Psi'', \Sigma'', \ell') \in \mathcal{VH}^N[\![\Psi''(\ell')]\!]$, which follows from IH 3) (smaller by index).

$\square$

4.3.3 Compatability Lemmas

Lemma 4.26 (**T-Var** compatibility). $\dfrac{[\![(x\!:\!\tau) \in \Gamma]\!]}{[\![\Gamma \vdash x : \tau]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(x)) \in \mathcal{E}^N[\![\tau]\!]$.
Since $x : \tau \in \Gamma$, we get that $\gamma(x) = \ell$.
Since $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$, we get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\tau]\!]$.
Then we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^N[\![\tau]\!]$ immediately since ℓ is already a value and we have as a premise that $\Sigma : (k, \Psi)$. □

Lemma 4.27 (**T-Nat** compatibility). $\dfrac{}{[\![\Gamma \vdash n : \mathsf{Nat}]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(n)) \in \mathcal{E}^N[\![\mathsf{Nat}]\!]$.
Note $\gamma(n) = n$.
By the OS, we have $(\Sigma, n) \longrightarrow_N (\Sigma[\ell \mapsto (n, \_)], \ell)$.
We get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N[\![\mathsf{Nat}]\!]$ immediately because $n \in \mathbb{N}$.
Since $\mathcal{V}^N[\![\mathsf{Nat}]\!]$ does not rely on Ψ or Σ, we have that $(k, \Psi[\ell \mapsto [\mathsf{Nat}]], \Sigma[\ell \mapsto (n, \_)], \ell) \in \mathcal{V}^N[\![\mathsf{Nat}]\!]$.

□

Lemma 4.28 (**T-Int** compatibility). $\dfrac{}{[\![\Gamma \vdash i : \mathsf{Int}]\!]}$

Proof. Not meaningfully different from **T-Int** □

Lemma 4.29 (**T-True** compatibility). $\dfrac{}{[\![\Gamma_1 \vdash \mathsf{True} : \mathsf{Bool}]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{True})) \in \mathcal{E}^N[\![\mathsf{Bool}]\!]$.
Note $\gamma(\mathsf{True}) = \mathsf{True}$.
By the OS, we have $(\Sigma, \mathsf{True}) \longrightarrow_N (\Sigma[\ell \mapsto (\mathsf{True}, \_)], \ell)$.
We get $(k, \Psi, \Sigma, \mathsf{True}) \in \mathcal{V}^N[\![\mathsf{Bool}]\!]$ immediately.
Since $\mathcal{V}^N[\![\mathsf{Bool}]\!]$ does not rely on Ψ or Σ, we have that $(k, \Psi[\ell \mapsto [\mathsf{Bool}]], \Sigma[\ell \mapsto (\mathsf{True}, \_)], \ell) \in \mathcal{V}^N[\![\mathsf{Bool}]\!]$.

□

Lemma 4.30 (**T-False** compatibility). $\dfrac{}{[\![\Gamma_1 \vdash \mathsf{False} : \mathsf{Bool}]\!]}$

Proof. Not meaningfully different from the previous case. □

Lemma 4.31 (**T-Lam** compatibility). $\dfrac{[\![\Gamma_1,\ (x_1\!:\!\tau_1) \vdash e_1 : \tau_2]\!]}{[\![\Gamma_1 \vdash \lambda(x_1\!:\!\tau_1).\, e_1 : \tau_1 \!\rightarrow\! \tau_2]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\lambda x_1 : \tau_1.\, e_1)) \in \mathcal{E}^N[\![\tau_1 \rightarrow \tau_2]\!]$.

Note that $\gamma(\lambda x_1 : \tau_1.\, e_1) = \lambda x_1 : \tau_1.\, \gamma(e_1)$.
Since $\lambda x_1 : \tau_1.\, \gamma(e_1)$ is a value, by the OS we have $(\Sigma, \lambda x_1 : \tau_1.\, \gamma(e_1)) \longrightarrow_N (\Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\, \gamma(e_1), \mathsf{none})])$, where $\ell \notin dom(\Sigma)$.
We choose our later Ψ' to be $\Psi[\ell \mapsto \tau_1 \to \tau_2]$.
We now have two obligations:

(1) $(k-1, \Psi[\ell \mapsto \tau_1 \to \tau_2], \Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\, \gamma(e_1), \mathsf{none}], \ell) \in \mathcal{V}^N[\![\tau_1 \to \tau_2]\!]$
(2) $\Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\, \gamma(e_1), \mathsf{none})] : (k-1, \Psi[\ell \mapsto \tau_1 \to \tau_2])$

For 1), unfolding the value relation:
Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto \tau_1 \to \tau_2])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\, \gamma(e_1), \mathsf{none})]$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
Let $\tau_0 \geqslant: \tau_2$.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
By Lemma 4.17, it suffices to show $(j-1, \Psi', \Sigma', \ell\ \ell_v) \in \mathcal{E}^N[\![\tau_0]\!]$.
By the OS, $(\Sigma', \ell\ \ell_v) \longrightarrow_N (\Sigma', \gamma(e_1)[\ell_v/x])$.
By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.
Note that $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^N[\![\Gamma, x : \tau_1]\!]$:

i) $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$ by Lemma 4.15.
ii) $\forall y \in dom(\gamma)$, $(j-1, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^N[\![\Gamma(y)]\!]$ by the premise about γ and Lemma 4.15.

Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma'$, and e_1 at $j-1$ to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Finally, we can apply Lemma 4.10 to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N[\![\tau_0]\!]$ which is what we wanted to show.

For 2), first note the domains are equal, since $dom(\Sigma) = dom(\Psi)$.
Then note $\vdash \Sigma[\ell \mapsto \lambda x_1 : \tau_1.\gamma(e_1)]$ since $\vdash \Sigma$.
Then let $j < k-1$ and let $\ell' \in dom(\Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\gamma(e_1), \mathsf{none})])$.
If $\ell' \neq \ell$, then we get the remaining conditions from $\Sigma : (k, \Psi)$ and Lemma 4.11.
If $\ell' = \ell$, then note the structural obligation on $\Psi[\ell \mapsto [\tau_1 \to \tau_2]]$ is immediate.
We want to show $(j, \Psi[\ell \mapsto \tau_1 \to \tau_2], \Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\gamma(e_1), \mathsf{none})], \ell) \in \mathcal{VH}^N[\![\tau_1 \to \tau_2]\!]$.
Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto \tau_1 \to \tau_2])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : \tau_1.\, \gamma(e_1), \mathsf{none})]$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$.
Let $\tau_0 \geqslant: \tau_2$.
By inspection of the value relation, we get immediately that $\Sigma'(\ell_v) \sim \tau_1$, so we want to show $(j, \Psi', \Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{EH}^V[\![\tau_0]\!]$.
By Lemma 4.17, it suffices to show $(j-1, \Psi', \Sigma', \ell\ \ell_v) \in \mathcal{EH}^V[\![\tau_0]\!]$.
By the OS, $(\Sigma', \ell\ \ell_v) \longrightarrow_N (\Sigma', \gamma(e_1)[\ell_v/x])$.
By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.
Note that $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^N[\![\Gamma, x : \tau_1]\!]$:

i) $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N[\![\tau_1]\!]$ by Lemma 4.15.
ii) $\forall y \in dom(\gamma)$, $(j-1, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^N[\![\Gamma(y)]\!]$ by the premise about γ and Lemma 4.15.

Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma'$, and e_1 at $j-1$ to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Then we can apply Lemma 4.24 to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{EH}^V[\![\tau_2]\!]$.
Finally, we can apply Lemma 4.10 to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{EH}^V[\![\tau_0]\!]$ which is what we wanted to show. □

Lemma 4.32 (**T-Pair compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \tau_1]\!] \quad [\![\Gamma_1 \vdash e_2 : \tau_2]\!]}{[\![\Gamma_1 \vdash \langle e_1, e_2\rangle : \tau_1 \times \tau_2]\!]}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\langle e_1, e_2\rangle)) \in \mathcal{E}^N[\![\tau_1 \times \tau_2]\!]$.
Note $\gamma(\langle e_1, e_2\rangle) = \langle \gamma(e_1), \gamma(e_2)\rangle$.
We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1]\!]$.
We can apply the second hypothesis to get $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Then by Lemma 4.19, $(k, \Psi, \Sigma, \langle \gamma(e_1), \gamma(e_2)\rangle) \in \mathcal{E}^N[\![\tau_1 \times \tau_2]\!]$, which is what we wanted to show. □

Lemma 4.33 (**T-App compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \tau_1 \to \tau_2]\!] \qquad [\![\Gamma_1 \vdash e_2 : \tau_1]\!]}{[\![\Gamma_1 \vdash \mathsf{app}\{\tau_2\}\, e_1\, e_2 : \tau_2]\!]}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{app}\{\tau_2\}\, e_1\, e_2)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Note $\gamma(\mathsf{app}\{\tau_2\}\, e_1\, e_2) = \mathsf{app}\{\tau_2\}\, \gamma(e_1)\, \gamma(e_2)$.
By the first hypothesis we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1 \to \tau_2]\!]$.
By the second hypothesis we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^N[\![\tau_1]\!]$.
Then we can apply Lemma 4.20 to get $(k, \Psi, \Sigma, \mathsf{app}\{\tau_2\}\, \gamma(e_1)\, \gamma(e_2)) \in \mathcal{E}^N[\![\tau_2]\!]$ which is what we wanted to show. □

Lemma 4.34 (**T-Fst compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \tau_1 \times \tau_2]\!]}{[\![\Gamma_1 \vdash \mathsf{fst}\{\tau_1\}\, e_1 : \tau_1]\!]}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma_1]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{fst}\{\tau_1\}\, e_1)) \in \mathcal{E}^N[\![\tau_1]\!]$.
Note $\gamma(\mathsf{fst}\{\tau_1\}\, e_1) = \mathsf{fst}\{\tau_1\}\, \gamma(e_1)$.
From the first hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1 \times \tau_2]\!]$.
Unfolding the expression relation, there are j, Σ', e_1' such that $(\Sigma, \gamma(e_1)) \longrightarrow_N^j (\Sigma'', e_1')$ and e_1' is irreducible.
If $e_1' = \mathsf{Err}^\bullet$ then we're done because the projection also steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j\Psi')$ and $(k-j, \Psi', \Sigma', e_1') \in \mathcal{V}^N[\![\tau_1 \times \tau_2]\!]$.
Unfolding the location and value relations, we get that $\Sigma'(e_1') = \langle \ell_1, \ell_2\rangle$.
By the OS, $(\Sigma, \mathsf{fst}\{\tau_1\}\, e_1) \longrightarrow_N^j (\Sigma' \mathsf{fst}\{\tau_1\}\, e_1') \longrightarrow_N (\Sigma', \mathsf{assert}\; \tau_1\; \ell_1) \longrightarrow_N (\Sigma', \ell_1)$.
We can apply Lemma 4.15 to the premise that $(k-j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$ to get $(k-j-2, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N[\![\tau_1]\!]$.
Finally, we can apply Lemma 4.11 to get that $\Sigma' : (k-j-2, \Psi')$, which is sufficient to complete the proof. □

Lemma 4.35 (**T-Snd compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \tau_1 \times \tau_2]\!]}{[\![\Gamma_1 \vdash \mathsf{snd}\{\tau_2\}\, e_1 : \tau_2]\!]}$$

Proof. Not meaningfully different from the previous lemma. □

Lemma 4.36 (**T-Binop compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \tau_1]\!] \qquad [\![\Gamma_1 \vdash e_2 : \tau_2]\!] \qquad \Delta(binop, \tau_1, \tau_2) = \tau_3}{[\![\Gamma_1 \vdash binop\, e_1\, e_2 : \tau_3]\!]}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(binop\, e_1\, e_2)) \in \mathcal{E}^N[\![\tau_3]\!]$.
Note $\gamma(binop\, e_1\, e_2) = binop\, \gamma(e_1)\, \gamma(e_2)$.
By the first hypothesis applied to γ we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1]\!]$.
Unfolding we get there are j, Σ', e_1' such that $(\Sigma, \gamma(e_1)) \longrightarrow_N^j (\Sigma', e_1')$ and e_1' is irreducible.
If $e_1' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e_1') \in \mathcal{V}^N[\![\tau_1]\!]$.

Note by Lemma 4.15 and Lemma 4.11, we have $(k - j, \Psi', \Sigma', \gamma) \in \mathcal{G}^N[\![\Gamma_1]\!]$ and $\Sigma' : (k - j, \Psi')$.
By the second hypothesis applied to γ we have $(k - j, \Psi', \Sigma', \gamma(e_2)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Unfolding we get there are j', Σ'', e_2' such that $(\Sigma', \gamma(e_2)) \longrightarrow_N^{j'} (\Sigma'', e_2')$ and e_2' is irreducible.
If $e_2' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in \mathcal{V}^N[\![\tau_2]\!]$.

From the definition of Δ, $\tau_3 = \mathsf{Int}$ or Nat the cases proceed identically, so without loss of generality assume $\tau_3 = \mathsf{Int}$.
$\tau_1 = \tau_2 = \mathsf{Int}$, and therefore $\Sigma''(e_1') = i_1$ and $\Sigma''(e_2') = i_2$.
If $binop = \mathsf{quotient}$ and $i_2 = 0$ then $(\Sigma'', binop\, e_1'\, e_2') \longrightarrow_N (\Sigma'', \mathsf{DivErr})$, so we're done.
If $binop = \mathsf{quotient}$ and $i_2 \neq 0$, then $(\Sigma'', binop\, e_1'\, e_2') \longrightarrow_N (\Sigma'', i_1/i_2) \longrightarrow_N (\Sigma''[\ell \mapsto (i_1/i_2, \mathsf{none})], \ell)$.
Since $i_1/i_2 \in \mathbb{Z}$, we're done.
If $binop = \mathsf{sum}$ then $(\Sigma'', binop\, e_1'\, e_2') \longrightarrow_N (\Sigma'', i_1 + i_2) \longrightarrow_N (\Sigma''[\ell \mapsto (i_1 + i_2, \mathsf{none})], \ell)$.
Since $i_1 + i_2 \in \mathbb{Z}$, we're done. □

Lemma 4.37 (**T-If compatibility**).
$$\frac{[\![\Gamma_1 \vdash e_1 : \mathsf{Bool}]\!] \qquad [\![\Gamma_1 \vdash e_2 : \tau]\!] \qquad [\![\Gamma_1 \vdash e_3 : \tau]\!]}{[\![\Gamma_1 \vdash \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 : \tau]\!]}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3)) \in \mathcal{E}^N[\![\tau]\!]$.
Note $\gamma(\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3) = \mathsf{if}\ \gamma(e_1)\ \mathsf{then}\ \gamma(e_2)\ \mathsf{else}\ \gamma(e_3)$.
From the first hypothesis applied to γ, we know $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\mathsf{Bool}]\!]$.
Unfolding, we have that there is Σ', e_1', j such that $(\Sigma, e_1) \longrightarrow_N^j (\Sigma', e_1')$ where e_1' is irreducible.
If $e_1' = \mathsf{Err}^\bullet$ then we're done, because the entire if statement errors.
Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e_1') \in \mathcal{V}^N[\![\mathsf{Bool}]\!]$.
Unfolding the location and then the value relation, we get that $\mathsf{pointsto}(\Sigma', e_1') = \mathsf{True}$ or $\mathsf{pointsto}(\Sigma', e_1') = \mathsf{False}$.

- pointsto(Σ', e_1') = True: Note by OS, $(\Sigma, \text{if } \gamma(e_1) \text{ then } \gamma(e_2) \text{ else } \gamma(e_3)) \longrightarrow_N^j (\Sigma', \text{if } e_1' \text{ then } \gamma(e_2) \text{ else } \gamma(e_3)) \longrightarrow_N (\Sigma', \gamma(e_2))$.
  By Lemma 4.15 and Lemma 4.11, we have $(k - j - 1, \Psi', \Sigma', \gamma) \in \mathcal{G}^N[\![\Gamma_1]\!]$ and $\Sigma' : (k - j - 1, \Psi')$.
  From the second hypothesis, we get $(k - j - 1, \Psi', \Sigma', \gamma(e_2)) \in \mathcal{E}^N[\![\tau]\!]$, which is sufficient to complete the proof.
- pointsto(Σ', e_1') = False: same as other case except replace e_2 with e_3.

□

Lemma 4.38 (**T-Cast** compatibility). $\dfrac{[\![\Gamma_1 \vdash e_1 : \tau_1]\!] \quad \tau_1 \sim \tau_2}{[\![\Gamma_1 \vdash \text{cast}\,\{\tau_2 \Leftarrow \tau_1\}\, e_1 : \tau_2]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\text{cast}\,\{\tau_2 \Leftarrow \tau_1\}\, e_1)) \in \mathcal{E}^N[\![\tau_2]\!]$.
Note $\gamma(\text{cast}\,\{\tau_2 \Leftarrow \tau_1\}\, e_1) = \text{cast}\,\{\tau_2 \Leftarrow \tau_1\}\, \gamma(e_1)$.
By the operational semantics, $(\Sigma, \text{cast}\,\{\tau_2 \Leftarrow \tau_1\}\, \gamma(e_1)) \longrightarrow_N (\Sigma, \text{mon}\,\{\tau_2 \Leftarrow \tau_1\}\, e_1)$.
By Lemma 4.11 and Lemma 4.15, $(k - 1, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ and $\Sigma : (k - 1, \Psi)$.
By the hypothesis, $(k - 1, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1]\!]$.
By Lemma 4.25, $(k - 1, \Psi, \Sigma, \text{mon}\,\{\tau_2 \Leftarrow \tau_1\}\, e_1) \in \mathcal{E}^N[\![\tau_2]\!]$, which is sufficient to complete the proof. □

Lemma 4.39 (**T-Sub** compatibility). $\dfrac{[\![\Gamma_1 \vdash e_1 : \tau_1]\!] \quad \tau_1 \leqslant: \tau_2}{[\![\Gamma_1 \vdash e_1 : \tau_2]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_2]\!]$.
From our hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N[\![\tau_1]\!]$.
We can apply Lemma 4.10 to finish the case. □

4.3.4 Fundamental Property / Vigilance

Theorem 4.40 (Vigilance). *If* $\Gamma \vdash e : \tau$ *then* $[\![\Gamma \vdash e : \tau]\!]_V^N$

Proof. By induction over the typing derivation, using the compatability lemmas. □

4.4 Vigilance Fundamental Property for Transient with Truer Transient Typing

In this subsection, we use $\Gamma \vdash e : \tau$ to mean $\Gamma \vdash_{\mathsf{tru}} e : \tau$.

The relation needs to be extended with a case to handle $\bot$:

$$\mathcal{V}^L[\![\bot]\!] = \emptyset$$

We also edit the function cases of the relation to insert a tag into the annotation of the app, and produce a value in the meet of the tag and the result type:

$$\begin{aligned}\mathcal{VH}^L[\![* \rightarrow \tau_1'', \tau_2, \ldots \tau_n]\!] = \{(k, \Psi, \Sigma, \ell) \mid{}& \forall (j, \Psi') \sqsupseteq (k, \Psi), \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi').\ \forall K.\\ & \forall \ell_v \text{ where } (j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^L[\![*]\!].\\ & (j, \Psi'\Sigma', \mathsf{app}\{\tau_0\}\ \ell\ \ell_v) \in \mathcal{EH}^L[\![[\tau_1'' \sqcap K, cod(\tau_2), \ldots cod(\tau_n)]]\!]\}\end{aligned}$$

$$\begin{aligned}\mathcal{V}^L[\![* \rightarrow \tau_2]\!] = \{(k, \Psi, \Sigma, w) \mid{}& \forall (j, \Psi') \sqsupseteq (k, \Psi).\ \forall \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi').\\ & \forall \ell \text{ where } (j, \Psi', \Sigma', \ell) \in \mathcal{V}^L[\![*]\!].\ \forall K.\\ & (j+1, \Psi', \Sigma', \mathsf{app}\{K\}\ w\ \ell) \in \mathcal{E}^L[\![\tau_2 \sqcap K]\!]\}\end{aligned}$$

We also need to edit the $\Sigma : (k, \Psi)$ judgement because we no longer have or need a correspondance between the from type of a guard and the type underneath the guard:

$$\begin{aligned}\Sigma : (k, \Psi) \triangleq\ & dom(\Sigma) = dom(\Psi) \wedge\ \vdash \Sigma\ \wedge\ \forall j < k, \ell \in dom(\Sigma).((j, \Psi, \Sigma, \ell) \in \mathcal{VH}^L[\![\Psi(\ell)]\!]\\ & \wedge (\Sigma(\ell) = (\ell', \mathsf{some}(\tau, \tau')) \Rightarrow \Psi(\ell) = [\tau, \tau', \Psi(\ell')] \wedge\\ & \wedge (\Sigma(\ell) = (v, \mathsf{none}) \wedge v \notin \mathbb{L} \Rightarrow \exists \tau. \Psi(\ell) = [\tau]))\end{aligned}$$

4.4.1 Lemmas Used Without Mention

Lemma 4.41 (Stepping to Error Implies Expression Relation). *If* $(\Sigma, e) \longrightarrow_T^j (\Sigma', \mathsf{Err}^\bullet)$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$

Proof. If $k < j$, then we're done because the condition in the expression relation is vacuously true. Otherwise, we can use j as our steps, Σ' as our ending value log, and $\mathsf{Err}^\bullet$ as our irreducible expression, and we satisfy the condition in the expression relation. □

Lemma 4.42 (Stepping to Error Implies Expression History). *If* $(\Sigma, e) \longrightarrow_T^j (\Sigma', \mathsf{Err}^\bullet)$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$

Proof. Similar to the previous proof. □

Lemma 4.43 (Anti-Reduction - Head Expansion - Expression Relation Commutes With Steps). *If* $(k, \Psi', \Sigma', e') \in \mathcal{E}^T[\![\tau]\!]$ *and* $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ *and* $\Sigma' : (k, \Psi')$ *then* $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$

Proof. Unfolding the expression relation in our hypothesis, there exists $(\Sigma'', e''), j'$ such that $(\Sigma', e') \longrightarrow_T^{j'} (\Sigma'', e'')$ and (Σ''', e'') is irreducible.
Either $e'' = \mathsf{Err}^\bullet$, in which case $(\Sigma, e) \longrightarrow_T^{j+j'} (\Sigma'', \mathsf{Err}^\bullet)$, so we're done.
Otherwise, there is a $(k - j', \Psi'') \sqsupseteq (k, \Psi')$ such that $\Sigma'' : (k - j', \Psi'')$, and $(k - j', \Psi'', \Sigma'', e'') \in \mathcal{V}^T[\![\tau]\!]$.
Using this information, we can show $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ by noting $(\Sigma, e) \longrightarrow_T^{j+j'} (\Sigma'', e'')$. □

Lemma 4.44 (Anti-Reduction - Head Expansion - Expression History Commutes With Steps). *If* $(k, \Psi', \Sigma', e') \in \mathcal{EH}^T[\![\overline{\tau}]\!]$ *and* $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ *and* $\Sigma' : (k, \Psi')$ *then* $(k + j, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$

Proof. Similar to the previous proof. □

Lemma 4.45 (The Operational Semantics Preserves Well Formed Value Logs). *If* $\vdash \Sigma$ *and* $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$ *then* $\vdash \Sigma'$.

Proof. The proof is immediate by inspection of the Operational Semantics. □

Lemma 4.46 (Not Enough Steps Implies Any Expression Relation). *If* $(\Sigma, e) \longrightarrow_T^k (\Sigma', e')$ *and* (Σ', e') *is not irreducible, then* $\forall j \leq k.\ (j, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ *and* $(j, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\tau]\!]$.

Proof. Both conclusions are immediate, since the implications in the relations are vacuously true. □

Lemma 4.47 (The Operational Semantics Only Grows Stores). *If* $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$ *then* $\Sigma' \supseteq \Sigma$.

Proof. This is a corollary of Lemma 4.48. □

4.4.2 Lemmas Used With Mention

Lemma 4.48 (The Operational Semantics Produces Value Log Extensions). *If* $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$, *then* $\exists \overline{\ell} \subseteq dom(\Sigma')$ *such that* $\overline{\ell \notin dom(\Sigma)}$ *and* $\Sigma' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.

Proof. By inspection of the Operational Semantics, no steps modify the value stored in the value log, meaning $\Sigma' \supseteq \Sigma$.
And also by the inspection of the Operational Semantics, there is exactly one rule to allocate new entries in the value log, meaning $\Sigma' \setminus \Sigma$ is a suitable choice for $\overline{[\ell \mapsto (v, \_)]}$. □

Lemma 4.49 (Steps are Preserved in Future Value Logs). *If* $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ *and* $\overline{\ell \notin dom(\Sigma')}$ *then* $(\Sigma\overline{[\ell \mapsto (v, \_)]}, e) \longrightarrow_T^j (\Sigma'\overline{[\ell \mapsto (v, \_)]}, e')$.

Proof. Since all of the added locations are not in Σ', and therefore also not in Σ, no rule that will lookup a label in the derivation tree for $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ will find a different value or type.
The only remaining notable reduction steps are those that allocate a new label and value entry, but since $\overline{\ell \notin dom(\Sigma')}$, we can allocate the same entry unchanged. □

Lemma 4.50 (Subtyping Preserves Logical Relations). $\forall \Sigma, k, \Psi, \tau, \tau'$. *where* $\Sigma : (k, \Psi)$ *and* $\tau \leqslant: \tau'$.

(1) *If* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau']\!]$
(2) *If* $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau']\!]$
(3) *If* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\tau, \overline{\tau}]\!]$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\tau', \overline{\tau}]\!]$
(4) *If* $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau, \overline{\tau}]\!]$ *then* $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau', \overline{\tau}]\!]$

Proof. Proceed by mutual induction on k and τ:

- $k = 0$: Both 1 and 3 are immediate if $e \neq \ell$.
  If $e = \ell$ then 1 and 3 follow immediately from 2 and 4.
  2 and 4 follow identically in the $k = 0$ case as they do in the $k > 0$ case, but the function case is vacuously true.

- $k > 0$:
  1. Unfolding our hypothesis, there is some $(\Sigma', e'), j$ such that $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$.
     If $e' = \mathsf{Err}^\bullet$ then we're done.
     Otherwise, there is some $(k-j, \Psi') \sqsupseteq (k, \Psi')$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau]\!]$.
     We now have two obligations:
     a) $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau']\!]$.
     b) $\Sigma' : (k-j, \Psi')$.
     For a) by IH 2) (not necessarily smaller by type or index), we have $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau']\!]$, which is what we wanted to show.

     For b), this is immediate from the premise.
  2. Case split on $\tau \leqslant: \tau'$:
     i) $\tau \leqslant: \tau$: immediate.
     ii) $\mathsf{Nat} \leqslant: \mathsf{Int}$: immediate because $\mathbb{T} \subseteq \mathbb{Z}$.
     iii) $\tau_1 \times \tau_2 \leqslant: \tau_1' \times \tau_2'$, with $\tau_1 \leqslant: \tau_1'$ and $\tau_2 \leqslant: \tau_2'$:
     We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau']\!]$.
     Unfolding our hypothesis, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
     We want to show $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T[\![\tau_1']\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T[\![\tau_2']\!]$.
     We can apply IH 2) (smaller by type) to both of these judgements to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T[\![\tau_1']\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T[\![\tau_2']\!]$.
     This is sufficient to show $(k, \Psi, \Sigma, \Sigma(\ell)) \in \mathcal{V}^T[\![\tau']\!]$.
     iv) $* \to \tau_2 \leqslant: * \to \tau_2'$, with $\tau_2 \leqslant: \tau_2'$:
     We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau']\!]$.
     Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
     Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
     Let K.
     We want to show $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_2' \sqcap K]\!]$.
     Then, we can apply our hypothesis about ℓ to get $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_2 \sqcap K]\!]$.
     Finally, we can apply IH 1) (smaller by type) to get $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_2' \sqcap K]\!]$ which is what we wanted to show.

