Bidirectional Type Checking for Relational Properties (Appendix)

Ezgi Çiçek Weihao Qu Gilles Barthe Marco Gaboardi Deepak Garg

Contents

List of Figures

List of Theorems and Lemmas

1 relSTLC

1.1 Syntax of relSTLC

$$\begin{array}{llll} \text{Types} & \tau & ::= & \text{bool}_r \mid \text{bool}_u \mid \tau_1 \to \tau_2 \\ \text{Expressions} & e & ::= & x \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \lambda x.e \mid e_1\ e_2 \\ \text{Value} & v & := & \text{true} \mid \text{false} \mid \lambda x.e \end{array}$$

Figure 1: Syntax of values and expressions in relSTLC

$$\begin{array}{llll} \text{Types} & \tau & ::= & \text{bool}_r \mid \text{bool}_u \mid \tau_1 \to \tau_2 \\ \text{Expressions} & e & ::= & x \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \lambda x.e \mid e_1\ e_2 \mid (e : \tau) \\ \text{Value} & v & := & \text{true} \mid \text{false} \mid \lambda x.e \end{array}$$

Figure 2: Syntax of values and expressions in BirelSTLC

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x : \tau}\ \textbf{r-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Gamma \vdash \mathtt{b} \backsim \mathtt{b} : \text{bool}_r}\ \textbf{r-bool} \qquad \frac{\mathtt{b}_1, \mathtt{b}_2 \in \{\text{true}, \text{false}\}}{\Gamma \vdash \mathtt{b}_1 \backsim \mathtt{b}_2 : \text{bool}_u}\ \textbf{r-u-bool}$$

$$\frac{\Gamma \vdash e \backsim e' : \text{bool}_r \qquad \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Gamma \vdash e_2 \backsim e_2' : \tau}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \backsim \text{if } e' \text{ then } e_1' \text{ else } e_2' : \tau}\ \textbf{r-if} \qquad \frac{\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 : \tau_2}{\Gamma \vdash \lambda x.e_1 \backsim \lambda x.e_2 : \tau_1 \to \tau_2}\ \textbf{r-lam}$$

$$\frac{\Gamma \vdash e_1 \backsim e_1' : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 \backsim e_2' : \tau_1}{\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' : \tau_2}\ \textbf{r-app} \qquad \frac{\Gamma \vdash e_1 \backsim e_2 : \tau \qquad \models \tau \sqsubseteq \tau'}{\Gamma \vdash e_1 \backsim e_2 : \tau'}\ \textbf{r-}\sqsubseteq$$

Figure 3: relSTLC typing rules

$$\frac{}{\models \mathrm{bool}_r \sqsubseteq \mathrm{bool}_u}\ \textbf{bool} \qquad \frac{\models \tau_1' \sqsubseteq \tau_1 \qquad \models \tau_2 \sqsubseteq \tau_2'}{\models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'}\ \to \qquad \frac{}{\models \tau \sqsubseteq \tau}\ \textbf{refl}$$

$$\frac{\models \tau_1 \sqsubseteq \tau_2 \qquad \models \tau_2 \sqsubseteq \tau_3}{\models \tau_1 \sqsubseteq \tau_3}\ \textbf{trans}$$

Figure 4: relSTLC subtyping rules

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x \uparrow \tau}\ \textbf{alg-r-var} \qquad \frac{\mathtt{b} \in \{\mathrm{true}, \mathrm{false}\}}{\Gamma \vdash \mathtt{b} \backsim \mathtt{b} \uparrow \mathrm{bool}_r}\ \textbf{alg-r-bool}$$

$$\frac{\mathtt{b}_1, \mathtt{b}_2 \in \{\mathrm{true}, \mathrm{false}\} \qquad b_1 \neq b_2}{\Gamma \vdash \mathtt{b}_1 \backsim \mathtt{b}_2 \downarrow \mathrm{bool}_u}\ \textbf{alg-r-u-bool}$$

$$\frac{\Gamma \vdash e \backsim e' \uparrow \mathrm{bool}_r \qquad \Gamma \vdash e_1 \backsim e_1' \downarrow \tau \qquad \Gamma \vdash e_2 \backsim e_2' \downarrow \tau}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \backsim \text{if } e' \text{ then } e_1' \text{ else } e_2' \downarrow \tau}\ \textbf{alg-r-if}$$

$$\frac{\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 \downarrow \tau_2}{\Gamma \vdash \lambda x.e_1 \backsim \lambda x.e_2 \downarrow \tau_1 \to \tau_2}\ \textbf{alg-r-lam} \qquad \frac{\Gamma \vdash e_1 \backsim e_1' \uparrow \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 \backsim e_2' \downarrow \tau_1}{\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' \uparrow \tau_2}\ \textbf{alg-r-app}$$

$$\frac{\Gamma \vdash e \backsim e' \uparrow \tau' \qquad \models \tau' \leq \tau}{\Gamma \vdash e \backsim e' \downarrow \tau}\ \textbf{alg-}\!\uparrow\!\downarrow \qquad \frac{\Gamma \vdash e \backsim e' \downarrow \tau}{\Gamma \vdash (e : \tau) \backsim (e' : \tau) \uparrow \tau}\ \textbf{alg-r-anno-}\!\uparrow$$

Figure 5: relSTLC agorithmic typing rules

$$\frac{}{\models \mathrm{bool}_r \leq \mathrm{bool}_r}\ \textbf{alg-bl-r} \qquad \frac{}{\models \mathrm{bool}_u \leq \mathrm{bool}_u}\ \textbf{alg-bl-u} \qquad \frac{}{\models \mathrm{bool}_r \leq \mathrm{bool}_u}\ \textbf{alg-bool}$$

$$\frac{\models \tau_1' \leq \tau_1 \qquad \models \tau_2 \leq \tau_2'}{\models \tau_1 \to \tau_2 \leq \tau_1' \to \tau_2'}\ \textbf{alg-}\!\to$$

Figure 6: relSTLC algorithmic subtyping rules

$$\begin{array}{lcl}
|.| & : & \text{Expression} \to \text{Expression} \\
\\
|x| & = & x \\
|e_1\ e_2| & = & |e_1|\ |e_2| \\
\vdots & & \\
|(e : \tau)| & = & |e|
\end{array}$$

Figure 7: relSTLC annotation erasure

1.2 relSTLC Lemmas

Lemma 1 (Reflexivity of algorithmic subtyping in relSTLC)
The reflexivity property holds for algorithm subtyping for relSTLC.
$\models \tau \le \tau$.

Proof. By induction on the binary type.

Case $\mathbf{bool}_r$
It is proved by **alg-bl-r**.

Case $\mathbf{bool}_u$
It is proved by **alg-bl-u**.

Case $\tau_1 \to \tau_2$
By **alg-**$\to$, TS: $\models \tau_1 \le \tau_1$ and $\models \tau_2 \le \tau_2$, which is obtained by IH on τ_1 and τ_2 respectively.

□

Lemma 2 (Transitivity of algorithmic subtyping in relSTLC)
The transitivity property holds for algorithm subtyping for relSTLC.
$\models \tau_1 \le \tau_2$ and $\models \tau_2 \le \tau_3$, then $\models \tau_1 \le \tau_3$.

Proof. By simultaneous induction on the first two subtyping derivation.

Case $\dfrac{}{\models \mathbf{bool}_r \le \mathbf{bool}_r}\ \textbf{alg-bl-r}\ ,\ \dfrac{}{\models \mathbf{bool}_r \le \mathbf{bool}_r}\ \textbf{alg-bl-r}$

TS: $\models \text{bool}_r \le \text{bool}_r$, which is proved by **alg-bl-r**.

Case $\dfrac{}{\models \mathbf{bool}_u \le \mathbf{bool}_u}\ \textbf{alg-bl-u},\ \dfrac{}{\models \mathbf{bool}_u \le \mathbf{bool}_u}\ \textbf{alg-bl-u}$
TS: $\models \text{bool}_u \le \text{bool}_u$, which is proved by **alg-bl-u**.

Case $\dfrac{}{\models \mathbf{bool}_r \le \mathbf{bool}_u}\ \textbf{alg-bool}\ ,\ \dfrac{}{\models \mathbf{bool}_u \le \mathbf{bool}_u}\ \textbf{alg-bl-u}$
TS: $\models \text{bool}_r \le \text{bool}_u$, which is proved by **alg-bool**.

Case $\dfrac{}{\models \mathbf{bool}_r \le \mathbf{bool}_r}\ \textbf{alg-bl-r}\ ,\ \dfrac{}{\models \mathbf{bool}_r \le \mathbf{bool}_u}\ \textbf{alg-bool}$
TS: TS: $\models \text{bool}_r \le \text{bool}_u$, which is proved by **alg-bool**.

Case $\dfrac{\models \tau_1' \le \tau_1 \qquad \models \tau_2 \le \tau_2'}{\models \tau_1 \to \tau_2 \le \tau_1' \to \tau_2'}\ \textbf{alg-}\!\to,\ \dfrac{\models \tau_1'' \le \tau_1' \qquad \models \tau_2' \le \tau_2''}{\models \tau_1' \to \tau_2' \le \tau_1'' \to \tau_2''}\ \textbf{alg-}\!\to$
TS: $\models \tau_1 \to \tau_2 \le \tau_1'' \to \tau_2''$.
By IH on $\models \tau_1'' \le \tau_1'$ and $\models \tau_1' \le \tau_1$, $\models \tau_1'' \le \tau_1$.

By IH on $\models \tau_2' \le \tau_1''$ and $\models \tau_2 \le \tau_2'$, $\models \tau_2 \le \tau_2''$.

It is proved by using the two statements and rule **alg-**$\to$.

□

Next, we can show that the algorithmic formulation of subtyping ($\tau_1 \le \tau_2$) coincides with the declarative formulation of subtyping ($\tau_1 \sqsubseteq \tau_2$). We do this in two steps.

Lemma 3 **(Soundness of algorithmic subtyping in relSTLC)**
If $\models \tau \leq \tau'$ then $\models \tau \sqsubseteq \tau'$.

Proof. Proof is by straightforward induction on the given algorithmic subtyping derivation.

Case $\dfrac{\models \tau_1' \leq \tau_1 \qquad \models \tau_2 \leq \tau_2'}{\models \tau_1 \to \tau_2 \leq \tau_1' \to \tau_2'}\ \textbf{alg-}\!\to$
TS: $\vDash \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'$
By IH on $\models \tau_1' \leq \tau_1$, $\vDash \tau_1' \sqsubseteq \tau_1$.
By IH on $\models \tau_2 \leq \tau_2'$, $\vDash \tau_2 \sqsubseteq \tau_2'$.
It is proved using these two statements and subtyping rule $\to$.

Case $\dfrac{}{\models \mathbf{bool}_r \leq \mathbf{bool}_u}\ \textbf{alg-bool}$
TS: $\vDash \mathrm{bool}_r \sqsubseteq \mathrm{bool}_u$, which is proved by subtyping rule **bool**.

Case $\dfrac{}{\models \mathbf{bool}_u \leq \mathbf{bool}_u}\ \textbf{alg-bl-u}$
TS: $\vDash \mathrm{bool}_u \sqsubseteq \mathrm{bool}_u$, which is proved by subtyping rule **refl**.

Case $\dfrac{}{\models \mathbf{bool}_r \leq \mathbf{bool}_r}\ \textbf{alg-bl-r}$
TS: $\vDash \mathrm{bool}_r \sqsubseteq \mathrm{bool}_r$, which is proved by subtyping rule **refl**.

□

Lemma 4 **(Completeness of algorithmic subtyping in relSTLC)**
If $\models \tau \sqsubseteq \tau'$ then $\models \tau \leq \tau'$.

Proof. Proof is by induction on the given subtyping derivation.

Case $\dfrac{\models \tau_1' \sqsubseteq \tau_1 \qquad \models \tau_2 \sqsubseteq \tau_2'}{\models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'}\ \to$
TS: $\vDash \tau_1 \to \tau_2 \leq \tau_1' \to \tau_2'$
By IH on $\models \tau_1' \sqsubseteq \tau_1$, $\vDash \tau_1' \leq \tau_1$.
By IH on $\models \tau_2 \sqsubseteq \tau_2'$, $\vDash \tau_2 \leq \tau_2'$.
It is proved using these two statements and algorithmic subtyping rule **alg-→**.

Case $\dfrac{}{\models \mathbf{bool}_r \sqsubseteq \mathbf{bool}_u}\ \textbf{bool}$
TS: $\vDash \mathrm{bool}_r \leq \mathrm{bool}_u$, which is proved by algorithmic subtyping rule **alg-bool**.

Case $\dfrac{}{\models \tau \sqsubseteq \tau}\textbf{refl}$
TS: $\vDash \tau \leq \tau$, which is proved by Lemma 1.

Case $\dfrac{\models \tau_1 \sqsubseteq \tau_2 \qquad \models \tau_2 \sqsubseteq \tau_3}{\models \tau_1 \sqsubseteq \tau_3}\textbf{trans}$
TS: $\vDash \tau_1 \leq \tau_3$.
By IH on $\models \tau_1 \sqsubseteq \tau_2$, $\models \tau_1 \leq \tau_2$.
By IH on $\models \tau_2 \sqsubseteq \tau_3$, $\models \tau_2 \leq \tau_3$.
It is proved using these two statements and Lemma 2.

□

Lemma 5 **(Soundness of algorithmic typing in relSTLC)**
The following holds, $|.|$ is the annotation erasure function.

1. If $\Gamma \vdash e_1 \backsim e_2 \downarrow \tau$ then $\Gamma \vdash |e_1| \backsim |e_2| : \tau$.
2. If $\Gamma \vdash e_1 \backsim e_2 \uparrow \tau$ then $\Gamma \vdash |e_1| \backsim |e_2| : \tau$.

Proof. Proof is by simultaneous induction on the given algorithmic checking and inference derivations and Lemma 3.

Case $\dfrac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x \uparrow \tau}$ **alg-r-var**

TS: $\Gamma \vdash |x| \backsim |x| : \tau$
It is proved by using the premise $\Gamma(x) = \tau$ and typing rule **r-var**.

Case $\dfrac{\Gamma \vdash e \backsim e' \uparrow \mathbf{bool}_r \qquad \Gamma \vdash e_1 \backsim e_1' \downarrow \tau \qquad \Gamma \vdash e_2 \backsim e_2' \downarrow \tau}{\Gamma \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 \backsim \mathbf{if}\ e'\ \mathbf{then}\ e_1'\ \mathbf{else}\ e_2' \downarrow \tau}$ **alg-r-if**

TS: $\Gamma \vdash |\text{if } e \text{ then } e_1 \text{ else } e_2| \backsim |\text{if } e' \text{ then } e_1' \text{ else } e_2'| : \tau$
By IH2 on $\Gamma \vdash e \backsim e' \uparrow \text{bool}_r$,$\Gamma \vdash |e| \backsim |e'| : \text{bool}_r$.
By IH on $\Gamma \vdash e_1 \backsim e_1' \downarrow \tau$, $\Gamma \vdash |e_1| \backsim |e_1'| : \tau$.
By IH on $\Gamma \vdash e_2 \backsim e_2' \downarrow \tau$, $\Gamma \vdash |e_2| \backsim |e_2'| : \tau$.
Because $|\text{if } e \text{ then } e_1 \text{ else } e_2| = \text{if } |e| \text{ then } |e_1| \text{ else } |e_2|$. By using these three statements and typing rule **r-if**, we get the conclusion.

Case $\dfrac{\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 \downarrow \tau_2}{\Gamma \vdash \lambda x.e_1 \backsim \lambda x.e_2 \downarrow \tau_1 \to \tau_2}$ **alg-r-lam**

TS: $\Gamma \vdash \lambda x.|e_1| \backsim \lambda x.|e_1| : \tau_1 \to \tau_2$.
By IH on $\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 \downarrow \tau_2$, $\Gamma, x : \tau_1 \vdash |e_1| \backsim |e_2| : \tau_2$.
It is proved by the above statment and tying rule **r-lam**.

Case $\dfrac{\Gamma \vdash e \backsim e' \uparrow \tau' \qquad \models \tau' \leq \tau}{\Gamma \vdash e \backsim e' \downarrow \tau}$ **alg-↑↓**

TS: $\Gamma \vdash |e| \backsim |e'| : \tau$.
By Lemma 3 on $\models \tau \leq \tau'$, $\models \tau \sqsubseteq \tau'$.
By IH2 on $\Gamma \vdash e \backsim e' \uparrow \tau$, $\Gamma, \vdash |e| \backsim |e'| : \tau$.
It is proved by the above statment and tying rule **r-$\sqsubseteq$**.

Case $\dfrac{\Gamma \vdash e \backsim e' \downarrow \tau}{\Gamma \vdash (e : \tau) \backsim (e' : \tau) \uparrow \tau}$ **alg-r-anno-↑**

TS: $\Gamma \vdash |(e : \tau)| \backsim |(e' : \tau)| : \tau$, which is simplified as $\Gamma \vdash e \backsim e' : \tau$.
By IH on $\Gamma \vdash e \backsim e' \downarrow \tau$, $\Gamma \vdash e \backsim e' : \tau$.
It is proved by the above statement.

□

Lemma 6 (Completeness of algorithmic typing in relSTLC)
If $\Gamma \vdash e_1 \backsim e_2 : \tau$ then there exists e_1' and e_2' such that $\Gamma \vdash e_1' \backsim e_2' \downarrow \tau$ and $|e_i'| = e_i$ for $i \in \{1, 2\}$.

Proof. Proof is by induction on the given typing derivation and Lemma 4.

Case $\dfrac{\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 : \tau_2}{\Gamma \vdash \lambda x.e_1 \backsim \lambda x.e_2 : \tau_1 \to \tau_2}$ **r-lam**

By IH on $\Gamma, x : \tau_1 \vdash e_1 \backsim e_2 : \tau_2$, $\exists e_1', e_2'.\Gamma, x : \tau_1 \vdash e_1' \backsim e_2' \downarrow \tau_2$ and $|e_i'| = e_i$.
Then, using the above statement and algorithmic typing rule **alg-r-lam**, we can construct the derivation where $e_1'' = \lambda x.e_1'$ and $e_2'' = \lambda x.e_2'$.

$$\Gamma \vdash e_1'' \backsim e_2'' \downarrow \tau_1 \to \tau_2$$

And we show that $|e_i''| = |\lambda x.e_i'| = \lambda x.|e_i'| = \lambda x.e_i'$.

Case $\dfrac{\Gamma \vdash e \backsim e' : \mathbf{bool}_r \qquad \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Gamma \vdash e_2 \backsim e_2' : \tau}{\Gamma \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 \backsim \mathbf{if}\ e'\ \mathbf{then}\ e_1'\ \mathbf{else}\ e_2' : \tau}$ **r-if**

By IH on $\Gamma \vdash e \backsim e' : \mathrm{bool}_r$, $\exists e'', e'''.\Gamma, \vdash e'' \backsim e''' \downarrow \mathrm{bool}_r$ and $|e''| = e, |e'''| = e'$.
By using algorithmic typing rule **alg-r-anno-↑**, $\Gamma \vdash (e'' : \mathrm{bool}_r) \backsim (e''' : \mathrm{bool}_r) \uparrow \mathrm{bool}_r$.
By IH on $\Gamma \vdash e_1 \backsim e_1' : \tau$, $\exists e_1'', e_1'''.\Gamma, \vdash e_1'' \backsim e_1''' \downarrow \tau$ and $|e_1''| = e_1, |e_1'''| = e_1'$.
By IH on $\Gamma \vdash e_2 \backsim e_2' : \tau$, $\exists e_2'', e_2'''.\Gamma, \vdash e_2'' \backsim e_2''' \downarrow \tau$ and $|e_2''| = e_2, |e_2'''| = e_2'$.
Then, using the above statement and algorithmic typing rule **alg-r-if**, we can construct the derivation where $e_3 = \mathrm{if}\ (e'' : \mathrm{bool}_r)\ \mathrm{then}\ e_1''\ \mathrm{else}\ e_2''$ and $e_4 = \mathrm{if}\ (e''' : \mathrm{bool}_r)\ \mathrm{then}\ e_1'''\ \mathrm{else}\ e_2'''$.

$$\Gamma \vdash e_3 \backsim e_4 \downarrow \tau$$

And we show that $|e_3| = |\mathrm{if}\ (e'' : \mathrm{bool}_r)\ \mathrm{then}\ e_1''\ \mathrm{else}\ e_2''| = \mathrm{if}\ |(e'' : \mathrm{bool}_r)|\ \mathrm{then}\ |e_1''|\ \mathrm{else}\ |e_2''| = \mathrm{if}\ e\ \mathrm{then}\ e_1\ \mathrm{else}\ e_2$, similarly for e_4.