  3. Unfolding our hypothesis, we get that there are some $(\Sigma', e'), j$ such that $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and (Σ', e') are irreducible.
     If $e' = \mathsf{Err}^\bullet$, then we're done.
     Otherwise, there is some $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{VH}^T[\![\tau, \overline{\tau}]\!]$, which means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.
     Then by IH 4) (not necessarily smaller by type or index) with $\tau \leqslant: \tau'$, we get $(k-j, \Psi', \Sigma', \ell) \in \mathcal{VH}^T[\![\tau', \overline{\tau}]\!]$, which is what we wanted to show.
  4. Unfolding the history relation, we want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau', \overline{\tau}]\!]$.
     We case split on $\tau \leqslant: \tau'$:
     i) $\tau = \tau'$: immediate by premise.

ii) Nat $\leqslant$: Int:
by our premise, we already get that $\forall \tau_o \in \overline{\tau}, (k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau_o]\!]$.
Therefore, it suffices to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\mathsf{Int}]\!]$ given $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\mathsf{Nat}]\!]$ which is immediate since $\mathbb{T} \subset \mathbb{Z}$.

iii) $\tau_1 \times \tau_2 \leqslant: \tau_1' \times \tau_2$ with $\tau_1 \leqslant: \tau_1'$ and $\tau_2 \leqslant: \tau_2'$:
by our premise, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$ and $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^T[\![\tau_1, fst(\overline{\tau})]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^T[\![\tau_2, snd(\overline{\tau})]\!]$.
We can apply IH 4) (smaller by type) to both to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^T[\![\tau_1', fst(\overline{\tau})]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^T[\![\tau_2', snd(\overline{\tau})]\!]$, which is what we wanted to show.

iv) $* \to \tau_2 \leqslant: * \to \tau_2'$ with $\tau_2 \leqslant: \tau_2'$:
unfolding what we want to show, let $\Sigma' \supseteq \Sigma$, $(j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
Let K.
We want to show $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{EH}^T[\![\tau' \sqcap K, cod(\overline{\tau})]\!]$.

We can then apply the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau, \overline{\tau}]\!]$ to get $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{EH}^T[\![\tau \sqcap K, cod(\overline{\tau})]\!]$.
Then we can apply IH 3) (smaller by type) to get $(j, \Psi', \Sigma', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{EH}^T[\![\tau' \sqcap K, cod(\overline{\tau})]\!]$, which is what we wanted to show.

□

Lemma 4.51 (RV-Monotonicity). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ and $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\overline{\tau}]\!]$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{VH}^T[\![\overline{\tau}]\!]$*

Proof. We want to show $(k - j, \Psi', \Sigma', \ell)\mathcal{VH}^T[\![\overline{\tau}]\!]$.
Let τ be the head of $\overline{\tau}$ so that $\overline{\tau} = [\tau, \ldots]$.
We proceed by induction over k and τ:

- $k = 0$: The function and dynamic cases are vacuously true, and the rest follow as in the other case.
- $k > 0$:
  i) $\tau = \mathsf{Int}$: immediate because $\Sigma(\ell) = \Sigma'(\ell)$.
  ii) $\tau = \mathsf{Nat}$: same as previous case.
  iii) $\tau = \mathsf{Bool}$: same as previous case.
  iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
  We want to show $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{VH}^L[\![\tau_1, \overline{fst(\tau)}]\!]$ and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{VH}^L[\![\tau_2, \overline{snd(\tau)}]\!]$.
  We have $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^L[\![\tau_1, \overline{fst(\tau)}]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^L[\![\tau_2, \overline{snd(\tau)}]\!]$.
  Both follow by IH (smaller by type).
  v) $\tau = * \to \tau_2$:
  Let $(j', Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ such that $\Sigma''(j', \Psi')$.
  Let $\ell_v \in dom(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
  Let K.
  We want to show $(j', \Psi'', \Sigma'', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_2 \sqcap K]\!]$.
  Since $(j', \Psi'') \sqsupseteq (k, \Psi)$ and $\Sigma'' \supseteq \Sigma$, we can apply our premise to finish the case.

vi) $\tau = *$: note by downward closure, $\Sigma' : (k - j - 1, \Psi')$.
   Then we want to show $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![\mathsf{Int}]\!]$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![* \times *]\!]$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![* \to *]\!]$.
   We know $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\mathsf{Int}]\!]$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![* \times *]\!]$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![* \to *]\!]$.
   The case follows by the IH (smaller by index).

□

Lemma 4.52 (Extensions Preserve Value Log Typing). *If $\Sigma : (k, \Psi)$ and $0 \le j \le k$ and $\Sigma' \sqsupseteq \Sigma$ and $(k-j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ and $\overline{\ell \notin dom(\Sigma')}$ and $\Sigma\overline{[\ell \mapsto (v, \_)]} : (k, \Psi\overline{[\ell \mapsto \tau]})$ then $\Sigma'\overline{[\ell \mapsto (v, \_)]} : (k - j, \Psi'\overline{[\ell \mapsto \tau]})$.*

Proof. Note that all of the conditions in $\Sigma'\overline{[\ell \mapsto (v, \_)]} : (k - j, \Psi'\overline{[\ell \mapsto \tau]})$ besides those concerning the history relation are immediate from the hypotheses.

Let $\Sigma'' = \Sigma'\overline{[\ell \mapsto (v, \_)]}$ and let $\Psi'' = \Psi'\overline{[\ell \mapsto \tau]}$.
We want to show $\forall j' < k - j$, and $\forall \ell \in dom(\Sigma'')$, $(j', \Psi'', \Sigma'', \ell) \in \mathcal{VH}^T[\![\Psi''(\ell)]\!]$.
Note by downward closure, $\Sigma'' : (j', \Psi'')$. If $\ell \in dom(\Sigma')$, then we can apply Lemma 4.51 with the fact that $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \sqsupseteq \Sigma'$.
If $\ell \notin dom(\Sigma')$, then $\ell \in \overline{\ell}$.
Then we can apply Lemma 4.51 with the fact that $(j', \Psi'') \sqsupseteq (k, \Psi\overline{[\ell \mapsto \tau]})$ and $\Sigma'' \sqsupseteq \Sigma\overline{[\ell \mapsto (v, \_)]}$ to get $(j', \Psi'', \Sigma'', \ell) \in \mathcal{VH}^T[\![\Psi''(\ell)]\!]$, which is what we wanted to show. □

Lemma 4.53 (Later Than Preserved By Lower Steps). *If $(j, \Psi') \sqsupseteq (k, \Psi)$ and $j' \le j$ then $(j - j', \Psi') \sqsupseteq (k - j', \Psi)$.*

Proof. Unfolding the world extension definition, we need to show $j - j' \le k - j'$ and $\forall \ell \in dom(\Psi)$, $\Psi'(\ell) = \Psi(\ell)$.
For the first condition, since $j \le k$ and $j' \le j$, $j - j' \le k - j'$.
For the second condition, we can unfold the hypothesis to get the statement we need. □

Lemma 4.54 (RE-Monotonicity). *If $\Sigma : (k, \Psi)$ and $0 \le j \le k$ and $\Sigma' \sqsupseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ and $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$.*

Proof. Unfolding the relation in our hypothesis, we get that there is some (Σ'', e'), j' such that $(\Sigma, e) \longrightarrow_T^{j'} (\Sigma'', e')$.
If $e' = \mathsf{Err}^\bullet$ then we're done.
Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{VH}^T[\![\overline{\tau}]\!]$.

By Lemma 4.48, $\Sigma'' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.
By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi\overline{[\ell \mapsto \tau]}$.
We also know from $\Sigma' \sqsupseteq \Sigma$ that $\Sigma' = \Sigma\overline{[\ell' \mapsto (v', \_)]}$.
And from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi[\ell' \mapsto \overline{\tau'}]$.
By alpha renaming, we can assume that $\overline{\ell' \notin dom(\Sigma'')}$.
Then by Lemma 4.49, we get that $(\Sigma', e) \longrightarrow_T^{j'} (\Sigma''\overline{[\ell' \mapsto (v', \_)]}, e')$.

Now, unfolding the expression relation in what we want to show, we have two obligations:

a) $\Sigma''\overline{[\ell' \mapsto (v', \_)]} : (k - j - j', \Psi''\overline{[\ell' \mapsto \tau']})$.
b) $(k - j - j', \Psi''[\ell' \mapsto \overline{\tau'}], \Sigma''\overline{[\ell' \mapsto (v', \_)]}, e') \in \mathcal{VH}^T[\![\overline{\tau}]\!]$.

For a) we can apply Lemma 4.52. We have a number of obligations:

i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
ii) $\Sigma'' \supseteq \Sigma$: immediate.
iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 4.53.
iv) $\Sigma'' : (k - j - j', \Psi'')$i: immediate by downward closure.
v) $\overline{\ell' \notin dom(\Sigma'')}$: assumed above by alpha renaming.
vi) $\Sigma\overline{[\ell' \mapsto (v', \_)]} : (k - j, \Psi\overline{[\ell' \mapsto \overline{\tau'}]})$: this is exactly $\Sigma' : (k - j, \Psi')$.

For b), we can apply Lemma 4.51 with the fact proven in a). □

Lemma 4.55 (E-V-Monotonicity). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ then*

(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^T[\![\tau]\!]$
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![\tau]\!]$

Proof. Proceed by simultaneous induction on k and τ:

- $k = 0$: 1) follows immediately from 2).
  Proceeds similarly to the other case, but function and dynamic cases are vacuously true.
- $k > 0$:
  1) Unfolding the expression relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \longrightarrow_T^{j'} (\Sigma'', e')$.
  If $e' = \mathsf{Err}^\bullet$ then we're done.
  Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{V}^T[\![\tau]\!]$.

  By Lemma 4.48, $\Sigma'' = \Sigma\overline{[\ell \mapsto (v, \_)]}$.
  By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi\overline{[\ell \mapsto \overline{\tau}]}$.
  We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma\overline{[\ell' \mapsto (v', \_)]}$, and from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi\overline{[\ell' \mapsto \overline{\tau'}]}$.
  By alpha renaming, we can assume that $\overline{\ell' \notin dom(\Sigma'')}$.
  Then by Lemma 4.49, we get that $(\Sigma', e) \longrightarrow_T^{j'} (\Sigma''\overline{[\ell' \mapsto (v', \_)]}, e')$.

  Now, unfolding the expression relation in what we want to show, we have two obligations:
  a) $\Sigma''\overline{[\ell' \mapsto (v', \_)]} : (k - j - j', \Psi''\overline{[\ell' \mapsto \overline{\tau'}]})$.
  b) $(k - j - j', \Psi''\overline{[\ell' \mapsto \overline{\tau'}]}, \Sigma''\overline{[\ell' \mapsto (v', \_)]}, e') \in \mathcal{V}^T[\![\tau]\!]$.
  For a) we can apply Lemma 4.52. We have a number of obligations:
  i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
  ii) $\Sigma'' \supseteq \Sigma$: immediate.
  iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 4.53.
  iv) $\Sigma'' : (k - j - j', \Psi'')$i: immediate by downward closure.
  v) $\overline{\ell' \notin dom(\Sigma'')}$: assumed above by alpha renaming.
  vi) $\Sigma\overline{[\ell' \mapsto (v', \_)]} : (k - j, \Psi\overline{[\ell' \mapsto \overline{\tau'}]})$: this is exactly $\Sigma' : (k - j, \Psi')$.

  For b), we can apply the IH 2) (not necessarily smaller by type or index) with the fact proven in a).

2) We want to show that $(k-j, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![\tau]\!]$.
   We case split on τ:
   i) $\tau = \mathsf{Nat}$: then $\Sigma(\ell) = (n, \_)$ where $n \in \mathbb{T}$, so the case is immediate.

   ii) $\tau = tint$: same as above.

   iii) $\tau = \mathsf{Bool}$: same as above.

   iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
   Unfolding our hypothesis gives us $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$.
   Applying IH 2) (smaller by type) to both gives us $(k-j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k-j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$, which is sufficient to complete the case.
   v) $\tau = * \to \tau_2$: Let $\Sigma'' \supseteq \Sigma'$ and $(j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (j', \Psi'')$.
   Let $\ell_v \in dom(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
   Let K.
   We want to show $(j', \Psi'', \Sigma'', \mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![K \sqcap \tau_2]\!]$.
   Since $\supseteq$ and $\sqsupseteq$ are both transitive, we have $\Sigma'' \supseteq \Sigma$, and $(j', \Psi'') \sqsupseteq (k, \Psi)$.
   Therefore we can apply the hypothesis to complete the case.
   vi) $\tau = *$: we want to show $(k-1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\![\mathsf{Int}]\!]$ or $\mathcal{V}^T[\![\mathsf{Bool}]\!]$ or $\mathcal{V}^T[\![* \times *]\!]$ or $\mathcal{V}^T[\![* \to *]\!]$.
   This follows from IH 2) (smaller by index).

□

Lemma 4.56 (Bot Relation If and Only If Error). *$(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\bot]\!]$ and $(\Sigma, e) \longrightarrow^j_T (\Sigma', e')$ where (Σ', e') is irreducible and $j \le k$, iff $e' = \mathsf{Err}^\bullet$.*

Proof.
- $\Rightarrow$: Unfolding our hypothesis about e in the expression relation, we get that either:
  - $e' = \mathsf{Err}^\bullet$ or
  - $\exists (k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\bot]\!]$

  Assume for sake of contradiction the second case holds.
  $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\bot]\!]$ implies $(k-j, \Psi', \Sigma', \Sigma'(e')) \in \mathcal{V}^T[\![\bot]\!]$, which is a contradiction.
  Therefore, $e' = \mathsf{Err}^\bullet$.
- $\Leftarrow$: immediate.

□

Lemma 4.57 (Tagmatch Makes Values In Relation At Meet). *If $K \sim \mathsf{pointsto}(\Sigma, \ell)$ and $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ then $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![K \sqcap \tau]\!]$*

Proof. There are three cases to consider:

(1) $K \sqcap \tau = \bot$: a contradiction.

(2) $K \sqcap \tau = \tau$: immediate by Lemma 4.55.

(3) $K \sqcap \tau = K$ and $\tau = *$: immediate by unfolding the value relation in our hypothesis, and noting that whichever type of Int, $* \times *$ or $* \to *$ we satisfy must be K.

□

Lemma 4.58 (Check Makes Terms In Relation At Meet). *If* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, \mathsf{assert}\, K\, e) \in \mathcal{E}^T[\![\tau \sqcap K]\!]$.

Proof. Unfolding the expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and (Σ', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done.
Otherwise $\exists (k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau]\!]$.
It suffices to show $(k - j, \Psi', \Sigma', \mathsf{assert}\, K\, e') \in \mathcal{E}^T[\![\tau \sqcap K]\!]$.
By the OS, if $\neg K \sim \mathsf{pointsto}(\Sigma', e')$ then $(\Sigma', \mathsf{assert}\, K\, e') \longrightarrow_T (\Sigma', \mathsf{Err}^\bullet)$ and we're done.
Otherwise, $(\Sigma', \mathsf{assert}\, K\, e') \longrightarrow_T (\Sigma', e')$ and $K \sim \mathsf{pointsto}(\Sigma', e')$.
By Lemma 4.57, we therefore get $(k - j - 1, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau \sqcap K]\!]$, which is sufficient to complete the proof. □

Lemma 4.59 (Tagmatch Makes Values In history relation At Meet). *If* $K \sim \mathsf{pointsto}(\Sigma, \ell)$ *and* $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau, \overline{\tau}]\!]$ *then* $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![K \sqcap \tau, \overline{\tau}]\!]$

Proof. There are three cases to consider:

(1) $K \sqcap \tau = \bot$: a contradiction because $K \sim \Sigma(\ell)$ and $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$.
(2) $K \sqcap \tau = \tau$: immediate by Lemma 4.51.
(3) $K \sqcap \tau = K$ and $\tau = *$: immediate by unfolding the erroring value relation in our hypothesis, and noting that whichever type of Int, $* \times *$ or $* \to *$ we satisfy must be K.

□

Lemma 4.60 (Check Makes Terms In history relation At Meet). *If* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\tau, \overline{\tau}]\!]$ *then* $(k, \Psi, \Sigma, \mathsf{assert}\, K\, e) \in \mathcal{EH}^T[\![\tau \sqcap K, \overline{\tau}]\!]$.

Proof. Unfolding the erroring expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and (Σ', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done.
Otherwise $\exists (k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{VH}^V[\![T]\!]\tau, \overline{\tau}$.
It suffices to show $(k - j, \Psi', \Sigma', \mathsf{assert}\, K\, e') \in \mathcal{EH}^T[\![\tau \sqcap K, \overline{\tau}]\!]$.
By the OS, if $\neg K \sim \mathsf{pointsto}(\Sigma', e')$ then $(\Sigma', \mathsf{assert}\, K\, e') \longrightarrow_T (\Sigma', \mathsf{Err}^\bullet)$ and we're done.
Otherwise, $(\Sigma', \mathsf{assert}\, K\, e') \longrightarrow_T (\Sigma', e')$ and $K \sim \mathsf{pointsto}(\Sigma', e')$.
By Lemma 4.59, we therefore get $(k - j - 1, \Psi', \Sigma', e') \in \mathcal{VH}^V[\![T]\!]\tau \sqcap K, \overline{\tau}$, which is sufficient to complete the proof. □

Lemma 4.61 (Lattice Ordering Preserves Relation). *If* $\tau \leq \tau'$ *then*

(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau']\!]$
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau']\!]$.

Proof. (1) Unfolding the expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and (Σ', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done.

Otherwise $\exists(k-j,\Psi') \sqsupseteq (k,\Psi)$ such that $\Sigma' : (k-j,\Psi')$ and $(k-j,\Psi',\Sigma',e') \in \mathcal{V}^T[\![\tau]\!]$.
It suffices to show $(k-j,\Psi',\Sigma',e') \in \mathcal{V}^T[\![\tau']\!]$, which follows by IH 2).

(2) Proceed by induction over the lattice ordering:
   (a) $\tau \lesssim \tau'$: follows from Lemma 4.50.
   (b) $\tau = \tau_1 \times \tau_2$, $\tau' = \tau_1' \times \tau_2'$, $\tau_1 \leq \tau_1'$, and $\tau_2 \leq \tau_2'$:
   Then unfolding the location relation in our hypothesis, we have that $\Sigma(\ell) = (\langle \ell_1, \ell_2\rangle, \_)$.
   We also have that $(k,\Psi,\Sigma,\ell_1) \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k,\Psi,\Sigma,\ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$.
   Unfolding the relation in what we want to show, we want to show $(k,\Psi,\Sigma,\ell_1) \in \mathcal{V}^T[\![\tau_2]\!]$ and $(k,\Psi,\Sigma,\ell_2) \in \mathcal{V}^T[\![\tau_2']\!]$, which follows by IH 2).
   (c) $\tau = * \to \tau_o$, $\tau' = * \to \tau_o'$, and $\tau_o \leq \tau_o'$:
   We want to show $(k,\Psi,\Sigma,\ell) \in \mathcal{V}^T[\![* \to \tau_o']\!]$.
   Let $(j,\Psi') \sqsupseteq (k,\Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j,\Psi')$.
   Let $\ell_v \in dom(\Sigma')$ such that $(j,\Psi',\Sigma',\ell_v) \in \mathcal{V}^T[\![*]\!]$.
   Let K.
   We want to show $(j,\Psi',\Sigma',\mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_o' \sqcap K]\!]$.
   From our hypothesis, we get that $(j,\Psi',\Sigma',\mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_o \sqcap K]\!]$.
   The proof follows from IH 1).
   (d) $\tau' = *$: Proceed by case analysis on τ:
      (i) $\tau = \mathsf{Nat}$: Immediate.
      (ii) $\tau = \mathsf{Int}$: Immediate.
      (iii) $\tau = \mathsf{Bool}$: Immediate.
      (iv) $\tau = \tau_1 \times \tau_2$: Then unfolding the location relation in our hypothesis, we have that $\Sigma(\ell) = (\langle \ell_1, \ell_2\rangle, \_)$.
      We also have that $(k,\Psi,\Sigma,\ell_1) \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k,\Psi,\Sigma,\ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$.
      Unfolding the relation in what we want to show, we want to show $(k-1,\Psi,\Sigma,\ell_1) \in \mathcal{V}^T[\![*]\!]$ and $(k-1,\Psi,\Sigma,\ell_2) \in \mathcal{V}^T[\![*]\!]$, which follows by IH 2) and Lemma 4.55.
      (v) $\tau = * \to \tau'$: We want to show $(k,\Psi,\Sigma,\ell) \in \mathcal{V}^T[\![* \to *]\!]$.
      Let $(j,\Psi') \sqsupseteq (k,\Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j,\Psi')$.
      Let $\ell_v \in dom(\Sigma')$ such that $(j,\Psi',\Sigma',\ell_v) \in \mathcal{V}^T[\![*]\!]$.
      Let K.
      We want to show $(j,\Psi',\Sigma',\mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![K]\!]$.
      From our hypothesis, we get that $(j,\Psi',\Sigma',\mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau' \sqcap K]\!]$.
      By the IH 1), we get that $(j,\Psi',\Sigma',\mathsf{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![K]\!]$ which is what we wanted to show.

□

Lemma 4.62 (Pairs of Semantically Well Typed Terms are Semantically Well Typed). *If* $(k,\Psi,\Sigma,e_1) \in \mathcal{E}^T[\![\tau_1]\!]$ *and* $(k,\Psi,\Sigma,e_2) \in \mathcal{E}^T[\![\tau_2]\!]$ *then* $(k,\Psi,\Sigma,\langle e_1,e_2\rangle) \in \mathcal{E}^T[\![\tau_1 \times \tau_2]\!]$.

Proof. Unfolding the expression relation in our hypothesis about e_1, we get that there are $(\Sigma, e_1'), j$ such that $(\Sigma, e_1) \longrightarrow_T^j (\Sigma, e_1')$ and (Σ', e_1') is irreducible.
If $e_1' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k-j,\Psi') \sqsupseteq (k,\Psi)$ such that $\Sigma' : (k-j,\Psi)$ and $(k-j,\Psi',\Sigma',e_1') \in \mathcal{V}^T[\![\tau_1]\!]$.
This means $e_1' = \ell_1$ for some $\ell_1 \in dom(\Sigma')$.

With this and by the OS, we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_T^j (\Sigma', \langle loc_1, e_2\rangle)$.

We can apply Lemma 4.55 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{E}^T[\![\tau_2]\!]$.
Unfolding the expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \longrightarrow_T^{j'} (\Sigma', e_2')$ and (Σ'', e_2') is irreducible.
If $e_2' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in \mathcal{V}^T[\![\tau_2]\!]$, which means $e_2' = \ell_2$ for some $\ell_2 \in dom(\Sigma'')$.

Putting everything together we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_T^{j'} (\Sigma'', \langle \ell_1, \ell_2\rangle)$, with $\Sigma'' : (k - j - j', \Psi'')$.
Note by OS, $(\Sigma'', \langle \ell_1, \ell_2\rangle) \longrightarrow_T (\Sigma''[\ell' \mapsto \langle \ell_1, \ell_2\rangle])$ where $\ell' \notin dom(\Sigma'')$.

We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)] : (k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
Note the only interesting part of this statement is that $\forall k' < k - j - j' - 1.\ (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \ell') \in \mathcal{VH}^T[\![\Psi''(\ell_1) \times \Psi''(\ell_2)]\!]$.
This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in \mathcal{VH}^T[\![\Psi''(\ell_1)]\!]$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^T[\![\Psi''(\ell_2)]\!]$.

We know that $(k - j, \Psi', \Sigma', \ell_1') \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$, and Lemma 4.55 with downward closure and the store typing judgement above.
From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \ell_1) \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2\rangle], \ell_2) \in \mathcal{V}^T[\![\tau_2]\!]$.
This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \langle \ell_1, \ell_2\rangle) \in \mathcal{V}^T[\![\tau_1 \times \tau_2]\!]$, which is what we wanted to prove. □

Lemma 4.63 (Pairs of Related Terms are Related). *If $(k, \Psi, \Sigma, e_1) \in \mathcal{EH}^T[\![fst(\overline{\tau})]\!]$ and $(k, \Psi, \Sigma, e_2) \in \mathcal{EH}^T[\![snd(\overline{\tau})]\!]$ then $(k, \Psi, \Sigma, \langle e_1, e_2\rangle) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$.*

Proof. Unfolding the erroring expression relation in our hypothesis about e_1, we get that there are $(\Sigma, e_1'), j$ such that $(\Sigma, e_1) \longrightarrow_T^j (\Sigma, e_1')$ and (Σ', e_1') is irreducible.
If $e_1' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi)$ and $(k - j, \Psi', \Sigma', e_1') \in \mathcal{VH}^T[\![fst(\overline{\tau})]\!]$.
This means $e_1' = \ell_1$ for some $\ell_1 \in dom(\Sigma')$.

With this and by the OS, we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_T^j (\Sigma', \langle loc_1, e_2\rangle)$.

We can apply Lemma 4.54 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{EH}^T[\![snd(\overline{\tau})]\!]$.
Unfolding the erroring expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \longrightarrow_T^{j'} (\Sigma', e_2')$ and (Σ'', e_2') is irreducible.
If $e_2' = \mathsf{Err}^\bullet$, then were done because the entire application steps to an error.
Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in$

$\mathcal{VH}^T[\![snd(\overline{\tau})]\!]$, which means $e_2' = \ell_2$ for some $\ell_2 \in dom(\Sigma'')$.

Putting everything together we get $(\Sigma, \langle e_1, e_2\rangle) \longrightarrow_T^{j'} (\Sigma'', \langle \ell_1, \ell_2\rangle)$, with $\Sigma'' : (k-j-j', \Psi'')$.
Note by OS, $(\Sigma'', \langle \ell_1, \ell_2\rangle) \longrightarrow_T (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)])$ where $\ell' \notin dom(\Sigma'')$.

We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)] : (k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
Note the only interesting part of this statement is that $\forall k' < k-j-j'-1.\ (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \ell') \in \mathcal{VH}^T[\![\Psi''(\ell_1) \times \Psi''(\ell_2)]\!]$.
This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in \mathcal{VH}^T[\![\Psi''(\ell_1)]\!]$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^T[\![\Psi''(\ell_2)]\!]$.

We know that $(k-j, \Psi', \Sigma', \ell_1') \in \mathcal{VH}^T[\![fst(\overline{\tau})]\!]$ and $(k-j-j', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^T[\![snd(\overline{\tau})]\!]$, and Lemma 4.51 with downward closure and the store typing judgement above.
From these facts we get that $(k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \ell_1) \in \mathcal{VH}^T[\![fst(\overline{\tau})]\!]$ and $(k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2\rangle], \ell_2) \in \mathcal{VH}^T[\![snd(\overline{\tau})]\!]$.
This is sufficient to show $(k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2\rangle, \_)], \langle \ell_1, \ell_2\rangle) \in \mathcal{VH}^T[\![\overline{\tau}]\!]$, which is what we wanted to prove. □

Lemma 4.64 (Applications of Semantically Well Typed Terms are Semantically Well Typed). *If* $(k, \Psi, \Sigma, e_f) \in \mathcal{E}^T[\![* \to \tau]\!]$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![*]\!]$ *then* $\forall K,\ (k, \Psi, \Sigma, \mathsf{app}\{K\}\, e_f\ e) \in \mathcal{E}^T[\![\tau \sqcap K]\!]$.

Proof. Unfolding the expression relation in our hypothesis about e_f, we get that there are (Σ', e_f'), j such that $(\Sigma, e_f) \longrightarrow_T^{j} (\Sigma', e_f')$ and (Σ', e_f') is irreducible.
If $e_f' = \mathsf{Err}^\bullet$, then we're done because the entire application steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e_f') \in \mathcal{V}^T[\![* \to \tau]\!]$.
This means $e_f' = \ell_f$ for some $\ell_f \in dom(\Sigma')$.

Using this, we know from the OS that $(\Sigma, \mathsf{app}\{K\}\, e_f\ e) \longrightarrow_T^{j} (\Sigma', \mathsf{app}\{K\}\, \ell_f\ e)$.

We can apply Lemma 4.55 with $\Sigma' : (k-j, \Psi')$ to our hypothesis about e to get $(k-j, \Psi', \Sigma', e) \in \mathcal{E}^T[\![*]\!]$.
Unfolding the expression relation, we get that there are (Σ'', e'), j' such that $(\Sigma', e) \longrightarrow_T^{j'} (\Sigma'', e')$ where (Σ'', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ than we're done, because the whole application errors.
Otherwise, there exists $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j'.\Psi'', \Sigma'', e') \in \mathcal{V}^T[\![*]\!]$.
This means $e' = \ell$ for some $\ell \in dom(\Sigma'')$.

Putting what we have together, by the OS, $(\Sigma, \mathsf{app}\{K\}\, e_f\ e) \longrightarrow_T^{j+j'} (\Sigma'', (\mathsf{app}\{K\}\, \ell_f\ \ell))$.
We have $(k-j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^T[\![* \to \tau]\!]$ and $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ and $\Sigma'' : (k-j-j', \Psi'')$.
We can combine these to get $(k-j-j', \Psi'', \Sigma'', \mathsf{app}\{K\}\, \ell_f\ \ell) \in \mathcal{E}^T[\![\tau \sqcap K]\!]$.
This is sufficient to complete the proof. □

Corollary 4.65. *If* $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^T[\![*]\!]$ *and* $\Sigma(\ell) = w$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![*]\!]$ *then* $(k-1, \Psi, \Sigma, \mathsf{app}\{*\}\ w\ e) \in \mathcal{E}^T[\![*]\!]$.

Lemma 4.66 (Applications of Related Terms are Related). *If* $(k, \Psi, \Sigma, e_f) \in \mathcal{EH}^T[\![\tau, \overline{\tau}]\!]$ *and* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![*]\!]$ *then* $\forall K$, $(k, \Psi, \Sigma, \mathsf{app}\{K\}\ e_f\ e) \in \mathcal{EH}^T[\![cod(\tau) \sqcap K, cod(\overline{\tau})]\!]$.

Proof. Unfolding the erroring expression relation in our hypothesis about e_f, we get that there are (Σ', e'_f), j such that $(\Sigma, e_f) \longrightarrow^j_T (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.
If $e'_f = \mathsf{Err}^\bullet$, then we're done because the entire application steps to an error.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e'_f) \in \mathcal{VH}^T[\![\tau, \overline{\tau}]\!]$.
This means $e'_f = \ell_f$ for some $\ell_f \in dom(\Sigma')$.

Using this, we know from the OS that $(\Sigma, \mathsf{app}\{K\}\ e_f\ e) \longrightarrow^j_T (\Sigma', \mathsf{app}\{K\}\ \ell_f\ e)$.

We can apply Lemma 4.55 with $\Sigma' : (k-j, \Psi')$ to our hypothesis about e to get $(k-j, \Psi', \Sigma', e) \in \mathcal{E}^T[\![*]\!]$.
Unfolding the expression relation, we get that there are (Σ'', e'), j' such that $(\Sigma', e) \longrightarrow^{j'}_T (\Sigma'', e')$ where (Σ'', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ than we're done, because the whole application errors.
Otherwise, there exists $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j'.\Psi'', \Sigma'', e') \in \mathcal{V}^T[\![*]\!]$.
This means $e' = \ell$ for some $\ell \in dom(\Sigma'')$.

Putting what we have together, by the OS, $(\Sigma, \mathsf{app}\{K\}\ e_f\ e) \longrightarrow^{j+j'}_T (\Sigma'', (\mathsf{app}\{K\}\ \ell_f\ \ell))$.
We have $(k-j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^T[\![* \to \tau]\!]$ and $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ and $\Sigma'' : (k-j-j', \Psi'')$.
We can combine these to get $(k-j-j', \Psi'', \Sigma'', \mathsf{app}\{K\}\ \ell_f\ \ell) \in \mathcal{EH}^T[\![cod(\tau) \sqcap K, cod(\overline{\tau})]\!]$.
This is sufficient to complete the proof. ◻

Corollary 4.67. *If* $(k, \Psi, \Sigma, e_f) \in \mathcal{EH}^T[\![*, \overline{\tau}]\!]$ *and* $(k-1, \Psi, \Sigma, e) \in \mathcal{E}^T[\![*]\!]$ *then* $(k-1, \Psi, \Sigma, \mathsf{app}\{\tau_0\}\ e_f\ e) \in \mathcal{EH}^T[\![*, cod(\overline{\tau})]\!]$.

Lemma 4.68 (Dynamic Checks Are Noops).

(1) If $(k+1, \Psi, \Sigma, \mathsf{assert}\ *\ e) \in \mathcal{E}^T[\![\tau]\!]$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$.
(2) If $(k+1, \Psi, \Sigma, \mathsf{assert}\ *\ e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$ *then* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$.

Proof.
(1) assume there is Σ', e', j such that $(\Sigma, e) \longrightarrow^j_T (\Sigma', e')$ where (Σ', e') is irreducible.
By the OS, we get that $(\Sigma, \mathsf{assert}\ *\ e) \longrightarrow^j_T (\Sigma', \mathsf{assert}\ *\ e')$.
Then by OS, we have $(\Sigma', \mathsf{assert}\ *\ e') \longrightarrow^j_T (\Sigma', e')$.
Therefore, we can apply our hypothesis to complete the proof.
(2) Same as previous case, just using the history relation.

◻

Lemma 4.69 (Monitor Compatibility). *If* $\Sigma : (k, \Psi)$, *then*

(1) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ *and* $\Sigma(\ell') = (\ell, \mathsf{some}(K', K))$, *then* $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^T[\![K' \sqcap K \sqcap \tau]\!]$
(2) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau \sqcap K \sqcap K']\!]$ *then* $(k, \Psi, \Sigma, \mathsf{mon}\ \{K' \Leftarrow K\}\ e) \in \mathcal{E}^T[\![\tau \sqcap K \sqcap K']\!]$.
(3) If $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\Psi(\ell)]\!]$ *and* $\Sigma' = \Sigma[\ell' \mapsto (\ell, \mathsf{some}(K', K))]$ *and* $\Psi' = [\ell' \mapsto K', K, \Psi(\ell)]\Psi$ *and* $\ell' \notin dom(\Sigma)$ *and* $\vdash \Sigma'$ *then* $(k, \Psi', \Sigma', \ell') \in \mathcal{VH}^T[\![K', K, \Psi(\ell)]\!]$

(4) *If* $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\overline{\tau}]\!]$ *then* $(k, \Psi, \Sigma, \mathsf{mon}\,\{* \Leftarrow *\}\, e) \in \mathcal{EH}^T[\![*, *, \overline{\tau}]\!]$

Proof. Proceed by simultaneous induction on k and τ.

- $k = 0$: 2) and 4) follow from 1) and 3) respectively.
  The proofs follow similarly to the other case, but any function or dynamic cases are vacuously true.
- $k > 0$:
  1) Unfolding the relation in the statement we want to prove, note from our hypothesis about Σ, we get that $\vdash \Sigma$.
     Proceed by case analysis on $\tau \sqcap K \sqcap K'$:
     i) $\tau = \tau \sqcap K \sqcap K'$: Immediate.

     ii) $\tau \sqcap K \sqcap K' = \bot$: then either K or K' is $\bot$, which is a contradiction since they both tagmatch $\mathsf{pointsto}(\Sigma, \ell)$.

     iii) $\tau \sqcap K \sqcap K' \leqslant: \tau$: then $\tau = \mathsf{Int}$ and K or $K' = \mathsf{Nat}$.
     Immediate because by $\vdash \Sigma$, $\mathsf{Nat} \sim \mathsf{pointsto}(\Sigma, \ell)$.
     iv) $\tau \sqcap K \sqcap K' \neq \tau$: then it must be the case that $\tau = *$ and K or $K' = * \to *$.
     Note K or K' cannot be $* \times *$, by $\vdash \Sigma$.
     Unfolding the relation in our hypothesis, we have that $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![* \to *]\!]$.
     We want to show that $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^T[\![* \to *]\!]$.
     Unfolding the relation, let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
     Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
     Let K.
     We want to show $(j, \Psi', \Sigma', \mathsf{app}\{K\}\, \ell'\, \ell_v) \in \mathcal{E}^T[\![K]\!]$.
     By the OS, $(\Sigma', \mathsf{app}\{K\}\, \ell'\, \ell_v) \longrightarrow^2_T (\Sigma', \mathsf{assert}\, K\, (\mathsf{mon}\,\{* \Leftarrow *\}\, (\ell\, (\mathsf{mon}\,\{* \Leftarrow *\}\, \ell_v))))$.
     By IH 2), we have $(j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow *\}\, \ell_v) \in \mathcal{E}^T[\![*]\!]$.
     By Lemma 4.64, we have that $(j, \Psi', \Sigma', \mathsf{app}\{K\}\, \ell\, (\mathsf{mon}\,\{* \Leftarrow *\}\, \ell_v)) \in \mathcal{E}^T[\![K]\!]$.
     Then by IH 2), we have $(j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow *\}\, (\mathsf{app}\{K\}\, \ell\, (\mathsf{mon}\,\{* \Leftarrow *\}\, \ell_v))) \in \mathcal{E}^T[\![K]\!]$.
     Note that $(j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow *\}\, (\mathsf{app}\{K\}\, \ell\, (\mathsf{mon}\,\{* \Leftarrow *\}\, \ell_v))) \in \mathcal{E}^T[\![K]\!]$ iff $(j, \Psi', \Sigma', \mathsf{assert}\, K\, (\mathsf{mon}\,\{* \Leftarrow *\}\, (\ell\, (\mathsf{mon}$
     $\mathcal{E}^T[\![K]\!]$.
     Therefore, this is sufficient to complete the case.
  2) Unfolding the expression relation in our hypothesis, we have that there are $(e', \Sigma'), j$ such that $(e, \Sigma) \longrightarrow^j_T (e', \Sigma')$ with (e', Σ') irreducible.
     If $e' = \mathsf{Err}^\bullet$ then we're done, because the monitor will step to an error as well.
     Otherwise, there is $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T[\![\tau \sqcap K \sqcap K']\!]$.
     This means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.

     We want to show $(k-j, \Psi', \Sigma', \mathsf{mon}\,\{K \Leftarrow \ell\} \in \mathcal{E}^T[\![\tau \sqcap K \sqcap K']\!]$.
     We destruct on whether $\Sigma'(\ell)$ is a pair.
     If $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$, then by the OS, $(\Sigma', \mathsf{mon}\,\{K \Leftarrow \ell\} \longrightarrow_T (\Sigma', \langle \mathsf{mon}\,\{* \Leftarrow *\}\, \ell_1, \mathsf{mon}\,\{* \Leftarrow *\}\, \ell_2 \rangle)$.
     Then by Lemma 4.62, it suffices to show $(k-j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow \ell_1\} \in \mathcal{E}^T[\![\mathit{fst}(\tau)]\!]$ and $(k-j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow \ell_2\} \in \mathcal{E}^T[\![\mathit{snd}(\tau)]\!]$

These both follow from IH 2) (smaller by index).

Otherwise, by the OS, $(\Sigma', \mathsf{mon}\,\{K \Leftarrow \ell\} \longrightarrow_T (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(K', K))], \ell')$.

Then by IH 3), we get $\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(K', K))] : (k - j - 1, \Psi'[\ell' \mapsto K', K, \Psi'(\ell)])$.

And by IH 1), we get $(k - j - 1, \Psi'[\ell' \mapsto K', K, \Psi'(\ell)], \Sigma'[\ell' \mapsto (\ell, \mathsf{some}(K', K))], \ell') \in \mathcal{V}^T[\![\tau \sqcap K \sqcap K']\!]$.

These two facts are sufficient to complete the case.

3) We proceed by case analysis on K' (note by the fact that $\vdash \Sigma', K \sim K'$):

(a) $K' = \mathsf{Nat}$: Since we already know $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^V[\![N]\!]\Psi(\ell)$, it suffices to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![K']\!]$ and $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N[\![K]\!]$.

This is immediate from $\vdash \Sigma'$, which implies $K' \sim \mathsf{pointsto}(\Sigma', \ell')$ and $K \sim \mathsf{pointsto}(\Sigma', \ell')$.

(b) $K' = \mathsf{Int}$: same as the Nat case.

(c) $K' = \mathsf{Bool}$: same as the Nat case.

(d) $K' = * \times *$: this case is a contradiction by the fact that $\vdash \Sigma$.

(e) $K' = * \to *$: Since $\mathsf{pointsto}(\Sigma, \ell) \sim K'$ and $\mathsf{pointsto}(\Sigma, \ell) \sim K$, $K = *$ or $* \to *$.

Also, since $\vdash \Sigma'$, we get that $\Psi(\ell) = [*, \overline{\tau'}]$ or $[* \to *, \overline{\tau'}]$.

From the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\Psi(\ell)]\!]$, we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![*, \overline{\tau'}]\!]$ or $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![* \to *, \overline{\tau'}]\!]$.

In the case of $*$, we can unfold and get $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![* \to *, \overline{\tau'}]\!]$.

Otherwise we can get the same using Lemma 4.51.

Similarly, we want to show that $(k, \Psi', \Sigma', \ell') \in \mathcal{VH}^T[\![K', K, \Psi(\ell)]\!]$.

By Lemma 4.51, in the $K' = *$ case, it suffices to show $(k, \Psi', \Sigma', \ell') \in \mathcal{VH}^T[\![* \to *, K, \Psi(\ell)]\!]$.

So let $(j, \Psi'') \sqsupseteq (k, \Psi')$, and let $\Sigma'' \supseteq \Sigma'$ such that $\Sigma'' : (j, \Psi'')$.

Let $\ell_v \in dom(\Sigma'')$ such that $(j, \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^T[\![*]\!]$.

Let K''.

We want to show $(j, \Psi'', \Sigma'', \mathsf{app}\{K''\}\,\ell'\,\ell_v) \in \mathcal{EH}^T[\![K'', *, cod(\Psi(\ell))]\!]$.

By the OS, $(\Sigma'', \mathsf{app}\{K''\}\,\ell'\,\ell_v) \longrightarrow_T (\Sigma'', \mathsf{assert}\,K''\,(\ell'\,\ell_v))$.

By Lemma 4.60, it suffices to show $(j - 1, \Psi'', \Sigma'', \ell'\,\ell_v) \in \mathcal{EH}^T[\![*, *, cod(\Psi(\ell))]\!]$.

By the OS, $(\Sigma'', \ell'\,\ell_v) \longrightarrow_T (\Sigma'', \mathsf{mon}\,\{* \Leftarrow *\}\,(\ell\,(\mathsf{mon}\,\{* \Leftarrow *\}\,\ell_v)))$.

By IH 2) (smaller by index), it suffices to show $(j-2, \Psi'', \Sigma'', \ell\,(\mathsf{mon}\,\{* \Leftarrow *\}\,\ell_v)) \in \mathcal{EH}^T[\![*, *, cod(\Psi(\ell))]\!]$.

By Lemma 4.68, it suffices to show $(j-1, \Psi'', \Sigma'', \mathsf{assert}\,*\,\ell\,(\mathsf{mon}\,\{* \Leftarrow *\}\,\ell_v)) \in \mathcal{EH}^T[\![*, *, cod(\Psi(\ell))]\!]$.

Then by the OS, it suffices to show $(j, \Psi'', \Sigma'', \mathsf{app}\{*\}\,\ell\,(\mathsf{mon}\,\{* \Leftarrow *\}\,\ell_v)) \in \mathcal{EH}^T[\![*, *, cod(\Psi(\ell))]\!]$.

By IH 2), $(j, \Psi'', \Sigma'', \mathsf{mon}\,\{* \Leftarrow *\}\,\ell_v) \in \mathcal{V}^T[\![*]\!]$.

Unfolding, we get that there exists some j', e'', Σ''' such that $(\Sigma'', \mathsf{mon}\,\{* \Leftarrow *\}) \longrightarrow_T^{j'} (\Sigma''', e')$.

If $e' = \mathsf{Err}^\bullet$, then we're done because the entire application errors.

Otherwise, we get that there exists a $(j - j', \Psi''') \sqsupseteq (j, \Psi'')$ such that $\Sigma''' : (j - j', \Psi''')$ and $(j - j', \Psi''', \Sigma''', e'') \in \mathcal{V}^T[\![*]\!]$.

Note by the operational semantics, $j' \geq 1$.

By Lemma 4.51, we get $(j - j', \Psi''', \Sigma''', \ell) \in \mathcal{VH}^T[\![* \to *, \overline{\tau'}]\!]$.

Finally we can apply this hypothesis to the fact about e'' to get that $(j - j', \Psi''', \Sigma''', \mathsf{app}\{*\}\,\ell\,e'') \in \mathcal{EH}^T[\![*, *, cod(\Psi(\ell))]\!]$, which is sufficient to complete the case.

(f) $K' = *$: unfolding the relation in what we want to show, the proof follows by IH 3) (smaller by index).

4) Unfolding the expression relation in our hypothesis, we have that there are (e', Σ'), j such that $(e, \Sigma) \longrightarrow^j_T (e', \Sigma')$ with (e', Σ') irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done, because the monitor will step to an error as well.
Otherwise, there is $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{VH}^T[\![\overline{\tau}]\!]$.
This means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.

We want to show $(k - j, \Psi', \Sigma', \mathsf{mon}\,\{* \Leftarrow *\}\,\ell) \in \mathcal{EH}^T[\![*, *, \Psi'(\ell)]\!]$.
For ii), by OS, if $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$, then $(\Sigma', \mathsf{mon}\,\{* \Leftarrow \ell\} \longrightarrow_T (\Sigma', \langle \mathsf{mon}\,\{* \Leftarrow *\}\,\ell_1, \mathsf{mon}\,\{* \Leftarrow *\}\,\ell_2 \rangle)$.
Then by Lemma 4.63, it suffices to show $(k - j - j' - 1, \Psi, \Sigma, \mathsf{mon}\,\{* \Leftarrow \ell_1\} \in \mathcal{VH}^T[\![*, *, \tau]\!]$ and $(k - j - j' - 1, \Psi, \Sigma, \mathsf{mon}\,\{* \Leftarrow \ell_2\} \in \mathcal{VH}^T[\![*, *, \tau]\!]$.
Both of these follow from (4) (smaller by index).

Otherwise, by the OS, $(\Sigma', \mathsf{mon}\,\{* \Leftarrow *\}\,) \longrightarrow_T (\Sigma'[\ell' \mapsto (\ell, \mathsf{some}(*, *))], \ell')$.
We can finish the proof by applying IH 3) (smaller by index).

□

Lemma 4.70 (Expression Relation implies Erroring Expression Relation). *(1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\![\tau]\!]$ then $(k, \Psi, \Sigma, e) \in \mathcal{EH}^T[\![\tau]\!]$.*
(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T[\![\tau]\!]$.

Proof. Proceed by induction on k and τ:

- $k = 0$: 1) is immediate from 2).
  - $\tau = \mathsf{Int}$: immediate.
  - $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
    The case follows from the IH on ℓ_1 and ℓ_2.
  - $\tau = \tau_1 \to \tau_2$: vacuously true.
  - $\tau = *$: vacuously true.
- $k > 0$: 1) is immediate from 2).
  - $\tau = \mathsf{Int}$: immediate.
  - $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, \_)$.
    The case follows from the IH on ℓ_1 and ℓ_2.
  - $\tau = \tau_1 \to \tau_2$: Follows from 1) from the IH (smaller by index).
  - $\tau = *$: Follows from 2) from the IH (smaller by index), using $* \times *$, $* \to *$, or Int.

□

4.4.3 Compatability Lemmas

Lemma 4.71 (**T-Var** compatibility). $\dfrac{(x_0 : K_0) \in \Gamma}{\Gamma \vdash x_0 : K_0}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(x)) \in \mathcal{E}^T[\![\tau]\!]$.
Since $x : \tau \in \Gamma$, we get that $\gamma(x) = \ell$.
Since $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$, we get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\tau]\!]$.
Then we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^T[\![\tau]\!]$ immediately since ℓ is already a value and we have as a premise that $\Sigma : (k, \Psi)$. □

Lemma 4.72 (**T-Nat** compatibility). $\dfrac{}{[\![\Gamma \vdash n_0 : \mathsf{Nat}]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(n)) \in \mathcal{E}^T[\![\mathsf{Nat}]\!]$.
Note $\gamma(n) = n$.
By the OS, we have $(\Sigma, n) \longrightarrow_T (\Sigma[\ell \mapsto (n, \mathsf{none})], \ell)$.
We get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\![\mathsf{Nat}]\!]$ immediately because $n \in \mathbb{T}$.
Since $\mathcal{V}^T[\![\mathsf{Nat}]\!]$ does not rely on Ψ or Σ, we have that $(k, \Psi[\ell \mapsto [\mathsf{Nat}]], \Sigma[\ell \mapsto (n, \mathsf{none})], \ell) \in \mathcal{V}^T[\![\mathsf{Nat}]\!]$.
Since $\ell \mapsto \mathsf{Nat}$, we have that $(k, \Psi[\ell \mapsto [\mathsf{Nat}]], \Sigma[\ell \mapsto (n, \mathsf{none})], \ell) \in \mathcal{V}^T[\![\mathsf{Nat}]\!]$.
Similarly we have $(k, \Psi[\ell \mapsto [\mathsf{Nat}]], \Sigma[\ell \mapsto (n, \mathsf{none})], \ell) \in \mathcal{VH}^V[\![T]\!]\mathsf{Nat}$.
Therefore, given we know $\Sigma : (k, \Psi)$, we know $\Sigma[\ell \mapsto (n, \mathsf{none})] : (k, \Psi[\ell \mapsto [\mathsf{Nat}]])$. □

Lemma 4.73 (**T-Int** compatibility). $\dfrac{}{[\![\Gamma \vdash i_0 : \mathsf{Int}]\!]}$

Proof. Not meaningfully different from **T-Nat** □

Lemma 4.74 (**T-True** compatibility). $\dfrac{}{[\![\Gamma \vdash \mathsf{True} : \mathsf{Bool}]\!]}$

Proof. Not meaningfully different from **T-Nat** □

Lemma 4.75 (**T-False** compatibility). $\dfrac{}{[\![\Gamma \vdash \mathsf{False} : \mathsf{Bool}]\!]}$

Proof. Not meaningfully different from **T-Nat** □

Lemma 4.76 (**T-Lam** compatibility). $\dfrac{[\![\Gamma_0,\ (x_0 : K_0) \vdash e_0 : \tau_1]\!]}{[\![\Gamma_0 \vdash \lambda(x_0 : K_0).\, e_0 : * \to \tau_1]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\lambda x_1 : K.\, e_1)) \in \mathcal{E}^T[\![* \to \tau_1]\!]$.
Note that $\gamma(\lambda x_1 : K.\, e_1) = \lambda x_1 : K.\, \gamma(e_1)$.
Since $\lambda x_1 : K.\, \gamma(e_1)$ is a value, by the OS we have $(\Sigma, \lambda x_1 : K.\, \gamma(e_1)) \longrightarrow_T (\Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \mathsf{none})])$, where $\ell \notin dom(\Sigma)$.
We choose our later Ψ' to be $\Psi[\ell \mapsto * \to *]$.
We now have two obligations:

(1) $(k-1, \Psi[\ell \mapsto * \to *], \Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \mathsf{none})], \ell) \in \mathcal{V}^T[\![* \to \tau_1]\!]$
(2) $\Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \mathsf{none})] : (k-1, \Psi[\ell \mapsto * \to *])$

For 1), we want to show $(k-1, \Psi[\ell \mapsto * \to *], \Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \text{none})], \lambda x_1 : K.\, \gamma(e_1)) \in \mathcal{V}^T[\![* \to \tau_1]\!]$.
Unfolding the value relation:
Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto * \to *])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \text{none})]$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
Let K.
We want to show $(j, \Psi', \Sigma', \text{app}\{K\}\ \ell\ \ell_v) \in \mathcal{E}^T[\![\tau_1 \sqcap K]\!]$.
By the OS, if $\neg K \sim \Sigma(\ell_v)$ then the application steps to an error and we're done.
Otherwise, $(\Sigma', \text{app}\{K\}\ \ell\ \ell_v) \longrightarrow_T (\Sigma', \text{assert}\, K\, \gamma(e_1)[\ell_v/x])$.
By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.
Note that $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^T[\![\Gamma, x : K]\!]$:

i) $(j-2, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![K]\!]$ by Lemma 4.55 and Lemma 4.57.
ii) $\forall y \in dom(\gamma)$, $(j-2, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^T[\![\Gamma(y)]\!]$ by the premise about γ and Lemma 4.55.

Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v]$, Ψ', Σ', and e_1 at $j-2$ to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T[\![\tau_1]\!]$.
Finally, we can apply Lemma 4.58 to get $(j-1, \Psi', \Sigma', \text{assert}\, K\, \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T[\![\tau_1 \sqcap K]\!]$ which is what we wanted to show.

For 2), first note the domains are equal, since $dom(\Sigma) = dom(\Psi)$.
Then note $\vdash \Sigma[\ell \mapsto (\lambda x_1 : K.\gamma(e_1), \text{none}]$ since $\vdash \Sigma$.
Then let $j < k-1$ and let $\ell' \in dom(\Sigma[\ell \mapsto (\lambda x_1 : K.\gamma(e_1), \text{none})])$.
If $\ell' \neq \ell$, then we get the remaining conditions from $\Sigma : (k, \Psi)$ and Lemma 4.51.
If $\ell' = \ell$, then note the structural obligation on $\Psi[\ell \mapsto [* \to *]]$ is immediate.
We want to show $(j, \Psi[\ell \mapsto * \to *], \Sigma[\ell \mapsto (\lambda x_1 : K.\gamma(e_1), \text{none})], \ell) \in \mathcal{VH}^T[\![* \to *]\!]$.
Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto * \to *])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : K.\, \gamma(e_1), \_)]$ such that $\Sigma' : (j, \Psi')$.
Let $\ell_v \in dom(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![*]\!]$.
Let K.
We get immediately that $\text{pointsto}(\Sigma', \ell_v) \sim *$, so we want to show $(j, \Psi', \Sigma', \text{app}\{K\}\ \ell\ \ell_v) \in \mathcal{EH}^V[\![* \sqcap K]\!]$.
By the OS, if $\neg K \sim \Sigma(\ell_v)$, then the application errors and we're done. Otherwise, $(\Sigma', \text{app}\{K\}\ \ell\ \ell_v) \longrightarrow_T (\Sigma', \text{assert}\, K\, \gamma(e_1)[\ell_v/x])$.
By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.
Note that $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^T[\![\Gamma, x : *]\!]$:

i) $(j-2, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T[\![K]\!]$ by Lemma 4.55 and Lemma 4.57.
ii) $\forall y \in dom(\gamma)$, $(j-2, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^T[\![\Gamma(y)]\!]$ by the premise about γ and Lemma 4.55.

Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v]$, Ψ', Σ', and e_1 at $j-2$ to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T[\![\tau_1]\!]$.
Then we can apply Lemma 4.70 to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{EH}^V[\![\tau_1]\!]$.
We can then apply Lemma 4.61 to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{EH}^V[\![*]\!]$.
Finally, we can apply Lemma 4.58 to get $(j-1, \Psi', \Sigma', \text{assert}\, K\, \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{EH}^V[\![* \sqcap K]\!]$ which is what we wanted to show.