Case $\dfrac{\Gamma \vdash e_1 \backsim e_1' : \tau_1 \rightarrow \tau_2 \qquad \Gamma \vdash e_2 \backsim e_2' : \tau_1}{\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' : \tau_2}$ **r-app**

By IH on $\Gamma \vdash e_1 \backsim e_1' : \tau_1 \to \tau_2$, $\exists e_1'', e_1'''.\Gamma, \vdash e_1'' \backsim e_1''' \downarrow \tau_1 \to \tau_2$ and $|e_1''| = e_1, |e_1'''| = e_1'$.
By using algorithmic typing rule **alg-r-anno-↑**, $\Gamma \vdash (e_1'' : \tau_1 \to \tau_2) \backsim (e_1''' : \tau_1 \to \tau_2) \uparrow \tau_1 \to \tau_2$.
By IH on $\Gamma \vdash e_2 \backsim e_2' : \tau_1$, $\exists e_2'', e_2'''.\Gamma, \vdash e_2'' \backsim e_2''' \downarrow \tau_1$ and $|e_2''| = e_2, |e_2'''| = e_2'$.
Then, using the above statement and algorithmic typing rule **alg-r-app**, we can construct the derivation where $e_3 = e_1''\ e_2''$ and $e_4 = e_1'''\ e_2'''$.

$$\Gamma \vdash e_3 \backsim e_4 \downarrow \tau_2$$

And we show that $|e_3| = |e_1''\ e_2''| = |e_1''|\ |e_2''| = e_1\ e_2$, , similarly for e_4.

Case $\dfrac{\Gamma \vdash e_1 \backsim e_2 : \tau \qquad \models \tau \sqsubseteq \tau'}{\Gamma \vdash e_1 \backsim e_2 : \tau'}$ **r-⊑**

By IH on $\Gamma \vdash e_1 \backsim e_2 : \tau$, $\exists e_1', e_2'.\Gamma, \vdash e_1' \backsim e_2' \downarrow \tau$ and $|e_1'| = e_1, |e_2'| = e_2$.
By using algorithmic typing rule **alg-r-anno-↑**, $\Gamma \vdash (e_1' : \tau) \backsim (e_2' : \tau) \uparrow \tau$.
By using Lemma 3 on premise $\models \tau \sqsubseteq \tau'$, $\models \tau \leq \tau'$.
Then, using the above statement and algorithmic typing rule **alg-r-↑↓**, we can construct the derivation where $e_3 = (e_1' : \tau)$ and $e_4 = (e_2' : \tau)$.

$$\Gamma \vdash e_3 \backsim e_4 \downarrow \tau$$

And we show that $|e_3| = |(e_1' : \tau)| = |e_1'| = e_1$, similarly for e_4.

$\square$

2 RelRef

2.1 Syntax of RelRef

Types	τ	$::=$	$\mathrm{bool}_r \mid \mathrm{bool}_u \mid \tau_1 \to \tau_2 \mid \mathrm{list}[n]^\alpha\, \tau \mid \forall i{::}S.\, \tau \mid \exists i{::}S.\, \tau \mid \square\, \tau \mid C \,\&\, \tau \mid C \supset \tau$
Expressions	e	$::=$	$x \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{if}\ e\ \mathrm{then}\ e_1\ \mathrm{else}\ e_2 \mid \mathrm{fix}\ f(x).e \mid e_1\ e_2 \mid \mathrm{nil} \mid \mathrm{cons}(e_1, e_2)$
			$\mid (\ \mathrm{case}\ e\ \mathrm{of}\ \mathrm{nil} \to e_1 \mid h :: tl \to e_2) \mid \Lambda e \mid e[\,] \mid \mathrm{let}\ x = e_1\ \mathrm{in}\ e_2$
			$\mid \mathrm{pack}\ e \mid \mathrm{unpack}\ e_1\ \mathrm{as}\ x\ \mathrm{in}\ e_2 \mid \mathrm{clet}\ e_1\ \mathrm{as}\ x\ \mathrm{in}\ e_2 \mid \mathrm{celim}\ e$
Value	v	$:=$	$\mathrm{true} \mid \mathrm{false} \mid \mathrm{fix}\ f(x).v \mid \mathrm{nil} \mid \mathrm{cons}(e_1, e_2) \mid \Lambda e \mid \mathrm{pack}\ v$
Index terms	I, n, α	$::=$	$i \mid 0 \mid I + 1 \mid I_1 + I_2 \mid I_1 - I_2 \mid \frac{I_1}{I_2} \mid I_1 \cdot I_2 \mid \lceil I \rceil \mid \lfloor I \rfloor \mid \mathtt{min}(I_1, I_2) \mid \mathtt{max}(I_1, I_2).$

Figure 8: Syntax of values and expressions,index terms in RelRef

	τ	$::=$	$\mathrm{bool}_r \mid \mathrm{bool}_u \mid \tau_1 \to \tau_2 \mid \mathrm{list}[n]^\alpha\, \tau \mid \forall i{::}S.\, \tau \mid \exists i{::}S.\, \tau \mid \square\, \tau \mid C \,\&\, \tau \mid C \supset \tau$
Expressions	e	$::=$	$x \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{if}\ e\ \mathrm{then}\ e_1\ \mathrm{else}\ e_2 \mid \mathrm{fix}\ f(x).e \mid \mathrm{fix}_{NC}\ f(x).e \mid e_1\ e_2 \mid \mathtt{NC}\ e$
			$\mid \mathtt{split}\ (e_1, e_2)\ \mathrm{with}\ C \mid \mathtt{contra}\ e \mid \mathtt{der}\ e \mid \Lambda i.e \mid e[I] \mid \mathrm{pack}\ e\ \mathrm{with}\ I$
			$\mid \mathrm{unpack}\ e_1\ \mathrm{as}\ (x, i)\ \mathrm{in}\ e_2 \mid \mathrm{clet}\ e_1\ \mathrm{as}\ x\ \mathrm{in}\ e_2 \mid \mathrm{celim}\ e \mid \mathrm{let}\ x = e_1\ \mathrm{in}\ e_2$
			$\mid \mathrm{cons}_{NC}(e_1, e_2) \mid \mathrm{cons}_C(e_1, e_2) \mid \begin{pmatrix} \mathrm{case}\ e\ \mathrm{of}\ \mathrm{nil}\ \to e_1 \\ \mid h\ ::_{NC}\ tl\ \to e_2 \mid h ::_C tl\ \to \\ e_3 \end{pmatrix}$
Value	v	$:=$	$\mathrm{true} \mid \mathrm{false} \mid \mathrm{fix}\ f(x).v \mid \mathrm{fix}_{NC}\ f(x).e \mid \mathrm{nil} \mid \mathrm{cons}(e_1, e_2) \mid \mathrm{pack}\ e\ \mathrm{with}\ I$
			$\mid \Lambda i.e \mid \mathrm{cons}_{NC}(v_1, v_2) \mid \mathrm{cons}_C(v_1, v_2)$

Figure 9: Syntax of values and expressions in RelRef Core

	τ	$::=$	$\mathrm{bool}_r \mid \mathrm{bool}_u \mid \tau_1 \to \tau_2 \mid \mathrm{list}[n]^\alpha\, \tau \mid \forall i{::}S.\, \tau \mid \exists i{::}S.\, \tau \mid \square\, \tau \mid C \,\&\, \tau \mid C \supset \tau$
Expressions	e	$::=$	$x \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{if}\ e\ \mathrm{then}\ e_1\ \mathrm{else}\ e_2 \mid \mathrm{fix}\ f(x).e \mid \mathrm{fix}_{NC}\ f(x).e \mid e_1\ e_2 \mid \mathtt{NC}\ e$
			$\mid \mathtt{split}\ (e_1, e_2)\ \mathrm{with}\ C \mid \mathtt{contra}\ e \mid \mathtt{der}\ e \mid \Lambda i.e \mid e[I] \mid \mathrm{pack}\ e\ \mathrm{with}\ I$
			$\mid \mathrm{unpack}\ e_1\ \mathrm{as}\ (x, i)\ \mathrm{in}\ e_2 \mid (e : \tau) \mid \mathrm{clet}\ e_1\ \mathrm{as}\ x\ \mathrm{in}\ e_2 \mid \mathrm{celim}\ e \mid \mathrm{let}\ x = e_1\ \mathrm{in}\ e_2$
			$\mid \mathrm{cons}_{NC}(e_1, e_2) \mid \mathrm{cons}_C(e_1, e_2) \mid \begin{pmatrix} \mathrm{case}\ e\ \mathrm{of}\ \mathrm{nil}\ \to e_1 \\ \mid h\ ::_{NC}\ tl\ \to e_2 \mid h ::_C tl\ \to \\ e_3 \end{pmatrix}$
Value	v	$:=$	$\mathrm{true} \mid \mathrm{false} \mid \mathrm{fix}\ f(x).v \mid \mathrm{fix}_{NC}\ f(x).e \mid \mathrm{nil} \mid \mathrm{cons}(e_1, e_2) \mid \mathrm{pack}\ e\ \mathrm{with}\ I$
			$\mid \Lambda i.e \mid \mathrm{cons}_{NC}(v_1, v_2) \mid \mathrm{cons}_C(v_1, v_2)$

Figure 10: Syntax of values and expressions in BiRelRef

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x : \tau}\ \textbf{rr-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Delta; \Phi_a; \Gamma \vdash \mathtt{b} \backsim \mathtt{b} : \text{bool}_r}\ \textbf{rr-bool}$$

$$\frac{\mathtt{b}_1, \mathtt{b}_2 \in \{\text{true}, \text{false}\}}{\Delta; \Phi_a; \Gamma \vdash \mathtt{b}_1 \backsim \mathtt{b}_2 : \text{bool}_u}\ \textbf{rr-u-bool}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \text{bool}_r \qquad \Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \tau}{\Delta; \Phi_a; \Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \backsim \text{if } e' \text{ then } e_1' \text{ else } e_2' : \tau}\ \textbf{rr-if}$$

$$\frac{\Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 : \tau_1 \to \tau_2}\ \textbf{rr-fix}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau_1 \to \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \tau_1}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' : \tau_2}\ \textbf{rr-app}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e : \tau \\ \forall x \in dom(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\, \Gamma(x)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \backsim e : \square\, \tau}\ \textbf{rr-nochange}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau'}\ \textbf{rr-}\ \boldsymbol{\sqsubseteq}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \mathtt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \square\, (\tau_1 \to \tau_2), \Gamma \vdash e \backsim e : \tau_2 \qquad \forall x \in dom(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\, \Gamma(x)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e : \square\, (\tau_1 \to \tau_2)}\ \textbf{rr-fixNC}$$

$$\frac{\begin{array}{c}i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau \\ i \notin \text{FIV}(\Phi_a; \Gamma)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \Lambda e \backsim \Lambda e' : \forall i{::}S.\, \tau}\ \textbf{rr-iLam} \qquad \frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \forall i{::}S.\, \tau \\ \Delta \vdash I :: S\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \backsim e'[\,] : \tau\{I/i\}}\ \textbf{rr-iApp}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \backsim \text{pack } e' : \exists i{::}S.\, \tau}\ \textbf{rr-pack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \exists i{::}S.\, \tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } x \text{ in } e_2' : \tau_2}\ \textbf{rr-unpack1}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : C \supset \tau}\ \textbf{rr-c-impII} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' : \tau}\ \textbf{rr-c-implE}$$

Figure 11: RelRef typing rules (Part 1)

$$\frac{\Delta; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{nil} \backsim \text{nil} : \text{list}[0]^\alpha \, \tau} \ \textbf{rr-nil}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \text{list}[n]^\alpha \, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') : \text{list}[n+1]^{\alpha+1} \, \tau} \ \textbf{rr-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \square \, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \text{list}[n]^\alpha \, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') : \text{list}[n+1]^{\alpha} \, \tau} \ \textbf{rr-cons2}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \text{list}[n]^\alpha \, \tau \\ \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \backsim e_1' : \tau' \qquad i, \Delta; \Phi_a \wedge n = i + 1; h : \square \, \tau, tl : \text{list}[i]^\alpha \, \tau, \Gamma \vdash e_2 \backsim e_2' : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i + 1 \wedge \alpha = \beta + 1; h : \tau, tl : \text{list}[i]^\beta \, \tau, \Gamma \vdash e_2 \backsim e_2' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \ \text{case } e \text{ of nil} \to e_1 \mid h :: tl \to e_2 \backsim \ \text{case } e' \text{ of nil} \to e_1' \mid h :: tl \to e_2' : \tau'} \ \textbf{rr-caseL}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : C \ \& \ \tau} \ \textbf{rr-c-andI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : C \ \& \ \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' : \tau_2} \ \textbf{rr-c-andE}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta; \Phi_a \wedge \neg C; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta \vdash C \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau} \ \textbf{rr-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau} \ \textbf{rr-contra}$$

Figure 12: RelRef typing rules (Part 2)

$$\frac{}{\models \text{bool}_r \sqsubseteq \text{bool}_u}\ \textbf{bool} \qquad \frac{\models \tau_1' \sqsubseteq \tau_1 \qquad \models \tau_2 \sqsubseteq \tau_2'}{\models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'} \to \qquad \frac{}{\models \tau \sqsubseteq \tau}\textbf{refl}$$

$$\frac{}{\models \Box(\tau_1 \to \tau_2) \sqsubseteq \Box\tau_1 \to \Box\tau_2} \to \Box\textbf{diff}$$

$$\frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \leq \alpha' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \tau \sqsubseteq \text{list}[n']^{\alpha'}\, \tau'}\ \textbf{l1}$$

$$\frac{\Delta; \Phi_a \models \alpha \doteq 0}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \tau \sqsubseteq \text{list}[n]^{\alpha}\, \Box\, \tau}\ \textbf{l2} \qquad \frac{}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \Box\, \tau \sqsubseteq \Box\, (\text{list}[n]^{\alpha}\, \tau)}\ \textbf{l}\Box$$

$$\frac{}{\Delta; \Phi_a \models \text{int}_r \sqsubseteq \Box\, \text{int}_r}\ \textbf{int-}\Box \qquad \frac{}{\Delta; \Phi_a \models \Box\, \tau \sqsubseteq \tau}\ \textbf{T} \qquad \frac{}{\Delta; \Phi_a \models \Box\, \tau \sqsubseteq \Box\, \Box\, \tau}\ \textbf{D}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2}{\Delta; \Phi_a \models \Box\, \tau_1 \sqsubseteq \Box\, \tau_2}\ \textbf{B-}\Box \qquad \frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_3}{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_3}\textbf{trans}$$

$$\frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \forall i{::}S.\, \tau \sqsubseteq \forall i{::}S.\, \tau'}\ \forall_{\textbf{diff}} \qquad \frac{}{\Delta; \Phi_a \models \Box\, (\forall i{::}S.\, \tau) \sqsubseteq \forall i{::}S.\, \Box\, \tau}\ \forall\ \Box$$

$$\frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \sqsubseteq \exists i{::}S.\, \tau'}\ \exists \qquad \frac{}{\Delta; \Phi_a \models \exists i{::}S.\, \Box\, \tau \sqsubseteq \Box\, (\exists i{::}S.\, \tau)}\ \exists\Box$$

$$\frac{\Delta; \Phi_a \wedge C' \models C \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C \supset \tau \sqsubseteq C' \supset \tau'}\textbf{c-impl} \qquad \frac{}{\Delta; \Phi_a \models \Box\, (C \supset \tau) \sqsubseteq C \supset \Box\, \tau}\ \textbf{c-impl-}\Box$$

$$\frac{\Delta; \Phi_a \wedge C \models C' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C\ \&\ \tau \sqsubseteq C'\ \&\ \tau'}\textbf{c-and} \qquad \frac{}{\Delta; \Phi_a \models C\ \&\ \Box\, \tau \sqsubseteq \Box\, (C\ \&\ \tau)}\ \textbf{c-and-}\Box$$

Figure 13: RelRef Subtyping rules

$\boxed{\Delta; \Phi_a \models \tau_1 \equiv \tau_2}$ checks whether τ_1 is equivalent to τ_2.

$$\frac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2'}{\Delta; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'} \ \textbf{eq-fun} \qquad \frac{\begin{array}{c}\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \doteq \alpha' \\ \Delta; \Phi_a \models \tau \equiv \tau'\end{array}}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \equiv \mathrm{list}[n']^{\alpha'}\, \tau'} \ \textbf{eq-list}$$

$$\frac{\Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \Box\, \tau \equiv \Box\, \tau'} \textbf{eq-B-}\Box \qquad \frac{i, \Delta; \Phi_a \models \tau \equiv \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \equiv \exists i{::}S.\, \tau'} \ \textbf{eq-}\exists$$

$$\frac{\Delta; C' \wedge \Phi_a \models C \qquad \Delta; C \wedge \Phi_a \models C' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \ \&\ \tau \equiv C' \ \&\ \tau'} \textbf{eq-c-prod}$$

$$\frac{\Delta; C \wedge \Phi_a \models C' \qquad \Delta; C' \wedge \Phi_a \models C \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \supset \tau \equiv C' \supset \tau'} \textbf{eq-c-impl}$$

$$\frac{i, \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau'} \textbf{eq-}\forall$$

Figure 14: RelRef Core type equivalence rules

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x :^c \tau}\ \textbf{c-r-var}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix}\ f(x).e_1 \backsim \text{fix}\ f(x).e_2 :^c \tau_1 \to \tau_2}\ \textbf{c-r-fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \square\,\Gamma \vdash e \backsim e :^c \tau_2}{\Delta; \Phi_a; \square\,\Gamma, \Gamma' \vdash \text{fix}_{NC}\ f(x).e \backsim \text{fix}_{NC}\ f(x).e :^c \square\,(\tau_1 \to \tau_2)}\ \textbf{c-r-fixNC}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_1, e_2) \backsim \text{cons}_C(e_1', e_2') :^c \text{list}[n+1]^{\alpha+1}\, \tau}\ \textbf{c-r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \square\,\tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_{NC}(e_1, e_2) \backsim \text{cons}_{NC}(e_1', e_2') :^c \text{list}[n+1]^{\alpha}\, \tau}\ \textbf{c-r-cons2}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \text{list}[n]^\alpha\, \tau \\ \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \backsim e_1' :^c \tau' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : \square\,\tau, tl : \text{list}[i]^\alpha\, \tau, \Gamma \vdash e_2 \backsim e_2' :^c \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h' : \tau, tl : \text{list}[i]^\beta\, \tau, \Gamma \vdash e_3 \backsim e_3' :^c \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case}\ e\ \text{of nil}\ \to e_1 \\ \mid h\ ::_N tl\ \to e_2 \\ \mid h'\ ::_C tl' \to e_3\end{array} \backsim \begin{array}{l}\text{case}\ e'\ \text{of nil}\ \to e_1' \\ \mid h\ ::_N tl\ \to e_2' \\ \mid h'\ ::_C tl' \to e_3'\end{array} :^c \tau'}\ \textbf{c-r-caseL}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \square\,\tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \backsim \texttt{der}\ e_2 :^c \tau}\ \textbf{c-der} \qquad \frac{\Delta; \Phi_a; \square\,\Gamma \vdash e \backsim e :^c \tau}{\Delta; \Phi_a; \square\,\Gamma, \Gamma' \vdash \texttt{NC}\ e \backsim \texttt{NC}\ e :^c \square\,\tau}\ \textbf{c-nochange}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau'}\ \textbf{c-r-}\equiv$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e_1 \backsim e_2 :^c \tau \qquad \Delta; \Phi_a \wedge \neg C; \Gamma \vdash e_1' \backsim e_2' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{split}\ (e_1, e_1')\ \text{with}\ C \backsim \texttt{split}\ (e_2, e_2')\ \text{with}\ C :^c \tau}\ \textbf{c-r-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \texttt{contra}\ e_1 \backsim \texttt{contra}\ e_2 :^c \tau}\ \textbf{c-r-contra}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \forall i{::}S.\, \tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[I] \backsim e'[I] :^c \tau\{I/i\}}\ \textbf{c-r-iApp}$$

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad i \notin \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda i.e \backsim \Lambda i.e' :^c \forall i{::}S.\, \tau}\ \textbf{c-r-iLam}$$

Figure 15: RelRef Core typing rules (Part 1)

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \supset \tau}\ \textbf{c-r-cimpI} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' :^c \tau}\ \textbf{c-r-cimplE}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \,\&\, \tau}\ \textbf{c-r-candI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c C \,\&\, \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' :^c \tau_2}\ \textbf{c-r-candE}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \rightarrow \tau_2 \\ \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \tau_1\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' :^c \tau_2}\ \textbf{c-r-app}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \backsim \text{let } x = e_1' \text{ in } e_2' :^c \tau_2}\ \textbf{c-r-let}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \text{ with } I \backsim \text{pack } e' \text{ with } I :^c \exists i{::}S.\, \tau}\ \textbf{c-r-pack}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \exists i{::}S.\, \tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' :^c \tau_2}\ \textbf{c-r-unpack1}$$