□

Lemma 4.77 (**T-Pair** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket \qquad \llbracket \Gamma \vdash e_1 : \tau_1 \rrbracket}{\llbracket \Gamma \vdash \langle e_0, e_1 \rangle : \tau_0 \times \tau_1 \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T\llbracket\Gamma\rrbracket$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\langle e_1, e_2 \rangle)) \in \mathcal{E}^T\llbracket \tau_1 \times \tau_2 \rrbracket$.
Note $\gamma(\langle e_1, e_2 \rangle) = \langle \gamma(e_1), \gamma(e_2) \rangle$.
We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T\llbracket\tau_1\rrbracket$.
We can apply the second hypothesis to get $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T\llbracket\tau_2\rrbracket$.
Then by Lemma 4.63, $(k, \Psi, \Sigma, \langle \gamma(e_1), \gamma(e_2) \rangle) \in \mathcal{E}^T\llbracket \tau_1 \times \tau_2 \rrbracket$, which is what we wanted to show. □

Lemma 4.78 (**T-Cast** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket}{\llbracket \Gamma \vdash \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; e_0 : K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T\llbracket\Gamma\rrbracket$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; e_0)) \in \mathcal{E}^T\llbracket K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket$.
Note $\gamma(\mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; e_0) = \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; \gamma(e_0)$.
We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T\llbracket\tau_0\rrbracket$.
Unfolding the expression relation, there are j, Σ', e' such that $(\Sigma, \gamma(e_0)) \longrightarrow_T^j (\Sigma', e')$ where (Σ', e') is irreducible.
If $e' = \mathsf{Err}^\bullet$ then we're done, because the entire boundary expression errors.
Otherwise, we know there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T\llbracket\tau_0\rrbracket$.
This means $\exists \ell \in dom(\Sigma')$ such that $e' = \ell$.
By the OS, $(\Sigma, \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; \gamma(e_0)) \longrightarrow_T^j (\Sigma', \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\; \ell) \longrightarrow_T (\Sigma', \mathsf{mon}\,\{K_1 \Leftarrow K_0\}\; \ell)$.
By Lemma 4.55, $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T\llbracket\tau_0\rrbracket$.
By Lemma 4.69, $(k - j - 1, \Psi', \Sigma', \mathsf{mon}\,\{K_1 \Leftarrow K_0\}\; \ell) \in \mathcal{E}^T\llbracket K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket$, which is what we wanted to show. □

Lemma 4.79 (**T-App** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : * \rightarrow \tau_1 \rrbracket \qquad \llbracket \Gamma \vdash e_1 : \tau_0' \rrbracket}{\llbracket \Gamma \vdash \mathsf{app}\{K_1\}\; e_0\; e_1 : K_1 \sqcap \tau_1 \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T\llbracket\Gamma\rrbracket$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{app}\{K_1\}\; e_1\; e_2)) \in \mathcal{E}^T\llbracket K_1 \sqcap \tau_1 \rrbracket$.
Note $\gamma(\mathsf{app}\{K_1\}\; e_1\; e_2) = \mathsf{app}\{K_1\}\; \gamma(e_1)\; \gamma(e_2)$.
By the first hypothesis we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T\llbracket * \rightarrow \tau_1 \rrbracket$.
By the second hypothesis we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T\llbracket \tau_0' \rrbracket$.
By Lemma 4.61, we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T\llbracket * \rrbracket$.
Then we can apply Lemma 4.64 to get $(k, \Psi, \Sigma, \mathsf{app}\{K_1\}\; \gamma(e_1)\; \gamma(e_2)) \in \mathcal{E}^T\llbracket \tau_1 \sqcap K_1 \rrbracket$ which is what we wanted to show. □

Lemma 4.80 (**T-AppBot** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : \bot \rrbracket \qquad \llbracket \Gamma \vdash e_1 : \tau_0' \rrbracket}{\llbracket \Gamma \vdash \mathsf{app}\{K_1\}\; e_0\; e_1 : \bot \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{app}\{K_1\}\, e_0\, e_1)) \in \mathcal{E}^T[\![\bot]\!]$.
By Lemma 4.56, we have that $(\Sigma, e_0) \longrightarrow_T^* (\Sigma', e_0')$ where $e_0' = \mathsf{Err}^\bullet$, which is sufficient to complete the case. □

Lemma 4.81 (**T-Fst compatibility**). $\dfrac{[\![\Gamma \vdash e_0 : \tau_0 \times \tau_1]\!]}{[\![\Gamma \vdash \mathsf{fst}\{K_0\}\, e_0 : K_0 \sqcap \tau_0]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma_1]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{fst}\{K_0\}\, e_0)) \in \mathcal{E}^T[\![\tau_0 \sqcap K_0]\!]$.
Note $\gamma(\mathsf{fst}\{K_0\}\, e_1) = \mathsf{fst}\{K_0\}\, \gamma(e_0)$.
From the first hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T[\![\tau_0 \times \tau_1]\!]$.
Unfolding the expression relation, there are j, Σ', e_0' such that $(\Sigma, \gamma(e_0)) \longrightarrow_T^j (\Sigma'', e_0')$ and e_0' is irreducible.
If $e_0' = \mathsf{Err}^\bullet$ then we're done because the projection also steps to an error.
Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j\Psi')$ and $(k - j, \Psi', \Sigma', e_0') \in \mathcal{V}^T[\![\tau_0 \times \tau_1]\!]$.
Unfolding the location and value relations, we get that $\Sigma'(e_0') = (\langle \ell_0, \ell_1 \rangle, \_)$.
By the OS, $(\Sigma, \mathsf{fst}\{K_0\}\, e_0) \longrightarrow_N^j (\Sigma' \mathsf{fst}\{K_0\}\, e_0') \longrightarrow_T (\Sigma', \mathsf{assert}\, K_0\, \ell_0)$.
We can apply Lemma 4.55 to the premise that $(k - j, \Psi', \Sigma', \ell_0) \in \mathcal{V}^T[\![\tau_0]\!]$ to get $(k - j - 1, \Psi', \Sigma', \ell_0) \in \mathcal{V}^T[\![\tau_0]\!]$.
Then we can apply Lemma 4.58 to get $(k - j - 1, \Psi', \Sigma', \mathsf{assert}\, K_0\, \ell_0) \in \mathcal{E}^T[\![\tau_0 \sqcap K_0]\!]$.
Finally, we can apply Lemma 4.51 to get that $\Sigma' : (k - j - 1, \Psi')$, which is sufficient to complete the proof. □

Lemma 4.82 (**T-FstBot compatibility**). $\dfrac{[\![\Gamma \vdash e_0 : \bot]\!]}{[\![\Gamma \vdash \mathsf{fst}\{K_0\}\, e_0 : \bot]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{fst}\{K_0\}\, e_0)) \in \mathcal{E}^T[\![\bot]\!]$.
By Lemma 4.56, we have that $(\Sigma, e_0) \longrightarrow_T^* (\Sigma', e_0')$ where $e_0' = \mathsf{Err}^\bullet$, which is sufficient to complete the case. □

Lemma 4.83 (**T-Snd compatibility**). $\dfrac{[\![\Gamma \vdash e_0 : \tau_0 \times \tau_1]\!]}{[\![\Gamma \vdash \mathsf{snd}\{K_1\}\, e_0 : K_1 \sqcap \tau_1]\!]}$

Proof. Not meaningfully different from the **T-Fst** case. □

Lemma 4.84 (**T-SndBot compatibility**). $\dfrac{[\![\Gamma \vdash e_0 : \bot]\!]}{[\![\Gamma \vdash \mathsf{snd}\{K_1\}\, e_0 : \bot]\!]}$

Proof. Not meaningfully different from the **T-FstBot** case. □

Lemma 4.85 (**T-Binop compatibility**). $\dfrac{[\![\Gamma \vdash e_0 : \tau_0]\!] \quad [\![\Gamma \vdash e_1 : \tau_1]\!]}{[\![\Gamma \vdash \mathit{binop}\, e_0\, e_1 : \Delta(\mathit{binop}, \tau_0, \tau_1)]\!]}$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathit{binop}\, e_0\, e_1)) \in \mathcal{E}^T[\![K_2]\!]$.
Note $\gamma(\mathit{binop}\, e_0\, e_1) = \mathit{binop}\, \gamma(e_0)\, \gamma(e_1)$.
By the first hypothesis applied to γ we have $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T[\![\tau_0]\!]$.

Unfolding we get there are j, Σ', e_0' such that $(\Sigma, \gamma(e_0)) \longrightarrow_T^j (\Sigma', e_0')$ and e_0' is irreducible.
If $e_0' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e_0') \in \mathcal{V}^T[\![\tau_0]\!]$.

Note by Lemma 4.55 and Lemma 4.51, we have $(k-j, \Psi', \Sigma', \gamma) \in \mathcal{G}^T[\![\Gamma_1]\!]$ and $\Sigma' : (k-j, \Psi')$.
By the second hypothesis applied to γ we have $(k-j, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T[\![\tau_1]\!]$.
Unfolding we get there are j', Σ'', e_1' such that $(\Sigma', \gamma(e_1)) \longrightarrow_T^{j'} (\Sigma'', e_1')$ and e_1' is irreducible.
If $e_1' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise, there is a $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi)$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j', \Psi'', \Sigma'', e_1') \in \mathcal{V}^T[\![\tau_1]\!]$.

From the definition of Δ, $K_2 = \mathsf{Int}$ or Nat or $\bot$.
In the case of $\bot$, we're done because either τ_0 or τ_1 is a $\bot$, which is a contradiction.
Otherwise, the cases proceed identically, so without loss of generality assume $K_2 = \mathsf{Int}$.
$\tau_0 = \tau_1 = \mathsf{Int}$, and therefore $\mathrm{pointsto}(\Sigma'', ()e_0') = i_0$ and $\mathrm{pointsto}(\Sigma'', e_1') = i_1$.
If $\mathit{binop} = \mathsf{quotient}$ and $i_1 = 0$ then $(\Sigma'', \mathit{binop}\, e_0'\, e_1') \longrightarrow_T (\Sigma'', \mathsf{DivErr})$, so we're done.
If $\mathit{binop} = \mathsf{quotient}$ and $i_1 \neq 0$, then $(\Sigma'', \mathit{binop}\, e_0'\, e_1') \longrightarrow_T (\Sigma'', i_0/i_1) \longrightarrow_T (\Sigma''[\ell \mapsto (i_0/i_1, \mathsf{none})], \ell)$.
Since $i_0/i_1 \in \mathbb{Z}$, we're done.
If $\mathit{binop} = \mathsf{sum}$ then $(\Sigma'', \mathit{binop}\, e_0'\, e_1') \longrightarrow_T (\Sigma'', i_0 + i_1) \longrightarrow_T (\Sigma''[\ell \mapsto (i_0 + i_1, \mathsf{none})], \ell)$.
Since $i_0 + i_1 \in \mathbb{Z}$, we're done. □

Lemma 4.86 (**T-If compatibility**).
$$\frac{[\![\Gamma \vdash e_0 : \mathsf{Bool}]\!] \quad [\![\Gamma \vdash e_1 : \tau_0]\!] \quad [\![\Gamma \vdash e_2 : \tau_1]\!]}{[\![\Gamma \vdash \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_0 \sqcup \tau_1]\!]}$$

Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2)) \in \mathcal{E}^T[\![\tau_0 \sqcup \tau_1]\!]$.
Note $\gamma(\mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2) = \mathsf{if}\ \gamma(e_0)\ \mathsf{then}\ \gamma(e_1)\ \mathsf{else}\ \gamma(e_2)$.
From the first hypothesis applied to γ, we know $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T[\![\mathsf{Bool}]\!]$.
Unfolding, we have that there is Σ', e_0', j such that $(\Sigma, e_0) \longrightarrow_T^j (\Sigma', e_0')$ where e_0' is irreducible.
If $e_0' = \mathsf{Err}^\bullet$ then we're done, because the entire if statement errors.
Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e_0') \in \mathcal{V}^T[\![\mathsf{Bool}]\!]$.
Unfolding the location and then the value relation, we get that $\mathrm{pointsto}(\Sigma', e_0') = \mathsf{True}$ or $\mathrm{pointsto}(\Sigma', e_0') = \mathsf{False}$.

- $\mathrm{pointsto}(\Sigma', e_0') = \mathsf{True}$: Note by OS, $(\Sigma, \mathsf{if}\ \gamma(e_0)\ \mathsf{then}\ \gamma(e_1)\ \mathsf{else}\ \gamma(e_2)) \longrightarrow_T^j (\Sigma', \mathsf{if}\ e_0'\ \mathsf{then}\ \gamma(e_1)\ \mathsf{else}\ \gamma(e_2)) \longrightarrow_T (\Sigma', \gamma(e_1))$.
  By Lemma 4.55 and Lemma 4.51, we have $(k-j-1, \Psi', \Sigma', \gamma) \in \mathcal{G}^T[\![\Gamma_1]\!]$ and $\Sigma' : (k-j-1, \Psi')$.
  From the second hypothesis, we get $(k-j-1, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T[\![\tau_0]\!]$.
  Finally, by Lemma 4.61, we get $(k-j-1, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T[\![\tau_0 \sqcup \tau_1]\!]$ which is sufficient to complete the proof.
- $\mathrm{pointsto}(\Sigma', e_0') = \mathsf{False}$: same as other case except replace e_1 with e_2.

Proof. ∎

Lemma 4.87 (**T-IfBot** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : \bot \rrbracket \quad \llbracket \Gamma \vdash e_1 : \tau_0 \rrbracket \quad \llbracket \Gamma \vdash e_2 : \tau_1 \rrbracket}{\llbracket \Gamma \vdash \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \bot \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T\llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(\mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2)) \in \mathcal{E}^T\llbracket \bot \rrbracket$.
By Lemma 4.56, we have that $(\Sigma, e_0) \longrightarrow_T^* (\Sigma', e_0')$ where $e_0' = \mathsf{Err}^\bullet$, which is sufficient to complete the case. ∎

Lemma 4.88 (**T-Sub** compatibility).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket \quad \tau_0 \leqslant: \tau_1}{\llbracket \Gamma \vdash e_0 : \tau_1 \rrbracket}$$

Proof. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T\llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.
We want to show $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T\llbracket \tau_2 \rrbracket$.
From our hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T\llbracket \tau_1 \rrbracket$.
We can apply Lemma 4.50 to finish the case. ∎

4.4.4 Transient with Truer Transient Typing is Vigilant

Theorem 4.89 (Transient with Truer Transient Typing is Vigilant). *If $\Gamma \vdash e : \tau$ then $\llbracket \Gamma \vdash e : \tau \rrbracket_V^T$*

Proof. By induction over the typing derivation, using the compatability lemmas. ∎

4.5 Vigilance Fundamental Property for Transient with Tag Typing

Theorem 4.90 (Transient is Tag Vigilant). *If* $\Gamma \vdash_{\mathsf{tag}} e : K$ *then* $[\![\Gamma \vdash_{\mathsf{tag}} e : K]\!]^T$

Proof. By Theorem 3.10, we have that there exists some $\tau \leq K$ such that $\Gamma \vdash_{\mathsf{tru}} e : \tau$.
By Theorem 4.89, we have that $[\![\Gamma \vdash_{\mathsf{tru}} e : \tau]\!]^T$.
Unfolding this result and what we want to prove, we note the only distinction is that in what we have, we get $(k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^T[\![\tau]\!]$, and what we want to prove is $(k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^T[\![K]\!]$.
This follows directly from Lemma 4.61. □

5 Contextual equivalence

5.1 Contextual Equivalence Logical Relation—No Store

$\mathsf{DivErr} \approx \mathsf{DivErr}$

$\mathsf{TypeErr}(\tau, v) \approx \mathsf{TypeErr}(\tau', v')$

$$[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \tau]\!]_C^{\mathcal{L}} \triangleq \forall (k, \gamma_1, \gamma_2) \in \mathcal{G}^{\mathcal{L}}[\![\Gamma]\!].\ (k, \gamma_1(e_1), \gamma_2(e_2)) \in \mathcal{E}^{\mathcal{L}}[\![\tau]\!]$$

$$[\![\Gamma \vdash_{\mathsf{tru}} e_1 \approx e_2 : \tau]\!]_C^{\mathcal{L}} \triangleq [\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \tau]\!]_C^{\mathcal{L}} \wedge [\![\Gamma \vdash_{\mathsf{tru}} e_2 \leq e_1 : \tau]\!]_C^{\mathcal{L}}$$

$$\begin{aligned}\mathcal{G}^{\mathcal{L}}[\![\Gamma, x : \tau]\!] \triangleq \{(k, \gamma_1[x \mapsto v_1], \gamma_2[x \mapsto v_2]) \mid{}& (k, \gamma_1, \gamma_2) \in \mathcal{G}^{\mathcal{L}}[\![\Gamma]\!] \\ & \wedge (k, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!]_k\}\end{aligned}$$

$$\mathcal{G}^{\mathcal{L}}[\![\bullet]\!] \triangleq \{(k, \emptyset, \emptyset)\}$$

$$\begin{aligned}\mathcal{E}^{\mathcal{L}}[\![\tau]\!] \triangleq \{(k, e_1, e_2) \mid{}& \forall j \leq k, e_1'.\ e_1 \longrightarrow_L^j e_1' \wedge \mathtt{irred}_L(e_1') \\ & \Rightarrow \exists e_2'.\ e_2 \longrightarrow_L^* e_2' \\ & \quad \wedge (e_1' \approx e_2' \in \mathsf{Err}^\bullet \vee (k - j, e_1', e_2') \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!])\}\end{aligned}$$

$$\mathcal{V}^{\mathcal{L}}[\![\mathsf{Int}]\!] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{Z}\}$$

$$\mathcal{V}^{\mathcal{L}}[\![\mathsf{Nat}]\!] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{N}\}$$

$$\mathcal{V}^{\mathcal{L}}[\![\mathsf{Bool}]\!] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{B}\}$$

$$\mathcal{V}^{\mathcal{L}}[\![\tau_1 \times \tau_2]\!] \triangleq \{(k, \langle v_{1,1}, v_{1,2}\rangle, \langle v_{2,1}, v_{2,2}\rangle) \mid (k, v_{1,1}, v_{2,1}) \in \mathcal{V}^{L}[\![\tau_1]\!] \wedge\ (k, v_{2,1}, v_{2,2}) \in \mathcal{V}^{L}[\![\tau_2]\!]\}$$

$$\begin{aligned}\mathcal{V}^{\mathcal{L}}[\![\tau_1 \rightarrow \tau_2]\!] \triangleq \{(k, v_1, v_2) \mid{}& \forall j \leq k, \\ & \forall v_1', v_2' \text{ where } (j, v_1', v_2') \in \mathcal{V}^{L}[\![\tau_1]\!]. \\ & \forall K, K' \text{ where } K \sqcap \tau_2 = K' \sqcap \tau_2. \\ & (j, \mathrm{app}\{K\}\, v_1\, v_1', \mathrm{app}\{K'\}\, v_2\, v_2') \in \mathcal{E}^{L}[\![K \sqcap \tau_2]\!]\}\end{aligned}$$

$$
\begin{aligned}
\mathcal{V}^{\mathcal{L}}[\![*]\!] \triangleq \{(k, \Sigma_1, \Sigma_2, \ell_1, \ell_2) \mid\ & (k-1, v_1, v_2) \in \mathcal{V}^{L}[\![\mathsf{Int}]\!] \\
& (k-1, v_1, v_2) \in \mathcal{V}^{L}[\![\mathsf{Bool}]\!] \\
& \vee (k-1, v_1, v_2) \in \mathcal{V}^{L}[\![* \times *]\!] \\
& \vee (k-1, v_1, v_2) \in \mathcal{V}^{L}[\![* \to *]\!] \}
\end{aligned}
$$

$$
\mathcal{V}^{\mathcal{L}}[\![\bot]\!] \triangleq \emptyset
$$

5.2 Context typing

Truer transient contexts:

$$
\begin{aligned}
E \ ::= \ & [\,] \mid \lambda(x\!:\!K).\, E \mid \langle e, E\rangle \mid \langle E, e\rangle \mid \mathsf{app}\{K\}\, e\, E \mid \mathsf{app}\{K\}\, E\, e \mid \mathsf{fst}\{K\}\, E \mid \mathsf{snd}\{K\}\, E \\
\mid \ & \mathit{binop}\, e\, E \mid \mathit{binop}\, E\, e \mid \mathsf{cast}\ \{K \Leftarrow K\}\ E \mid \mathsf{if}\ E\ \mathsf{then}\ e\ \mathsf{else}\ e \mid \mathsf{if}\ e\ \mathsf{then}\ E\ \mathsf{else}\ e \mid \mathsf{if}\ e\ \mathsf{then}\ e\ \mathsf{else}\ E
\end{aligned}
$$

$$\textsc{T-Ctx-Hole}\quad \frac{\Gamma' \subseteq \Gamma}{\Gamma \vdash_{\mathsf{tru}} [\,] : (\Gamma' \triangleright \tau) \leadsto \tau}$$

$$\textsc{T-Ctx-Lam}\quad \frac{\Gamma,\ (x{:}K) \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \tau'}{\Gamma \vdash_{\mathsf{tru}} \lambda(x{:}K).\,E : (\Gamma',\ (x{:}K) \triangleright \tau) \leadsto {*} \to \tau'}$$

$$\textsc{T-Ctx-Pair-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \langle E, e\rangle : (\Gamma' \triangleright \tau) \leadsto \tau_1 \times \tau_2}$$

$$\textsc{T-Ctx-Pair-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \langle e, E\rangle : (\Gamma' \triangleright \tau) \leadsto \tau_1 \times \tau_2}$$

$$\textsc{T-Ctx-App-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto {*} \to \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, E\, e : (\Gamma' \triangleright \tau) \leadsto K \sqcap \tau_1}$$

$$\textsc{T-Ctx-AppBot-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \bot \qquad \Gamma \vdash_{\mathsf{tru}} e : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, E\, e : (\Gamma' \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-App-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e : {*} \to \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, e\, E : (\Gamma' \triangleright \tau) \leadsto K \sqcap \tau_1}$$

$$\textsc{T-Ctx-AppBot-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e : \bot \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma' \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, e\, E : (\Gamma' \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-Fst}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_1 \times \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, E : (\Gamma \triangleright \tau) \leadsto K \sqcap \tau_1}$$

$$\textsc{T-Ctx-FstBot}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \bot}{\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, E : (\Gamma \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-Snd}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_1 \times \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K\}\, E : (\Gamma \triangleright \tau) \leadsto K \sqcap \tau_2}$$

$$\textsc{T-Ctx-SndBot}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \bot}{\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K\}\, E : (\Gamma \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-Binop-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, E\, e : (\Gamma \triangleright \tau) \leadsto \Delta(\mathit{binop}, \tau_1, \tau_2)}$$

$$\textsc{T-Ctx-Binop-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, E\, e : (\Gamma \triangleright \tau) \leadsto \Delta(\mathit{binop}, \tau_1, \tau_2)}$$

$$\textsc{T-Ctx-Bnd-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{cast}\,\{K_2 \Leftarrow K_1\}\, E : (\Gamma \triangleright \tau) \leadsto K_2 \sqcap K_1 \sqcap \tau'}$$

$$\textsc{T-Ctx-If-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \mathsf{Bool} \qquad \Gamma \vdash_{\mathsf{tru}} e_1 : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e_2 : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ E\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : (\Gamma \triangleright \tau) \leadsto \tau_1 \sqcup \tau_2}$$

$$\textsc{T-Ctx-IfBot-1}\quad \frac{\Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \bot \qquad \Gamma \vdash_{\mathsf{tru}} e_1 : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e_2 : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ E\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : (\Gamma \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-If-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e_b : \mathsf{Bool} \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e_2 : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ E\ \mathsf{else}\ e_2 : (\Gamma \triangleright \tau) \leadsto \tau_1 \sqcup \tau_2}$$

$$\textsc{T-Ctx-IfBot-2}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e_b : \bot \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} e_2 : \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ E\ \mathsf{else}\ e_2 : (\Gamma \triangleright \tau) \leadsto \bot}$$

$$\textsc{T-Ctx-If-3}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e_b : \mathsf{Bool} \qquad \Gamma \vdash_{\mathsf{tru}} e_1 : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ E : (\Gamma \triangleright \tau) \leadsto \tau_1 \sqcup \tau_2}$$

$$\textsc{T-Ctx-IfBot-3}\quad \frac{\Gamma \vdash_{\mathsf{tru}} e_b : \bot \qquad \Gamma \vdash_{\mathsf{tru}} e_1 : \tau_1 \qquad \Gamma \vdash_{\mathsf{tru}} E : (\Gamma \triangleright \tau) \leadsto \tau_2}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ E : (\Gamma \triangleright \tau) \leadsto \bot}$$

5.3 Contextual equivalence statement

We define a logical relation for contexts:

$$[\![\Gamma \vdash_{\mathsf{tru}} C_1 \approx C_2 : (\Gamma' \triangleright \tau) \rightsquigarrow \tau']\!] \triangleq \forall e_1, e_2. [\![\Gamma' \vdash_{\mathsf{tru}} e_1 \approx e_2 : \tau]\!] \Rightarrow [\![\Gamma \vdash_{\mathsf{tru}} C_1[e_1] \approx C_2[e_2] : \tau']\!]$$

We define an abbreviation for the notion that an expression reduces to an eventual value without encountering an error: $e \Downarrow \triangleq \exists e'.\ e \longrightarrow_L^* e' \wedge (\mathtt{val}(e'))$

Theorem 5.1 (Expression relation implies reduction equivalence). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \approx e_2 : \tau]\!]$*, then* $e_1 \Downarrow \Leftrightarrow e_2 \Downarrow$.

Proof. By applying Lemm 5.2 in both directions. □

Lemma 5.2 (Expression relation implies reduction equivalence). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \tau]\!]$*, then* $e_1 \Downarrow \Rightarrow e_2 \Downarrow$.

Proof. Since $e_1 \Downarrow$, then there exists some e_1', k s.t. $e_1 \longrightarrow_L^k e_1'$ and e_1' is a value and hence irreducible.

We want to show that $e_2' \Downarrow$. Instantiate the premise with $(k, \emptyset, \emptyset)$, obtaining that $(k, e_1, e_2) \in \mathcal{E}^{\mathcal{L}}[\![\tau]\!]$. Instantiate j with k and e_1' with e_1', observing that e_1' being a value entails it is irreducible. Then e_2' from this relation is just what we need, since e_2 reduces to it, and it is syntactically a value. □

The usual definition of contextual equivalence is then:

$$\Gamma \vdash_{\mathsf{tru}} e_1 \approx^{\mathrm{ctx}} e_2 : \tau \triangleq \forall C, \bullet \vdash_{\mathsf{tru}} C : (\Gamma \triangleright \tau) \rightsquigarrow \tau' \Rightarrow (C[e_1] \Downarrow \Leftrightarrow C[e_2] \Downarrow)$$

Theorem 5.3 (Binary relation is sound for contextual equivalence). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \approx e_2 : \tau]\!]$*, then* $\Gamma \vdash_{\mathsf{tru}} e_1 \approx^{ctx} e_2 : \tau$.

Proof. Consider an arbitrary type τ' and context C s.t. $\bullet \vdash_{\mathsf{tru}} C : (\Gamma \triangleright \tau) \rightsquigarrow \tau'$. Then we must show that $C[e_1] \Downarrow \Leftrightarrow C[e_2] \Downarrow$. By Theorem 5.1, it is sufficient to show that $[\![\bullet \vdash_{\mathsf{tru}} C[e_1] \approx C[e_2] : \tau']\!]$.

By Theorem 5.71, $[\![\bullet \vdash_{\mathsf{tru}} C \approx C : (\Gamma \triangleright \tau) \rightsquigarrow \tau']\!]$. Unfolding this definition and instantiating it with e_1, e_2, and our hypothesis about them, we obtain precisely the required conclusion. □

5.4 Binary relation—Proofs

5.4.1 Lemmas Used Without Mention

Lemma 5.4 (Values are in the $\mathcal{E}$-relation). *If* $(k, v, v') \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!]$*, then* $(k, v, v') \in \mathcal{E}^{\mathcal{L}}[\![\tau]\!]$.

Proof. Consider arbitrary j s.t. $v \longrightarrow^j v_f \wedge \mathtt{irred}_{\mathcal{L}}(v_f)$. Note that j must be equal to 0 since values do not reduce. Then choose v' as the e_2' of the expression relation; it is easy to see that v' reduces to v' in some number (0) of steps. By our assumption, $(k - 0, v, v') \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!]$, so we are done. □

Lemma 5.5 (Anti-Reduction - Head Expansion - Expression Relation Commutes With Steps). *If* $(k, e_1', e_2') \in \mathcal{E}^T[\![\tau]\!]$ *and* $e_1 \longrightarrow_T^j e_1'$ *and* $e_2 \longrightarrow_T^{j'} e_2'$*, then* $(k + j, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$

Proof. Consider arbitrary j', e_1'' s.t. $e_1 \longrightarrow_T^{j'} e_1''$. If $j' \leq j$, by determinism of the operational semantics, e_1'' must not be irreducible and so we are trivially done. Otherwise, assume $\mathtt{irred}_T(e_1'')$ and $j' \leq k + j$; we must show that $\exists e_2''.e_2 \longrightarrow_T^* e_2'' \wedge (e_1'' \approx e_2'' \in \mathsf{Err}^\bullet \vee (k + j - j', e_1'', e_2'') \in \mathcal{V}^T[\![\tau]\!]$.

Instantiate the hypothesis with $(k + j' - j, e_1'')$. Since $k + j' - j \leq k$ and the operational semantics are deterministic, this gives us that $\exists e_2''.e_2' \longrightarrow_T^* e_2'' \wedge (e_1'' \approx e_2'' \in \mathsf{Err}^\bullet \vee (k + j - j', e_1'', e_2'') \in \mathcal{V}^T[\![\tau]\!]$, from which our conclusion follows immediately. □

Lemma 5.6 (Anti-Reduction - Head Expansion - Steps Commute With Expression Relation). *If* $(k + j, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$ *and* $e_1 \longrightarrow_T^j e_1'$ *and* $e_2 \longrightarrow_T^{j'} e_2'$, *then* $(k, e_1', e_2') \in \mathcal{E}^T[\![\tau]\!]$

Proof. Consider arbitrary j', e_1'' s.t. $j' \leq k \wedge \mathtt{irred}_T(e_1'') \wedge e_1' \longrightarrow_T^{j'} e_1''$.
We must show that $\exists e_2''.e_2' \longrightarrow_T^* e_2'' \wedge (e_1'' \approx e_2'' \in \mathsf{Err}^\bullet \vee (k - j', e_1'', e_2'') \in \mathcal{V}^T[\![\tau]\!]$.
Instantiate the hypothesis with $j + j', e_1''$. Since $j' \leq k$, $j + j' \leq k + j$. Since the operational semantics are deterministic and transitive, the other conditions apply. Then the hypothesis provides precisely the appropriate e_2'' and conditions on it and e_1''. □

We define a notion of tags extended with bottom that are compatible with the usual lattice:

$$K^\perp = K \mid \perp$$

$$\lfloor K^\perp \rfloor^\perp = \begin{cases} \perp & \text{if } K^\perp = \perp \\ \lfloor K^\perp \rfloor & \text{otherwise} \end{cases}$$

$$\sim^\perp (K^\perp, v) = \begin{cases} \mathsf{False} & \text{if } K^\perp = \perp \\ v \sim K^\perp & \text{otherwise} \end{cases}$$

Lemma 5.7 (Tagof-bot is compatible with meet). $\lfloor K_1^\perp \sqcap K_2^\perp \rfloor^\perp = \lfloor K_1^\perp \rfloor^\perp \sqcap \lfloor K_2^\perp \rfloor^\perp$.

Proof. Immediate, by unfolding definitions and case analysis. □

Lemma 5.8 (Relation implies tagmatch). *If* $(k, v, v') \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!]$ *and* $K^\perp \leq \lfloor \tau \rfloor^\perp$, *then* $\sim^\perp (K^\perp, v)$.

Proof. By case analysis on τ and $K^\perp$; in each case this follows immediately from unfolding the definitions of $\mathcal{V}$ and tagmatch. □

5.4.2 Lemmas Used With Mention

Lemma 5.9 (Related values have matching constructors). *If* $(k, v, v') \in \mathcal{V}^{\mathcal{L}}[\![\tau]\!]$, *then either*

- $v = v'$
- *There exist some* v_1, v_2, v_1', v_2' *s.t.* $v = \langle v_1, v_2 \rangle$ *and* $v' = \langle v_1', v_2' \rangle$
- *There exist some* w, w' *s.t.* $v = w$ *and* $v' = w'$.

Proof. By induction on τ, unfolding the definition of $\mathcal{V}$ in each case. □

Lemma 5.10 (Tagmatch is up to approximation). *If* $(k, v, v') \in \mathcal{V}^T[\![\tau]\!]$, *then* $\sim^\perp (K^\perp, v) \Leftrightarrow \sim^\perp (K^\perp, v')$.

Proof. By Lemma 5.9 and inspection of the definition of $\sim^\perp (K^\perp, v)$. □

Lemma 5.11 (Tagmatch respects meets). $\sim^\perp (K_1^\perp \sqcap K_2^\perp, v) \Leftrightarrow \sim^\perp (K_1^\perp, v) \wedge \sim^\perp (K_2^\perp, v)$.

Proof. By case analysis on $K_1^\perp, K_2^\perp$; in each case the conclusion follows immediately by unfolding. □

Lemma 5.12 (Tagmatch implies values in relation at meet). *If* $(k, v, v') \in \mathcal{V}^T[\![\tau]\!]$ *and* $\sim^\perp (K^\perp, v)$, *then* $(k - 1, v, v') \in \mathcal{V}^T[\![K^\perp \sqcap \tau]\!]$.

Proof. Proceed by case analysis on $K^\perp$:

$*$ By lattice properties, $K^\perp \sqcap \tau = \tau$, so this is trivial by Lemma **??**.

Nat By the definition of tagmatch, v must be a natural number. By inspection, this is possible only when τ is $*$, Int, or Nat; in each case, $K^\perp \sqcap \tau =$ Nat. By inspection on the relation, v always satisfied what is needed.

Int Analogous to the Nat case above.

$*\times*$ By the definition of tagmatch, v must be a pair; by inspection this is possible only if τ is $*$ or some pair type. If the latter, $K^\perp \sqcap \tau = \tau$, and so the conclusion is immediate; otherwise, $K^\perp \sqcap \tau = *\times*$, and the conclusion is immediate from the definition of the $*$ case of the relation.

$*\to*$ By the definition of tagmatch, v must be a w; by inspection this is possible only if τ is $*$ or some function type. If the latter, $K^\perp \sqcap \tau = \tau$, and so the conclusion is immediate; otherwise, $K^\perp \sqcap \tau = *\to*$, and the conclusion is immediate from the definition of the $*$ case of the relation.e

$\perp$ Contradiction

□

Lemma 5.13 ($\mathcal{E}$-$\mathcal{V}$-monotonicity). *(1) If $(k, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$ and $j \le k$, then $(j, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$.*
(2) If $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$ and $j \le k$, then $(j, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$.

Proof. Proceed by simultaneous induction on k and τ:

- $k = 0$: 1) follows immediately from 2).
  Proceeds similarly to the other case, but function and dynamic cases are vacuously true.
- $k > 0$:
  1) Unfolding the expression relation in our hypothesis, we get that there is some e_1', j' such that $e_1 \longrightarrow_T^{j'} e_1'$, and some e_2' such that $e_2 \longrightarrow_T^{*} e_2'$.
     If $e_1' = \mathsf{Err}^\bullet$ then we're done.
     Otherwise, $(k - j', e_1', e_2') \in \mathcal{V}^T[\![\tau]\!]$.

     Now, unfolding the expression relation, we want to show $(k - j - j', e_1', e_2') \in \mathcal{V}^T[\![\tau]\!]$.
     We can apply the IH 2) with the fact proven in a).
  2) We want to show that $(k - j, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$.
     We case split on τ:
     i) $\tau =$ Nat: then where $n \in \mathbb{N}$, so the case is immediate.

     ii) $\tau = tint$: same as above.

     iii) $\tau =$ Bool: same as above.

     iv) $\tau = \tau_1 \times \tau_2$: Then unfolding our hypothesis gives us $v_1 = \langle v_1', v_1'' \rangle$ and $v_2 = \langle v_1', v_1'' \rangle$ with $(k, v_1', v_2') \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k, v_1'', v_2'') \in \mathcal{V}^T[\![\tau_2]\!]$.
     The case follows by applying the IH 2) to both premises.
     v) $\tau = * \to \tau_2$: Let $j' \le k - j$.
     Let $(j', v_1', v_2') \in \mathcal{V}^T[\![*]\!]$.
     Let K, K'.

We want to show $(j', \mathsf{app}\{K\}\, v_1\, v_1', \mathsf{app}\{K'\}\, v_2\, v_2') \in \mathcal{E}^T[\![K \sqcap \tau_2]\!]$.

Since $j' \le k - j \le k$, we can apply the hypothesis to complete the case.

vi) $\tau = *$: we want to show $(k-1, v_1, v_2) \in \mathcal{V}^T[\![\mathsf{Int}]\!]$ or $\mathcal{V}^T[\![\mathsf{Bool}]\!]$ or $\mathcal{V}^T[\![* \times *]\!]$ or $\mathcal{V}^T[\![* \to *]\!]$.

This follows from IH 2) (smaller by index).

□

Lemma 5.14 (Monadic bind). *Suppose that E_1, E_2 are any evaluation contexts (n.b. not a general context, as used elsewhere in these proofs), $(k, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$, and for all k', v_1, v_2, if $k' \le k \wedge (k', v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$ then $(k', E_1[v_1], E_2[v_2]) \in \mathcal{E}^T[\![\tau']\!]$.*

Then $(k, E_1[e_1], E_2[e_2]) \in \mathcal{E}^T[\![\tau']\!]$.

Proof. Consider arbitrary j, e_1' s.t. $j \le k \wedge E_1[e_1] \longrightarrow_T^j e_1' \wedge \mathtt{irred}_T(e_1')$. Then we must show that must show that $\exists e_2'. E_2[e_2] \longrightarrow_T^* e_2' \wedge (e_1' \approx e_2' \in \mathsf{Err}^\bullet \vee (k-j, e_1', e_2') \in \mathcal{V}^T[\![\tau]\!])$.

Because $E_1[e_1]$ reaches an irreducible term in at most j steps, by our operational semantics e_1 must itself reduce to some irreducible term e_3 in some smaller number of steps $j' \le j$. Then since $j' \le j \wedge e_1 \longrightarrow_T^{j'} e_3 \wedge \mathtt{irred}_T(e_3)$, we can instantiate our first assumption, obtaining that there similarly exists e_4 s.t. $e_2 \longrightarrow_T^* e_4 \wedge (e_3 \approx e_4 \in \mathsf{Err}^\bullet \vee (k-j', e_3, e_4) \in \mathcal{V}^T[\![\tau]\!]$.

Suppose that $e_3 \approx e_4 \in \mathsf{Err}^\bullet$. Then by the operational semantics, $E_1[e_1]$ and $E_2[e_2]$ reduce to the same errors, so instantiating e_1' and e_2' with them proves our goal.

Otherwise, we know that $(k-j', e_3, e_4) \in \mathcal{V}^T[\![\tau]\!]$. We may therefore instantiate our other assumption with $k-j', e3, e_4$ and this fact, obtaining that $(k-j', E_1[e_3], E_2[e_4]) \in \mathcal{E}^T[\![\tau]\!]$. We still must show that $\exists e_2'. E_2[e_2] \longrightarrow_T^* e_2' \wedge (e_1' \approx e_2 \in \mathsf{Err}^\bullet \vee (k-j, e_1', e_2') \in \mathcal{V}^T[\![\tau]\!])$.

Instantiate the result of our assumption with step index $j - j' \le k - j'$ and e_1'. By determinism of the operational semantics, $E_1[e_3] \longrightarrow_T^{j-j'} e_1'$, so we obtain that $\exists e_2'. E_2[e_4] \longrightarrow_T^* e_2' \wedge (e_1' \approx e_2' \in \mathsf{Err}^\bullet \vee (k-j'-(j-j'), e_1', e_2') \in \mathcal{V}^T[\![\tau]\!])$. Note that $k - j' - (j - j') = k - j$, and that since $E_2[e_4] \longrightarrow_T^* e_2'$ and $e_2 \longrightarrow_T^* e_4$, then $E_2[e_2] \longrightarrow_T^* e_2'$, so this is precisely the e_2' that we needed to show the existence of. □

Lemma 5.15 (Check compatibility). *If $(k, v, v') \in \mathcal{E}^T[\![\tau]\!]$ and $\tau' = K \sqcap \tau = K' \sqcap \tau$, then $(k, \mathsf{assert}\, K\, v, \mathsf{assert}\, K'\, v') \in \mathcal{E}^T[\![\tau']\!]$.*

Proof. Proceed by case analysis on $K \sqcap \tau$:

$K \sqcap \tau = \tau$ Then it must be the case that $K \sim v$ and $K' \sim v'$, meaning $\mathsf{assert}\, K\, v \longrightarrow_T v$ and $\mathsf{assert}\, K'\, v' \longrightarrow_T v'$, which is sufficient to complete the case.

$K \sqcap \tau = \mathsf{Nat}$ **and** $\tau = \mathsf{Int}$ Unfolding our hypothesis, we get that $v = v'$ and $v \in \mathbb{Z}$.

If $v \in \mathbb{N}$, then $\mathsf{assert}\, K\, v \longrightarrow_T v$ and $\mathsf{assert}\, K'\, v' \longrightarrow_T v'$, which is sufficient to complete the case.

Otherwise, $\mathsf{assert}\, K\, v \longrightarrow_T \mathsf{TypeErr}(\mathsf{Nat}, v)$ and $\mathsf{assert}\, K'\, v' \longrightarrow_T \mathsf{TypeErr}(\mathsf{Nat}, v')$, which is sufficient to complete the case.

$K \sqcap \tau = \bot$ Then $\mathsf{assert}\, K\, v \longrightarrow_T \mathsf{TypeErr}(\mathsf{Nat}, v)$ and $\mathsf{assert}\, K\, v' \longrightarrow_T \mathsf{TypeErr}(\mathsf{Nat}, v')$, which is sufficient to complete the case.

$K \sqcap \tau = K$ **and** $\tau \ne K$ Then $\tau = *$ and $K = K'$.

We can unfold our hypothesis to get that $(k-1, v, v') \in \mathcal{V}^T[\![K'']\!]$ for some K'', which implies $v' \sim v$.

By the OS, either $\mathsf{assert}\, K\, v \longrightarrow v$ and $v \sim K$, or $\mathsf{assert}\, K\, v \longrightarrow \mathsf{TypeErr}(K, v)$ and $\neg v \sim K$.

In either case, we have the corresponding property needed to complete the case.

□

Lemma 5.16 (Dynamic Checks Are No-ops). *If* $(k+1, \mathsf{assert} * v, \mathsf{assert} * v') \in \mathcal{E}^T[\![\tau]\!]$, *then* $(k, v, v') \in \mathcal{E}^T[\![\tau]\!]$

Proof. By the OS, $\mathsf{assert} * v \longrightarrow v$ and $\mathsf{assert} * v' \longrightarrow v'$.
Then by our hypothesis, $(k, v, v') \in \mathcal{V}^T[\![\tau]\!]$, which is sufficient to complete the proof. □

Lemma 5.17 (Subtyping Compatibility). *(1) If* $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$ *and* $\tau \leqslant: \tau'$ *then* $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau']\!]$
(2) If $(k, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$ *and* $\tau \leqslant: \tau'$ *then* $(k, e_1, e_2) \in \mathcal{E}^T[\![\tau']\!]$.

Proof. Proceed by mutual induction on k and τ:

- $k = 0$: 2 is immediate if $e \neq v$.
  If $e = v$ then 2 follows immediately from 1.
  1 follows identically in the $k = 0$ case as it does in the $k > 0$ case, but the function case is vacuously true.
- $k > 0$:
  (1) Case split on $\tau \leqslant: \tau'$:
    i) $\tau \leqslant: \tau$: immediate.
    ii) $\mathsf{Nat} \leqslant: \mathsf{Int}$: immediate because $\mathbb{T} \subseteq \mathbb{Z}$.
    iii) $\tau_1 \times \tau_2 \leqslant: \tau_1' \times \tau_2'$, with $\tau_1 \leqslant: \tau_1'$ and $\tau_2 \leqslant: \tau_2'$:
      We want to show $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau']\!]$.
      Unfolding our hypothesis, we get that $v_1 = \langle v_1', v_1'' \rangle$ and similarly for v_2.
      We want to show $(k, v_1', v_2') \in \mathcal{V}^T[\![\tau_1']\!]$ and $(k, v_1'', v_2'') \in \mathcal{V}^T[\![\tau_2']\!]$.
      We can apply IH 1) to both of judgements in our hypothesis to get $(k, v_1', v_2') \in \mathcal{V}^T[\![\tau_1']\!]$ and $(k, v_1'', v_2'') \in \mathcal{V}^T[\![\tau_2']\!]$.
      This is sufficient to show $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau']\!]$.
    iv) $* \to \tau_2 \leqslant: * \to \tau_2'$, with $\tau_2 \leqslant: \tau_2'$:
      We want to show $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau']\!]$.
      Let $j \leq k$ and $(j, v_1', v_2') \in \mathcal{V}^T[\![*]\!]$.
      Let K.
      We want to show $(j, \mathsf{app}\{K\}\, v_1\, v_1', \mathsf{app}\{K\}\, v_2\, v_2') \in \mathcal{E}^T[\![\tau_2' \sqcap K]\!]$.
      Then, we can apply our hypothesis about v_1, v_2 to get $(j, \mathsf{app}\{K\}\, v_1\, v_1', \mathsf{app}\{K\}\, v_2\, v_2') \in \mathcal{E}^T[\![\tau_2 \sqcap K]\!]$.
      Finally, we can apply IH 1) to get $(j, \mathsf{app}\{K\}\, v_1\, v_1', \mathsf{app}\{K\}\, v_2\, v_2') \in \mathcal{E}^T[\![\tau_2' \sqcap K]\!]$ which is what we wanted to show.

  (2) Unfolding our hypothesis, there is some $j \leq k$ and irreducible e_1', e_2' such that $e_1 \longrightarrow_T^j e_1'$ and $e_2 \longrightarrow_T^* e_2'$.
    If $e_1', e_2' \in \mathsf{Err}^\bullet$ then we're done.
    Otherwise, $(k - j, e_1', e_2') \in \mathcal{V}^T[\![\tau]\!]$.
    By IH 1), we have $(k - j, e_1', e_2') \in \mathcal{V}^T[\![\tau']\!]$, which is what we wanted to show.

□

Lemma 5.18 (Monitor compatibility). *If* $(k, v, v') \in \mathcal{V}^T[\![\tau]\!]$, *then* $(k+1, \mathsf{mon}\,\{K_1' \Leftarrow K_1\}, \mathsf{mon}\,\{K_2' \Leftarrow K_2\}\, v') \in \mathcal{E}^T[\![\tau]\!]$.

Proof. By induction on k and v:

$k = 0$ By case analysis on v, v':

i, i' By OS, $\mathsf{mon}\,\{K_1' \Leftarrow K_1\}\; i \longrightarrow i$ and $\mathsf{mon}\,\{K_2' \Leftarrow K_2\}\; i' \longrightarrow i'e$, so this is immediate.

$\mathsf{True}, \mathsf{True}$ As in case i above.

$\mathsf{False}, \mathsf{False}$ As in case True above.

$\langle v_1, v_2\rangle, \langle v_1', v_2'\rangle$ Since $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$, by inspection τ must be either $\tau_1{\times}\tau_2$ or $*$:

$\tau_1{\times}\tau_2$ Note that $\mathsf{mon}\,\{K_1' \Leftarrow K_1\}\,\langle v_1, v_2\rangle \longrightarrow \langle \mathsf{mon}\,\{\mathit{fst}(K_1') \Leftarrow \mathit{fst}(K_1)\}\, v_1, \mathsf{mon}\,\{\mathit{snd}(K_1') \Leftarrow \mathit{snd}(K_1)\}\, v_2\rangle$, and similarly $\mathsf{mon}\,\{K_2' \Leftarrow K_2\}\,\langle v_1', v_2'\rangle \longrightarrow \langle \mathsf{mon}\,\{\mathit{fst}(K_2') \Leftarrow \mathit{fst}(K_2)\}\, v_1', \mathsf{mon}\,\{\mathit{snd}(K_2') \Leftarrow \mathit{snd}(K_2)\}\, v_2'\rangle$

It is therefore sufficient to show that

$$(k, \langle \mathsf{mon}\,\{\mathit{fst}(K_1') \Leftarrow \mathit{fst}(K_1)\}\, v_1, \mathsf{mon}\,\{\mathit{snd}(K_1') \Leftarrow \mathit{snd}(K_1)\}\, v_2\rangle, \langle \mathsf{mon}\,\{\mathit{fst}(K_2') \Leftarrow \mathit{fst}(K_2)\}\, v_1', \mathsf{mon}\,\{\mathit{snd}(K_2') \Leftarrow \mathit{snd}(K_2)\}\, v_2'\rangle) \in \mathcal{E}^T[\![\tau_1{\times}\tau_2]$$

By unfolding, this is the same as showing $(k, \mathsf{mon}\,\{\mathit{fst}(K_1') \Leftarrow \mathit{fst}(K_1)\}\, v_1, \mathsf{mon}\,\{\mathit{fst}(K_2') \Leftarrow \mathit{fst}(K_2)\}\, v_1') \in \mathcal{E}^T[\![\tau_1]\!]$ and $(k, \mathsf{mon}\,\{\mathit{snd}(K_1') \Leftarrow \mathit{snd}(K_1)\}\, v_2, \mathsf{mon}\,\{\mathit{snd}(K_2') \Leftarrow \mathit{snd}(K_2)\}\, v_2') \in \mathcal{E}^T[\![\tau_2]\!]$.

By Lemma 5.13, it suffices to show $(k{+}1, \mathsf{mon}\,\{\mathit{fst}(K_1') \Leftarrow \mathit{fst}(K_1)\}\, v_1, \mathsf{mon}\,\{\mathit{fst}(K_2') \Leftarrow \mathit{fst}(K_2)\}\, v_1') \in \mathcal{E}^T[\![\tau_1]\!]$ and $(k + 1, \mathsf{mon}\,\{\mathit{snd}(K_1') \Leftarrow \mathit{snd}(K_1)\}\, v_2, \mathsf{mon}\,\{\mathit{snd}(K_2') \Leftarrow \mathit{snd}(K_2)\}\, v_2') \in \mathcal{E}^T[\![\tau_2]\!]$.

In both cases, IH applies and hence it suffices to show $(k, v_1, v_1') \in \mathcal{E}^T[\![\tau_1]\!]$ and $(k, v_2, v_2') \in \mathcal{E}^T[\![\tau_2]\!]$.

These are both obtained by unfolding our assumption.

$*$ Impossible, since $k = 0$.

w, w' Since $(k, w, w') \in \mathcal{V}^T[\![\tau]\!]$, by inspection τ must be either $*{\rightarrow}\tau'$ or $*$:

$*{\rightarrow}\tau'$ Note that $\mathsf{mon}\,\{K_1' \Leftarrow K_1\}\; w \longrightarrow \mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w$, and similarly $\mathsf{mon}\,\{K_2' \Leftarrow K_2\}\; w' \longrightarrow \mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w'$.

Consequently, it is sufficient to show that $(k, \mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w, \mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w') \in \mathcal{E}^T[\![*{\rightarrow}\tau']\!]$.

Consider arbitrary $j \le k, v, v'$ s.t. $(j, v, v') \in \mathcal{V}^T[\![*]\!], K, K'$. Then we must show that

$(j, \mathsf{app}\{K\}\; (\mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w)\; v, \mathsf{app}\{K'\}\; (\mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w')\; v') \in \mathcal{E}^T[\![K \sqcap \tau']\!]$.

By assumption, $k = 0$, so $j = 0$. Therefore, this is vacuously true.

$*$ Impossible, since $k = 0$.

otherwise Impossible by Lemma 5.9.

$k > 0$ By case analysis on v, v':

i, i' As in $k = 0$ case.

$\mathsf{True}, \mathsf{True}$ As in $k = 0$ case.

$\mathsf{False}, \mathsf{False}$ As in $k = 0$ case.

$\langle v_1, v_2\rangle, \langle v_1', v_2'\rangle$ Since $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$, by inspection τ must be either $\tau_1{\times}\tau_2$ or $*$:

$\tau_1{\times}\tau_2$ As in $k = 0$ case.

$*$ By unfolding, $(k - 1, w, w') \in \mathcal{V}^T[\![*{\times}*]\!]$. By an argument essentially identical to the previous case, merely reducing one application of monotonicity by one is sufficient to show what is needed.

w, w' Since $(k, w, w') \in \mathcal{V}^T[\![\tau]\!]$, by inspection τ must be either $*{\rightarrow}\tau'$ or $*$:

$*{\rightarrow}\tau'$ Note that $\mathsf{mon}\,\{K_1' \Leftarrow K_1\}\; w \longrightarrow \mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w$, and similarly $\mathsf{mon}\,\{K_2' \Leftarrow K_2\}\; w' \longrightarrow \mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w'$.

Consequently, it is sufficient to show that $(k, \mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w, \mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w') \in \mathcal{E}^T[\![*{\rightarrow}\tau']\!]$.

Consider arbitrary $j \le k, v, v'$ s.t. $(j, v, v') \in \mathcal{V}^T[\![*]\!], K, K'$ s.t. $K \sqcap \tau' = K' \sqcap \tau'$. Then we must show that

$(j, \mathsf{app}\{K\}\; (\mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w)\; v, \mathsf{app}\{K'\}\; (\mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w')\; v') \in \mathcal{E}^T[\![K \sqcap \tau']\!]$.

By OS, it suffices to show that

$(j - 1, \mathsf{assert}\, K\; ((\mathsf{grd}\,\{K_1' \Leftarrow K_1\}\; w)\; v), \mathsf{assert}\, K'\; ((\mathsf{grd}\,\{K_2' \Leftarrow K_2\}\; w')\; v')) \in \mathcal{E}^T[\![K \sqcap \tau']\!]$.

By Lemma 5.15, it suffices to show that $(j-1, (\text{grd}\,\{K_1' \Leftarrow K_1\}\, w)\ v, (\text{grd}\,\{K_2' \Leftarrow K_2\}\, w')\ v') \in \mathcal{E}^T[\![\tau']\!]$.

By OS, it suffices to show that

$(j-2, \text{mon}\,\{cod(K_1') \Leftarrow cod(K_1)\}\ w\ \text{mon}\,\{dom(K_1) \Leftarrow dom(K_1')\}\ v,$
$\text{mon}\,\{cod(K_2') \Leftarrow cod(K_2)\}\ w'\ \text{mon}\,\{dom(K_2) \Leftarrow dom(K_2')\}\ v')$
$\in \mathcal{E}^T[\![\tau']\!]$.

By IH, it suffices to show that $(j-3, w\ \text{mon}\,\{dom(K_1) \Leftarrow dom(K_1')\}\ v, w'\ \text{mon}\,\{dom(K_2) \Leftarrow dom(K_2')\}\ v') \in \mathcal{E}^T[\![\tau']\!]$.

By Lemma 5.16, it suffices to show that

$(j-2, \text{assert}\ *\ w\ \text{mon}\,\{dom(K_1) \Leftarrow dom(K_1')\}\ v, \text{assert}\ *\ w'\ \text{mon}\,\{dom(K_2) \Leftarrow dom(K_2')\}\ v') \in \mathcal{E}^T[\![\tau']\!]$.

By the definition of meet and OS, this is equivalent to

$(j-1, \text{app}\{*\}\ w\ \text{mon}\,\{dom(K_1) \Leftarrow dom(K_1')\}\ v, \text{app}\{*\}\ w'\ \text{mon}\,\{dom(K_2) \Leftarrow dom(K_2')\}\ v') \in \mathcal{E}^T[\![* \sqcap \tau']\!]$.

By unfolding the assumption that $(k, w, w') \in \mathcal{E}^T[\![* \to \tau']\!]$, it suffices to show that

$(j-1, \text{mon}\,\{dom(K_1) \Leftarrow dom(K_1')\}\ v, \text{mon}\,\{dom(K_2) \Leftarrow dom(K_2')\}\ v') \in \mathcal{E}^T[\![*]\!]$.

By IH, it suffices to show that $(j-2, v, v') \in \mathcal{E}^T[\![*]\!]$.

By Lemma 5.13, it suffices to show that $(j, v, v') \in \mathcal{E}^T[\![*]\!]$.

This is immediate from the assumption that $(j, v, v') \in \mathcal{V}^T[\![*]\!]$.

* By unfolding, $(k-1, w, w') \in \mathcal{V}^T[\![* \to *]\!]$. By an argument essentially identical to the previous case, merely reducing one application of monotonicity by one is sufficient to show what is needed.

otherwise Impossible by Lemma 5.9.

□

Corollary 5.19. *If* $(k, e_1, e_2) \in \mathcal{E}^T[\![\tau]\!]$, *then* $(k+1, \text{mon}\,\{K_1' \Leftarrow K_1\}, \text{mon}\,\{K_2' \Leftarrow K_2\}\ e_2) \in \mathcal{E}^T[\![\tau]\!]$.