Figure 16: RelRef Core typing rules (Part 2)

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x \rightsquigarrow x \backsim x : \tau} \textbf{ e-r-var}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 \rightsquigarrow \text{fix } f(x).{e_1}^* \backsim \text{fix } f(x).{e_2}^* : \tau_1 \to \tau_2} \textbf{ e-r-fix}$$

$$\frac{\begin{array}{c} \Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \Box\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2 \qquad \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,\Gamma(x_i)} \\ \forall x_i \in dom(\Gamma), \quad \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \Box\,\Gamma(x) \qquad e^{**} = \text{let } \overline{y_i = e_i \ x_i} \text{ in fix}_{NC} \ f(x).e^*\overline{[\texttt{der} \ y_i/x_i]} \end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e \rightsquigarrow e^{**} \backsim e^{**} : \Box\,(\tau_1 \to \tau_2)} \textbf{ e-r-fixNC}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \supset \tau} \textbf{ e-r-c-impI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' \rightsquigarrow \text{celim } e^* \backsim \text{celim } {e'}^* : \tau} \textbf{ e-r-c-implE}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \ \& \ \tau} \textbf{ e-r-andI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : C \ \& \ \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow \text{clet } e_1^* \text{ as } x \text{ in } e_2^* \backsim \text{clet } e_1'^* \text{ as } x \text{ in } e_2'^* : \tau_2} \textbf{ e-r-c-andE}$$

Figure 17: RelRef embedding rules (Part 1)

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \text{list}[n]^{\alpha}\,\tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') \rightsquigarrow \text{cons}_C(e_1^*, e_2^*) \backsim \text{cons}_C(e_1'^*, e_2'^*) : \text{list}[n+1]^{\alpha+1}\,\tau}\ \textbf{e-r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \Box\,\tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \text{list}[n]^{\alpha}\,\tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') \rightsquigarrow \text{cons}_{NC}(e_1', e_2') \backsim \text{cons}_{NC}(e_1^*, e_2^*) : \text{list}[n+1]^{\alpha}\,\tau}\ \textbf{e-r-cons2}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau \\ \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,\Gamma(x_i)} \qquad \forall x_i \in dom(\Gamma), \quad \Delta; \Phi_a \models \Gamma(x_i) \sqsubseteq \Box\,\Gamma(x_i)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \backsim e \rightsquigarrow \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC}\ e^*[\overline{\texttt{der}\ y_i/x_i}] \backsim \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC}\ e^*[\overline{\texttt{der}\ y_i/x_i}] : \Box\,\tau}\ \textbf{e-nochange}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \text{list}[n]^{\alpha}\,\tau \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \tau' \\ i, \Delta; \Phi_a \wedge n = i+1; h : \Box\,\tau, tl : \text{list}[i]^{\alpha}\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h : \tau, tl : \text{list}[i]^{\beta}\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^{**} \backsim e_2'^{**} : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil} \to e_1 \\ \mid h\ ::\ tl \to e_2\end{array} \backsim \begin{array}{l}\text{case } e \text{ of nil} \to e_1' \\ \mid h\ ::\ tl \to e_2'\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^* \text{ of nil } \to e_1^* \\ \mid h\ ::_{NC}\ tl\ \to e_2^* \backsim \\ \mid h\ ::_C\ tl \quad \to e_2^{**}\end{array} \begin{array}{l}\text{case } e'^* \text{ of nil } \to e_1'^* \\ \mid h\ ::_{NC}\ tl\ \to e_2'^* \\ \mid h\ ::_C\ tl \quad \to e_2'^{**}\end{array} : \tau'}\ \textbf{e-r-caseL}$$

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau \qquad i \notin \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda.e \backsim \Lambda.e' \rightsquigarrow \Lambda i.e^* \backsim \Lambda i.e'^* : \forall i{::}S.\,\tau}\ \textbf{e-r-iLam}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \forall i{::}S.\,\tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[\,] \backsim e'[\,] \rightsquigarrow e^*[I] \backsim e'^*[I] : \tau\{I/i\}}\ \textbf{e-r-iApp}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \texttt{coerce}_{\tau, \tau'}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\sqsubseteq$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \backsim \text{pack } e' \rightsquigarrow \text{pack } e^* \text{ with } I \backsim \text{pack } e'^* \text{ with } I : \exists i{::}S.\,\tau}\ \textbf{e-r-pack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \exists i{::}S.\,\tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2 \\ i \notin FV(\Phi_a; \Gamma, \tau_2, t_2) \qquad e_1^{**} = \text{unpack } e_1^* \text{ as } (x, i) \text{ in } e_2^* \qquad e_2^{**} = \text{unpack } e_1'^* \text{ as } (x, i) \text{ in } e_2'^*\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau_2}\ \textbf{e-r-unpack}$$

$$\frac{\Delta; C \wedge \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta; \neg C \wedge \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau \qquad \Delta \vdash C\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{split}\ (e_1^*, e_1^{**}) \text{ with } C \backsim \texttt{split}\ (e_2^*, e_2^{**}) \text{ with } C : \tau}\ \textbf{e-r-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{contra}\ e_1 \backsim \texttt{contra}\ e_2 : \tau}\ \textbf{e-r-contra}$$

Figure 18: RelRef embedding rules (Part 2)

$$\frac{\Delta;\psi_a;\Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta;\psi_a;\Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2 \qquad \Phi = \Phi_1 \wedge \Phi_2}{\Delta;\psi_a;\Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2' \Rightarrow \Phi}\ \textbf{alg-r-fun}$$

$$\frac{i,\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta;\psi_a;\Phi_a \models \forall i{::}S.\,\tau \equiv \forall i{::}S.\,\tau' \Rightarrow \forall i :: S.\Phi}\ \textbf{alg-r-forall}$$

$$\frac{\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad \Phi' = \Phi \wedge n \doteq n' \wedge \alpha \doteq \alpha'}{\Delta;\psi_a;\Phi_a \models \mathrm{list}[n]^{\alpha}\,\tau \equiv \mathrm{list}[n']^{\alpha'}\,\tau' \Rightarrow \Phi'}\ \textbf{alg-r-list}$$

$$\frac{i,\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad i \notin FV(\Phi_a)}{\Delta;\psi_a;\Phi_a \models \exists i{::}S.\,\tau \equiv \exists i{::}S.\,\tau' \Rightarrow \forall i :: S.\Phi}\ \textbf{alg-r-}\exists$$

$$\frac{\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta;\psi_a;\Phi_a \models C \;\&\; \tau \equiv C' \;\&\; \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\textbf{alg-r-cprod}$$

$$\frac{\Delta;\psi_a;\Phi_a \models \tau_1 \equiv \tau_2 \Rightarrow \Phi}{\Delta;\psi_a;\Phi_a \models \Box\,\tau_1 \equiv \Box\,\tau_2 \Rightarrow \Phi}\textbf{alg-r-}\Box \qquad \frac{\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta;\psi_a;\Phi_a \models C \supset \tau \equiv C' \supset \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\textbf{alg-r-cimpl}$$

Figure 19: RelRef algorithmic type equivalence rules

$$\frac{\Delta; \psi_a; \Phi_a; \square\,\Gamma \vdash e \ominus e \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\,\Gamma \vdash \texttt{NC}\ e \ominus \texttt{NC}\ e \downarrow \square\,\tau \Rightarrow \Phi} \ \textbf{alg-r-nochange-}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \square\,\tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \ominus \texttt{der}\ e_2 \uparrow \tau \Rightarrow \Phi} \ \textbf{alg-r-der-}\!\uparrow \qquad \frac{\Gamma(x) = \tau}{\Delta; \psi_a; \Phi_a; \Gamma \vdash x \ominus x \uparrow \tau \Rightarrow \top} \ \textbf{alg-r-var-}\!\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \tau_1 \to \tau_2, x : \tau_1, \Gamma \vdash e \ominus e' \downarrow \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{fix}\ f(x).e \ominus \text{fix}\ f(x).e' \downarrow \tau_1 \to \tau_2 \Rightarrow \Phi} \ \textbf{alg-r-fix-}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \square\,(\tau_1 \to \tau_2), x : \tau_1, \square\,\Gamma \vdash e \ominus e' \downarrow \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\,\Gamma \vdash \text{fix}_{NC}\ f(x).e \ominus \text{fix}_{NC}\ f(x).e \downarrow \square\,(\tau_1 \to \tau_2) \Rightarrow \Phi} \ \textbf{alg-r-fix-}\!\downarrow\square$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \tau_1 \to \tau_2 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_1 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \uparrow \tau_2 \Rightarrow \Phi_1 \wedge \Phi_2} \ \textbf{alg-r-app-}\!\uparrow$$

$$\frac{\Delta, \psi_a; \Phi \vdash \tau\ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{nil} \ominus \text{nil} \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow n \doteq 0} \ \textbf{alg-r-nil-}\!\downarrow$$

$$\frac{\begin{array}{c} i, \beta \in \text{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \Rightarrow \Phi_1 \\ \Delta; i, \beta, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^\beta\,\tau \Rightarrow \Phi_2 \qquad \Phi_2' = n \doteq (i+1) \wedge \exists \beta :: \mathbb{N}.\Phi_2 \wedge \alpha \doteq \beta + 1 \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{cons}_C(e_1, e_2) \ominus \text{cons}_C(e_1', e_2') \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'} \ \textbf{alg-r-consC-}\!\downarrow$$

$$\frac{\begin{array}{c} i \in \text{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \square\,\tau \Rightarrow \Phi_1 \\ \Delta; i, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^\alpha\,\tau \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{cons}_{NC}(e_1, e_2) \ominus \text{cons}_{NC}(e_1', e_2') \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'} \ \textbf{alg-r-consNC-}\!\downarrow$$

$$\frac{\begin{array}{c} \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_e \qquad \Delta; \psi_a; n \doteq 0 \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau' \Rightarrow \Phi_1 \\ i :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \Phi_a; h : \square\,\tau, tl : \text{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau' \Rightarrow \Phi_2 \\ i :: \mathbb{N}, \beta :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \alpha \doteq \beta + 1 \wedge \Phi_a; h : \tau, tl : \text{list}[i]^\beta\,\tau, \Gamma \vdash e_3 \ominus e_3' \downarrow \tau' \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_1) \wedge (\forall i :: \mathbb{N}.(n \doteq i+1) \to (\Phi_2 \wedge \forall \beta :: \mathbb{N}.(\alpha \doteq \beta+1) \to \Phi_3)) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \begin{array}{l} \text{case}\ e\ \text{of nil}\ \to e_1 \\ \mid h\ ::_{NC}\ tl\ \to e_2 \\ \mid h\ ::_C\ tl\ \ \to e_3 \end{array} \ominus \begin{array}{l} \text{case}\ e'\ \text{of nil}\ \to e_1' \\ \mid h\ ::_{NC}\ tl\ \to e_2' \\ \mid h\ ::_C\ tl\ \ \to e_3' \end{array} \downarrow \tau' \Rightarrow (\Phi_e \wedge \Phi_{body})} \ \textbf{alg-r-caseL-}\!\downarrow$$

Figure 20: RelRef algorithmic typing rules (Part 1)

$$\frac{i :: S, \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \Lambda i.e \ominus \Lambda i.e' \downarrow \forall i{::}S.\,\tau \Rightarrow (\forall i :: S.\Phi)}\ \textbf{alg-r-iLam-}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \forall i{::}S.\,\tau' \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e\,[I] \ominus e'\,[I] \uparrow \tau'\{I/i\} \Rightarrow \Phi}\ \textbf{alg-r-iApp-}\!\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau\{I/i\} \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{pack } e \text{ with } I \ominus \text{pack } e' \text{ with } I \downarrow \exists i{::}S.\,\tau \Rightarrow \Phi}\ \textbf{alg-r-pack-}\!\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \exists i{::}S.\,\tau_1 \Rightarrow \Phi_1 \\ i :: S, \Delta; \psi_a; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2) \qquad \Phi = (\Phi_1 \wedge \forall i :: S.\Phi_2)\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' \downarrow \tau_2 \Rightarrow \Phi}\ \textbf{alg-r-unpack-}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Gamma \vdash e_1 \ominus e_1 \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow C \,\&\, \tau \Rightarrow C \wedge (C \to \Phi)}\ \textbf{alg-r-c-andI}\!\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow C \,\&\, \tau_1 \Rightarrow \Phi_1 \\ \Delta; \psi_a; \Phi \wedge C; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2 \Rightarrow \Phi_2 \qquad \Phi_2' = C \to \Phi_2\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \ominus \text{clet } e_1' \text{ as } x \text{ in } e_2' \downarrow \tau_2 \Rightarrow (\Phi_1 \wedge \Phi_2')}\ \textbf{alg-r-c-andE}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Gamma \vdash e_1 \ominus e_1 \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow C \supset \tau \Rightarrow (C \to \Phi)}\ \textbf{alg-r-c-implI}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow C \supset \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{celim } e \ominus \text{celim } e' \uparrow \tau \Rightarrow (C \wedge \Phi)}\ \textbf{alg-r-c-implE}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \tau' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau' \equiv \tau \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-}\!\uparrow\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash \tau \;\texttt{wf} \qquad \text{FIV}(\tau) \in \Delta}{\Delta; \psi_a; \Phi_a; \Gamma \vdash (e : \tau) \ominus (e' : \tau) \uparrow \tau \Rightarrow \Phi}\ \textbf{alg-r-anno-}\!\uparrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; C \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \tau \Rightarrow \Phi_2 \qquad \Delta \vdash C \;\texttt{wf}\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{split}\ (e_1, e_2) \text{ with } C \ominus \texttt{split}\ (e_1', e_2') \text{ with } C \downarrow \tau \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2}\ \textbf{alg-r-split}\!\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a \models \bot}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{contra}\ e \ominus \texttt{contra}\ e' \downarrow \tau \Rightarrow \top}\ \textbf{alg-r-contra}\!\downarrow$$

$$\frac{\Gamma \vdash e \ominus e' \uparrow \text{bool}_r \Rightarrow \Phi_1 \qquad \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \Rightarrow \Phi_2 \qquad \Gamma \vdash e_2 \ominus e_2' \downarrow \tau \Rightarrow \Phi_3}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \ominus \text{if } e' \text{ then } e_1' \text{ else } e_2' \downarrow \tau \Rightarrow \Phi_1 \wedge \Phi_2 \wedge \Phi_3}\ \textbf{alg-r-if}$$

Figure 21: RelRef algorithmic typing rules (Part 2)

2.2 RelRef Lemmas

Lemma 7 (Substitution of RelRef Core)
Assume that

1. $\Delta; \Psi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau$ (1).
2. $\Delta; \Psi_a; \Gamma, x : \tau \vdash e'_1 \backsim e'_2 :^c \tau'$ (2).

then $\Delta; \Psi_a; \Gamma \vdash e'_1[e_1/x] \backsim e'_2[e_2/x] :^c \tau'$.

Proof. By induction on the typing derivation of the second assumption (2).

Case $\dfrac{(\Gamma, x : \tau)(z) = \tau'}{\Delta; \Phi_a; \Gamma, x : \tau \vdash z \backsim z :^c \tau'}$ **c-r-var**

Subcase: $z = x$.
TS: $\Delta; \Phi_a; \Gamma \vdash z[e_1/x] \backsim z[e_2/x] :^c \tau'$.
Because $x = z$,$z[e_1/x] = e_1$, $z[e_2/x] = e_1$, STS: $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau'$.
From $(\Gamma, x : \tau)(z) = \tau'$ and $x = z$, we know $\tau = \tau'$.
By the above statements, STS: $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau$, which is proved by the assumption.
Subcase: $z \neq x$.
we know: $\Gamma(z) = \tau'$ $(\star)$.
TS: $\Delta; \Phi_a; \Gamma \vdash z[e_1/x] \backsim z[e_2/x] :^c \tau'$.
Because $x \neq z$, $z[e_1/x] = z$, $z[e_2/x] = z$, STS: $\Delta; \Phi_a; \Gamma \vdash z \backsim z :^c \tau'$.
By the core rule **c-r-vars** and $(\star)$, we construct the derivation :

$\dfrac{\Gamma(z) = \tau'}{\Delta; \Phi_a; \Gamma \vdash z \backsim z :^c \tau'}$ **c-r-var**

Case $\dfrac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e''_1 \backsim e''_2 :^c \Box\,\tau' \quad (3)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \mathtt{der}\ e''_1 \backsim \mathtt{der}\ e''_2 :^c \tau'}$ **c-der**

$e'_1 = \mathtt{der}\ e''_1$, $e'_2 = \mathtt{der}\ e''_2$.
TS: $\Delta; \Phi_a; \Gamma \vdash \mathtt{der}\ (e''_1[e_1/x]) \backsim \mathtt{der}\ (e''_2[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Psi_a; \Gamma \vdash e''_1[e_1/x] \backsim e''_2[e_2/x] :^c \Box\tau'$ (4).
By core rule **c-der** and (4), we can derive

$\dfrac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e''_1[e_1/x] \backsim e''_2[e_2/x] :^c \Box\,\tau' \quad (4)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \mathtt{der}\ (e''_1[e_1/x]) \backsim \mathtt{der}\ (e''_2[e_2/x]) :^c \tau'}$ **c-der**.

Case $\dfrac{\Delta; \Phi_a; z : \tau_1, f : \tau_1 \to \tau_2, \Gamma, x : \tau \vdash e''_1 \backsim e''_2 :^c \tau_2 \quad (3)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \mathbf{fix}\ f(z).e''_1 \backsim \mathbf{fix}\ f(z).e''_2 :^c \tau_1 \to \tau_2}$ **c-r-fix**

$e'_1 = \text{fix}\ f(z).e''_1$, $e'_2 = \text{fix}\ f(z).e''_2, \tau' = \tau_1 \to \tau_2$.
TS: $\Delta; \Phi_a; \Gamma \vdash \text{fix}\ f(z).(e''_1[e_1/x]) \backsim \text{fix}\ f(z).(e''_2[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Psi_a; z : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e''_1[e_1/x] \backsim e''_2[e_2/x] :^c \tau_2$ (4).
By core rule **c-r-fix** and (4), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \text{fix}\ f(z).(e''_1[e_1/x]) \backsim \text{fix}\ f(z).(e''_2[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; z : \tau_1, f : \Box\,(\tau_1 \to \tau_2), \Box\,\Gamma, x : \tau \vdash e \backsim e :^c \tau_2 \quad (3)}{\Delta; \Phi_a; \Box\,\Gamma, \Gamma', x : \tau \vdash \mathbf{fix}_{NC}\ f(z).e \backsim \mathbf{fix}_{NC}\ f(z).e :^c \Box\,(\tau_1 \to \tau_2)}$ **c-r-fixNC**

$e'_1 = \text{fix}\ f(z).e$, $e'_2 = \text{fix}\ f(z).e, \tau' = \Box\,(\tau_1 \to \tau_2)$.
TS: $\Delta; \Phi_a; \Box\,\Gamma, \Gamma' \vdash \text{fix}\ f(z).(e[e_1/x]) \backsim \text{fix}\ f(z).(e[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Phi_a; z : \tau_1, f : \Box\,(\tau_1 \to \tau_2), \Box\,\Gamma \vdash e[e_1/x] \backsim e[e_2/x] :^c \tau_2$ (4).
By core rule **c-r-fixNC** and (4), we can derive :
$\Delta; \Phi_a; \Box\,\Gamma, \Gamma' \vdash \text{fix}\ f(z).(e[e_1/x]) \backsim \text{fix}\ f(z).(e[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e_3 \backsim e'_3 :^c \tau'' \quad (3) \qquad \Delta; \Phi_a; \Gamma, x : \tau \vdash e_4 \backsim e'_4 :^c \mathbf{list}[n]^\alpha \tau'' \quad (4)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \mathbf{cons}_C(e_3, e_4) \backsim \mathbf{cons}_C(e'_3, e'_4) :^c \mathbf{list}[n+1]^{\alpha+1} \tau''}$ **c-r-cons1**

$e'_1 = \text{cons}_C(e_3, e_4)$, $e'_2 = \text{cons}_C(e'_3, e'_4), \tau' = \text{list}[n+1]^{\alpha+1} \tau''$.