Proof. Unfolding the expression relation in our hypothesis, we get that there is a j and e_1' such that $e_1 \longrightarrow_T^j e'$ such that e' is irreducible, and an e_2' such that $e_2 \longrightarrow_T^* e_2'$ and either they're errors, or $(k-j, e_1', e_2') \in \mathcal{V}^T[\![\tau]\!]$.
If they're errors, then we're done because the monitors will also step to errors.
Otherwise, we have $\text{mon}\,\{K_1' \Leftarrow K_1\} \longrightarrow_T^j \text{mon}\,\{K_1' \Leftarrow K_1\}$ and $\text{mon}\,\{K_2' \Leftarrow K_2\} \longrightarrow_T^j \text{mon}\,\{K_2' \Leftarrow K_2\}$.
By Lemma 5.18, we have that $(k-j, \text{mon}\,\{K_1' \Leftarrow K_1\}, \text{mon}\,\{K_2' \Leftarrow K_2\}) \in \mathcal{E}^T[\![\tau]\!]$, which is sufficient to complete the proof. □

Lemma 5.20 (Boundary compatibility). *If* $(k, v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$ *and* $\tau' = K_1' \sqcap K_1 \sqcap \tau = K_2' \sqcap K_2 \sqcap \tau$, *then* $(k+1, \text{cast}\,\{K_1' \Leftarrow K_1\}\ v_1, \text{cast}\,\{K_2' \Leftarrow K_2\}\ v_2) \in \mathcal{E}^T[\![\tau']\!]$.

Proof. By Lemma 5.10, notice that $\sim^\perp (\lfloor\tau'\rfloor^\perp, v_1) \Leftrightarrow \sim^\perp (\lfloor\tau'\rfloor^\perp, v_2)$. By Lemma 5.11 and our assumption, therefore, $\sim^\perp (K_1', v_1) \wedge \sim^\perp (K_1, v_1) \wedge \sim^\perp (\lfloor\tau\rfloor^\perp, v_1) \Leftrightarrow \sim^\perp (K_2', v_2) \wedge \sim^\perp (K_2, v_2) \wedge \sim^\perp (\lfloor\tau\rfloor^\perp, v_2)$. By Lemma 5.10, $\sim^\perp (\lfloor\tau\rfloor^\perp, v_1) \Leftrightarrow \sim^\perp (\lfloor\tau\rfloor^\perp, v_2)$. Consequently, $\sim^\perp (K_1', v_1) \wedge \sim^\perp (K_1, v_1) \Leftrightarrow \sim^\perp (K_2', v_2) \wedge \sim^\perp (K_2, v_2)$—which is to say, either both of the values match both of their annotated tags, or both of them do not match at least one of their annotated tags.

Consider then each case:

Tags match By the operational semantics, it is sufficient to show that $(k, \mathsf{mon}\,\{K_1' \Leftarrow K_1\}\, v_1, \mathsf{mon}\,\{K_2' \Leftarrow K_2\}\, v_2) \in \mathcal{E}^T[\![\tau']\!]$.

By Lemma 5.18, it is sufficient to show that $(k-1, v_1, v_2) \in \mathcal{E}^T[\![\tau']\!]$.

By Lemma 5.12, it is sufficient to show that $(k, v_1, v_2) \in \mathcal{E}^T[\![\tau]\!]$, which is our assumption.

Tags do not match Inspection of the operational semantics shows that both terms step to a boundary error, and so are trivially in the relation.

□

Lemma 5.21 (Boundary compatibility—open relation). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \tau]\!]_C^T$ *and* $\tau' = K_1' \sqcap K_1 \sqcap \tau = K_2' \sqcap K_2 \sqcap \tau$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{cast}\,\{K_1' \Leftarrow K_1'\}\, e_1 \leq \mathsf{cast}\,\{K_2' \Leftarrow K_2\}\, e_1 : \tau']\!]$.

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show that $(k, \gamma(\mathsf{cast}\,\{K_1' \Leftarrow K_1\}\, e_1), \gamma'(\mathsf{cast}\,\{K_2' \Leftarrow K_2\}\, e_2)) \in \mathcal{E}^T[\![\tau']\!]$.

By the definition of substitution, it suffices to show that $(k, \mathsf{cast}\,\{K_1' \Leftarrow K_1\}\, \gamma(e_1), \mathsf{cast}\,\{K_2' \Leftarrow K_2\}\, \gamma'(e_2)) \in \mathcal{E}^T[\![\tau']\!]$.

Instantiate the hypothesis with (k, γ, γ'), providing that $(k, \gamma(e_1), \gamma'(e_2)) \in \mathcal{E}^T[\![\tau]\!]$.

Then Lemma 5.14 applies. Consider arbitrary (k', v_1, v_2) s.t. $(k', v_1, v_2) \in \mathcal{V}^T[\![\tau]\!]$; we must show that $(k', \mathsf{cast}\,\{K_1' \Leftarrow K_1\}\, v_1, \mathsf{cast}\,\{K_2' \Leftarrow K_2\}\, v_2) \in \mathcal{E}^T[\![\tau]\!]$. This is immediate by Lemma 5.20 and Lemma 5.13. □

Lemma 5.22 (Application compatibility). *If* $(k, v_f, v_f') \in \mathcal{V}^T[\![* \to \tau_2]\!]$ *and* $(k, v_a, v_a') \in \mathcal{V}^T[\![\tau_1]\!]$ *and* $\tau' = K \sqcap \tau_2 = K' \sqcap \tau_2$, *then* $(k, \mathsf{app}\{K\}\, v_f\, v_a, \mathsf{app}\{K'\}\, v_f'\, v_a') \in \mathcal{E}^T[\![\tau']\!]$

Proof. Unfolding the $\mathcal{V}$ relation on our first assumption and instantiating with $j = k$, $v_1' = v_a$, $v_2' = v_a'$, $K = K$, $K' = K'$ gives precisely what is to be shown. □

Lemma 5.23 (Application compatibility—open relation). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_{f1} \leq e_{f2} : * \to \tau_2]\!]_C^T$ *and* $\tau' = K_1 \sqcap \tau_2 = K_2 \sqcap \tau_2$ *and* $[\![\Gamma \vdash_{\mathsf{tru}} e_{a1} \leq e_{a2} : \tau_1]\!]_C^T$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\, e_{f1}\, e_{a1} \leq \mathsf{app}\{K_2\}\, e_{f2}\, e_{a2} : \tau']\!]_C^T$.

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show that $(k, \gamma(\mathsf{app}\{K_1\}\, e_{f1}\, e_{a1}), \gamma'(\mathsf{app}\{K_2\}\, e_{f2}\, e_{a2})) \in \mathcal{E}^T[\![\tau']\!]$.

By the definition of substitution, it suffices to show that $(k, \mathsf{app}\{K_1\}\, \gamma(e_{f1})\; \gamma(e_{a1}), \mathsf{app}\{K_2\}\, \gamma'(e_{f2})\; \gamma'(e_{a2})) \in \mathcal{E}^T[\![\tau']\!]$.

Instantiate the first hypothesis with (k, γ, γ'), providing $(k, \gamma(e_{f1}), \gamma'(e_{f2})) \in \mathcal{E}^T[\![* \to \tau_2]\!]$. Similarly, the second provides $(k, \gamma(e_{a1}), \gamma'(e_{a2})) \in \mathcal{E}^T[\![\tau_1]\!]$.

Then Lemma 5.14 applies. Consider arbitrary $(k', v_{f1}, v_{f2}) \in \mathcal{V}^T[\![* \to \tau_2]\!]$ with $k' \leq k$. Then by Lemma 5.13, $(k', \gamma(e_{a1}), \gamma'(e_{a2})) \in \mathcal{E}^T[\![\tau_1]\!]$, Lemma 5.14 again applies. Consider arbitrary $(k'', v_{a1}, v_{a2} \in \mathcal{V}^T[\![\tau_1]\!]$ with $k'' \leq k'$. We must show that $(k'', \mathsf{app}\{K_1\}\, v_{f1}\, v_{a1}, \mathsf{app}\{K_2\}\, v_{f2}\, v_{a2}) \in \mathcal{E}^T[\![\tau']\!]$; this is immmediate by Lemma 5.22. □

Lemma 5.24 (Application compatibility–function is bottom). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_{f1} \leq e_{f2} : \bot]\!]_C^T$ *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\, e_{f1}\, e_{a1} \leq \mathsf{app}\{K_2\}\, e_{f2}\, e_{a2} : \bot]\!]_C^T$.

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show that $(k, \gamma(\mathsf{app}\{K_1\}\, e_{f1}\, e_{a1}), \gamma'(\mathsf{app}\{K_2\}\, e_{f2}\, e_{a2})) \in \mathcal{E}^T[\![\tau']\!]$.

By the definition of substitution, it suffices to show that $(k, \mathsf{app}\{K_1\}\, \gamma(e_{f1})\; \gamma(e_{a1}), \mathsf{app}\{K_2\}\, \gamma'(e_{f2})\; \gamma'(e_{a2})) \in \mathcal{E}^T[\![\tau']\!]$.

Instantiate the first hypothesis with (k, γ, γ'), providing $(k, \gamma(e_{f1}), \gamma'(e_{f2})) \in \mathcal{E}^T[\![\bot]\!]$.

Then Lemma 5.14 applies. Consider arbitrary $(k', v_{f1}, v_{f2}) \in \mathcal{V}^T[\![\bot]\!]$ with $k' \leq k$. By unfolding of $\mathcal{V}$ no such values can exist, so we are done. □

Lemma 5.25 (Fst compatibility). *If* $(k, v, v') \in \mathcal{V}^T[\![\tau_1 \times \tau_2]\!]$ *and* $\tau' = K \sqcap \tau_1 = K' \sqcap \tau_1$, *then* $(k, \mathsf{fst}\{K\}\, v, \mathsf{fst}\{K'\}\, v') \in \mathcal{E}^T[\![\tau']\!]$.

Proof. Unfolding the definition of $\mathcal{V}$ tells us that there must be some v_1, v_2, v_1', v_2' s.t. $v = \langle v_1, v_2\rangle$, $v' = \langle v_1', v_2'\rangle$, $(k, v_1, v_1') \in \mathcal{V}^T[\![\tau_1]\!]$, and $(k, v_2, v_2') \in \mathcal{V}^T[\![\tau_2]\!]$. We must show that $(k, \mathsf{fst}\{K\}\,\langle v_1, v_2\rangle, \mathsf{fst}\{K'\}\,\langle v_1', v_2'\rangle) \in \mathcal{E}^T[\![\tau']\!]$.

By the OS, it suffices to show that $(k-1, \mathsf{assert}\, K\, v_1, \mathsf{assert}\, K'\, v_1') \in \mathcal{E}^T[\![\tau']\!]$.

By Lemma 5.15, it suffices to show that $(k-1, v_1, v_1') \in \mathcal{E}^T[\![\tau_1]\!]$. This is immediate by Lemma 5.13. □

Lemma 5.26 (Fst compatibility—open relation). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e \leq e' : \tau_1 \times \tau_2]\!]_C^T$ *and* $\tau' = K \sqcap \tau_1 = K' \sqcap \tau_1$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, e \leq \mathsf{fst}\{K'\}\, e' : \tau']\!]_C^T$.

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show that $(k, \gamma(\mathsf{fst}\{K\}\, e), \gamma'(\mathsf{fst}\{K'\}\, e')) \in \mathcal{E}^T[\![\tau']\!]$.

By the definition of substitution, it suffices to show that $(k, \mathsf{fst}\{K\}\, \gamma(e), \mathsf{fst}\{K'\}\, \gamma'(e')) \in \mathcal{E}^T[\![\tau']\!]$.

Instantiate the hypothesis with (k, γ, γ'), providing $(k, \gamma(e), \gamma'(e')) \in \mathcal{E}^T[\![\tau_1 \times \tau_2]\!]$.

Then Lemma 5.14 applies. Consider arbitrary $(k', v, v') \in \mathcal{V}^T[\![\tau_1 \times \tau_2]\!]$. We must show that $(k', \mathsf{fst}\{K\}\, v, \mathsf{fst}\{K'\}\, v') \in \mathcal{E}^T[\![\tau']\!]$; this is immediate by Lemma 5.25. □

Lemma 5.27 (Fst compatibility–pair is bottom). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \bot]\!]_C^T$ *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K_1\}\, e_1 \leq \mathsf{fst}\{K_2\}\, e_2 : \bot]\!]_C^T$.

Proof. By the same reasoning as Lemma 5.24. □

Lemma 5.28 (Snd compatibility).

Proof. Nearly identical to that of Lemma 5.25. □

Lemma 5.29 (Fst compatibility—open relation). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e \leq e' : \tau_1 \times \tau_2]\!]_C^T$ *and* $\tau' = K \sqcap \tau_2 = K' \sqcap \tau_2$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K\}\, e \leq \mathsf{snd}\{K'\}\, e' : \tau]\!]_C^T$.

Proof. Nearly identical to that of Lemma 5.26, using Lemma 5.28. □

Lemma 5.30 (Snd compatibility–pair is bottom). *If* $[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_2 : \bot]\!]_C^T$ *then* $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\, e_1 \leq \mathsf{snd}\{K_2\}\, e_2 : \bot]\!]_C^T$.

Proof. By the same reasoning as Lemma 5.24. □

5.4.3 Binary relation: Compatibility Lemmata

Lemma 5.31 (T-Var compatibility). $\dfrac{(x : K) \in \Gamma}{[\![\Gamma \vdash_{\mathsf{tru}} x \leq x : K]\!]_C^{\mathcal{L}}}$

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\![\Gamma]\!]$.

We must show that $(k, \gamma(x), \gamma'(x)) \in \mathcal{E}^{\mathcal{L}}[\![K]\!]$.

Since $x : K \in \Gamma$, we know that there exist some values v, v' s.t. $\gamma(x) = v$ and $\gamma'(x) = v'$. Since $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\![\Gamma]\!]$, we know that $(k, v, v') \in \mathcal{V}^{\mathcal{L}}[\![K]\!]$. Then we get $(k, v, v') \in \mathcal{E}^{\mathcal{L}}[\![\Gamma]\!]$ immediately since v, v' are already values. □

Lemma 5.32 (T-Nat compatibility). $\overline{\llbracket \Gamma \vdash_{\mathsf{tru}} n \leq n : \mathsf{Nat} \rrbracket^{\mathcal{L}}_{C}}$

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}\llbracket \Gamma \rrbracket$.
We must show $(k, \gamma(n), \gamma'(n)) \in \mathcal{E}^{\mathcal{L}}\llbracket \mathsf{Nat} \rrbracket$.
Note that $\gamma(n) = n$.
Since n is already a value, it suffices to show that $(k, n, n) \in \mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Nat} \rrbracket$.
Unfolding the definition of $\mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Nat} \rrbracket$, this is true. □

Lemma 5.33 (T-Int compatibility). $\overline{\llbracket \Gamma \vdash_{\mathsf{tru}} i \leq i : \mathsf{Int} \rrbracket^{\mathcal{L}}_{C}}$

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}\llbracket \Gamma \rrbracket$.
We must show $(k, \gamma(i), \gamma'(i)) \in \mathcal{E}^{\mathcal{L}}\llbracket \mathsf{Nat} \rrbracket$.
Note that $\gamma(i) = i$.
Since i is already a value, it suffices to show that $(k, i, i) \in \mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Int} \rrbracket$.
Unfolding the definition of $\mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Nat} \rrbracket$, this is true. □

Lemma 5.34 (T-True compatibility). $\overline{\llbracket \Gamma_0 \vdash_{\mathsf{tru}} \mathsf{True} \leq \mathsf{True} : \mathsf{Bool} \rrbracket^{\mathcal{L}}_{C}}$

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}\llbracket \Gamma \rrbracket$.
We must show $(k, \gamma(\mathsf{True}), \gamma'(\mathsf{True})) \in \mathcal{E}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$.
Note that $\gamma(\mathsf{True}) = \mathsf{True}$.
Since True is already a value, it suffices to show that $(k, \mathsf{True}, \mathsf{True}) \in \mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$.
Unfolding the definition of $\mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$, this is true. □

Lemma 5.35 (T-False compatibility). $\overline{\llbracket \Gamma_0 \vdash_{\mathsf{tru}} \mathsf{False} \leq \mathsf{False} : \mathsf{Bool} \rrbracket^{\mathcal{L}}_{C}}$

Proof. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}\llbracket \Gamma \rrbracket$.
We must show $(k, \gamma(\mathsf{False}), \gamma'(\mathsf{False})) \in \mathcal{E}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$.
Note that $\gamma(\mathsf{False}) = \mathsf{False}$.
Since False is already a value, it suffices to show that $(k, \mathsf{False}, \mathsf{False}) \in \mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$.
Unfolding the definition of $\mathcal{V}^{\mathcal{L}}\llbracket \mathsf{Bool} \rrbracket$, this is true. □

Lemma 5.36 (T-Lam compatibility). $\dfrac{\llbracket \Gamma_0, (x_0 : K_0) \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_1 \rrbracket^{\mathcal{L}}_{C}}{\llbracket \Gamma_0 \vdash_{\mathsf{tru}} \lambda(x_0 : K_0).\, e_0 \leq \lambda(x_0 : K_0).\, e_0' : * \to \tau_1 \rrbracket^{\mathcal{L}}_{C}}$

Proof. Let $(k, \gamma, \gamma') \in \mathcal{G}^{T}\llbracket \Gamma_0 \rrbracket$.
We want to show $(k, \gamma(\lambda x_0 : K_0.\, e_0), \gamma'(\lambda x_0.\, K_0 e_0')) \in \mathcal{E}^{T}\llbracket * \to \tau_1 \rrbracket$.
Note that $\gamma(\lambda x_0 : K_0.\, e_0) = \lambda x_0 : K_0.\, \gamma(e_0)$ and similarly for the other.
We want to show $(k - 1, \lambda x_0 : K_0.\, \gamma(e_0), \lambda x_0 : K_0.\, \gamma(e_0')) \in \mathcal{V}^{T}\llbracket * \to \tau_1 \rrbracket$.
Unfolding the value relation:
Let $j \leq k$.
Let $(j, v, v') \in \mathcal{V}^{T}\llbracket * \rrbracket$.
Let K.

We want to show $(j, \mathsf{app}\{K\}\ (\lambda x_0 : K_0.\, \gamma(e_0))\ v, \mathsf{app}\{K\}\ (\lambda x_0 : K_0.\, \gamma(e_0'))\ v') \in \mathcal{E}^T[\![\tau_1 \sqcap K]\!]$.
By the OS, if $\neg K \sim v$ then the application steps to an error and we're done.
Otherwise, $\mathsf{app}\{K\}\ (\lambda x_0 : K_0.\, \gamma(e_0))\ v \longrightarrow_T \mathsf{assert}\, K\ ((\lambda x_0 : K_0.\, \gamma(e_0))\ v) \longrightarrow \mathsf{assert}\, K\ \gamma(e_0)[v/x]$.
By the definition of substitution, $\gamma(e_0)[v/x] = \gamma[x \mapsto v](e_0)$.
Note that $(j-2, \gamma[x \mapsto v](e_0), \gamma'[x \mapsto v](e_0')) \in \mathcal{G}^T[\![\Gamma, x : K]\!]$ by Lemma 4.55 and Lemma 4.57.
Therefore, we can apply the hypothesis to $\gamma[x \mapsto v]$, $\gamma'[x \mapsto v']$, and e_0, e_0' at $j-2$ to get $(j-2, \gamma[x \mapsto v](e_0), \gamma'[x \mapsto v']e_0') \in \mathcal{E}^T[\![\tau_1]\!]$.
Finally, we can apply Lemma 4.58 to get $(j-1, \mathsf{assert}\, K\ \gamma[x \mapsto v](e_0), \mathsf{assert}\, K\ \gamma'[x \mapsto v'](e_0')) \in \mathcal{E}^T[\![\tau_1 \sqcap K]\!]$ which is what we wanted to show. ◻

Lemma 5.37 (T-Pair compatibility).
$$\frac{[\![\Gamma \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_1]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} e_2 \leq e_2' : \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \langle e_1, e_2\rangle \leq \langle e_1', e_2'\rangle : \tau_1 \times \tau_2]\!]^T_C}$$

Proof. Consider arbitrary $(k, \gamma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show $(k, \gamma(\langle e_1, e_2\rangle), \gamma'(\langle e_1', e_2'\rangle)) \in \mathcal{E}^T[\![\tau_1 \times \tau_2]\!]$.

Note that $\gamma(\langle e_1, e_2\rangle) = \langle \gamma(e_1), \gamma(e_2)\rangle$, and similarly for γ', e_1', e_2'. We want to show that $(k, \langle \gamma(e_1), \gamma(e_2)\rangle, \langle \gamma'(e_1'), \gamma'(e_2')\rangle) \in \mathcal{E}^T[\![\tau_1 \times \tau_2]\!]$.

Notice that by instantiating our hypothesis with (k, γ, γ'), we know that $(k, \gamma(e_1), \gamma'(e_1')) \in \mathcal{E}^T[\![\tau_1]\!]$ and $(k, \gamma(e_2), \gamma'(e_2')) \in \mathcal{E}^T[\![\tau_2]\!]$.

By Lemma 5.14, it suffices to show that for any $(k', v_1, v_1') \in \mathcal{V}^T[\![\tau_1]\!]$ where $k' \leq k$, $(k', \langle v_1, e_2\rangle, \langle v_1', e_2'\rangle) \in \mathcal{E}^T[\![\tau_1 \times \tau_2]\!]$.

By Lemma 5.13, we know that $(k', \gamma(e_2), \gamma'(e_2')) \in \mathcal{E}^T[\![\tau_2]\!]$. Again by Lemma 5.14, therefore, it suffices to show that for any $k'' \leq k'$ and v_2, v_2' s.t. $(k'', v_2, v_2') \in \mathcal{V}^T[\![\tau_2]\!]$, $(k'', \langle v_1, v_2\rangle, \langle v_1', v_2'\rangle) \in \mathcal{E}^T[\![\tau_1 \times tau_2]\!]$.

Since these terms are values, it suffices to show that $(k'', \langle v_1, v_2\rangle, \langle v_1', v_2'\rangle) \in \mathcal{V}^T[\![\tau_1 \times \tau_2]\!]$.

Unfolding the definition of $\mathcal{V}$, it suffices to show that $(k'', v_1, v_1') \in \mathcal{V}^T[\![\tau_1]\!]$ and $(k'', v_2, v_2') \in \mathcal{V}^T[\![\tau_2]\!]$; both of these are immediate by Lemma 5.13 from our assumptions. ◻

Lemma 5.38 (T-Cast compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_0]\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\ e_0 \leq \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\ e_0' : K_1 \sqcap K_0 \sqcap \tau_0]\!]^T_C}$$

Proof. Follows immediately from Lemma 5.21. ◻

Lemma 5.39 (T-App compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : * \to \tau_1]\!]^T_C \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_0']\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 \leq \mathsf{app}\{K_1\}\ e_0'\ e_1' : K_1 \sqcap \tau_1]\!]^T_C}$$

Proof. Follows immediately from Lemma 5.23. ◻

Lemma 5.40 (T-AppBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \bot]\!]^T_C \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_0']\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 \leq \mathsf{app}\{K_1\}\ e_0'\ e_1' : \bot]\!]^T_C}$$

Proof. Consider arbitrary $(k, \gamma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show $(k, \gamma(\mathsf{app}\{K_1\}\ e_0\ e_1), \gamma'(\mathsf{app}\{K_1\}\ e_0'\ e_1')) \in \mathcal{E}^T[\![\bot]\!]$.

Apply the first hypothesis to get $(k, \gamma(e_0), \gamma'(e_0')) \in \mathcal{E}^T[\![\bot]\!]$.

Unfolding, there exists some $j \leq k, e_2, e_3$ such that $\gamma(e_0) \longrightarrow_T^j e_2$ and $\gamma'(e_0') \longrightarrow_T^j e_3$ where e_2 and e_3 are irreducible. Either $e_2 = e_3 \in \mathsf{Err}^\bullet$, or $(j, e_2, e_3) \in \mathcal{V}^T[\![\bot]\!]$.

By inversion, it must be the case that $e_2 = e_3 \in \mathsf{Err}^\bullet$, which means that by the OS, $\gamma(\mathsf{app}\{K_1\}\, e_0\, e_1 \longrightarrow_T^{j+1} e_2$ and $\gamma'(\mathsf{app}\{K_1\}\, e_0'\, e_1') \longrightarrow_T^{j+1} e_3$.

Then either, $j+1 > k$, in which case we're done, and otherwise both applications step to the same error within k steps, in which case we're done. □

Lemma 5.41 (T-Fst compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_0 \times \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\, e_0 \leq \mathsf{fst}\{K_0\}\, e_0' : K_0 \sqcap \tau_0]\!]_C^T}$$

Proof. Consider arbitrary $(k, \gamma, \gamma) \in \mathcal{G}^T[\![\Gamma]\!]$.

We must show $(k, \gamma(\mathsf{fst}\{K_0\}\, e_0), \gamma'(\mathsf{fst}\{K_1\}\, e_0')) \in \mathcal{E}^T[\![K_0 \sqcap \tau_0]\!]$.

Note that $\gamma(\mathsf{fst}\{K_0\}\, e_0) = \mathsf{fst}\{K_0\}\, \gamma(e_0)$ and similarly for e_0'.

Assume that there are $j \leq k, e_1$ such that $\mathsf{fst}\{K_0\}\, e_0 \longrightarrow_T^j e_1$ and e_1 is irreducible.

By the OS, it must be the case that there are irreducible e_1', e_1'' such that $\mathsf{fst}\{K_0\}\, e_0 \longrightarrow^{j-2} \mathsf{fst}\{K_0\}\, e_1' \longrightarrow \mathsf{assert}\, K_0\, e_1'' \longrightarrow e_1$.

Unfolding our hypothesis and applying it to the reduction $e_0 \longrightarrow^{j-2} e_1'$, we get that there is an irreducible e_2' such that $e_0' \longrightarrow_T^* e_2'$ and $(k-j+2, e_1', e_2') \in \mathcal{V}^T[\![\tau_0 \times \tau_1]\!]$.

Unfolding the value relation, we get that both e_1' and e_2' are pairs.

Therefore, we have by the OS that there exists e_2'', e_2 such that $\mathsf{fst}\{K_0\}\, e_0' \longrightarrow_T^* \mathsf{fst}\{K_0\}\, e_2' \longrightarrow_T \mathsf{assert}\, K_0\, e_2'' \longrightarrow_T e_2$.

Unfolding the fact that $(k-j+2, e_1', e_2') \in \mathcal{V}^T[\![\tau_0 \times \tau_1]\!]$ gives us that $(k-j+2, e_1'', e_2'') \in \hat{\mathcal{V}}^T[\![\tau_0]\!]$.

Finally, by Lemma 5.15, we get that $(k-j+2, \mathsf{assert}\, K_0\, e_1'', \mathsf{assert}\, K_0\, e_2'') \in \mathcal{E}^T[\![\tau_0 \sqcap K_0]\!]$, which is sufficient to complete the proof. □

Lemma 5.42 (T-FstBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \bot]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\, e_0 \leq \mathsf{fst}\{K_0\}\, e_0' : \bot]\!]_C^T}$$

Proof. Similar reasoning to T-AppBot. □

Lemma 5.43 (T-Snd compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_0 \times \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\, e_0 \leq \mathsf{snd}\{K_1\}\, e_0' : K_1 \sqcap \tau_1]\!]_C^T}$$

Proof. Almost identical to T-Fst. □

Lemma 5.44 (T-SndBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \bot]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\, e_0 \leq \mathsf{snd}\{K_1\}\, e_0' : \bot]\!]_C^T}$$

Proof. Similar reasoning to T-AppBot. □

Lemma 5.45 (T-Binop compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_0]\!]_C^T \quad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathit{binop}\, e_0\, e_1 \leq \mathit{binop}\, e_0'\, e_1' : \Delta(\mathit{binop}, \tau_0, \tau_1)]\!]_C^T}$$

Proof. Let $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.
We want to show $(k, \gamma(\mathit{binop}\, e_0\, e_1), \gamma(\mathit{binop}\, e_0'\, e_1')) \in \mathcal{E}^T[\![\Delta(\mathit{binop} \tau_0, \tau_1)]\!]$.
Note $\gamma(\mathit{binop}\, e_0\, e_1) = \mathit{binop}\, \gamma(e_0)\, \gamma(e_1)$, and similarly for e_0', e_1'.

By the first hypothesis applied to γ, γ' we have $(k, \gamma(e_0), \gamma'(e_0')) \in \mathcal{E}^T[\![\tau_0]\!]$.
Unfolding we get there is a $j \leq k$, and irreducible e_2, e_2' such that $\gamma(e_0) \longrightarrow_T^j e_2$ and $\gamma'(e_0') \longrightarrow_T^* e_2'$.
If $e_2 = e_2' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise $(k - j, e_2, e_2') \in \mathcal{V}^T[\![\tau_0]\!]$.

Note by Lemma 5.13 $(k - j, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma_1]\!]$.
By the second hypothesis applied to γ, γ' and $k - j$, we have $(k - j, \gamma(e_1), \gamma'(e_1')) \in \mathcal{E}^T[\![\tau_1]\!]$.
Unfolding we get there are j', and irreducible e_3, e_3' such that $\gamma(e_1) \longrightarrow_T^{j'} e_3$ and $\gamma'(e_1') \longrightarrow_T^* e_3'$.
If $e_3 = e_3' = \mathsf{Err}^\bullet$ then we're done, because the whole operation errors.
Otherwise $(k - j - j', e_3, e_3') \in \mathcal{V}^T[\![\tau_1]\!]$.

From the definition of Δ, $K_2 = \mathsf{Int}$ or Nat or $\bot$.
In the case of $\bot$, we're done because either τ_0 or τ_1 is a $\bot$, which is a contradiction.
Otherwise, the cases proceed identically, so without loss of generality assume $K_2 = \mathsf{Int}$.
$\tau_0 = \tau_1 = \mathsf{Int}$, and therefore $e_2 = e_2' = i_0$ and $e_3 = e_3' = i_1$.
If $\mathit{binop} = \mathsf{quotient}$ and $i_1 = 0$ then $\mathit{binop}\, i_0\, i_1 \longrightarrow_T \mathsf{DivErr}$, so we're done.
If $\mathit{binop} = \mathsf{quotient}$ and $i_1 \neq 0$, then $\mathit{binop}\, i_0\, i_1 \longrightarrow_T (i_0/i_1)$.
Since $i_0/i_1 \in \mathbb{Z}$, we're done.
If $\mathit{binop} = \mathsf{sum}$ then $\mathit{binop}\, i_0\, i_1 \longrightarrow_T i_0 + i_1$.
Since $i_0 + i_1 \in \mathbb{Z}$, we're done. □

Lemma 5.46 (T-If compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \mathsf{Bool}]\!]_C^T \quad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_0]\!]_C^T \quad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_2 \leq e_2' : \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \leq \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \tau_0 \sqcup \tau_1]\!]_C^T}$$

Proof. Let $(k, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma]\!]$.
We want to show $(k, \gamma(\mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2), \gamma'(\mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2')) \in \mathcal{E}^T[\![\tau_0 \sqcup \tau_1]\!]$.
Note $\gamma(\mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2) = \mathsf{if}\ \gamma(e_0)\ \mathsf{then}\ \gamma(e_1)\ \mathsf{else}\ \gamma(e_2)$ and similarly for e_0', e_1', e_2.
From the first hypothesis applied to γ, γ', we know $(k, \gamma(e_0), \gamma'(e_0')) \in \mathcal{E}^T[\![\mathsf{Bool}]\!]$.
Unfolding, we have that there is a $j \leq k$ and irreducible e_4, e_4' such that $e_0 \longrightarrow_T^j e_4$ and $e_0' \longrightarrow_T^* e_4'$.
If $e_4, e_4' \in \mathsf{Err}^\bullet$ then we're done, because the entire if statement errors.
Otherwise, $(k - j, e_4, e_4') \in \mathcal{V}^T[\![\mathsf{Bool}]\!]$.
Unfolding the location and then the value relation, we get that $e_4 = e_4' = \mathsf{True}$ or $e_4 = e_4' = \mathsf{False}$.

- $e_4 = e_4' = \mathsf{True}$: Note by OS, if $\gamma(e_0)$ then $\gamma(e_1)$ else $\gamma(e_2) \longrightarrow_T^j$ if e_4 then $\gamma(e_1)$ else $\gamma(e_2) \longrightarrow_T \gamma(e_1)$, and similarly for if $\gamma'(e_0')$ then $\gamma'(e_1')$ else $\gamma(e_2')$.
  By Lemma 5.13, we have $(k - j - 1, \gamma, \gamma') \in \mathcal{G}^T[\![\Gamma_1]\!]$.
  From the second hypothesis, we get $(k - j - 1, \gamma(e_1), \gamma'(e_1')) \in \mathcal{E}^T[\![\tau_0]\!]$.
  Finally, by Lemma 4.61, we get $(k - j - 1, \gamma(e_1), \gamma'(e_1')) \in \mathcal{E}^T[\![\tau_0 \sqcup \tau_1]\!]$ which is sufficient to complete the proof.
- $e_4 = e_4' = \mathsf{False}$: same as other case except replace e_1 with e_2.

□

LEMMA 5.47 (T-IfBot compatibility).
$$\frac{\llbracket\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \bot\rrbracket_C^T \quad \llbracket\Gamma_0 \vdash_{\mathsf{tru}} e_1 \leq e_1' : \tau_0\rrbracket_C^T \quad \llbracket\Gamma_0 \vdash_{\mathsf{tru}} e_2 \leq e_2' : \tau_1\rrbracket_C^T}{\llbracket\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \leq \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \bot\rrbracket_C^T}$$

PROOF. Similar reasoning to T-AppBot. □

LEMMA 5.48 (T-Sub compatibility).
$$\frac{\llbracket\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_0\rrbracket_C^T \quad \tau_0 \leqslant: \tau_1}{\llbracket\Gamma_0 \vdash_{\mathsf{tru}} e_0 \leq e_0' : \tau_1\rrbracket_C^T}$$

PROOF. Follows directly from Lemma 5.17. □

5.4.4 Binary relation: Fundamental Property

THEOREM 5.49 (Binary relation is reflexive). *If* $\Gamma \vdash_{\mathsf{tru}} e : \tau$ *then* $\llbracket\Gamma \vdash_{\mathsf{tru}} e \approx e : \tau\rrbracket_C^T$

PROOF. By induction over the typing derivation, using the compatibility lemmata. □

5.5 Context relation—Proofs

5.5.1 Context relation: Compatibility Lemmata

LEMMA 5.50 (T-Ctx-Hole compatibility).
$$\frac{\Gamma' \subseteq \Gamma}{\llbracket\Gamma \vdash_{\mathsf{tru}} [] \approx [] : (\Gamma' \triangleright \tau) \rightsquigarrow \tau\rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket\Gamma' \vdash_{\mathsf{tru}} e \approx e' : \tau\rrbracket$.
We want to show $\llbracket\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau\rrbracket$.
Note $\forall(k, \gamma, \gamma') \in \mathcal{G}^T\llbracket\Gamma\rrbracket$, $(k, \gamma|_{dom(\Gamma')}, \gamma'|_{dom(\Gamma')}) \in \mathcal{G}^T\llbracket\Gamma'\rrbracket$.
And note $\gamma(e) = \gamma|_{dom(\Gamma')}(e)$ and similarly for e'.
Then given such k, γ, γ', we can apply the hypothesis to get that $(k, \gamma(e), \gamma'(e')) \in \mathcal{E}^T\llbracket\tau\rrbracket$, which is sufficient to complete the proof. □

LEMMA 5.51 (T-Ctx-Lam compatibility).
$$\frac{\llbracket\Gamma, (x{:}K) \vdash_{\mathsf{tru}} E \approx E' : (\Gamma', (x{:}K) \triangleright \tau) \rightsquigarrow \tau'\rrbracket_C^T}{\llbracket\Gamma \vdash_{\mathsf{tru}} \lambda(x{:}K).\, E \approx \lambda(x{:}K).\, E' : (\Gamma', (x{:}K) \triangleright \tau) \rightsquigarrow * \to \tau'\rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket\Gamma', (x{:}K) \vdash_{\mathsf{tru}} e \approx e' : \tau\rrbracket$.
We want to show $\llbracket\Gamma \vdash_{\mathsf{tru}} \lambda(x{:}K).\, e \approx \lambda(x{:}K).\, e' : * \to \tau'\rrbracket$.
From our hypothesis we get $\llbracket\Gamma', (x{:}K) \vdash_{\mathsf{tru}} E[e] \approx E[e'] : \tau'\rrbracket$.
Then the case follows from Lemma 5.36. □

LEMMA 5.52 (T-Ctx-Pair-1 compatibility).
$$\frac{\llbracket\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1\rrbracket_C^T \qquad \llbracket\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_2\rrbracket_C^T}{\llbracket\Gamma \vdash_{\mathsf{tru}} \langle E, e\rangle \approx \langle E', e'\rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2\rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket\Gamma' \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau\rrbracket$.

We want to show $[\![\Gamma' \vdash_{\mathsf{tru}} \langle E[e_1], e\rangle \approx \langle E'[e_1'], e\rangle : \tau_1 \times \tau_2]\!]$.
From our first hypothesis, we have $[\![\Gamma' \vdash_{\mathsf{tru}} E[e_1] \approx E'[e_1'] : \tau_1]\!]$.
Then the case follows by Lemma 5.37. □

Lemma 5.53 (T-Ctx-Pair-2 compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_1]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \langle e, E\rangle \approx \langle e', E'\rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2]\!]^T_C}$

Proof. Analagous to T-Ctx-Pair-1. □

Lemma 5.54 (T-Ctx-App-1 compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow * \to \tau_1]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, E\, e \approx \mathsf{app}\{K\}\, E'\, e' : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1]\!]^T_C}$

Proof. Let e, e' such that $[\![\Gamma' \vdash_{\mathsf{tru}} e_1 \approx e_1' : * \to \tau_1]\!]$.
We want to show $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, E[e_1]\, e \approx \mathsf{app}\{K\}\, E'[e_1']\, e' : K \sqcap \tau_1]\!]$.
By the first hypothesis, we have $[\![\Gamma \vdash_{\mathsf{tru}} E[e_1] \approx E'[e_1'] : * \to \tau_1]\!]$.
Then the case follows by Lemma 5.22. □

Lemma 5.55 (T-Ctx-AppBot-1 compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \bot]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, E\, e \approx \mathsf{app}\{K\}\, E'\, e' : (\Gamma' \triangleright \tau) \rightsquigarrow \bot]\!]^T_C}$

Proof. Analogous to T-Ctx-App-1. □

Lemma 5.56 (T-Ctx-App-2 compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : * \to \tau_1]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, e\, E \approx \mathsf{app}\{K\}\, e'\, E' : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1]\!]^T_C}$

Proof. Analogous to T-Ctx-App-1. □

Lemma 5.57 (T-Ctx-AppBot-2 compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \bot]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{app}\{K\}\, e\, E \approx \mathsf{app}\{K\}\, e'\, E' : (\Gamma' \triangleright \tau) \rightsquigarrow \bot]\!]^T_C}$

Proof. Analogous to T-Ctx-App-1. □

Lemma 5.58 (T-Ctx-Fst compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, E \approx \mathsf{fst}\{K\}\, E' : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_1]\!]^T_C}$

Proof. Let e, e' such that $[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_1 \times \tau_2]\!]$.
We want to show $[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, E[e] \approx \mathsf{fst}\{K\}\, E'[e'] : K \sqcap \tau_1]\!]$.
By the hypothesis, we get $[\![\Gamma \vdash_{\mathsf{tru}} E[e] \approx E'[e'] : \tau_1 \times \tau_2]\!]$.
Then the case follows by Lemma 5.25. □

Lemma 5.59 (T-Ctx-FstBot compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\{K\}\, E \approx \mathsf{fst}\{K\}\, E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot]\!]^T_C}$

Proof. Analogous to T-Ctx-Fst. □

Lemma 5.60 (T-Ctx-Snd compatibility). $\dfrac{[\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K\}\, E \approx \mathsf{snd}\{K\}\, E' : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_2]\!]^T_C}$

Proof. Analogous to T-Ctx-Fst. □

Lemma 5.61 (T-Ctx-SndBot compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\{K\}\, E \approx \mathsf{snd}\{K\}\, E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-Fst. □

Lemma 5.62 (T-Ctx-Binop-1 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, E\, e \approx \mathit{binop}\, E'\, e' : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\mathit{binop}, \tau_1, \tau_2) \rrbracket^T_C}$$

Proof. Let e_1, e_1' such that $\llbracket \Gamma \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau \rrbracket$.
We want to show $\llbracket \Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, E[e_1]\, e \approx \mathit{binop}\, E'[e_1']\, e' : \Delta(\mathit{binop}, \tau_1, \tau_2) \rrbracket$.
By the first hypothesis, $\llbracket \Gamma \vdash_{\mathsf{tru}} E[e_1] \approx E'[e_1'] : \tau_1 \rrbracket$.
Then the case follows by Lemma 5.45.

□

Lemma 5.63 (T-Ctx-Binop-2 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, E\, e \approx \mathit{binop}\, E'\, e' : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\mathit{binop}, \tau_1, \tau_2) \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-Binop-1. □

Lemma 5.64 (T-Ctx-Bnd-1 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau' \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{cast}\, \{K_2 \Leftarrow K_1\}\, E \approx \mathsf{cast}\, \{K_2 \Leftarrow K_1\}\, E' : (\Gamma \triangleright \tau) \rightsquigarrow K_2 \sqcap K_1 \sqcap \tau' \rrbracket^T_C}$$

Proof. □

Lemma 5.65 (T-Ctx-If-1 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \mathsf{Bool} \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ E\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \approx \mathsf{if}\ E'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket^T_C}$$

Proof. Let e_0, e_0' such that $\llbracket \Gamma \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau \rrbracket$.
We want to show $\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ E[e_0]\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \approx \mathsf{if}\ E'[e_0']\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \tau_1 \sqcup \tau_2 \rrbracket$.
By the first hypothesis, $\llbracket \Gamma \vdash_{\mathsf{tru}} E[e_0] \approx E'[e_0'] : \mathsf{Bool} \rrbracket$.
The case follows by Lemma 5.46. □

Lemma 5.66 (T-Ctx-IfBot-1 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ E\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \approx \mathsf{if}\ E'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : (\Gamma \triangleright \tau) \rightsquigarrow \bot \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-If-1 □

Lemma 5.67 (T-Ctx-If-2 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} e_b \approx e_b' : \mathsf{Bool} \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ E\ \mathsf{else}\ e_2 \approx \mathsf{if}\ e_b'\ \mathsf{then}\ E'\ \mathsf{else}\ e_2' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-If-1 □

Lemma 5.68 (T-Ctx-IfBot-2 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} e_b \approx e_b' : \bot \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ E\ \mathsf{else}\ e_2 \approx \mathsf{if}\ e_b'\ \mathsf{then}\ E'\ \mathsf{else}\ e_2' : (\Gamma \triangleright \tau) \rightsquigarrow \bot \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-If-1 □

Lemma 5.69 (T-Ctx-If-3 compatibility).
$$\frac{\llbracket \Gamma \vdash_{\mathsf{tru}} e_b \approx e_b' : \mathsf{Bool} \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1 \rrbracket^T_C \qquad \llbracket \Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket^T_C}{\llbracket \Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ E \approx \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket^T_C}$$

Proof. Analagous to T-Ctx-If-1 □

Lemma 5.70 (T-Ctx-IfBot-3 compatibility).
$$\frac{[\![\Gamma \vdash_{\mathsf{tru}} e_b \approx e_b' : \bot]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1]\!]^T_C \qquad [\![\Gamma \vdash_{\mathsf{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ E \approx \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ E' : (\Gamma \triangleright \tau) \rightsquigarrow \bot]\!]^T_C}$$

Proof. Analagous to T-Ctx-If-1 □

5.5.2 Context relation: Fundamental Property

Theorem 5.71 (Context relation is reflexive). *If* $\Gamma \vdash_{\mathsf{tru}} C : (\Gamma' \triangleright \tau) \rightsquigarrow \tau'$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} C \approx C : (\Gamma' \vdash_{\mathsf{tru}} \tau) \rightsquigarrow \tau']\!]$.

Proof. By induction over the typing derivation, using the compatibility lemmata. □

5.6 Check optimization

$$K \setminus \tau = \begin{cases} * & \text{if } \tau \leq K \\ K & \text{otherwise} \end{cases}$$

$\boxed{\Gamma \vdash_{\mathsf{tru}} e : \tau \rightsquigarrow e}$ optimization

$$\textsc{T-Var}\ \frac{(x_0 : K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\mathsf{tru}} x_0 : K_0 \rightsquigarrow x_0} \qquad \textsc{T-Nat}\ \frac{}{\Gamma_0 \vdash_{\mathsf{tru}} n_0 : \mathsf{Nat} \rightsquigarrow n_0} \qquad \textsc{T-Int}\ \frac{}{\Gamma_0 \vdash_{\mathsf{tru}} i_0 : \mathsf{Int} \rightsquigarrow i_0} \qquad \textsc{T-True}\ \frac{}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{True} : \mathsf{Bool} \rightsquigarrow \mathsf{True}}$$

$$\textsc{T-False}\ \frac{}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{False} : \mathsf{Bool} \rightsquigarrow \mathsf{False}} \qquad \textsc{T-Lam}\ \frac{\Gamma_0,\ (x_0 : K_0) \vdash_{\mathsf{tru}} e_0 : \tau_1 \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \lambda(x_0 : K_0).\, e_0 : * \to \tau_1 \rightsquigarrow \lambda(x_0 : K_0).\, e_0'} \qquad \textsc{T-Pair}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_1 \rightsquigarrow e_1'}{\Gamma_0 \vdash_{\mathsf{tru}} \langle e_0, e_1 \rangle : \tau_0 \times \tau_1 \rightsquigarrow \langle e_0', e_1' \rangle}$$

$$\textsc{T-Cast}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{cast}\,\{K_1 \Leftarrow K_0\}\ e_0 : K_1 \sqcap K_0 \sqcap \tau_0 \rightsquigarrow \mathsf{cast}\,\{K_1 \setminus (K_0 \sqcap \tau_0) \Leftarrow K_0 \setminus \tau_0\}\ e_0'}$$

$$\textsc{T-App}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : * \to \tau_1 \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_0' \rightsquigarrow e_1'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 : K_1 \sqcap \tau_1 \rightsquigarrow \mathsf{app}\{K_1 \setminus \tau_1\}\ e_0'\ e_1'} \qquad \textsc{T-AppBot}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \bot \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_0' \rightsquigarrow e_1'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\ e_0\ e_1 : \bot \rightsquigarrow \mathsf{app}\{K_1 \setminus \bot\}\ e_0'\ e_1'}$$

$$\textsc{T-Fst}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \times \tau_1 \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\ e_0 : K_0 \sqcap \tau_0 \rightsquigarrow \mathsf{app}\{K_0 \setminus \tau_0\}\ e_0'} \qquad \textsc{T-FstBot}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \bot \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\ e_0 : \bot \rightsquigarrow \mathsf{fst}\{K_0 \setminus \bot\}\ e_0'}$$

$$\textsc{T-Snd}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \times \tau_1 \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\ e_0 : K_1 \sqcap \tau_1 \rightsquigarrow \mathsf{snd}\{K_1 \setminus \tau_1\}\ e_0'} \qquad \textsc{T-SndBot}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \bot \rightsquigarrow e_0'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\ e_0 : \bot \rightsquigarrow \mathsf{snd}\{K_1 \setminus \bot\}\ e_0'}$$

$$\textsc{T-Binop}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_1 \rightsquigarrow e_1'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathit{binop}\, e_0\, e_1 : \Delta(\mathit{binop}, \tau_0, \tau_1) \rightsquigarrow \mathit{binop}\, e_0'\, e_1'} \qquad \textsc{T-If}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \mathsf{Bool} \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_0 \rightsquigarrow e_1' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_2 : \tau_1 \rightsquigarrow e_2'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_0 \sqcup \tau_1 \rightsquigarrow \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

$$\textsc{T-IfBot}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \bot \rightsquigarrow e_0' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_1 : \tau_0 \rightsquigarrow e_1' \quad \Gamma_0 \vdash_{\mathsf{tru}} e_2 : \tau_1 \rightsquigarrow e_2'}{\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \bot \rightsquigarrow \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'} \qquad \textsc{T-Sub}\ \frac{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_0 \rightsquigarrow e_0' \quad \tau_0 \leqslant: \tau_1}{\Gamma_0 \vdash_{\mathsf{tru}} e_0 : \tau_1 \rightsquigarrow e_0'}$$

Theorem 5.72 (Check-Elision Correctness). *If* $\Gamma \vdash_{\mathsf{tru}} e : \tau \rightsquigarrow e'$, *then* $\Gamma \vdash_{\mathsf{tru}} e \approx^{\mathrm{ctx}} e' : \tau$.

Proof. Consider arbitrary Γ, e, τ, e' s.t. $\Gamma \vdash_{\mathsf{tru}} e : \tau \rightsquigarrow e'$. By Lemma 5.92, $[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau]\!]^{T}_{C}$. By Theorem 5.3, $\Gamma \vdash_{\mathsf{tru}} e \approx^{\mathrm{ctx}} e' : \tau$, which is what was to be shown. □

5.7 Check-elision—Proofs

Lemma 5.73 ($K \setminus \tau$ preserves meets). $K \sqcap \tau = (K \setminus \tau) \sqcap \tau$.

Proof. Immediate by unfolding and lattice properties. □

5.7.1 Check-elision: Compatibility Lemmata

Lemma 5.74 (T-Var compatibility).
$$\frac{(x_0 : K_0) \in \Gamma_0}{[\![\Gamma_0 \vdash_{\mathsf{tru}} x_0 \approx x_0 : K_0]\!]^T_C}$$

Proof. By unfolding and Lemma 5.31. □

Lemma 5.75 (T-Nat compatibility).
$$\frac{}{[\![\Gamma_0 \vdash_{\mathsf{tru}} n_0 \approx n_0 : \mathsf{Nat}]\!]^T_C}$$

Proof. By unfolding and Lemma 5.32. □

Lemma 5.76 (T-Int compatibility).
$$\frac{}{[\![\Gamma_0 \vdash_{\mathsf{tru}} i_0 \approx i_0 : \mathsf{Int}]\!]^T_C}$$

Proof. By unfolding and Lemma 5.32. □

Lemma 5.77 (T-True compatibility).
$$\frac{}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{True} \approx \mathsf{True} : \mathsf{Bool}]\!]^T_C}$$

Proof. By unfolding and Lemma 5.34. □

Lemma 5.78 (T-False compatibility).
$$\frac{}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{False} \approx \mathsf{False} : \mathsf{Bool}]\!]^T_C}$$

Proof. By unfolding and Lemma 5.35. □

Lemma 5.79 (T-Lam compatibility).
$$\frac{[\![\Gamma_0, (x_0 : K_0) \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_1]\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \lambda(x_0 : K_0).\, e_0 \approx \lambda(x_0 : K_0).\, e_0' : * \!\rightarrow\! \tau_1]\!]^T_C}$$

Proof. By unfolding and Lemma 5.36. □

Lemma 5.80 (T-Pair compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_0]\!]^T_C \quad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1]\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \langle e_0, e_1 \rangle \approx \langle e_0', e_1' \rangle : \tau_0 \!\times\! \tau_1]\!]^T_C}$$

Proof. By unfolding and Lemma 5.37. □

Lemma 5.81 (T-Cast compatibility).
$$\frac{[\![\Gamma \vdash_{\mathsf{tru}} e_1 \approx e_2 : \tau]\!]^T_C}{[\![\Gamma \vdash_{\mathsf{tru}} \mathsf{cast}\,\{K' \Leftarrow K\}\; e_1 \approx \mathsf{cast}\,\{K' \setminus (K \sqcap \tau) \Leftarrow K \setminus \tau\}\; e_2 : K' \sqcap K \sqcap \tau]\!]^T_C}$$

Proof. Follows immediately from lattice properties and Lemma 5.21. □

Lemma 5.82 (T-App compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : * \!\rightarrow\! \tau_1]\!]^T_C \quad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_0']\!]^T_C}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\; e_0\; e_1 \approx \mathsf{app}\{K_1 \setminus \tau_1\}\; e_0'\; e_1' : K_1 \sqcap \tau_1]\!]^T_C}$$

Proof. Follows immediately from lattice properties and Lemma 5.23. □

Lemma 5.83 (T-AppBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \bot]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_0']\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{app}\{K_1\}\, e_0\, e_1 \approx \mathsf{app}\{K_1 \setminus \bot\}\, e_0'\, e_1' : \bot]\!]_C^T}$$

Proof. Follows immediately from Lemma 5.24. □

Lemma 5.84 (T-Fst compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_0 \times \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\, e_0 \approx \mathsf{fst}\{K_0 \setminus \tau_0\}\, e_0' : K_0 \sqcap \tau_0]\!]_C^T}$$

Proof. Follows immediately from lattice properties and Lemma 5.26. □

Lemma 5.85 (T-FstBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \bot]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{fst}\{K_0\}\, e_0 \approx \mathsf{fst}\{K_0 \setminus \bot\}\, e_0' : \bot]\!]_C^T}$$

Proof. Follows immediately from Lemma 5.27. □

Lemma 5.86 (T-Snd compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_0 \times \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\, e_0 \approx \mathsf{snd}\{K_1 \setminus \tau_1\}\, e_0' : K_1 \sqcap \tau_1]\!]_C^T}$$

Proof. Follows immediately from lattice properties and Lemma 5.29. □

Lemma 5.87 (T-SndBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \bot]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{snd}\{K_1\}\, e_0 \approx \mathsf{snd}\{K_1 \setminus \bot\}\, e_0' : \bot]\!]_C^T}$$

Proof. Follows immediately from Lemma 5.30. □

Lemma 5.88 (T-Binop compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_0]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathit{binop}\, e_0\, e_1 \approx \mathit{binop}\, e_0'\, e_1' : \Delta(\mathit{binop}, \tau_0, \tau_1)]\!]_C^T}$$

Proof. By unfolding and Lemma 5.45. □

Lemma 5.89 (T-If compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \mathsf{Bool}]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_0]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \approx \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \tau_0 \sqcup \tau_1]\!]_C^T}$$

Proof. By unfolding and Lemma 5.46. □

Lemma 5.90 (T-IfBot compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \bot]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_1 \approx e_1' : \tau_0]\!]_C^T \qquad [\![\Gamma_0 \vdash_{\mathsf{tru}} e_2 \approx e_2' : \tau_1]\!]_C^T}{[\![\Gamma_0 \vdash_{\mathsf{tru}} \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \approx \mathsf{if}\ e_0'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \bot]\!]_C^T}$$

Proof. By unfolding and Lemma 5.47. □

Lemma 5.91 (T-Sub compatibility).
$$\frac{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_0]\!]_C^T \qquad \tau_0 \leqslant: \tau_1}{[\![\Gamma_0 \vdash_{\mathsf{tru}} e_0 \approx e_0' : \tau_1]\!]_C^T}$$

Proof. By unfolding and Lemma 5.48. □

5.7.2 Check-elision: Fundamental Property

Theorem 5.92 (Check-elision is correct for Binary LR). *If* $\Gamma \vdash_{\mathsf{tru}} e : \tau \rightsquigarrow e'$, *then* $[\![\Gamma \vdash_{\mathsf{tru}} e \approx e' : \tau]\!]^{T}_{C}$.