TS: $\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_3[e_1/x], e_4[e_1/x]) \backsim \text{cons}_C(e'_3[e_2/x], e'_4[e_2/x]) :^c \tau'$.
By IH on (1) and (3),$\Delta; \Phi_a; \Gamma \vdash e_3[e_1/x] \backsim e'_3[e_2/x] :^c \tau'' \quad (5)$.
By IH on (1) and (4),$\Delta; \Phi_a; \Gamma \vdash e_4[e_1/x] \backsim e'_4[e_2/x] :^c \text{list}[n]^\alpha \tau'' \quad (6)$.
By core rule **c-r-cons1** and (5),(6), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_3[e_1/x], e_4[e_1/x]) \backsim \text{cons}_C(e'_3[e_2/x], e'_4[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; \Box\,\Gamma, x : \tau \vdash e \backsim e :^c \tau'' \quad (3)}{\Delta; \Phi_a; \Box\,\Gamma, \Gamma', x : \tau \vdash \mathtt{NC}\ e \backsim \mathtt{NC}\ e :^c \Box\,\tau''}$ **c-nochange**

$e'_1 = \mathtt{NC}\ e$, $e'_2 = \mathtt{NC}\ e, \tau' = \Box\,\tau''$.
TS: $\Delta; \Phi_a; \Gamma \vdash \mathtt{NC}\ e[e_1/x] \backsim \mathtt{NC}\ e[e_2/x] :^c \tau'$.
By IH on (1) and (3),$\Delta; \Phi_a; \Gamma \vdash e[e_1/x] \backsim e[e_2/x] :^c \tau'' \quad (4)$.
By core rule **c-nochange** and (4), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \mathtt{NC}\ e[e_1/x] \backsim \mathtt{NC}\ e[e_2/x]) :^c \tau'$

□

Lemma 8 (Reflexivity of Algorithmic Binary Type Equivalence in RelRef)
$\forall \Delta, \psi_a, \Phi_a, \tau.$ there exists Φ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$.

Proof. By induction on the binary type.

Case $\forall i{::}S.\, \tau$
Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ where $\Delta' = i, \Delta$, $\exists \Phi'$ s.t $\Delta'; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi'$ and $\Delta'; \psi_a; \Phi_a \models \Phi'$.
By the above statement and algorithmic type eq rules **alg-r-forall**, we conclude the following statement where $\Phi = \forall i{::}S.\, \Phi'$.
$\Delta; \psi_a; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $\tau_1 \to \tau_2$
Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1 \to \tau_2 \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ_1, $\exists \Phi_1$ s.t $\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1 \Rightarrow \Phi_1$ and $\Delta; \psi_a; \Phi_a \models \Phi_1$.
By IH on τ_2, $\exists \Phi_2$ s.t $\Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2 \Rightarrow \Phi_2$ and $\Delta; \psi_a; \Phi_a \models \Phi_2$.
By the above statement and algorithmic type eq rules **alg-r-fun**, we conclude the following statement where $\Phi = \Phi_1 \wedge \Phi_2$.
$\Delta; \psi_a; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1 \to \tau_2 \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $\mathbf{list}[n]^\alpha \tau$
Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models \text{list}[n]^\alpha \tau \equiv \text{list}[n]^\alpha \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ, $\exists \Phi_1$ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi_1$ and $\Delta; \psi_a; \Phi_a \models \Phi_1$.
By the above statement and algorithmic type eq rules **alg-r-list**, we conclude the following statement where $\Phi = \Phi_1 \wedge n = n \wedge \alpha = \alpha$.
$\Delta; \psi_a; \Phi_a \models \text{list}[n]^\alpha \tau \equiv \text{list}[n]^\alpha \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $\exists i{::}S.\,\tau$

Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models \exists i{::}S.\,\tau \equiv \exists i{::}S.\,\tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ, choose $i \notin FV(\Phi_a)$, $\exists \Phi_1$ s.t $i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi_1$ and $i, \Delta; \psi_a; \Phi_a \models \Phi_1$.
By the above statement and algorithmic type eq rules **alg-r-$\exists$**, we conclude the following statement where $\Phi = \forall i{::}S.\,\Phi_1$.
$\Delta; \psi_a; \Phi_a \models \exists i{::}S.\,\tau \equiv \exists i{::}S.\,\tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $\Box\,\tau$

Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models \Box\,\tau \equiv \Box\,\tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ,$\exists \Phi_1$ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi_1$ and $\Delta; \psi_a; \Phi_a \models \Phi_1$.
By the above statement and algorithmic type eq rules **B-$\Box$**, we conclude the following statement where $\Phi = \Phi_1$.
$\Delta; \psi_a; \Phi_a \models \Box\,\tau \equiv \Box\,\tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $C \,\&\, \tau$

Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models C \,\&\, \tau \equiv C \,\&\, \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ,$\exists \Phi_1$ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi_1$ and $\Delta; \psi_a; \Phi_a \models \Phi_1$.
By the above statement and algorithmic type eq rules **alg-r-cprod**, we conclude the following statement where $\Phi = C \leftrightarrow C \wedge \Phi_1$.
$\Delta; \psi_a; \Phi_a \models C \,\&\, \tau \equiv C \,\&\, \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

Case $C \supset \tau$

Assume Δ, ψ_a, Φ_a.
TS: $\exists \Phi$ s.t $\Delta; \psi_a; \Phi_a \models C \supset \tau \equiv C \supset \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$
By IH on τ,$\exists \Phi_1$ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi_1$ and $\Delta; \psi_a; \Phi_a \models \Phi_1$.
By the above statement and algorithmic type eq rules **alg-r-cimpl**, we conclude the following statement where $\Phi = C \leftrightarrow C \wedge \Phi_1$.
$\Delta; \psi_a; \Phi_a \models C \supset \tau \equiv C \supset \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$

$\square$

Theorem 9 (Soundness of the Algorithmic Binary Type Equality in RelRef) Assume that

1. $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$
2. $\mathrm{FIV}(\Phi_a, \tau, \tau') \subseteq \Delta, \psi_a$
3. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable and $\Delta \;\triangleright\; \theta_a : \psi_a$ is derivable.

Then $\Delta; \Phi_a[\theta_a] \models \tau[\theta_a] \equiv \tau'[\theta_a]$.

Proof. By induction on the algorithmic binary type equivalence derivation.

Case
$$\dfrac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2 \qquad \Phi = \Phi_1 \wedge \Phi_2}{\Delta; \psi_a; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2' \Rightarrow \Phi}\ \textbf{alg-r-fun}$$
Assume $\mathrm{FIV}(\Phi_a, \tau_1 \to \tau_2, \tau_1' \to \tau_2') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable and $\Delta \;\triangleright\; \theta_a : \psi_a$ is

derivable.
TS: $\Delta; \Phi_a[\theta_a] \models (\tau_1 \to \tau_2)[\theta_a] \equiv (\tau_1' \to \tau_2')[\theta_a]$.
We know $\Phi = \Phi_1 \wedge \Phi_2$ s.t $\mathrm{FIV}(\Phi_a, \tau_1, \tau_1') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a]$ is provable.
Similaryly, $\mathrm{FIV}(\Phi_a, \tau_2, \tau_2') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a]$ is provable.
By IH on $\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1$ with the same substitution θ_a and the above statements, $\Delta; \Phi_a[\theta_a] \models \tau_1[\theta_a] \equiv \tau_1'[\theta_a]$.
By IH on $\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_2$ with the same substitution θ_a and the above statements , $\Delta; \Phi_a[\theta_a] \models \tau_2[\theta_a] \equiv \tau_2'[\theta_a]$.
By the above statements and RelRef type equivalence rule **eq-fun**, we conclude that

$$\Delta; \Phi_a[\theta_a] \models (\tau_1 \to \tau_2)[\theta_a] \equiv (\tau_1' \to \tau_2')[\theta_a]$$

.

Case $\dfrac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad \Phi' = \Phi \wedge n \doteq n' \wedge \alpha \doteq \alpha'}{\Delta; \psi_a; \Phi_a \models \mathbf{list}[n]^\alpha\, \tau \equiv \mathbf{list}[n']^{\alpha'}\, \tau' \Rightarrow \Phi'}$ **alg-r-list**

Assume $\mathrm{FIV}(\Phi_a, \mathrm{list}[n]^\alpha\, \tau, \mathrm{list}[n']^{\alpha'}\, \tau') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi'[\theta_a]$ is provable and $\Delta \rhd \theta_a : \psi_a$ is derivable.
TS: $\Delta; \Phi_a[\theta_a] \models (\mathrm{list}[n]^\alpha\, \tau)[\theta_a] \equiv (\mathrm{list}[n']^{\alpha'}\, \tau')[\theta_a]$.
We know $\Phi' = \Phi \wedge n \doteq n' \wedge \alpha \doteq \alpha'$ s.t
From assumption $\mathrm{FIV}(\Phi_a, \mathrm{list}[n]^\alpha\, \tau, \mathrm{list}[n']^{\alpha'}\, \tau') \subseteq \Delta, \psi_a$, we know $\mathrm{FIV}(\Phi_a, \tau, n, \alpha, \tau', n', \alpha') \subseteq \Delta, \psi_a$.
From assumption $\Delta; \Phi_a[\theta_a] \models \Phi'[\theta_a]$, we get $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable and $\Delta; \Phi_a[\theta_a] \models (n \doteq n')[\theta_a]$ and $\Delta; \Phi_a[\theta_a] \models (\alpha \doteq \alpha')[\theta_a]$.
By IH on the first premise $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$ with the same substitution θ_a and the above statements, $\Delta; \Phi_a[\theta_a] \models \tau[\theta_a] \equiv \tau'[\theta_a]$.
By the above statements and RelRef type equivalence rule **eq-list**, we conclude that

$$\Delta; \Phi_a[\theta_a] \models (\mathrm{list}[n]^\alpha\, \tau)[\theta_a] \equiv (\mathrm{list}[n']^{\alpha'}\, \tau')[\theta_a]$$

.

Case $\dfrac{i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau' \Rightarrow \forall i :: S.\Phi}$ **alg-r-forall**

Assume $\mathrm{FIV}(\Phi_a, \forall i{::}S.\, \tau, \forall i{::}S.\, \tau') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi'[\theta_a]$ is provable and $\Delta \rhd \theta_a : \psi_a$ is derivable.
TS: $\Delta; \Phi_a[\theta_a] \models \forall i{::}S.\, \tau[\theta_a] \equiv \forall i{::}S.\, \tau'[\theta_a]$.
We know $\Phi' = \forall i{::}S.\, \Phi$ s.t
From assumption $\mathrm{FIV}(\Phi_a, \forall i{::}S.\, \tau, \forall i{::}S.\, \tau') \subseteq \Delta, \psi_a$, we know $\mathrm{FIV}(\Phi_a, \tau, \tau', i) \subseteq (i, \Delta), \psi_a$.
From assumption $\Delta; \Phi_a[\theta_a] \models \forall i{::}S.\, \Phi[\theta_a]$, we get $i, \Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable.
By IH on the first premise $i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$ with the above statements, $i, \Delta; \Phi_a[\theta_a] \models \tau[\theta_a] \equiv \tau'[\theta_a]$.
By the above statements and RelRef type equivalence rule **eq-$\forall$**, we conclude that

$$\Delta; \Phi_a[\theta_a] \models \forall i{::}S.\, \tau[\theta_a] \equiv \forall i{::}S.\, \tau'[\theta_a]$$

.

Case $\dfrac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models C \,\&\, \tau \equiv C' \,\&\, \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}$ **alg-r-cprod**

Assume $\mathrm{FIV}(\Phi_a, C \,\&\, \tau, C' \,\&\, \tau') \subseteq \Delta, \psi_a$ and $\Delta; \Phi_a[\theta_a] \models \Phi'[\theta_a]$ is provable and $\Delta \rhd \theta_a : \psi_a$ is derivable.

TS: $\Delta; \Phi_a[\theta_a] \models (C \ \& \ \tau)[\theta_a] \equiv (C' \ \& \ \tau')[\theta_a]$.
We know $\Phi' = C \leftrightarrow C' \wedge \Phi$ s.t
From assumption $\text{FIV}(\Phi_a, C \ \& \ \tau, C' \ \& \ \tau') \subseteq \Delta, \psi_a$, we know $\text{FIV}(\Phi_a, \tau, \tau', C, C') \subseteq \Delta, \psi_a$.
From assumption $\Delta; \Phi_a[\theta_a] \models (C \leftrightarrow C' \wedge \Phi)[\theta_a]$, we get $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable and $\Delta; \Phi_a[\theta_a] \models C[\theta_a]$ and $\Delta; \Phi_a[\theta_a] \models C'[\theta_a]$.
By IH on the first premise $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$ with the above statements, $\Delta; \Phi_a[\theta_a] \models \tau[\theta_a] \equiv \tau'[\theta_a]$.
By the above statements and RelRef type equivalence rule **eq-c-prod**, we conclude that

$$\Delta; \Phi_a[\theta_a] \models (C \ \& \ \tau)[\theta_a] \equiv (C' \ \& \ \tau')[\theta_a]$$

.

$\square$

Theorem 10 (Completeness of the Binary Algorithmic Type Equivalence in RelRef) Assume that $\Delta; \Phi_a \models \tau \equiv \tau'$. Then $\exists \Phi$. such that $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the binary equivalence typing derivation.

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2'}{\Delta; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'}$ **eq-fun**

By IH on the first premise $\Delta; \Phi_a \models \tau_1 \equiv \tau_1'$, there exists Φ_1 s.t $\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models \Phi_1$.
By IH on the second premise $\Delta; \Phi_a \models \tau_2 \equiv \tau_2'$, there exists Φ_2 s.t $\Delta; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2$ and $\Delta; \Phi_a \models \Phi_2$.
By the above statements and algorithmic type equivalence rules **alg-r-fun** with an empty Ψ_a, we conclude the following statement where $\Phi = \Phi_1 \wedge \Phi_2$.
$\Delta; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1 \to \tau_2 \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Case $\dfrac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \doteq \alpha' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \mathbf{list}[n]^\alpha \tau \equiv \mathbf{list}[n']^{\alpha'} \tau'}$ **eq-list**

By IH on the third premise $\Delta; \Phi_a \models \tau \equiv \tau'$, there exists Φ_1 s.t $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models \Phi_1$.
By the above statements, the first and second premises and algorithmic type equivalence rules **alg-r-list** with an empty Ψ_a, we conclude the following statement where $\Phi = \Phi_1 \wedge n \doteq n' \wedge \alpha \doteq \alpha'$.
$\Delta; \Phi_a \models \text{list}[n]^\alpha \tau \equiv \text{list}[n']^{\alpha'} \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Case $\dfrac{\Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \square \tau \equiv \square \tau'}$ **eq-B-□**

By IH on the premise $\Delta; \Phi_a \models \tau \equiv \tau'$, there exists Φ_1 s.t $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models \Phi_1$.
By the above statements and algorithmic type equivalence rules **alg-r-□** with an empty Ψ_a, we conclude the following statement where $\Phi = \Phi_1$.
$\Delta; \Phi_a \models \square \tau \equiv \square \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Case $\dfrac{i, \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau'}$ **eq-∀**

By IH on the premise $i, \Delta; \Phi_a \models \tau \equiv \tau'$, there exists Φ_1 s.t $i, \Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1$ and $i, \Delta; \Phi_a \models \Phi_1$.
By the above statements and algorithmic type equivalence rules **alg-r-forall** with an empty Ψ_a, we

conclude the following statement where $\Phi = \forall i{::}S.\, \Phi_1$.
$\Delta; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Case $\dfrac{\Delta; C' \wedge \Phi_a \models C \qquad \Delta; C \wedge \Phi_a \models C' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \,\&\, \tau \equiv C' \,\&\, \tau'}$ **eq-c-prod**

By IH on the third premise $\Delta; \Phi_a \models \tau \equiv \tau'$, there exists Φ_1 s.t $i, \Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models \Phi_1$.
From the first two premises $\Delta; C' \wedge \Phi_a \models C$ and $\Delta; C \wedge \Phi_a \models C'$, we know that $\Delta; \Phi_a \models C \leftrightarrow C'$.
By the above statements and the algorithmic type equivalence rules **alg-r-cprod** with an empty Ψ_a, we conclude the following statement where $\Phi = C \leftrightarrow C' \wedge \Phi_1$.
$\Delta; \Phi_a \models C \,\&\, \tau \equiv C' \,\&\, \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

$\square$

Lemma 11 (Existence of coercions for relational subtyping in RelRef) If $\Delta; \Phi_a \models \tau \sqsubseteq \tau'$ then there exists $e \in$ RelRef Core s.t. $\Delta; \Phi_a; \cdot \vdash e \backsim e :^c \tau \to \tau'$.

Proof. Proof is by induction on the subtyping derivation. We denote the witness e of type $\tau \to \tau'$ as $\texttt{coerce}_{\tau,\tau'}$ for clarity.

Case $\dfrac{\Delta; \Phi_a \models \tau_1' \sqsubseteq \tau_1 \quad (\star) \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2' \quad (\diamond)}{\Delta; \Phi_a \models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'} \to$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1',\tau_1}$. $\Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau_1',\tau_1} \backsim \texttt{coerce}_{\tau_1',\tau_1} :^c \tau_1' \to \tau_1$
By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau_2,\tau_2'}$. $\Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau_2,\tau_2'} \backsim \texttt{coerce}_{\tau_2,\tau_2'} :^c \tau_2 \to \tau_2'$

Then, using these two statements,we can construct the following derivation where $e = \text{fix } f(x).\text{fix } f(y).\texttt{coerce}_{\tau_2,\tau_2'}\ (x\ (\texttt{c}$
$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (\tau_1 \to \tau_2) \to \tau_1' \to \tau_2'$

Case $\dfrac{}{\Delta; \Phi_a \models \textbf{int}_r \sqsubseteq \square\, \textbf{int}_r}$ **int-$\square$**

Then, we can construct the derivation using the primitive function $\text{box}_{\text{int}} : \text{int}_r \to \square\, \text{int}_r$

$$\dfrac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\text{box}_{\text{int}} x \backsim \text{fix } f(x).\text{box}_{\text{int}} x :^c \text{int}_r \to \square\, \text{int}_r}$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \tau}$ **T**

Then, we can immediately construct the derivation using the rule **c-der**.

$$\dfrac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\texttt{der}\ x \backsim \text{fix } f(x).\texttt{der}\ x :^c \square\, \tau \to \tau}$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \square\, \square\, \tau}$ **D**

Then, we can immediately construct the derivation using the rule **c-nochange**.

$$\dfrac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\texttt{NC}\ x \backsim \text{fix } f(x).\texttt{NC}\ x :^c \square\, \tau \to \square\, \square\, \tau}$$

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 (\star)}{\Delta; \Phi_a \models \square\, \tau_1 \sqsubseteq \square\, \tau_2}$ **B-$\square$**

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1,\tau_2}$. $\Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau_1,\tau_2} \backsim \texttt{coerce}_{\tau_1,\tau_2} :^c \tau_1 \to \tau_2$

Then, using $(\star)$ and the rules **c-der** and **c-nochange**, we can construct the derivation

$$\frac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\texttt{NC}\,(\texttt{coerce}_{\tau_1,\tau_2}\ (\texttt{der}\ x)) \backsim \text{fix } f(x).\texttt{NC}\,(\texttt{coerce}_{\tau_1,\tau_2}\ (\texttt{der}\ x)) :^c \square\,\tau_1 \to \square\,\tau_2}$$

Case $\dfrac{}{\Delta; \Phi_a \models \tau \sqsubseteq \tau}\,\textbf{refl}$

Then, we can immediately construct the derivation

$$\frac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).x \backsim \text{fix } f(x).x :^c \tau \to \tau}$$

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 \quad (\star) \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_3 \quad ({\color{red}\diamond})}{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_3}\,\textbf{trans}$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1,\tau_2}\ .\ \ i :: S, \Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau_1,\tau_2}\ \backsim \texttt{coerce}_{\tau_1,\tau_2}\ :^c \tau_1 \to \tau_2$
By IH on $({\color{red}\diamond})$, $\exists \texttt{coerce}_{\tau_2,\tau_3}\ .\ \ i :: S, \Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau_2,\tau_3}\ \backsim \texttt{coerce}_{\tau_2,\tau_3}\ :^c \tau_2 \to \tau_3$

Then, using $(\star)$ and $({\color{red}\diamond})$, we can construct the derivation simply by function composition

$$\frac{}{\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) \backsim \text{fix } f(x).\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) :^c \tau_1 \to \tau_3}$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,(\tau_1 \xrightarrow{\textbf{diff}(t)} \tau_2) \sqsubseteq \square\,\tau_1 \xrightarrow{\textbf{diff}(0)} \square\,\tau_2}\, \to \square\textbf{diff}$

Then, we can immediately construct the derivation where $e = \text{fix } f(x).\text{fix } (y).\texttt{NC}\,(\texttt{der}\ x)\ (\texttt{der}\ y))$
$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c \square\,(\tau_1 \to \tau_2) \to \square\,\tau_1 \to \square\,\tau_2$

Case $\dfrac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\star) \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \forall i{::}S.\,\tau \sqsubseteq \forall i{::}S.\,\tau'}\,\forall\textbf{diff}$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ i :: S, \Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau,\tau'}\ \backsim \texttt{coerce}_{\tau,\tau'}\ :^c \tau \to \tau'$

Then, using this, and the **c-r-iLam** and **c-r-iApp** rules, we can construct the following derivation:

$\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\Lambda i.\texttt{coerce}_{\tau,\tau'}\ (x\ [i]) \backsim \text{fix } f(x).\Lambda i.\texttt{coerce}_{\tau,\tau'}\ (x\ [i]) :^c (\forall i{::}S.\,\tau) \to \forall i{::}S.\,\tau'$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,(\forall i{::}S.\,\tau) \sqsubseteq \forall i{::}S.\,\square\,\tau}\,\forall\,\square$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-iLam** and **c-r-iApp** rules.

$\Delta; \Phi_a; \cdot \vdash \text{fix } f(x).\Lambda i.\texttt{NC}\,((\texttt{der}\ x)\,[i]) \backsim \text{fix } f(x).\Lambda i.\texttt{NC}\,((\texttt{der}\ x)\,[i]) :^c \square\,(\forall i{::}S.\,\tau) \to \forall i{::}S.\,\square\,\tau$

Case $\dfrac{\Delta; \Phi_a \models n \doteq n' \quad (\star) \qquad \Delta; \Phi_a \models \alpha \leq \alpha' \quad ({\color{red}\diamond}) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad ({\color{blue}\dagger})}{\Delta; \Phi_a \models \textbf{list}[n]^\alpha\,\tau \sqsubseteq \textbf{list}[n']^{\alpha'}\,\tau'}\,\textbf{l1}$

By IH on $({\color{blue}\dagger})$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ \Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau,\tau'}\ \backsim \texttt{coerce}_{\tau,\tau'}\ :^c \tau \to \tau'$
We first construct the more generic term for type:

$$\text{unit}_r \to \forall n{::}\mathbb{N}.\,\forall n'{::}\mathbb{N}.\,\forall \alpha{::}\mathbb{N}.\,\forall \alpha'{::}\mathbb{N}.\,((n = n' \wedge \alpha \leq \alpha')\ \&\ \text{list}[n]^\alpha\,\tau) \to \text{list}[n']^{\alpha'}\,\tau' \tag{2.1}$$

and then instantiate the term for eq. (5.1) later.
It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda n'.\Lambda\alpha.\Lambda\alpha'.\text{fix } f(x).\text{clet } x \text{ as } e \text{ in}$
$\quad\text{case } e \text{ of}$
$\quad\;\text{nil} \to \text{nil}$
$\quad\mid h \;::_N tl \;\to \text{let } r = \text{fList } () \, [n-1] \, [n'-1] \, [\alpha] \, [\alpha'] \; tl$
$\quad\text{in } \text{cons}_{NC}(\texttt{NC}\,(\texttt{coerce}_{\tau,\tau'}\;\; \texttt{der}\; h), r)$
$\quad\mid h \;::_C tl \;\to \text{let } r = \text{fList } () \, [n-1] \, [n'-1] \, [\alpha-1] \, [\alpha'-1] \; tl$
$\quad\text{in } \text{cons}_{C}(\texttt{coerce}_{\tau,\tau'}\;\; h, r)$

Then, we can instantiate fList using ($\star$) and ($\diamond$) as follows where
$e'' = \text{fix } f(x).\text{fList } () \; [n][n'][\alpha][\alpha'] \; x$

$\Delta; \Phi_a; \cdot \vdash e'' \backsim e'' :^c \text{list}[n]^\alpha \, \tau \to \text{list}[n']^{\alpha'} \, \tau'$

Case $\dfrac{}{\Delta; \Phi_a \models \textbf{list}[n]^\alpha \,\square\, \tau \sqsubseteq \square\,(\textbf{list}[n]^\alpha \, \tau)}$ **l**$\square$

We first construct the more generic term for type

$$\text{unit}_r \to \forall n{::}\mathbb{N}.\, \forall \alpha{::}\mathbb{N}.\, \text{list}[n]^\alpha \,\square\, \tau \to \square\,(\text{list}[n]^\alpha \, \tau) \tag{2.2}$$

and then instantiate the term for eq. (5.2) later. It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda\alpha.\text{fix } f(x).$
$\quad\text{case } e \text{ of}$
$\quad\;\text{nil} \to \texttt{NC}\,(\text{nil})$
$\quad\mid h \;::_N tl \;\to \text{let } r = \text{fList } () \, [n-1] \, [\alpha] \; tl \text{ in}$
$\qquad\qquad\qquad\texttt{NC}\,(\text{cons}_{NC}(\texttt{der}\; h, \texttt{der}\; r))$
$\quad\mid h \;::_C tl \;\to \text{let } r = \text{fList } () \, [n-1] \, [\alpha-1] \; tl \text{ in}$
$\qquad\qquad\qquad\texttt{NC}\,(\text{cons}_{C}(\texttt{der}\; h, \texttt{der}\; r))$

Then, we can instantiate fList with a concrete n and α as follows where $e'' = \text{fix } f(x).\text{fList } () \; [n][\alpha] \; x$
$\Delta; \Phi_a; \cdot \vdash e'' \ominus e'' \lesssim 0 :^c \text{list}[n]^\alpha \,\square\, \tau \to \square\,(\text{list}[n]^\alpha \, \tau)$

Case $\dfrac{\Delta; \Phi_a \models \alpha \doteq 0}{\Delta; \Phi_a \models \textbf{list}[n]^\alpha \, \tau \sqsubseteq \textbf{list}[n]^\alpha \,\square\, \tau}$ **l2**

We first construct the more generic term for type

$$\text{unit}_r \to \forall n{::}\mathbb{N}.\, \forall \alpha{::}\mathbb{N}.\, (\alpha = 0 \;\&\; \text{list}[n]^\alpha \, \tau) \to \text{list}[n]^\alpha \,\square\, \tau \tag{2.3}$$

and then instantiate the term for eq. (5.3) later. It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda\alpha.\text{fix } f(x).\text{clet } x \text{ as } e \text{ in}$
$\quad\text{case } e \text{ of}$
$\quad\;\text{nil} \to \text{nil}$
$\quad\mid h \;::_N tl \;\to \text{let } r = \text{fList } () \, [n-1][\alpha] \; tl \text{ in } \text{cons}_{NC}(\texttt{NC}\; h, r)$
$\quad\mid h \;::_C tl \;\to \texttt{contra}$

Then, we can instantiate fList with a concrete n and α (note the premise $\alpha = 0$) as follows where
$e'' = \text{fix } f(x).\text{fList } () \; [n][\alpha] \; x$
$\Delta; \Phi_a; \cdot \vdash e'' \backsim e'' :^c \text{list}[n]^0 \, \tau \to \text{list}[n]^0 \,\square\, \tau$

Case $\dfrac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\star) \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \sqsubseteq \exists i{::}S.\, \tau'}\ \exists$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ i :: S, \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \backsim \texttt{coerce}_{\tau,\tau'} :^c \tau \to \tau'$

Then, using this and the **c-r-pack** and **c-r-unpack** rules, we can construct the following derivation where $e = \text{fix } f(x).\text{unpack } x \text{ as } (y, i) \text{ in pack } (\texttt{coerce}_{\tau,\tau'}\ \ y) \text{ with } i$

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (\exists i{::}S.\, \tau) \to \exists i{::}S.\, \tau'$

Case $\dfrac{}{\Delta; \Phi_a \models \exists i{::}S.\, \square\, \tau \sqsubseteq \square\, (\exists i{::}S.\, \tau)}\ \exists\square$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-pack** and **c-r-unpack** rules in in Figures 80 and 83 where $e = \text{fix } f(x).\text{unpack } x \text{ as } (y, i) \text{ in } \texttt{NC}\ (\text{pack } \texttt{der}\ y \text{ with } i)$.

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (\exists i{::}S.\, \square\, \tau) \to \square\, (\exists i{::}S.\, \tau)$

Case $\dfrac{\Delta; \Phi_a \wedge C' \models C \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond)}{\Delta; \Phi_a \models C \supset \tau \sqsubseteq C' \supset \tau'}\textbf{c-impl}$

By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ \Delta; \Phi_a; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \backsim \texttt{coerce}_{\tau,\tau'} :^c \tau \to \tau'$

Then, using this and the premise $(\star)$ along with the **c-r-c-implI** and **c-r-c-implE** rules, we can construct the following derivation where $e = \text{fix } f(x).\texttt{coerce}_{\tau,\tau'}\ \ (\text{celim } x)$

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (C \supset \tau) \to C' \supset \tau'$

Case $\dfrac{}{\Delta; \Phi_a \models \square\, (C \supset \tau) \sqsubseteq C \supset \square\, \tau}\ \textbf{c-impl-}\square$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange** and **c-r-c-implE** rules where $e = \lambda x.\texttt{NC}\ (\text{celim } \texttt{der}\ x)$.

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c \square\, (C \supset \tau) \to (C \supset \square\, \tau)$

Case $\dfrac{\Delta; \Phi_a \wedge C \models C' \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond)}{\Delta; \Phi_a \models C \,\&\, \tau \sqsubseteq C' \,\&\, \tau'}\textbf{c-and}$

By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \backsim \texttt{coerce}_{\tau,\tau'} :^c \tau \to \tau'$

Then, using this and the premise $(\star)$ along with the **c-r-c-prodI** and **c-r-c-prodE** rules, we can construct the following derivation where $e = \text{fix } f(x).\text{clet } x \text{ as } y \text{ in } \texttt{coerce}_{\tau,\tau'}\ \ y$

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (C \,\&\, \tau) \to C' \,\&\, \tau'$

Case $\dfrac{}{\Delta; \Phi_a \models C \,\&\, \square\, \tau \sqsubseteq \square\, (C \,\&\, \tau)}\ \textbf{c-and-}\square$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-c-prodI** and **c-r-c-prodE** rules where $e = \text{fix } f(x).\text{clet } x \text{ as } y \text{ in } \texttt{NC}\ (\texttt{der}\ y)$.

$\Delta; \Phi_a; \cdot \vdash e \backsim e :^c (C \,\&\, \square\, \tau) \to \square\, (C \,\&\, \tau)$

$\square$

Theorem 12 **(Types are preserved by embedding for RelRef)**

1. If $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$, then $\Delta; \Phi_a; \Gamma \vdash e_1^* \backsim e_2^* :^c \tau$ and $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau$.

Proof. Proof is by induction on the embedding derivations.

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond) \qquad e' = \texttt{coerce}_{\tau,\tau'} \quad (\dagger)}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}$ **e-r-$\sqsubseteq$**

By IH on $(\star)$, $\Delta; \Phi_a; \Gamma \vdash e_1^* \backsim e_2^* :^c \tau \quad (\star\star)$ and $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau \quad (\star\star\star)$.
By Lemma 11 using $(\diamond)$, we know that $\Delta; \Phi_a; \cdot \vdash e' \backsim e' :^c \tau \to \tau' \quad (\diamond\diamond)$.
By applying **c-r-app** rule and $(\star\star)$ and $(\diamond\diamond)$, we get $\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau' \quad (\spadesuit)$.
By reflexivity of binary type equivalence, we know $\Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit)$.
Then, we conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau' \quad (\spadesuit) \qquad \Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit)}{\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau'}\ \textbf{c-r}-\sqsubseteq$$

By $(\star\star\star)$ and $(\diamond)$ and **rr-$\sqsubseteq$**, we also conclude $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau'$.

Case $\dfrac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \textbf{fix}\ f(x).e_1 \backsim \textbf{fix}\ f(x).e_2 \rightsquigarrow \textbf{fix}\ f(x).{e_1}^* \backsim \textbf{fix}\ f(x).{e_2}^* : \tau_1 \to \tau_2}$ **e-r-fix**

By IH on the premise $\Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau_2$, $\Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1^* \backsim e_2^* :^c \tau_2 \quad (\star)$.
By applying **c-r-fix** rule and $(\star)$, we get $\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1^* \backsim \text{fix } f(x).e_2^* :^c \tau_1 \to \tau_2 \quad (\spadesuit)$.
Similarly, using **rr-fix**, we conclude $\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 : \tau_1 \to \tau_2$.

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2 \\ \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \square\, \Gamma(x_i)} \\ \forall x_i \in dom(\Gamma), \quad \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\, \Gamma(x) \qquad e^{**} = \textbf{let}\ \overline{y_i = e_i\ x_i}\ \textbf{in fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]}\end{array}}{\Delta; \Phi_a; \Gamma \vdash \textbf{fix}\ f(x).e \backsim \textbf{fix}\ f(x).e \rightsquigarrow e^{**} \backsim e^{**} : \square\,(\tau_1 \to \tau_2)}$ **e-r-fixNC**

By IH on the premise $\Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2$, we have :
$\Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \Gamma \vdash e^* \backsim e^* :^c \tau_2 \quad (\star)$
and $\Delta; \Phi_a; \Gamma \vdash e \backsim e : \tau_2 \quad (\star\star)$.
TS: $\Delta; \Phi_a; \Gamma \vdash e^{**} \backsim e^{**} :^c \square(\tau_1 \to \tau_2)$
By applying **c-r-let** rule on e^{**}.

$$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_i\ x_i \backsim e_i\ x_i :^c \square\, \Gamma(x_i) \\ \Delta; \Phi_a; \overline{y_i : \square\, \Gamma(x_i)}, \Gamma \vdash \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} \backsim \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} :^c \square\,(\tau_1 \to \tau_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} \backsim \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} :^c \square\,(\tau_1 \to \tau_2)}\ \textbf{c-r-let}$$

Set $\Gamma = \overline{x_i : \Gamma(x_i)}$, $\Gamma_1 = \overline{y_i : \Gamma(x_i)}$.
STS: $\Delta; \Phi_a; \square\Gamma_1, \Gamma \vdash \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} \backsim \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} :^c \square\,(\tau_1 \to \tau_2)$.

By **c-r-fixNC**,

$$\dfrac{\Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \square\,\Gamma_1 \vdash e^*\overline{[\texttt{der}\ y_i/x_i]} \backsim e^*\overline{[\texttt{der}\ y_i/x_i]} :^c \tau_2 \quad (\diamond)}{\Delta; \Phi_a; \square\Gamma_1, \Gamma \vdash \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} \backsim \text{fix}_{NC}\ f(x).e^*\overline{[\texttt{der}\ y_i/x_i]} :^c \square\,(\tau_1 \to \tau_2)}\ \textbf{c-r-fixNC}$$

STS: $\Delta; \Phi_a; x : \tau_1, f : \square(\tau_1 \to \tau_2), \square\Gamma_1 \vdash e^*[\overline{\texttt{der } y_i/x_i}] \backsim e^*[\overline{\texttt{der } y_i/x_i}] :^c \tau_2$.
From $(\star)$, we extend the context and get $\Delta; \Phi_a; x : \tau_1, f : \square(\tau_1 \to \tau_2), \square\Gamma_1, \Gamma \vdash e^* \backsim e^* :^c \tau_2 \quad (\star\star)$.
By Lemma 7 on $(\star\star)$ for the number of x_i in Γ times, we conclude :
$\Delta; \Phi_a; x : \tau_1, f : \square(\tau_1 \to \tau_2), \square\Gamma_1 \vdash e^*[\overline{\texttt{der } y_i/x_i}] \backsim e^*[\overline{\texttt{der } y_i/x_i}] :^c \tau_2$.

Similarly, using $(\star\star)$ and the third premise and **rr-fixNC**, we conclude $\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e : \square\tau_1 \to \tau_2$.

Case $$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \tau \quad (\star) \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \textbf{list}[n]^\alpha\,\tau \quad (\diamond)}{\Delta; \Phi_a; \Gamma \vdash \textbf{cons}(e_1, e_2) \backsim \textbf{cons}(e_1', e_2') \rightsquigarrow \textbf{cons}_C(e_1^*, e_2^*) \backsim \textbf{cons}_C(e_1'^*, e_2'^*) : \textbf{list}[n+1]^{\alpha+1}\,\tau}\ \textbf{e-r-cons1}$$
By IH on the premise $\star$, $\Delta; \Phi_a; \Gamma \vdash e_1^* \backsim e_1'^* :^c \tau \quad (\star\star)$
By IH on the premise $\diamond$, $\Delta; \Phi_a; \Gamma \vdash e_2^* \backsim e_2'^* :^c \text{list}[n]^\alpha\,\tau \quad (\diamond\diamond)$
By applying **c-r-con1** rule using $\star\star$ and $\diamond\diamond$. we get
$\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_1^*, e_2^*) \backsim \text{cons}_C(e_1^*, e_2^*) :^c \text{list}[n+1]^{\alpha+1}\,\tau \quad (\spadesuit)$.

Case $$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau \\ \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \square\Gamma(x_i)} \qquad \forall x_i \in dom(\Gamma), \quad \Delta; \Phi_a \models \Gamma(x_i) \sqsubseteq \square\Gamma(x_i)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \backsim e \rightsquigarrow \textbf{let } \overline{y_i = e_i\ x_i} \textbf{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \textbf{let } \overline{y_i = e_i\ x_i} \textbf{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] : \square\tau}\ \textbf{e-nochange}$$
TS:$\Delta; \Phi_a; \Gamma, \Gamma' \vdash \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] :^c \square\tau$.
By IH on the first premise $\Delta; \Phi_a; \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau$,
$\Delta; \Phi_a; \Gamma \vdash e^* \backsim e^* :^c \tau \quad (\star)$
and $\Delta; \Phi_a; \Gamma \vdash e \backsim e : \tau \quad (\star\star)$.
By applying **c-r-let** rule.

$$\frac{\begin{array}{c}\overline{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e_i\ x_i \backsim e_i\ x_i :^c \square\Gamma(x_i)} \\ \Delta; \Phi_a; \square\Gamma_1, \Gamma, \Gamma' \vdash \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] :^c \square\tau\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \text{let } \overline{y_i = e_i\ x_i} \text{ in } \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] :^c \square\tau}\ \textbf{c-r-let}$$

Set $\Gamma = \overline{x_i : \Gamma(x_i)}$, $\Gamma_1 = \overline{y_i : \Gamma(x_i)}$.
STS:$\Delta; \Phi_a; \square\Gamma_1, \Gamma, \Gamma' \vdash \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] :^c \square\tau$.
By applying **c-nochange**.

$$\frac{\Delta; \Phi_a; \square\Gamma_1 \vdash e^*[\overline{y_i/x_i}] \backsim e^*[\overline{y_i/x_i}] :^c \tau}{\Delta; \Phi_a; \square\Gamma_1, \Gamma, \Gamma' \vdash \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] \backsim \texttt{NC } e^*[\overline{\texttt{der } y_i/x_i}] :^c \square\tau}\ \textbf{c-nochange}$$

STS: $\Delta; \Phi_a; \square\Gamma_1 \vdash e^*[\overline{y_i/x_i}] \backsim e^*[\overline{y_i/x_i}] :^c \tau$.
From $(\star)$, we extend the context and get
$\Delta; \Phi_a; \Gamma, \square\Gamma_1 \vdash e^* \backsim e^* :^c \tau \quad (\star\star)$
By Lemma 7 on $(\star\star)$ for the number of x_i of Γ times, we conclude :
$\Delta; \Phi_a; \square\Gamma_1 \vdash e^*[\overline{y_i/x_i}] \backsim e^*[\overline{y_i/x_i}] :^c \tau$.
Similarly, using $(\star\star)$ and the third premise and **rr-nochange**, we conclude $\Delta; \Phi_a; \Gamma \vdash e \backsim e : \square\tau$.

□

Theorem 13 **(Completeness of embedding in RelRef)**

1. If $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau$, then there exist e_1^*, e_2^* such that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$.