Proof. By induction over the check-elision judgment derivation, using the compatibility lemmata. □

6 Surface

$\boxed{\text{Surface language}}$

$$
\begin{array}{lll}
e & ::= & x \mid n \mid i \mid \mathsf{True} \mid \mathsf{False} \mid \lambda(x{:}K) \rightarrow \tau.\, e \mid \langle e, e\rangle \mid e\ e \mid \mathsf{fst}\, e \mid \mathsf{snd}\, e \mid \mathit{binop}\, e\, e \mid \mathsf{if}\ e\ \mathsf{then}\ e\ \mathsf{else}\ e \\
\tau & ::= & \mathsf{Nat} \mid \mathsf{Int} \mid \mathsf{Bool} \mid \tau \times \tau \mid * \rightarrow \tau \mid * \\
\mathit{binop} & ::= & \mathsf{sum} \mid \mathsf{quotient} \\
\Gamma & ::= & \cdot \mid \Gamma, (x{:}\tau) \\
n & ::= & \mathbb{N} \\
i & ::= & \mathbb{Z}
\end{array}
$$

$$
\Delta^{-1}(\mathit{binop}, \tau) = \begin{cases} \mathsf{Int}, \mathsf{Int} & \text{if } \tau = \mathsf{Int} \\ \mathsf{Nat}, \mathsf{Nat} & \text{if } \tau = \mathsf{Nat} \end{cases}
$$

6.1 Simple Translation

$\boxed{\Gamma \vdash_{\mathsf{sim}} e : \tau \rightsquigarrow e'}$

$$\frac{(x{:}\tau) \in \Gamma}{\Gamma \vdash_{\mathsf{sim}} x : \tau \rightsquigarrow x} \qquad \frac{}{\Gamma \vdash_{\mathsf{sim}} n : \mathsf{Nat} \rightsquigarrow n} \qquad \frac{}{\Gamma \vdash_{\mathsf{sim}} i : \mathsf{Int} \rightsquigarrow i}$$

$$\frac{\Gamma,\ (x{:}\tau) \vdash_{\mathsf{sim}} e : \tau'' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \lambda(x{:}\tau) \to \tau'.\, e : \tau \to \tau' \rightsquigarrow \lambda(x{:}\tau).\, ([\tau' \swarrow \tau'']e')} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} \langle e_1, e_2\rangle : \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2'\rangle}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau \to \tau' \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau'' \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} e_1\ e_2 : \tau' \rightsquigarrow \mathsf{app}\{\tau'\}\, e_1'\ ([\tau \swarrow \tau'']e_2')} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e_1 : * \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau'}{\Gamma \vdash_{\mathsf{sim}} e_1\ e_2 : * \rightsquigarrow \mathsf{app}\{*\}\, (\mathsf{cast}\,\{* \to * \Leftarrow *\}\, e_1)\ e_2}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e : \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{fst}\, e : \tau \rightsquigarrow \mathsf{fst}\{\tau\}\, e'} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e : * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{fst}\, e : * \rightsquigarrow \mathsf{fst}\{*\}\, (\mathsf{cast}\,\{* \times * \Leftarrow *\}\, e')} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e : \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{snd}\, e : \tau' \rightsquigarrow \mathsf{snd}\{\tau'\}\, e'}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e : * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{snd}\, e : * \rightsquigarrow \mathsf{snd}\{*\}\, (\mathsf{cast}\,\{* \times * \Leftarrow *\}\, e')}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2' \qquad \Delta(\mathit{binop}, \tau_1 \sqcup_{\leqslant} \tau_2, \tau_1 \sqcup_{\leqslant} \tau_2) = \tau' \qquad \tau_1 \leqslant: \mathsf{Int} \wedge \tau_2 \leqslant: \mathsf{Int}}{\Gamma \vdash_{\mathsf{sim}} \mathit{binop}\, e_1\, e_2 : \tau' \rightsquigarrow \mathit{binop}\, e_1'\, e_2'}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2' \qquad \tau_1 = * \vee \tau_1 \leqslant: \mathsf{Int} \qquad \tau_2 = * \vee \tau_2 \leqslant: \mathsf{Int}}{\Gamma \vdash_{\mathsf{sim}} \mathit{binop}\, e_1\, e_2 : \tau' \rightsquigarrow \mathit{binop}\, ([\mathsf{Int} \swarrow \tau_1]e_1')\ ([\mathsf{Int} \swarrow \tau_2]e_2')}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_b : \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_1 \sqcup_{\leqslant} \tau_2 \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

$\boxed{[\tau \swarrow \tau']e}$

$$[\tau \swarrow \tau']e = \begin{cases} e & \text{if } \tau \geqslant: \tau' \\ \mathsf{cast}\,\{\tau \Leftarrow \tau'\}\ e & \text{if } \tau \not\geqslant: \tau' \wedge \tau \sim \tau' \end{cases}$$

$\boxed{\tau \sqcup_{\leqslant} \tau'}$

$$\tau \sqcup_{\leqslant} \tau' = \begin{cases} \tau_1 & \text{if } \tau_2 \leqslant: \tau_1 \\ \tau_2 & \text{if } \tau_1 \leqslant: \tau_2 \end{cases}$$

Lemma 6.1 (Typed Translation Imply Simple Typing). *If $\Gamma \vdash_{\mathsf{sim}} e : \tau \rightsquigarrow e'$ then $\Gamma \vdash_{\mathsf{sim}} e' : \tau'$ with $\tau' \leq \tau$.*

Proof. Proceed by induction on the typed translation.

$$\frac{(x{:}\tau) \in \Gamma}{\Gamma \vdash_{\mathsf{sim}} x : \tau \rightsquigarrow x} \qquad \frac{}{\Gamma \vdash_{\mathsf{sim}} n : \mathsf{Nat} \rightsquigarrow n} \qquad \frac{}{\Gamma \vdash_{\mathsf{sim}} i : \mathsf{Int} \rightsquigarrow i}$$

These cases are all immediate.

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} \langle e_1, e_2 \rangle : \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2' \rangle} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e : \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{fst}\, e : \tau \rightsquigarrow \mathsf{fst}\{\tau\}\, e'} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e : \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{snd}\, e : \tau' \rightsquigarrow \mathsf{snd}\{\tau'\}\, e'}$$

These cases are all immediate by the IH applied to their premises and their corresponding typing rule in sim.

$$\frac{\Gamma,\ (x{:}\tau) \vdash_{\mathsf{sim}} e : \tau'' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \lambda(x{:}\tau) \to \tau'.\, e : \tau \to \tau' \rightsquigarrow \lambda(x{:}\tau).\, ([\tau' \swarrow \tau'']e')} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau \to \tau' \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau'' \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} e_1\ e_2 : \tau' \rightsquigarrow \mathsf{app}\{\tau'\}\, e_1'\ ([\tau \swarrow \tau'']e_2')}$$

These cases proceed similarly.
First we apply the IH to all premises.
Then we either, use subsumption to typecheck the body or argument respectively if the types are subtype related, or use T-Cast if they're instead compatible.
Finally, we use the corresponding typing rule to typecheck the elimination form.

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : * \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau'}{\Gamma \vdash_{\mathsf{sim}} e_1\ e_2 : * \rightsquigarrow \mathsf{app}\{*\}\, (\mathsf{cast}\, \{* \to * \Leftarrow *\}\ e_1)\ e_2} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e_1 : * \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau'}{\Gamma \vdash_{\mathsf{sim}} e_1\ e_2 : * \rightsquigarrow \mathsf{app}\{*\}\, (\mathsf{cast}\, \{* \to * \Leftarrow *\}\ e_1)\ e_2}$$

$$\frac{\Gamma \vdash_{\mathsf{sim}} e : * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{fst}\, e : * \rightsquigarrow \mathsf{fst}\{*\}\, (\mathsf{cast}\, \{* \times * \Leftarrow *\}\ e')} \qquad \frac{\Gamma \vdash_{\mathsf{sim}} e : * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{snd}\, e : * \rightsquigarrow \mathsf{snd}\{*\}\, (\mathsf{cast}\, \{* \times * \Leftarrow *\}\ e')}$$

All of these cases proceed similarly.
First, we apply the IH to all premises.
Then we typecheck the casts with T-Cast, where all compatibility constraints are either given as a premise or immediate.
Finally we use the corresponding typing rule to typecheck the elimination form.

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2' \qquad \Delta(\mathit{binop}, \tau_1 \sqcup_{\leqslant} \tau_2, \tau_1 \sqcup_{\leqslant} \tau_2) = \tau' \qquad \tau_1 \leqslant: \mathsf{Int} \land \tau_2 \leqslant: \mathsf{Int}}{\Gamma \vdash_{\mathsf{sim}} \mathit{binop}\, e_1\ e_2 : \tau' \rightsquigarrow \mathit{binop}\, e_1'\ e_2'}$$

By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_1' : \tau_1$.
By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_2' : \tau_2$.
Then we can use subsumption to get both $\Gamma \vdash_{\mathsf{sim}} e_1' : \tau_1 \sqcup_{\leqslant} \tau_2$ and $\Gamma \vdash_{\mathsf{sim}} e_2' : \tau_1 \sqcup_{\leqslant} \tau_2$.
Finally we can typecheck with T-Binop.

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2' \qquad \tau_1 = * \lor \tau_1 \leqslant: \mathsf{Int} \qquad \tau_2 = * \lor \tau_2 \leqslant: \mathsf{Int}}{\Gamma \vdash_{\mathsf{sim}} \mathit{binop}\, e_1\ e_2 : \tau' \rightsquigarrow \mathit{binop}\, ([\mathsf{Int} \swarrow \tau_1]e_1')\ ([\mathsf{Int} \swarrow \tau_2]e_2')}$$

By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_1' : \tau_1$.
By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_2' : \tau_2$.
If $\tau_1 = *$, then $[\mathsf{Int} \swarrow \tau_1]e_1' = \mathsf{cast}\, \{\mathsf{Int} \Leftarrow *\}\ e_1'$, and by the IH we have $\Gamma \vdash_{\mathsf{sim}} \mathsf{cast}\, \{\mathsf{Int} \Leftarrow *\}\ e_1' : \mathsf{Int}$.
Otherwise, $[\mathsf{Int} \swarrow \tau_1]e_1' = e_1'$.
If $\tau_2 = *$, then $[\mathsf{Int} \swarrow \tau_2]e_2' = \mathsf{cast}\, \{\mathsf{Int} \Leftarrow *\}\ e_2'$, and by the IH we have $\Gamma \vdash_{\mathsf{sim}} \mathsf{cast}\, \{\mathsf{Int} \Leftarrow *\}\ e_2' : \mathsf{Int}$.

Otherwise, $[\mathsf{Int} \swarrow \tau_2]e_2' = e_2'$.
Finally we can typecheck with T-Binop and potentially T-Subsumption.

$$\frac{\Gamma \vdash_{\mathsf{sim}} e_b : \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash_{\mathsf{sim}} e_1 : \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{sim}} e_2 : \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{sim}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau_1 \sqcup_{\leqslant} \tau_2 \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_b' : \mathsf{Bool}$.
By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_1' : \tau_1$.
By the IH, we have $\Gamma \vdash_{\mathsf{sim}} e_2' : \tau_2$.
Then by subsumption, we have $\Gamma \vdash_{\mathsf{sim}} e_1' : \tau_1 \sqcup_{\leqslant} \tau_2$ and $\Gamma \vdash_{\mathsf{sim}} e_2' : \tau_1 \sqcup_{\leqslant} \tau_2$.
Finally, we can typecheck with T-If. □

6.2 Truer Transient Translation

$$\tau \setminus K = \begin{cases} * & \text{if } K \leq \tau \\ \tau & \text{otherwise} \end{cases}$$

$\boxed{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow \tau \rightsquigarrow e'}$

$$\frac{(x{:}K) \in \Gamma}{\Gamma \vdash_{\mathsf{tru}} x \Rightarrow K \rightsquigarrow x} \qquad \frac{}{\Gamma \vdash_{\mathsf{tru}} n \Rightarrow \mathsf{Nat} \rightsquigarrow n} \qquad \frac{}{\Gamma \vdash_{\mathsf{tru}} i \Rightarrow \mathsf{Int} \rightsquigarrow i}$$

$$\frac{\Gamma,\ (x{:}K) \vdash_{\mathsf{tru}} e \Leftarrow^{+} \tau \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \lambda(x{:}K) \to \tau.\, e \Rightarrow * \to \tau \rightsquigarrow \lambda(x{:}K).\, e'} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Rightarrow \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{tru}} \langle e_1, e_2 \rangle \Rightarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2' \rangle}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Rightarrow * \to \tau \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Rightarrow \tau'}{\Gamma \vdash_{\mathsf{tru}} e_1\ e_2 \Rightarrow \tau \rightsquigarrow \mathsf{app}\{*\}\ e_1\ e_2} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Rightarrow * \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Rightarrow \tau'}{\Gamma \vdash_{\mathsf{tru}} e_1\ e_2 \Rightarrow * \rightsquigarrow \mathsf{app}\{*\}\ (\mathsf{cast}\ \{* \to * \Leftarrow *\}\ e_1)\ e_2}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\, e \Rightarrow \tau \rightsquigarrow \mathsf{fst}\{*\}\, e'} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\, e \Rightarrow * \rightsquigarrow \mathsf{fst}\{*\}\ (\mathsf{cast}\ \{* \times * \Leftarrow *\}\ e')} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\, e \Rightarrow \tau \rightsquigarrow \mathsf{snd}\{*\}\, e'}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\, e \Rightarrow * \rightsquigarrow \mathsf{snd}\{*\}\ (\mathsf{cast}\ \{* \times * \Leftarrow *\}\ e')}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Rightarrow \tau_2 \rightsquigarrow e_2' \qquad \Delta(\mathit{binop}, \tau_1, \tau_2) = \tau'}{\Gamma \vdash_{\mathsf{tru}} \mathit{binop}\, e_1\, e_2 \Rightarrow \tau' \rightsquigarrow \mathit{binop}\, e_1'\, e_2'}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e_b \Rightarrow \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash_{\mathsf{tru}} e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Rightarrow \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \Rightarrow \tau_1 \sqcup \tau_2 \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

$\boxed{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'}$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow \tau' \rightsquigarrow e' \qquad \tau' \leq \tau}{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e \Rightarrow \tau' \rightsquigarrow e' \qquad \tau' \not\leq K}{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{\Rightarrow} K \rightsquigarrow \mathsf{cast}\ \{K \Leftarrow \lfloor \tau' \rfloor\}\ e'}$$

$\boxed{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{+} \tau \rightsquigarrow e'}$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow \tau \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{+} \tau \rightsquigarrow e'} \qquad \frac{\neg(\exists e'.\, \Gamma \vdash_{\mathsf{tru}} e \Leftarrow \tau \rightsquigarrow e') \qquad \Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{+} \tau \rightsquigarrow e'}$$

$$\boxed{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow \tau \rightsquigarrow e'}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Leftarrow^{+} \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Leftarrow^{+} \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{tru}} \langle e_1, e_2\rangle \Leftarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2'\rangle} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{+} (\tau \setminus \lfloor \tau \rfloor) \times * \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{fst}\, e \Leftarrow \tau \rightsquigarrow \mathsf{fst}\{\lfloor \tau \rfloor\}\, e'}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e \Leftarrow^{+} * \times (\tau \setminus \lfloor \tau \rfloor) \rightsquigarrow e'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{snd}\, e \Leftarrow \tau \rightsquigarrow \mathsf{snd}\{\lfloor \tau \rfloor\}\, e'} \qquad \frac{\Gamma \vdash_{\mathsf{tru}} e_b \Leftarrow^{+} \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash_{\mathsf{tru}} e_1 \Leftarrow^{+} \tau \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Leftarrow^{+} \tau \rightsquigarrow e_2'}{\Gamma \vdash_{\mathsf{tru}} \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \Leftarrow \tau \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

$$\frac{\Gamma \vdash_{\mathsf{tru}} e_1 \Leftarrow^{+} \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash_{\mathsf{tru}} e_2 \Leftarrow^{+} \tau_2 \rightsquigarrow e_2' \qquad \Delta^{-1}(binop, \tau') = \tau_1, \tau_2}{\Gamma \vdash_{\mathsf{tru}} binop\, e_1\, e_2 \Leftarrow \tau' \rightsquigarrow binop\, e_1'\, e_2'}$$

For the purpose of the following proof, assume the tru rules are used in each judgement.

Lemma 6.2 (Typed Translations Imply Truer Transient Typing).

(1) If $\Gamma \vdash e \Rightarrow \tau \rightsquigarrow e'$ then $\Gamma \vdash e' : \tau'$ with $\tau' \leq \tau$.
(2) If $\Gamma \vdash e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'$ then $\Gamma \vdash e' : \tau'$ with $\tau' \leq \tau$.
(3) If $\Gamma \vdash e \Leftarrow^{+} \tau \rightsquigarrow e'$ then $\Gamma \vdash e' : \tau'$ with $\tau' \leq \tau$.
(4) If $\Gamma \vdash e \Leftarrow \tau \rightsquigarrow e'$ then $\Gamma \vdash e' : \tau'$ with $\tau' \leq \tau$.

Proof. All cases proceed by induction over their respective judgement derivations.
This is well founded by the size of the term e, with the caveat that (2) will call into (1) with the same term, but (1) will then reduce the size before calling back into (2) (in the lambda case, through (3)).
Similarly, (3) will call into (2), but by the time it gets back to (3), the term will have been reduced in size in (1) (in the lambda case).
And similarly, (3) will call into (4), but by the time it gets back to (3), the term will have reduced in size.

$$\frac{(x : K) \in \Gamma}{\Gamma \vdash x \Rightarrow K \rightsquigarrow x} \qquad \frac{}{\Gamma \vdash n \Rightarrow \mathsf{Nat} \rightsquigarrow n} \qquad \frac{}{\Gamma \vdash i \Rightarrow \mathsf{Int} \rightsquigarrow i}$$

All of the above cases follow immediately.

$$\frac{\Gamma \vdash e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Rightarrow \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash \langle e_1, e_2\rangle \Rightarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2'\rangle}$$

Follows immediately by the induction hypotheses.

$$\frac{\Gamma \vdash e_1 \Rightarrow * \rightarrow \tau \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Rightarrow \tau'}{\Gamma \vdash e_1\ e_2 \Rightarrow \tau \rightsquigarrow \mathsf{app}\{*\}\, e_1\, e_2} \qquad \frac{\Gamma \vdash e \Rightarrow \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash \mathsf{fst}\, e \Rightarrow \tau \rightsquigarrow \mathsf{fst}\{*\}\, e'} \qquad \frac{\Gamma \vdash e \Rightarrow \tau \times \tau' \rightsquigarrow e'}{\Gamma \vdash \mathsf{snd}\, e \Rightarrow \tau \rightsquigarrow \mathsf{snd}\{*\}\, e'}$$

All of the above cases follow similar reasoning.
We apply the induction hypothesis to each premise.
If the term being eliminated is at type $\bot$, then we use the corresponding $\bot$ rule.
Otherwise we use the corresponding elimination rule with check $*$.

$$\frac{\Gamma \vdash e_1 \Rightarrow * \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Rightarrow \tau'}{\Gamma \vdash e_1\ e_2 \Rightarrow * \rightsquigarrow \mathsf{app}\{*\}\ (\mathsf{cast}\ \{* \to * \Leftarrow *\}\ e_1)\ e_2} \qquad \frac{\Gamma \vdash e \Rightarrow * \rightsquigarrow e'}{\Gamma \vdash \mathsf{fst}\ e \Rightarrow * \rightsquigarrow \mathsf{fst}\{*\}\ (\mathsf{cast}\ \{* \times * \Leftarrow *\}\ e')}$$

$$\frac{\Gamma \vdash e \Rightarrow * \rightsquigarrow e'}{\Gamma \vdash \mathsf{snd}\ e \Rightarrow * \rightsquigarrow \mathsf{snd}\{*\}\ (\mathsf{cast}\ \{* \times * \Leftarrow *\}\ e')}$$

All of the above cases follow similar reasoning.
The reasoning is identical to the previous case, with the note that the boundary term also sends the type below the tag corresponding to the kind of elimination form.

$$\frac{\Gamma \vdash e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Rightarrow \tau_2 \rightsquigarrow e_2' \qquad \Delta(\mathit{binop}, \tau_1, \tau_2) = \tau}{\Gamma \vdash \mathit{binop}\, e_1\, e_2 \Rightarrow \tau' \rightsquigarrow \mathit{binop}\, e_1'\, e_2'}$$

From (1) we get that there is a $\tau_1' \leq \tau_1$ such that $\Gamma \vdash e_1' : \tau_1'$.
From (1) we get that there is a $\tau_2' \leq \tau_2$ such that $\Gamma \vdash e_2' : \tau_2'$.
If $\tau_1' = \bot$ or $\tau_2' = \bot$ then we're done, because $\Delta(\mathit{binop}, \tau_1', \tau_2') = \bot$.
Otherwise, $\tau_1' = \mathsf{Int}$ or Nat and $\tau_2' = \mathsf{Int}$ or Nat. If $\tau_1' \neq \tau_2'$, we can use subsumption to get both e_1' and e_2' at Int to complete the case.
Otherwise they're both at Nat or Int, which is sufficient to complete the case.

$$\frac{\Gamma \vdash e_b \Rightarrow \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash e_1 \Rightarrow \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Rightarrow \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \Rightarrow \tau_1 \sqcup \tau_2 \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

By (1) we have $\exists \tau_b \leq \mathsf{Bool}$ such that $\Gamma \vdash e_b' : \tau_b$.
By (1) we have $\exists \tau_1 \leq \tau$ such that $\Gamma \vdash e_1' : \tau_1$.
By (1) we have $\exists \tau_2 \leq \tau$ such that $\Gamma \vdash e_2' : \tau_2$.
If $\tau_b = \bot$, then we're done by the if bot rule.
Otherwise, we get by the if rule that $\Gamma \vdash \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \tau_1 \sqcup \tau_2$, and that $\tau_1 \sqcup \tau_2 \leq \tau$ by the fact that $\sqcup$ is a greatest lower bound.

$$\frac{\Gamma,\ (x{:}K) \vdash e \Leftarrow^{+} \tau \rightsquigarrow e'}{\Gamma \vdash \lambda(x{:}K) \to \tau.\, e \Rightarrow * \to \tau \rightsquigarrow \lambda(x{:}K).\, e'}$$

By the lambda typing rule for truer typing, we want to show there is a $\tau' \leq \tau$ such that $\Gamma,\ (x{:}K) \vdash e' : \tau'$.
This is immediate from (3) applied to the premise.

$$\frac{\Gamma \vdash e \Rightarrow \tau' \rightsquigarrow e' \qquad \tau' \leq \tau}{\Gamma \vdash e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'}$$

By (1), we have there is a $\tau'' \leq \tau'$ such that $\Gamma \vdash e : \tau''$.
Since $\leq$ is transitive, this completes the case.

$$\frac{\Gamma \vdash e \Rightarrow \tau' \rightsquigarrow e' \qquad \tau' \not\leq K}{\Gamma \vdash e \Leftarrow^{\Rightarrow} K \rightsquigarrow \mathsf{cast}\,\{K \Leftarrow \lfloor \tau' \rfloor\}\ e'}$$

From (1) we have $\tau'' \leq \tau'$ such that $\Gamma \vdash e' : \tau''$.
We want to show there is a $tau''' \leq K$ such that $\Gamma \vdash \mathsf{cast}\,\{K \Leftarrow \lfloor \tau' \rfloor\}\ e' : \tau'''$.
Set $\tau''' \sqcap \lfloor \tau' \rfloor \sqcap K$ to be τ'''.
By the boundary typing rule of truer typing, this typechecks.
The last condition is that $\tau''' \leq K$, which is immediate by the fast that $\sqcap$ is the greatest lower bound.

$$\frac{\neg(\exists e'.\, \Gamma \vdash e \Leftarrow \tau \rightsquigarrow e') \qquad \Gamma \vdash e \Leftarrow^{\Rightarrow} \tau \rightsquigarrow e'}{\Gamma \vdash e \Leftarrow^{+} \tau \rightsquigarrow e'}$$

Immediate by (2).

$$\frac{\Gamma \vdash e \Leftarrow \tau \rightsquigarrow e'}{\Gamma \vdash e \Leftarrow^{+} \tau \rightsquigarrow e'}$$

Immediate by (4).

$$\frac{\Gamma \vdash e_1 \Leftarrow^{+} \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Leftarrow^{+} \tau_2 \rightsquigarrow e_2'}{\Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1', e_2' \rangle}$$

Immediate by (3) and induction.

$$\frac{\Gamma \vdash e \Leftarrow^{+} (\tau \setminus \lfloor \tau \rfloor) \times * \rightsquigarrow e'}{\Gamma \vdash \mathsf{fst}\, e \Leftarrow \tau \rightsquigarrow \mathsf{fst}\{\lfloor \tau \rfloor\}\, e'}$$

By our induction hypothesis, we have that there is some $\tau' \leq (\tau \setminus \lfloor \tau \rfloor) \times *$ such that $\Gamma \vdash e' : \tau'$.
If $\tau' = \bot$, then we're done by the fst bot rule.
Otherwise, $\tau' = \tau_1' \times \tau_2'$, and $\tau_1' \leq \tau \setminus \lfloor \tau \rfloor$.
By the fst projection typing rule, we have that $\Gamma \vdash \mathsf{fst}\{\lfloor \tau \rfloor\}\, e' : \tau_1' \sqcap \lfloor \tau \rfloor$.
It suffices to show that $\tau_1' \sqcap \lfloor \tau \rfloor \leq \tau$.
If $\tau \setminus \lfloor \tau \rfloor = *$, then $\lfloor \tau \rfloor \leq \tau$, which means $\tau_1' \sqcap \lfloor \tau \rfloor \leq \lfloor \tau \rfloor \leq \tau$.
Otherwise, $\tau \setminus \lfloor \tau \rfloor = \tau$, which means $\tau_1' \leq \tau$ and therefore $\tau_1' \sqcap \lfloor \tau \rfloor \leq \tau$.

$$\frac{\Gamma \vdash e \Leftarrow^{+} * \times (\tau \setminus \lfloor \tau \rfloor) \rightsquigarrow e'}{\Gamma \vdash \mathsf{snd}\, e \Leftarrow \tau \rightsquigarrow \mathsf{snd}\{\lfloor \tau \rfloor\}\, e'}$$

Not meaningfully different from the previous case regarding fst .

$$\frac{\Gamma \vdash e_b \Leftarrow^+ \mathsf{Bool} \rightsquigarrow e_b' \qquad \Gamma \vdash e_1 \Leftarrow^+ \tau \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Leftarrow^+ \tau \rightsquigarrow e_2'}{\Gamma \vdash \mathsf{if}\ e_b\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \Leftarrow \tau \rightsquigarrow \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2'}$$

By (3) we have $\exists \tau_b \leq \mathsf{Bool}$ such that $\Gamma \vdash e_b' : \tau_b$.
By (3) we have $\exists \tau_1 \leq \tau$ such that $\Gamma \vdash e_1' : \tau_1$.
By (3) we have $\exists \tau_2 \leq \tau$ such that $\Gamma \vdash e_2' : \tau_2$.
If $\tau_b = \bot$, then we're done by the if bot rule.
Otherwise, we get by the if rule that $\Gamma \vdash \mathsf{if}\ e_b'\ \mathsf{then}\ e_1'\ \mathsf{else}\ e_2' : \tau_1 \sqcup \tau_2$, and that $\tau_1 \sqcup \tau_2 \leq \tau$ by the fact that $\sqcup$ is a greatest lower bound.

$$\frac{\Gamma \vdash e_1 \Leftarrow^+ \tau_1 \rightsquigarrow e_1' \qquad \Gamma \vdash e_2 \Leftarrow^+ \tau_2 \rightsquigarrow e_2' \qquad \Delta^{-1}(\mathit{binop}, \tau') = \tau_1, \tau_2}{\Gamma \vdash \mathit{binop}\, e_1\, e_2 \Leftarrow \tau' \rightsquigarrow \mathit{binop}\, e_1'\, e_2'}$$

By (3) we have $\exists \tau_1' \leq \tau_1$ such that $\Gamma \vdash e_1' : \tau_1'$.
By (3) we have $\exists \tau_2' \leq \tau_2$ such that $\Gamma \vdash e_2' : \tau_2'$.
By the definition of Δ^{-1}, either $\tau_1 = \tau_2 = \mathsf{Int}$ or $\tau_1 = \tau_2 = \mathsf{Nat}$.
If $\tau_1' = \bot$ or $\tau_2' = \bot$, then we're done because $\Delta(\mathit{binop}, \tau_1', \tau_2') = \bot$.
Otherwise, we have $\tau_1' = \mathsf{Int}$ or Nat and similarly for τ_2'.
If $\tau_1' \neq \tau_2'$, then we can use subsumption to get both at Int and complete the case.
Otherwise, we get that both are Int or Nat, which is sufficient to complete the case. □

Theorem 6.3 (Typed Translation Implies Truer Transient Typing).
If $\Gamma \vdash e \Rightarrow \tau \rightsquigarrow e'$ *then* $\Gamma \vdash e : \tau'$ *where* $\tau' \leq \tau$.