Proof. By induction on the typing derivations.

$$\textbf{Case}\quad \frac{\begin{array}{c}\Delta;\Phi_a;\Gamma\vdash e\backsim e:\tau\\ \forall x\in dom(\Gamma).\ \ (\star)\Delta;\Phi_a\models\Gamma(x)\sqsubseteq\square\,\Gamma(x)\ \ \ (\diamond)\end{array}}{\Delta;\Phi_a;\Gamma,\Gamma'\vdash e\backsim e:\square\,\tau}\ \textbf{rr-nochange}$$

By IH on $(\star)$, we get $\exists e^*$ such that $\Delta;\Phi_a;\Gamma\vdash e\backsim e\rightsquigarrow e^*\backsim e^*:\tau$ $(\dagger)$.
By Lemma 11 on $(\diamond)$, we get $\exists e_i=\texttt{coerce}_{\Gamma(x_i),\square\,(\Gamma(x_i)}$ for all $x_i\in dom(\Gamma)$ $(\dagger\dagger)$.
By **e-nochange** embedding rule using $(\dagger)$ and $(\dagger\dagger)$, we can conclude as follows:

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma\vdash e\backsim e\rightsquigarrow e^*\backsim e^*:\tau\quad(\dagger)\\ \forall x_i\in dom(\Gamma),\quad e_i=\texttt{coerce}_{\Gamma(x_i),\square\,\Gamma(x_i)}\quad(\dagger\dagger)\qquad \forall x_i\in dom(\Gamma),\quad \Delta;\Phi_a\models\Gamma(x_i)\sqsubseteq\square\,\Gamma(x_i)\end{array}}{\Delta;\Phi_a;\Gamma,\Gamma'\vdash e\backsim e\rightsquigarrow \text{let }\overline{y_i=e_i\ x_i}\text{ in }\texttt{NC}\ e^*[\overline{\texttt{der}\ y_i/x_i}]\backsim \text{let }\overline{y_i=e_i\ x_i}\text{ in }\texttt{NC}\ e^*[\overline{\texttt{der}\ y_i/x_i}]:\square\,\tau}\ \textbf{e-nochange}.$$

$$\textbf{Case}\quad \frac{\begin{array}{c}\Delta;\Phi_a;\Gamma\vdash e\backsim e':\textbf{list}[n]^\alpha\,\tau\quad(\star)\qquad \Delta;\Phi_a\wedge n=0;\Gamma\vdash e_1\backsim e_1':\tau'\quad(\diamond)\\ i,\Delta;\Phi_a\wedge n=i+1;h:\square\,\tau,tl:\textbf{list}[i]^\alpha\,\tau,\Gamma\vdash e_2\backsim e_2':\tau'\quad(\dagger)\\ i,\beta,\Delta;\Phi_a\wedge n=i+1\wedge\alpha=\beta+1;h:\tau,tl:\textbf{list}[i]^\beta\,\tau,\Gamma\vdash e_2\backsim e_2':\tau'\quad(\spadesuit)\end{array}}{\Delta;\Phi_a;\Gamma\vdash\ \textbf{case}\ e\ \textbf{of nil}\rightarrow e_1\mid h::tl\rightarrow e_2\backsim\ \textbf{case}\ e'\ \textbf{of nil}\rightarrow e_1'\mid h::tl\rightarrow e_2':\tau'}\ \textbf{rr-caseL}$$

By IH on $(\star)$, we get $\exists e^*$ and $\exists e'^*$ s.t. $\Delta;\Phi_a;\Gamma\vdash e\backsim e'\rightsquigarrow e^*\backsim e'^*:\text{list}[n]^\alpha\,\tau$ $(\star\star)$.
By IH on $(\diamond)$, we get $\exists e_1^*$ and $\exists e_1'^*$ s.t. $\Delta;n\doteq 0\wedge\Phi_a;\Gamma\vdash e_1\backsim e_1'\rightsquigarrow e_1^*\backsim e_1'^*:\tau'$ $(\diamond\diamond)$.
By IH on $(\dagger)$, we get $\exists e_2^*$ and $\exists e_2'^*$ s.t.
$i::S,\Delta;n\doteq i+1\wedge\Phi_a;h:\square\,\tau,tl:\text{list}[i]^\alpha\,\tau,\Gamma\vdash e_2\backsim e_2'\rightsquigarrow e_2^*\backsim e_2'^*:\tau'$ $(\dagger\dagger)$.
By IH on $(\spadesuit)$, we get $\exists e_2^{**}$ and $\exists e_2'^{**}$ s.t.
$i::S,\beta::S,\Delta;n\doteq i+1\wedge\alpha\doteq\beta+1\wedge\Phi_a;h:\tau,tl:\text{list}[i]^\beta\,\tau,\Gamma\vdash e_2\backsim e_2'\rightsquigarrow e_2^{**}\backsim e_2'^{**}:\tau'$ $(\spadesuit\spadesuit)$.

By **e-caseL** embedding rule using $(\star\star)$, $(\diamond\diamond)$, and $(\spadesuit\spadesuit)$, we can conclude as follows

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma\vdash e\backsim e'\rightsquigarrow e^*\backsim e'^*:\text{list}[n]^\alpha\,\tau\qquad \Delta;\Phi_a\wedge n=0;\Gamma\vdash e_1\backsim e_1'\rightsquigarrow {e_1}^*\backsim {e_1'}^*:\tau'\\ i,\Delta;\Phi_a\wedge n=i+1;h:\square\,\tau,tl:\text{list}[i]^\alpha\,\tau,\Gamma\vdash e_2\backsim e_2'\rightsquigarrow {e_2}^*\backsim {e_2'}^*:\tau'\\ i,\beta,\Delta;\Phi_a\wedge n=i+1\wedge\alpha=\beta+1;h:\tau,tl:\text{list}[i]^\beta\,\tau,\Gamma\vdash e_2\backsim e_2'\rightsquigarrow e_2^{**}\backsim e_2'^{**}:\tau'\end{array}}{\Delta;\Phi_a;\Gamma\vdash\begin{array}{l}\text{case }e\text{ of nil}\rightarrow e_1\\ \ \mid h\ ::tl\rightarrow e_2\end{array}\backsim\begin{array}{l}\text{case }e\text{ of nil}\rightarrow e_1'\\ \ \mid h\ ::tl\rightarrow e_2'\end{array}\rightsquigarrow\begin{array}{l}\text{case }e^*\text{ of nil }\rightarrow e_1^*\\ \mid h\ ::_{NC}tl\ \rightarrow e_2^*\\ \mid h::_C tl\ \ \rightarrow e_2^{**}\end{array}\backsim\begin{array}{l}\text{case }e'^*\text{ of nil }\rightarrow e_1'^*\\ \mid h\ ::_{NC}tl\ \rightarrow e_2'^*\\ \mid h::_C tl\ \ \rightarrow e_2'^{**}\end{array}:\tau'}\ \textbf{e-r-caseL}$$

$$\textbf{Case}\quad \frac{\begin{array}{c}\Delta;\Phi_a\vdash\tau_1\rightarrow\tau_2\ \texttt{wf}\\ \Delta;\Phi_a;x:\tau_1,f:\square\,(\tau_1\rightarrow\tau_2),\Gamma\vdash e\backsim e:\tau_2\quad(\star)\qquad \forall x\in dom(\Gamma).\Delta;\Phi_a\models\Gamma(x)\sqsubseteq\square\,\Gamma(x)\quad(\diamond)\end{array}}{\Delta;\Phi_a;\Gamma\vdash\textbf{fix}\ f(x).e\backsim\textbf{fix}\ f(x).e:\square\,(\tau_1\rightarrow\tau_2)}\ \textbf{rr-fixNC}$$

By IH on $(\star)$, we get $\exists e^*$ such that $\Delta;\Phi_a;x:\tau_1,f:\square\,(\tau_1\rightarrow\tau_2),\Gamma\vdash e\backsim e\rightsquigarrow e^*\backsim e^*:\tau_2$ $(\star\star)$.
By Lemma 11 on $(\diamond)$, we get $\exists e_i=\texttt{coerce}_{\Gamma(x_i),\square\,(\Gamma(x_i)}$ for all $x_i\in dom(\Gamma)$ $(\diamond\diamond)$.

By **e-fixNC** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\frac{\begin{array}{c}\Delta;\Phi_a\vdash\tau_1\rightarrow\tau_2\ \texttt{wf}\\ \Delta;\Phi_a;x:\tau_1,f:\square\,(\tau_1\rightarrow\tau_2),\Gamma\vdash e\backsim e\rightsquigarrow e^*\backsim e^*:\tau_2\qquad \forall x_i\in dom(\Gamma),\quad e_i=\texttt{coerce}_{\Gamma(x_i),\square\,\Gamma(x_i)}\\ \forall x_i\in dom(\Gamma),\quad \Delta;\Phi_a\models\Gamma(x)\sqsubseteq\square\,\Gamma(x)\qquad e^{**}=\text{let }\overline{y_i=e_i\ x_i}\text{ in fix}_{NC}\ f(x).e^*[\overline{\texttt{der}\ y_i/x_i}]\end{array}}{\Delta;\Phi_a;\Gamma\vdash\text{fix }f(x).e\backsim\text{fix }f(x).e\rightsquigarrow e^{**}\backsim e^{**}:\square\,(\tau_1\rightarrow\tau_2)}\ \textbf{e-r-fixNC}.$$

$$\textbf{Case}\quad \frac{\begin{array}{c}i::S,\Delta;\Phi_a;\Gamma\vdash e\backsim e':\tau\quad(\star)\\ i\not\in\textbf{FIV}(\Phi_a;\Gamma)\end{array}}{\Delta;\Phi_a;\Gamma\vdash\Lambda e\backsim\Lambda e':\forall i{::}S.\,\tau}\ \textbf{rr-iLam}$$

By IH on $(\star)$, we get $\exists e^*$ and $\exists e'^*$ such that $i::S,\Delta;\Phi_a;\Gamma\vdash e\backsim e'\rightsquigarrow e^*\backsim e'^*:\tau$ $(\star\star)$.
By **e-iLam** embedding rule using $(\star\star)$, we can conclude as follows:

$$\frac{i::S,\Delta;\Phi_a;\Gamma\vdash e\backsim e'\rightsquigarrow e^*\backsim {e'}^*:\tau\qquad i\not\in\text{FIV}(\Phi_a;\Gamma)}{\Delta;\Phi_a;\Gamma\vdash\Lambda.e\backsim\Lambda.e'\rightsquigarrow\Lambda i.e^*\backsim\Lambda i.e'^*:\forall i{::}S.\,\tau}\ \textbf{e-r-iLam}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \forall i{::}S.\,\tau \quad (\star) \\ \Delta \vdash I : S \quad ({\color{red}\diamond})\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \backsim e'[\,] : \tau\{I/i\}}$ **rr-iApp**

By IH on $(\star)$, we get $\exists e^*$ such that $\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \forall i{::}S.\,\tau \quad (\star\star)$.

By **e-iApp** embedding rule using $(\star\star)$ and $({\color{red}\diamond})$, we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \forall i{::}S.\,\tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[] \backsim e'[] \rightsquigarrow e^*[I] \backsim e'^*[I] : \tau\{I/i\}}\ \textbf{e-r-iApp}.$$

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau\{I/i\} \quad (\star) \qquad \Delta \vdash I :: S \quad ({\color{red}\diamond})}{\Delta; \Phi_a; \Gamma \vdash \textbf{pack}\ e \backsim \textbf{pack}\ e' : \exists i{::}S.\,\tau}$ **rr-pack**

By IH on $(\star)$, we get $\exists e^*$ and $\exists e'^*$ such that $\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau\{I/i\} \quad (\star\star)$.

By **e-pack** embedding rule using $(\star\star)$ and $({\color{red}\diamond})$, we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack}\ e \backsim \text{pack}\ e' \rightsquigarrow \text{pack}\ e^* \text{ with } I \backsim \text{pack}\ e'^* \text{ with } I : \exists i{::}S.\,\tau}\ \textbf{e-r-pack}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \exists i{::}S.\,\tau_1 \quad (\star) \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2 \quad ({\color{red}\diamond}) \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \textbf{unpack}\ e_1\ \textbf{as}\ x\ \textbf{in}\ e_2 \backsim \textbf{unpack}\ e_1'\ \textbf{as}\ x\ \textbf{in}\ e_2' : \tau_2}$ **rr-unpack1**

By IH on $(\star)$, we get $\exists e_1^*$ and $\exists e_1'^*$ such that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow e_1^* \backsim e_1'^* : \exists i{::}S.\,\tau_1 \quad (\star\star)$.

By IH on $({\color{red}\diamond})$, we get $\exists e_2^*$ and $\exists e_2'^*$ such that $i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^* \backsim e_2'^* : \tau_2 \quad ({\color{red}\diamond\diamond})$.

By **e-unpack** embedding rule using $(\star\star)$ and $({\color{red}\diamond\diamond})$, we can conclude as follows:

$$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \exists i{::}S.\,\tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2 \\ i \notin FV(\Phi_a; \Gamma, \tau_2, t_2) \qquad e_1^{**} = \text{unpack}\ e_1^* \text{ as } (x, i) \text{ in } e_2^* \qquad e_2^{**} = \text{unpack}\ e_1'^* \text{ as } (x, i) \text{ in } e_2'^*\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack}\ e_1 \text{ as } x \text{ in } e_2 \backsim \text{unpack}\ e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau_2}\ \textbf{e-r-unpack}.$$

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad ({\color{red}\diamond})}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau'}$ **rr-$\sqsubseteq$**

By IH on $(\star)$, we get $\exists e_1^*, e_2^*$ such that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \quad (\star\star)$.

By Lemma 11 on $({\color{red}\diamond})$, we can show that $\exists e' = \texttt{coerce}_{\tau,\tau'} \quad ({\color{red}\diamond\diamond})$.

By **e-r-$\sqsubseteq$** rule using $(\star\star)$, $({\color{red}\diamond\diamond})$ and $({\color{red}\diamond})$, we conclude as follows

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \texttt{coerce}_{\tau,\tau'}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\sqsubseteq$$

□

Theorem 14 (Soundness of algorithmic typechecking in RelRef)

1. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow {\color{red}\Phi}$ and $\text{FIV}(\Phi_a, \Gamma, \tau) \subseteq dom(\Delta, \psi_a)$ and θ_a is a valid substitution for ψ_a such that $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \backsim |e'| :^c \tau[\theta_a]$.
2. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow {\color{red}\tau} \Rightarrow {\color{red}\Phi}$ and $\text{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ and θ_a is a valid substitution for ψ_a such that $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \backsim |e'| :^c \tau[\theta_a]$.

Proof. By simultaneous induction on the given algorithmic typing derivations.

Proof of Theorem 14.1:

Case $\dfrac{\Delta; \psi_a; \Phi_a; \square\Gamma \vdash e \ominus e \downarrow \tau \Rightarrow {\color{red}\Phi}}{\Delta; \psi_a; \Phi_a; \Gamma', \square\Gamma \vdash \texttt{NC}\ e \ominus \texttt{NC}\ e \downarrow \square\tau \Rightarrow {\color{red}\Phi}}$ **alg-r-nochange-$\downarrow$**

TS: $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\Gamma[\theta_a] \vdash \texttt{NC}\ |e| \ominus \texttt{NC}\ |e| : \square\tau[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma', \Box\,\Gamma, \tau) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$
Using $(\star)$, we can show that

a) $\text{FIV}(\Phi_a, \Box\,\Gamma, \tau) \subseteq dom(\Delta, \psi_a)$.

By IH1 on the premise using a) and $(\star\star)$, we can show that

$$\Delta; \Phi_a[\theta_a]; \Box\,\Gamma[\theta_a] \vdash |e| \ominus |e| : \tau[\theta_a] \tag{2.4}$$

By the **c-nochange** rule using eq. (2.4), we obtain $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \Box\,\Gamma[\theta_a] \vdash \texttt{NC}\ |e| \ominus \texttt{NC}\ |e| : \Box\,\tau[\theta_a]$.

Case
$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \textbf{list}[n]^{\alpha}\,\tau \Rightarrow \Phi_e \qquad \Delta; \psi_a; n \doteq 0 \land \Phi_a; \Gamma \vdash e_1 \ominus e'_1 \downarrow \tau' \Rightarrow \Phi_1 \\ i :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \land \Phi_a; h : \Box\,\tau, tl : \textbf{list}[i]^{\alpha}\,\tau, \Gamma \vdash e_2 \ominus e'_2 \downarrow \tau' \Rightarrow \Phi_2 \\ i :: \mathbb{N}, \beta :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \land \alpha \doteq \beta+1 \land \Phi_a; h : \tau, tl : \textbf{list}[i]^{\beta}\,\tau, \Gamma \vdash e_3 \ominus e'_3 \downarrow \tau' \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_1) \land (\forall i :: \mathbb{N}.(n \doteq i+1) \to (\Phi_2 \land \forall\beta :: \mathbb{N}.(\alpha \doteq \beta+1) \to \Phi_3))\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \begin{array}{l}\textbf{case}\ e\ \textbf{of}\ \textbf{nil}\ \to e_1 \\ \mid h\ ::_{NC}\ tl\ \to e_2 \\ \mid h\ ::_{C}\ tl\ \to e_3\end{array} \ominus \begin{array}{l}\textbf{case}\ e'\ \textbf{of}\ \textbf{nil}\ \to e'_1 \\ \mid h\ ::_{NC}\ tl\ \to e'_2 \\ \mid h\ ::_{C}\ tl\ \to e'_3\end{array} \downarrow \tau' \Rightarrow (\Phi_e \land \Phi_{body})}\ \textbf{alg-r-caseL-}\!\downarrow$$

TS:

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \begin{array}{l}\text{case}\ e\ \text{of nil}\ \to |e_1| \\ \mid h\ ::_N\ tl\ \to |e_2| \\ \mid h\ ::_C\ tl \to |e_3|\end{array} \ominus \begin{array}{l}\text{case}\ e'\ \text{of nil}\ \to |e'_1| \\ \mid h\ ::_N\ tl\ \to |e'_2| \\ \mid h\ ::_C\ tl \to |e'_3|\end{array} : \tau'[\theta_a]$$

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, \tau') \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models (\Phi_e \land \Phi_{body})[\theta_a]$ $(\star\star)$
Using $(\star)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta; \Phi_a[\theta_a] \models \Phi_e[\theta_a]$
b) $\Delta; n[\theta_a] \doteq 0 \land \Phi_a[\theta_a] \models \Phi_1[\theta_a]$
c) $i :: S, \Delta; n[\theta_a] \doteq i+1 \land \Phi_a[\theta_a] \models \Phi_2[\theta_a]$
d) $i :: S, \beta :: S, \Delta; n[\theta_a] \doteq i+1 \land \alpha[\theta_a] \doteq \beta+1 \land \Phi_a[\theta_a] \models \Phi_3[\theta_a]$

By IH2 on the first premise using a) and $(\star)$, we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \text{list}[n[\theta_a]]^{\alpha[\theta_a]}\,\tau[\theta_a] \tag{2.5}$$

By IH1 on the second premise using b) and $(\star)$, we can show that

$$\Delta; n[\theta_a] \doteq 0 \land \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1| \ominus |e'_1| : \tau'[\theta_a] \tag{2.6}$$

By IH1 on the third premise using c) and ($\star$), we can show that

$$i :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_2| \ominus |e_2'| : \tau'[\theta_a] \tag{2.7}$$

By IH1 on the fourth premise using d) and ($\star$), we can show that

$$i :: S, \beta :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \alpha[\theta_a] \doteq \beta + 1 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_3| \ominus |e_3'| : \tau'[\theta_a] \tag{2.8}$$

Then by **c-r-caseL** rule using eqs. (2.5) to (2.8), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \begin{array}{l} \text{case } e \text{ of nil } \rightarrow |e_1| \\ \mid h \ ::_N \ tl \ \rightarrow |e_2| \\ \mid h ::_C tl \rightarrow |e_3| \end{array} \ominus \begin{array}{l} \text{case } e' \text{ of nil } \rightarrow |e_1'| \\ \mid h \ ::_N \ tl \ \rightarrow |e_2'| \\ \mid h ::_C tl \rightarrow |e_3'| \end{array} : \tau'[\theta_a]$$

Case $\dfrac{\begin{array}{c} i \in \mathbf{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \square\, \tau \Rightarrow \Phi_1 \\ \Delta; i, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \mathbf{list}[i]^\alpha\, \tau \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathbf{cons}_{NC}(e_1, e_2) \ominus \mathbf{cons}_{NC}(e_1', e_2') \downarrow \mathbf{list}[n]^\alpha\, \tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}. \Phi_2'}$ **alg-r-consNC-$\downarrow$**

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \text{cons}_{NC}(|e_1|, |e_2|) \ominus \text{cons}_{NC}(|e_1'|, |e_2'|) : \text{list}[n[\theta_a]]^{\alpha[\theta_a]}\, \tau[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, \text{list}[n]^\alpha\, \tau) \subseteq dom(\Delta, \psi_a)$ ($\star$) and $\Delta; \Phi_a[\theta_a] \models (\Phi_1 \wedge \exists i :: \mathbb{N}. \Phi_2')[\theta_a]$ ($\star\star$)
Using ($\star$), ($\star\star$)'s derivation must be in a form such that we have

a) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a]$
b) $\Delta \vdash I :: \mathbb{N}$
c) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, i \mapsto I]$
d) $\Delta; \Phi_a[\theta_a] \models (I + 1) \doteq n[\theta_a]$

By IH1 on the second premise using ($\star$) and a), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1| \ominus |e_1'| : \square\, \tau[\theta_a] \tag{2.9}$$

By IH1 on the third premise using ($\star$) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_2| \ominus |e_2'| : \text{list}[I]^{\alpha[\theta_a]}\, \tau[\theta_a] \tag{2.10}$$

By **c-r-cons2** typing rule using eqs. (2.9) and (2.10), we obtain
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \text{cons}_{NC}(|e_1|, |e_2|) \ominus \text{cons}_{NC}(|e_1'|, |e_2'|) : \text{list}[I+1]^{\alpha[\theta_a]}\, \tau[\theta_a]$.
We conclude by applying **c-r-$\sqsubseteq$** rule to this using d) .

Proof of Theorem 14.2:

Case $\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash \tau \texttt{ wf} \qquad \textbf{FIV}(\tau) \in \Delta}{\Delta; \psi_a; \Phi_a; \Gamma \vdash (e : \tau) \ominus (e' : \tau) \uparrow \tau \Rightarrow \Phi}$ **alg-r-anno-↑**

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |(e : \tau, t)| \ominus |(e' : \tau, t)| : \tau[\theta_a]$.

Since by definition, $\forall e.\ |(e :\_)| = |e|$, STS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \tau[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$

Using the third premise, we can show that

a) $\text{FIV}(\Phi_a, \Gamma, \tau) \subseteq dom(\Delta, \psi_a)$.

By IH2 on the first premise using $(\star\star)$ and a), we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \tau[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \Box\tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{der } e_1 \ominus \texttt{der } e_2 \uparrow \tau \Rightarrow \Phi}$ **alg-r-der-↑**

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \texttt{der } |e| \backsim \texttt{der } |e'| :^c \tau[\theta\, \theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$ such that $\Delta \ \triangleright\ \theta_a : \psi_a$ are derivable.

By IH2 on the first premise using $(\star)$ and $\star\star$, we obtain

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \Box\, \tau[\theta_a] \tag{2.11}$$

Then, by **c-der** rule using eq. (5.30), we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \texttt{der } |e| \ominus \texttt{der } |e'| : \tau[\theta\, \theta_a]$.

□

Theorem 15 (Completeness of algorithmic typechecking in RelRef)

1. Assume that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau$. Then, there exist e_1', e_2' such that $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$ and $|e_1'| = e_1$ and $|e_2'| = e_2$.

Proof. By simultaneous induction on the given Core typing derivations.

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \Box\tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{der } e_1 \backsim \texttt{der } e_2 :^c \tau}$ **c-der**

By IH on the premise, $\exists e_1', e_2'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau, t \Rightarrow \Phi$
b) $\Delta; \Phi_a \models \Phi$
c) $|e_1'| = e_1$ and $|e_2'| = e_2$

Then, we can conclude by using a), b) and c) as follows:
$\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \Box\tau \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der } e_1' \ominus \texttt{der } e_2' \downarrow \tau \Rightarrow \Phi}$ **alg-r-der-↓** and

1.

$$\dfrac{\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \square\tau \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der}\ e_1' \ominus \texttt{der}\ e_2' \downarrow \tau \Rightarrow \Phi}\ \textbf{alg-r-der-}\downarrow}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e_1' : \tau) \ominus (\texttt{der}\ e_2' : \tau) \uparrow \tau \Rightarrow \Phi}\ \textbf{alg-r-anno-}\uparrow \quad \Delta; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi' \text{ by } \textit{Lemma } 8}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e_1' : \tau) \ominus (\texttt{der}\ e_2' : \tau) \downarrow \tau \Rightarrow \Phi \wedge \Phi'}\ \textbf{alg-r-}\uparrow\downarrow$$

2. By c), $|(\texttt{der}\ e_i' : \tau)| = \texttt{der}\ |e_i'|$.
3. By b) and Lemma 8.

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau'}\ \textbf{c-r-}\equiv$

By IH on the first premise, $\exists e_1', e_2'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau \Rightarrow \Phi_1$
b) $\Delta; \Phi_a \models \Phi_1$
c) $|e_1'| = e$ and $|e_2'| = e'$

By IH on the second premise,

d) $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2$
e) $\Delta; \Phi_a \models \Phi_2$.

Then, we can conclude as follows

1. By using a) and d)

$$\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau \Rightarrow \Phi_1}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e_1' : \tau) \ominus (e_2' : \tau) \uparrow \tau \Rightarrow \Phi_1}\ \textbf{alg-r-anno-}\uparrow \qquad \Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e_1' : \tau) \ominus (e_2' : \tau) \downarrow \tau' \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-}\uparrow\downarrow$$

2. By using b), e) ,we can show that $\Delta; \Phi_a \models \Phi_1 \wedge \Phi_2$
3. By c), $|(e_1' : \tau)| = e_1'$ and $(e_2' : \tau) = e_2'$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \rightarrow \tau_2 \\ \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \tau_1\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' :^c \tau_2}\ \textbf{c-r-app}$

By IH on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \rightarrow \tau_2 \Rightarrow \Phi_1$
b) $\Delta; \Phi_a \models \Phi_1$
c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By IH on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

d) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1 \Rightarrow \Phi_2$
e) $\Delta; \Phi_a \models \Phi_2$
f) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

Then, we can conclude as follows

1.

$$\cfrac{\cfrac{\cfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \to \tau_2 \Rightarrow \color{red}{\Phi_1}}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\overline{e}_1 : \tau_1 \to \tau_2) \ominus \overline{e}_1' : \tau_1 \to \tau_2) \uparrow \color{red}{\tau_1 \to \tau_2} \Rightarrow \color{red}{\Phi_1}} \textbf{ alg-r-anno-}\uparrow \quad \Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1 \Rightarrow \color{red}{\Phi_2}}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E_2 \uparrow \color{red}{\tau_2} \Rightarrow \color{red}{\Phi_1 \wedge \Phi_2}} \textbf{ c-r-app}}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E_2 \downarrow \tau_2 \Rightarrow \color{red}{\Phi_1 \wedge \Phi_2}} \textbf{ alg-r-}\uparrow\downarrow$$

where $E_1 = (\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2$ and $E_2 = (\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'$.

2. By using b) and e) .
3. Using c) and f), $|(\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2| = e_1\ e_2$ and $|(\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'| = e_1'\ e_2'$.

Case $$\cfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \exists i{::}S.\, \tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)}{\Delta; \Phi_a; \Gamma \vdash \textbf{unpack } e_1 \textbf{ as } (x, i) \textbf{ in } e_2 \backsim \textbf{unpack } e_1' \textbf{ as } (x, i) \textbf{ in } e_2' :^c \tau_2} \textbf{ c-r-unpack1}$$

By IH on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \exists i{::}S.\, \tau_1 \Rightarrow \color{red}{\Phi_1}$
b) $\Delta; \Phi_a \models \Phi_1$
c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By IH on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

d) $i :: S, \Delta; \cdot; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_2 \Rightarrow \color{red}{\Phi_2}$
e) $i :: S, \Delta; \Phi_a \models \Phi_2$
f) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

Then, we can conclude as follows

1.

$$\cfrac{\cfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \exists i{::}S.\, \tau_1 \Rightarrow \color{red}{\Phi_1} \quad (a)}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E_1' \uparrow \color{red}{\exists i{::}S.\, \tau_1} \Rightarrow \color{red}{\Phi_1}} \textbf{ alg-r-anno-}\uparrow \quad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_2 \Rightarrow \color{red}{\Phi_2} \qquad \Phi' = \Phi_1 \wedge \Phi_2'}{\Delta; \cdot; \Phi_a; \Gamma \vdash \text{unpack } E_1 \text{ as } (x, i) \text{ in } \overline{e}_2 \ominus \text{unpack } E_1' \text{ as } (x, i) \text{ in } \overline{e}_2' \downarrow \tau_2 \Rightarrow \color{red}{\Phi'}} \textbf{ alg-r-unpack-}\downarrow$$

where $E_1 = (\overline{e}_1 : \exists i{::}S.\, \tau_1)$ and $E_2 = (\overline{e}_1' : \exists i{::}S.\, \tau_1)$

2. By using b) and e).
3. Using c) and f), $|\text{unpack } (\overline{e}_1 : \exists i{::}S.\, \tau_1) \text{ as } (x, i) \text{ in } \overline{e}_2| = \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2$ and $|\text{unpack } (\overline{e}_1' : \exists i{::}S.\, \tau_1) \text{ as } (x, i) \text{ in } \overline{e}_2'| = \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2'$.

$\square$

3 RelInf

3.1 Syntax of RelInf

$$
\begin{array}{llll}
\text{Unary types} & A & ::= & \text{bool} \mid \text{int} \mid A_1 \rightarrow A_2 \\
\text{Relational types} & \tau & ::= & \text{bool}_r \mid \text{int}_r \mid \tau_1 \rightarrow \tau_2 \mid U\,(A_1, A_2) \\
\text{Expressions} & e & ::= & x \mid n \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fix } f(x).e \mid e_1\ e_2 \\
\text{Value} & v & := & n \mid \text{true} \mid \text{false} \mid \text{fix } f(x).e
\end{array}
$$

Figure 22: Syntax of values and expressions in RelInf

$$
\begin{array}{llll}
\text{Unary types} & A & ::= & \text{bool} \mid \text{int} \mid A_1 \rightarrow A_2 \\
\text{Relational types} & \tau & ::= & \text{bool}_r \mid \text{int}_r \mid \tau_1 \rightarrow \tau_2 \mid U\,(A_1, A_2) \\
\text{Expressions} & e & ::= & x \mid n \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fix } f(x).e \mid e_1\ e_2 \mid \texttt{switch}\ e \\
\text{Value} & v & := & n \mid \text{true} \mid \text{false} \mid \text{fix } f(x).e
\end{array}
$$

Figure 23: Syntax of values and expressions in RelInf Core

$$
\begin{array}{llll}
\text{Unary types} & A & ::= & \text{bool} \mid \text{int} \mid A_1 \rightarrow A_2 \\
\text{Relational types} & \tau & ::= & \text{bool}_r \mid \text{int}_r \mid \tau_1 \rightarrow \tau_2 \mid U\,(A_1, A_2) \\
\text{Expressions} & e & ::= & \ldots \mid \texttt{switch}\ e \mid (e : \tau) \mid (e : A) \\
\text{Value} & v & := & n \mid \text{true} \mid \text{false} \mid \text{fix } f(x).e
\end{array}
$$

Figure 24: Syntax of values and expressions in BiRelInf

$$\frac{\Omega(x) = A}{\Omega \vdash x : A}\ \textbf{var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Omega \vdash \mathtt{b} : \text{bool}}\ \textbf{bool} \qquad \frac{}{\Omega \vdash \mathtt{n} : \text{int}}\ \textbf{const}$$

$$\frac{\vdash^{\mathsf{A}} A_1 \to A_2\ \mathtt{wf} \qquad x : A_1, f : A_1 \to A_2, \Omega \vdash e : A_2}{\Omega \vdash \text{fix } f(x).e : A_1 \to A_2}\ \textbf{fix} \qquad \frac{\Omega \vdash e_1 : A_1 \to A_2 \qquad \Omega \vdash e_2 : A_1}{\Omega \vdash e_1\ e_2 : A_2}\ \textbf{app}$$

$$\frac{\Omega \vdash e : \text{bool} \qquad \Omega \vdash e_1 : \tau \qquad \Omega \vdash e_2 : \tau}{\Omega \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 : \tau}\ \textbf{if} \qquad \frac{\Omega \vdash e : A \qquad \models^{\mathsf{A}} A \sqsubseteq A'}{\Omega \vdash e : A'}\ \sqsubseteq_{\text{exec}}$$

Figure 25: RelInf unary typing rules

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x : \tau}\ \textbf{r-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Gamma \vdash \mathtt{b} \backsim \mathtt{b} : \text{bool}_r}\ \textbf{r-bool} \qquad \frac{|\Gamma|_1 \vdash e_1 : A_1 \qquad |\Gamma|_2 \vdash e_2 : A_2}{\Gamma \vdash e_1 \backsim e_2 : U\,(A_1, A_2)}\ \textbf{r-switch}$$

$$\frac{\Gamma \vdash e \backsim e' : \text{bool}_r \qquad \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Gamma \vdash e_2 \backsim e_2' : \tau}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \backsim \text{if } e' \text{ then } e_1' \text{ else } e_2' : \tau}\ \textbf{r-if}$$

$$\frac{\Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 : \tau_1 \to \tau_2}\ \textbf{r-fix}$$

$$\frac{\Gamma \vdash e_1 \backsim e_1' : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 \backsim e_2' : \tau_1}{\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' : \tau_2}\ \textbf{r-app} \qquad \frac{\Gamma \vdash e_1 \backsim e_2 : \tau \qquad \models \tau \sqsubseteq \tau'}{\Gamma \vdash e_1 \backsim e_2 : \tau'}\ \textbf{r-}\sqsubseteq$$

Figure 26: RelInf binary typing rules

$$\frac{\models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A_1' \to A_2'} \to \textbf{exec} \qquad \frac{}{\models^{\mathsf{A}} A \sqsubseteq A}\textbf{u-refl} \qquad \frac{\models^{\mathsf{A}} A_1 \sqsubseteq A_2 \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_3}{\models^{\mathsf{A}} A_1 \sqsubseteq A_3}\textbf{u-tran}$$

Figure 27: RelInf unary subtyping rules

$$\frac{}{\models U\,(A_1 \to A_2, A_1' \to A_2') \sqsubseteq U\,(A_1, A_1') \to U\,(A_2, A_2')} \to \textbf{execdiff}$$

$$\frac{\models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\models U\,(A_1, A_2) \sqsubseteq U\,(A_1', A_2')}\ \textbf{U} \qquad \frac{}{\models \tau \sqsubseteq U\,(|\tau|_1, |\tau|_2)}\ \textbf{W} \qquad \frac{\models \tau_1' \sqsubseteq \tau_1 \qquad \models \tau_2 \sqsubseteq \tau_2'}{\models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'} \to$$

$$\frac{}{\models \tau \sqsubseteq \tau}\textbf{refl} \qquad \frac{\models \tau_1 \sqsubseteq \tau_2 \qquad \models \tau_2 \sqsubseteq \tau_3}{\models \tau_1 \sqsubseteq \tau_3}\textbf{trans}$$

Figure 28: RelInf binary subtyping rules

$$
\begin{array}{rcl}
|.|_{i \in \{1,2\}} & : & \textsf{Binary type} \to \textsf{Unary type} \\
|\mathrm{int}_r|_i & = & \mathrm{int} \\
|\mathrm{bool}_r|_i & = & \mathrm{bool} \\
|\tau_1 \to \tau_2|_i & = & |\tau_1|_i \to |\tau_2|_i \\
|U\,(A_1, A_2)|_i & = & A_i \\
|\Gamma, x : \tau|_i & = & |\Gamma|_i, x : |\tau|_i \\
|\emptyset|_i & = & \emptyset
\end{array}
$$

Figure 29: RelInf refinement removal operation

$$
\frac{}{\models \mathrm{bool}_r \equiv \mathrm{bool}_r}\ \textbf{eq-boolr}
\qquad
\frac{}{\models \mathrm{int}_r \equiv \mathrm{int}_r}\ \textbf{eq-intr}
\qquad
\frac{\models \tau_1 \equiv \tau_1' \qquad \models \tau_2 \equiv \tau_2'}{\models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'}\ \textbf{eq-fun}
$$

$$
\frac{\models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \qquad \models^{\mathsf{A}} A_2' \sqsubseteq A_2}{\models U\,(A_1, A_2) \equiv U\,(A_1', A_2')}\ \textbf{eq-U}
$$

Figure 30: RelInf relational type equivalence rules

$$
\frac{x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2}{\Omega \vdash \mathrm{fix}\ f(x).e :^c A_1 \to A_2}\ \textbf{c-u-fix}
\qquad
\frac{\Omega(x) = A}{\Omega \vdash x :^c A}\ \textbf{c-u-var}
\qquad
\frac{}{\Omega \vdash \mathtt{n} :^c \mathrm{int}}\ \textbf{c-u-const}
$$

$$
\frac{\Omega \vdash e :^c A \qquad \models A \sqsubseteq A'}{\Omega \vdash_{k'}^{t'} e :^c A'}\ \textbf{c-u-}\sqsubseteq
$$

Figure 31: $\mathsf{RelInf\ Core}$ unary typing rules

$$
\frac{|\Gamma|_1 \vdash e_1 :^c A_1 \qquad |\Gamma|_2 \vdash e_2 :^c A_2}{\Gamma \vdash \mathtt{switch}\ e_1 \backsim \mathtt{switch}\ e_2 :^c U\,(A_1, A_2)}\ \textbf{c-switch}
$$

$$
\frac{\vdash \tau_1 \to \tau_2\ \mathtt{wf} \qquad x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 :^c \tau_2}{\Gamma \vdash \mathrm{fix}\ f(x).e_1 \backsim \mathrm{fix}\ f(x).e_2 :^c \tau_1 \to \tau_2}\ \textbf{c-r-fix}
\qquad
\frac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x :^c \tau}\ \textbf{c-r-var}
$$

$$
\frac{}{\Gamma \vdash \mathtt{n} \backsim \mathtt{n} :^c \mathrm{int}_r}\ \textbf{c-r-const}
\qquad
\frac{\Gamma \vdash e \backsim e' :^c \tau \qquad \models \tau \equiv \tau'}{\Gamma \vdash e \backsim e' :^c \tau'}\ \textbf{c-r-}\sqsubseteq
$$

Figure 32: $\mathsf{RelInf\ Core}$ binary typing rules

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x \backsim x \rightsquigarrow x \backsim x : \tau}\ \textbf{e-r-var} \qquad \frac{}{\Gamma \vdash \mathtt{n} \backsim \mathtt{n} \rightsquigarrow \mathtt{n} \backsim \mathtt{n} : \text{int}_r}\ \textbf{e-r-const}$$

$$\frac{\vdash \tau_1 \to \tau_2\ \mathtt{wf} \qquad x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau_2}{\Gamma \vdash \text{fix}\ f(x).e_1 \backsim \text{fix}\ f(x).e_2 \rightsquigarrow \text{fix}\ f(x).{e_1}^* \backsim \text{fix}\ f(x).{e_2}^* : \tau_1 \to \tau_2}\ \textbf{e-r-fix}$$

$$\frac{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \models \tau \sqsubseteq \tau' \qquad e' = \mathtt{coerce}_{\tau,\tau'}}{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\sqsubseteq$$

$$\frac{|\Gamma|_1 \vdash e_1 \rightsquigarrow e_1^* : A_1 \qquad |\Gamma|_2 \vdash e_2 \rightsquigarrow e_2^* : A_2}{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \mathtt{switch}\ e_1^* \backsim \mathtt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}$$

Figure 33: RelInf binary embedding rules

$$\frac{\Omega(x) = A}{\Omega \vdash x \rightsquigarrow x : A}\ \textbf{e-u-var} \qquad \frac{}{\Omega \vdash \mathtt{n} \rightsquigarrow \mathtt{n} : \text{int}}\ \textbf{e-u-const}$$

$$\frac{\vdash^{\mathsf{A}} A_1 \to A_2\ \mathtt{wf} \qquad ; x : A_1, f : A_1 \to A_2, \Omega \vdash e \rightsquigarrow e^* : A_2}{\Omega \vdash \text{fix}\ f(x).e \rightsquigarrow \text{fix}\ f(x).e^* : A_1 \to A_2}\ \textbf{e-u-fix}$$

$$\frac{\Omega \vdash e \rightsquigarrow e^* : A \qquad \models^{\mathsf{A}} A \sqsubseteq A'}{\Omega \vdash e \rightsquigarrow e^* : A'}\ \textbf{e-u-}\sqsubseteq$$

Figure 34: RelInf unary embedding rules

$$\frac{\Omega(x) = \tau}{\Omega \vdash x \uparrow A}\ \textbf{alg-u-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Omega \vdash \mathtt{b} \uparrow \text{bool}}\ \textbf{alg-u-bool} \qquad \frac{\Omega \vdash e \uparrow A' \qquad \models A' \sqsubseteq A}{\Omega \vdash e \downarrow A}\ \textbf{alg-u-}\uparrow\downarrow$$

$$\frac{\Omega \vdash e \downarrow A}{\Omega \vdash (e : A) \uparrow A}\ \textbf{alg-u-anno-}\uparrow \qquad \frac{\Omega \vdash e \uparrow \text{bool} \qquad \Omega \vdash e_1 \downarrow A \qquad \Omega \vdash e_2 \downarrow A}{\Omega \vdash \text{if}\ e\ \text{then}\ e_1\ \text{else}\ e_2 \downarrow A}\ \textbf{alg-u-if}$$

$$\frac{\Omega, x : A_1, f : A_1 \to A_2 \vdash e_1 \downarrow A_2}{\Omega \vdash \text{fix}\ f(x).e_1 \downarrow A_1 \to A_2}\ \textbf{alg-u-fix}$$

Figure 35: RelInf unary algorithmic typing rules

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x \ominus x \uparrow \color{red}{\tau}}\ \textbf{alg-r-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Gamma \vdash \mathtt{b} \ominus \mathtt{b} \uparrow \color{red}{\text{bool}_r}}\ \textbf{alg-r-bool}$$

$$\frac{|\Gamma|_1 \vdash e_1 \uparrow \color{red}{A_1} \qquad |\Gamma|_2 \vdash e_2 \uparrow \color{red}{A_2}}{\Gamma \vdash \mathtt{switch}\ e_1 \ominus \mathtt{switch}\ e_2 \uparrow \color{red}{U\,(A_1, A_2)}}\ \textbf{alg-r-switch}\!\uparrow$$

$$\frac{|\Gamma|_1 \vdash e_1 \downarrow A_1 \qquad |\Gamma|_2 \vdash e_2 \downarrow A_2}{\Gamma \vdash \mathtt{switch}\ e_1 \ominus \mathtt{switch}\ e_2 \downarrow U\,(A_1, A_2)}\ \textbf{alg-r-switch}\!\downarrow$$

$$\frac{\Gamma \vdash e \ominus e' \uparrow \color{red}{\text{bool}_r} \qquad \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \qquad \Gamma \vdash e_2 \ominus e_2' \downarrow \tau}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \ominus \text{if } e' \text{ then } e_1' \text{ else } e_2' \downarrow \tau}\ \textbf{alg-r-if}$$

$$\frac{\Gamma, x : \tau_1, f : \tau_1 \to \tau_2 \vdash e_1 \ominus e_2 \downarrow \tau_2}{\Gamma \vdash \text{fix } f(x).e_1 \ominus \text{fix } f(x).e_2 \downarrow \tau_1 \to \tau_2}\ \textbf{alg-r-fix}$$

$$\frac{\Gamma \vdash e_1 \ominus e_1' \uparrow \color{red}{\tau_1 \to \tau_2} \qquad \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_1}{\Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \uparrow \color{red}{\tau_2}}\ \textbf{alg-r-app} \qquad \frac{\Gamma \vdash e \ominus e' \uparrow \color{red}{\tau'} \qquad \models \tau' \equiv \tau}{\Gamma \vdash e \ominus e' \downarrow \tau}\ \textbf{alg-}\!\uparrow\downarrow$$

$$\frac{\Gamma \vdash e \ominus e' \downarrow \tau}{\Gamma \vdash (e : \tau) \ominus (e : \tau) \uparrow \color{red}{\tau}}\ \textbf{alg-r-anno-}\!\uparrow$$

Figure 36: RelInf binary algorithmic typing rules

$$\frac{}{\models \text{bool}_r \equiv \text{bool}_r}\ \textbf{alg-boolr} \qquad \frac{}{\models \text{int}_r \equiv \text{int}_r}\ \textbf{alg-intr} \qquad \frac{\models^{\mathsf{A}} A \sqsubseteq A' \qquad \models^{\mathsf{A}} A' \sqsubseteq A}{\models U\ A \equiv U\ A'}\ \textbf{alg-u}$$

$$\frac{\models \tau_1' \equiv \tau_1 \qquad \models \tau_2 \equiv \tau_2'}{\models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'}\ \textbf{alg-}\!\to$$

Figure 37: RelInf algorithmic subtyping rules

3.2 RelInf Lemmas

Lemma 16 (Reflexivity of Algorithmic Binary Type Equivalence in RelInf)
$\models \tau \equiv \tau$.

Proof. By induction on the binary type.

Case $\mathbf{bool}_r$
It is proved by algorithmic binary type equivalence rule **alg-boolr**.

Case $\tau_1 \to \tau_2$
By IH on τ_1,$\models \tau_1 \equiv \tau_1$.
By IH on τ_2, $\models \tau_2 \equiv \tau_2$
By the above statements and rule **alg-**$\to$, we conclude
$\models \tau_1 \to \tau_2 \equiv \tau_1 \to \tau_2$.

Case $U(A_1, A_2)$
TS: $\models U(A_1, A_2) \equiv U(A_1, A_2)$.
By unary subtyping rule **u-refl**, $\models^{\mathsf{A}} A_1 \sqsubseteq A_1$ and $\models^{\mathsf{A}} A_2 \sqsubseteq A_2$.
By the above statements and rule **alg-u**, we conclude
$\models U(A_1, A_2) \equiv U(A_1, A_2)$.

$\square$

Lemma 17 (Existence of coercions for relational subtyping in RelInf)
If $\models \tau \sqsubseteq \tau'$ then there exists $\texttt{coerce}_{\tau,\tau'} \in$ Core s.t. $\cdot \vdash \texttt{coerce}_{\tau,\tau'} \backsim \texttt{coerce}_{\tau,\tau'} :^{c} \tau \to \tau'$.

Proof. Proof is by induction on the subtyping derivation. We denote the witness e of type $\tau \to \tau'$ as $\texttt{coerce}_{\tau,\tau'}$ for clarity.

Case $$\dfrac{\models \tau_1' \sqsubseteq \tau_1 \quad (\star) \qquad \models \tau_2 \sqsubseteq \tau_2' \quad (\diamond)}{\models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'} \to$$
By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1',\tau_1}$. $\cdot \vdash \texttt{coerce}_{\tau_1',\tau_1} \backsim \texttt{coerce}_{\tau_1',\tau_1} :^{c} \tau_1' \to \tau_1$
By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau_2,\tau_2'}$. $\cdot \vdash \texttt{coerce}_{\tau_2,\tau_2'} \backsim \texttt{coerce}_{\tau_2,\tau_2'} :^{c} \tau_2 \to \tau_2'$

Then, using these two statements,we can construct the following derivation where $e = \text{fix } f(x).\text{fix } f(y).\texttt{coerce}_{\tau_2,\tau_2'}\ (x\ (\texttt{c}$
$\cdot \vdash e \backsim e :^{c} (\tau_1 \to \tau_2) \to \tau_1' \to \tau_2'$

Case $$\dfrac{}{\cdot \models \tau \sqsubseteq U(|\tau|_1, |\tau|_2)}\ \mathbf{W}$$
Then, we can immediately construct the derivation using the rule **c-switch**.

$$\dfrac{}{\cdot \vdash \text{fix } f(x).\texttt{switch}\ x \backsim \text{fix } f(x).\texttt{switch}\ x :^{c} \tau \to U(|\tau|_1, |\tau|_2)}$$

Case $$\dfrac{\models \tau_1 \sqsubseteq \tau_2 \quad (\star) \qquad \models \tau_2 \sqsubseteq \tau_3 \quad (\diamond)}{\models \tau_1 \sqsubseteq \tau_3}\mathbf{tran}$$
By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1,\tau_2}$. $i :: S, \cdot \vdash \texttt{coerce}_{\tau_1,\tau_2} \backsim \texttt{coerce}_{\tau_1,\tau_2} :^{c} \tau_1 \to \tau_2$
By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau_2,\tau_3}$. $i :: S, \cdot \vdash \texttt{coerce}_{\tau_2,\tau_3} \backsim \texttt{coerce}_{\tau_2,\tau_3} :^{c} \tau_2 \to \tau_3$

Then, using $(\star)$ and $(\diamond)$, we can construct the derivation simply by function composition

$$\dfrac{}{\cdot \vdash \text{fix } f(x).\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) \backsim \text{fix } f(x).\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) :^{c} \tau_1 \to \tau_3}$$

Case $\dfrac{}{\models U\,(A_1 \to A_2, A_1' \to A_2') \sqsubseteq U\,(A_1, A_1') \to U\,(A_2, A_2')} \to \textbf{execdiff}$
Then, we can immediately construct the following derivation where $e = \text{fix } f(x).\text{fix } f(y).\texttt{switch}\ (x\ y)$ using the **c-switch** and **c-app** rules.
$\cdot \vdash e \backsim e :^c (U\,(A_1 \to A_2, A_1' \to A_2')) \to U\,(A_1, A_1') \to U\,(A_2, A_2')$

Case $\dfrac{\models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\models U\,(A_1, A_2) \sqsubseteq U\,(A_1', A_2')}\ \mathbf{U}$
Then, we can immediately construct the following derivation where $e = \text{fix } f(x).\texttt{switch}\ x$ using the **c-switch** and **c-u-$\sqsubseteq$** rules.
$\cdot \vdash e \backsim e :^c (U\,(A_1, A_2)) \to U\,(A_1', A_2')$.

□

Theorem 18 **(Types are preserved by embedding in RelInf)**

1. If $\Omega \vdash e \rightsquigarrow e^* : A$, then $\Omega \vdash e^* :^c A$ and $\Omega \vdash e : A$.
2. If $\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$, then $\Gamma \vdash e_1^* \backsim e_2^* :^c \tau$ and $\Gamma \vdash e_1 \backsim e_2 : \tau$.

Proof. By simultaneous induction on the given derivations.
Proof of Theorem 18.1:

Case $\dfrac{\vdash^{\mathsf{A}} A_1 \to A_2\ \texttt{wf} \qquad ; x : A_1, f : A_1 \to A_2, \Omega \vdash e \rightsquigarrow e^* : A_2}{\Omega \vdash \textbf{fix } f(x).e \rightsquigarrow \textbf{fix } f(x).e^* : A_1 \to A_2}\ \textbf{e-u-fix}$
By Theorem 18.1 on the premise, we get $x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2$. Then, by unary core rule **c-fix**,we conclude: $\dfrac{x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2}{\Omega \vdash \text{fix } f(x).e :^c A_1 \to A_2}\ \textbf{c-u-fix}$.

Proof of Theorem 18.2:

Case $\dfrac{|\Gamma|_1 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow e_1^* : A_1 \quad (\star) \qquad |\Gamma|_2 \vdash^{t_2}_{k_2} e_2 \rightsquigarrow e_2^* : A_2 \quad (\diamond)}{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{switch}\ e_1^* \backsim \texttt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}$
By Theorem 18.1 on $(\star)$, we get $\Delta; \Phi; \Omega \vdash e_1^* :^c A_1 \quad (\star\star)$.
By Theorem 18.1 on $(\diamond)$, we get $\Delta; \Phi; \Omega \vdash e_2^* :^c A_2 \quad (\diamond\diamond)$.
Then, we conclude as follows: $\dfrac{|\Gamma|_1 \vdash e_1 :^c A_1 \qquad |\Gamma|_2 \vdash e_2 :^c A_2}{\Gamma \vdash \texttt{switch}\ e_1 \backsim \texttt{switch}\ e_2 :^c U\,(A_1, A_2)}\ \textbf{c-switch}$.

□

Theorem 19 **(Completeness of embedding in RelInf)**

1. If $\Omega \vdash e : A$, then there exists an e^* such that $\Omega \vdash e \rightsquigarrow e^* : A$.
2. If $\Gamma \vdash e_1 \backsim e_2 : \tau$, then there exist e_1^*, e_2^* such that $\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$.

Proof. By simultaneous induction on the given typing derivations.
Proof of Theorem 19.1:

Case $\dfrac{\Omega \vdash^t_k e : A \quad (\star) \qquad \models^{\mathsf{A}} A \sqsubseteq A' \quad (\diamond)}{\Omega \vdash e : A'} \sqsubseteq$

By Theorem 19.1 on $(\star)$, we get $\exists\, e^*$ such that $\Omega \vdash e \rightsquigarrow e^* : A \quad (\star\star)$.

By **e-u-**$\sqsubseteq$ rule using $(\star\star)$, $(\diamond)$, we conclude as follows

$$\dfrac{\Omega \vdash e \rightsquigarrow e^* : A \qquad \models^{\mathsf{A}} A \sqsubseteq A'}{\Omega \vdash e \rightsquigarrow e^* : A'}\ \textbf{e-u-}\sqsubseteq$$

Proof of Theorem 19.2:

Case $\dfrac{|\Gamma|_1 \vdash e_1 : A_1 \quad (\star) \qquad |\Gamma|_2 \vdash e_2 : A_2 \quad (\diamond)}{\Gamma \vdash e_1 \backsim e_2 : U\,(A_1, A_2)}$ **r-switch**

By ??.1 on $(\star)$, we get $\exists\, e_1^*$ such that $|\Gamma|_1 \vdash e_1 \rightsquigarrow e_1{}^* : A_1 \quad (\star\star)$.

By ??.1 on $(\diamond)$, we get $\exists\, e_2^*$ such that $|\Gamma|_2 \vdash e_2 \rightsquigarrow e_2{}^* : A_2 \quad (\diamond\diamond)$.

By **e-switch** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\dfrac{|\Gamma|_1 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow e_1^* : A_1 \quad (\star\star) \qquad |\Gamma|_2 \vdash^{t_2}_{k_2} e_2 \rightsquigarrow e_2^* : A_2 \quad (\diamond\diamond)}{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{switch}\ e_1^* \backsim \texttt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}.$$

$\square$

Theorem 20 **(Soundness of algorithmic typechecking in RelInf)**

1. Assume that $\Omega \vdash e \downarrow A$, then, $\Omega \vdash |e| :^c A$.
2. Assume that $\Omega \vdash e \uparrow A$,then, $\Omega \vdash |e| :^c A$.
3. Assume that $\Gamma \vdash e \ominus e' \downarrow \tau$,then, $\Gamma \vdash |e| \backsim |e'| :^c \tau$.
4. Assume that $\Gamma \vdash e \ominus e' \uparrow \tau$,then, $\Gamma \vdash |e| \backsim |e'| :^c \tau$.

Proof. By simultaneous induction on the given algorithmic typing derivations.

Proof of Theorem 20.1:

Case $\dfrac{\Omega, x : A_1, f : A_1 \to A_2 \vdash e_1 \downarrow A_2}{\Omega \vdash \textbf{fix}\ f(x).e_1 \downarrow A_1 \to A_2}$ **alg-u-fix**

By IH1 on the premise, we get $\Omega, x : A_1, f : A_1 \to A_2 \vdash |e_1| :^c A_2 \quad (\star)$.

By the unary core rule **c-u-fix** and $(\star)$, We conclude $\Omega \vdash \text{fix}\ f(x).|e_1| :^c A_1 \to A_2$.

Proof of Theorem 20.2:

Case $\dfrac{\Omega \vdash e \downarrow A}{\Omega \vdash (e : A) \uparrow A}$ **alg-u-anno-**$\uparrow$

By IH1 on the premise, we get $\Omega, \vdash |e| :^c A \quad (\star)$.

Because $|(e : A)| = |e|$, We conclude $\Omega \vdash |(e : A)| :^c A$.

Proof of Theorem 20.3:

Case $\dfrac{|\Gamma|_1 \vdash e_1 \downarrow A_1 \qquad |\Gamma|_2 \vdash e_2 \downarrow A_2}{\Gamma \vdash \texttt{switch}\ e_1 \backsim \texttt{switch}\ e_2 \downarrow U\,(A_1, A_2)}$ **alg-r-switch↓**

By IH1 on the first premise, we get $|\Gamma|_1 \vdash |e_1| :^c A_1 \quad (\star)$.

By IH1 on the second premise, we get $|\Gamma|_2 \vdash |e_2| :^c A_2 \quad (\diamond)$.

By relational core rule **c-switch** and $(\star)$ and $(\diamond)$, We conclude $\Gamma \vdash \texttt{switch}\ |e_1| \backsim \texttt{switch}\ e_1 |e_2| :^c U\,(A_1, A_2)$.

Proof of Theorem 20.4:

Case $\dfrac{|\Gamma|_1 \vdash e_1 \uparrow A_1 \qquad |\Gamma|_2 \vdash e_2 \uparrow A_2}{\Gamma \vdash \texttt{switch}\ e_1 \backsim \texttt{switch}\ e_2 \uparrow U\,(A_1, A_2)}$ **alg-r-switch↑**

By IH2 on the first premise, we get $|\Gamma|_1 \vdash |e_1| :^c A_1 \quad (\star)$.

By IH2 on the second premise, we get $|\Gamma|_2 \vdash |e_2| :^c A_2 \quad (\diamond)$.

By relational core rule **c-switch** and $(\star)$ and $(\diamond)$, We conclude $\Gamma \vdash \texttt{switch}\ |e_1| \backsim \texttt{switch}\ e_1 |e_2| :^c U\,(A_1, A_2)$.

□

Theorem 21 **(Completeness of algorithmic typechecking in RelInf)**

1. Assume that $\Omega \vdash e :^c A$. Then, there exists e' such that $\Omega \vdash e' \downarrow A$ and $|e'| = e$.
2. Assume that $\Gamma \vdash e_1 \backsim e_2 :^c \tau$. Then, there exist e'_1, e'_2 such that $\Gamma \vdash e'_1 \ominus e'_2 \downarrow \tau$ and $|e'_1| = e_1$ and $|e'_2| = e_2$.

Proof. By simultaneous induction on the given Core typing derivations.

Proof of Theorem 21.1:

Case $\dfrac{\Omega(x) = A}{\Omega \vdash x :^c A}$ **c-u-var**

We can conclude as follows

$$\dfrac{\dfrac{\Omega(x) = \tau}{\Omega \vdash x \uparrow A}\ \textbf{alg-u-var} \qquad \models A \sqsubseteq A}{\Omega \vdash x \downarrow A}\ \textbf{alg-u-}\uparrow\downarrow$$

Case $\dfrac{x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2}{\Omega \vdash \textbf{fix}\ f(x).e :^c A_1 \to A_2}$ **c-u-fix**

By IH1 on the premise, we get exists e'' s.t $\Omega \vdash e'' \downarrow A_2 \quad (\star)$.

By algorithmic unary rule **alg-u-fix** and $(\star)$, we conclude where $e' = \text{fix}\ f(x).e''$ and $|e''| = e''$ $\Omega \vdash e' \downarrow A_1 \to A_2$. and $|e'| = \text{fix}\ f(x).|e''|$.

Proof of Theorem 21.2:

Case $\dfrac{\Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 \backsim e_2' :^c \tau_1}{\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' :^c \tau_2}$ **c-r-app**

By IH on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \to \tau_2$

b) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By IH on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

c) $\Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1$

d) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

Then, we can conclude as follows

1.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \to \tau_2}{\Gamma \vdash (\overline{e}_1 : \tau_1 \to \tau_2) \ominus \overline{e}_1' : \tau_1 \to \tau_2) \uparrow \tau_1 \to \tau_2}\ \textbf{alg-r-anno-}\uparrow \qquad \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1 \Rightarrow \Phi_2}{\Gamma \vdash E_1 \ominus E_2 \uparrow \tau_2}\ \textbf{c-r-app}}{\Gamma \vdash E_1 \ominus E_2 \downarrow \tau_2}\ \textbf{alg-r-}\uparrow\downarrow$$

where $E_1 = (\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2$ and $E_2 = (\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'$.

2. By using b) and e) .
3. Using c) and f), $|(\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2| = e_1\ e_2$ and $|(\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'| = e_1'\ e_2'$.

$\square$

4 RelRefU

4.1 Syntax of RelRefU

Unary types	A	$::=$	$\text{bool} \mid \text{int} \mid A_1 \to A_2 \mid \text{list}[n]\, A \mid \forall i{::}S.\, A \mid \exists i{::}S.\, A \mid C\ \&\ A \mid C \supset A$
Relational types	τ	$::=$	$\text{bool}_r \mid \text{int}_r \mid \tau_1 \to \tau_2 \mid U\,(A_1, A_2) \mid \text{list}[n]^\alpha\, \tau \mid \forall i{::}S.\, \tau \mid$ $\exists i{::}S.\, \tau \mid \square\, \tau \mid C\ \&\ \tau \mid C \supset \tau$
Expressions	e	$::=$	$x \mid n \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fix } f(x).e \mid e_1\ e_2 \mid \text{nil} \mid \text{cons}(e_1, e_2) \mid$ $\text{case } e \text{ of nil} \to e_1 \mid h :: tl \to e_2 \mid \Lambda.e \mid e[\,] \mid$ $\text{pack } e \mid \text{unpack } e_1 \text{ as } x \text{ in } e_2 \mid \text{let } x = e_1 \text{ in } e_2 \mid$ $\text{clet } e_1 \text{ as } x \text{ in } e_2 \mid \text{celim } e$
Value	v	$:=$	$\mathtt{n} \mid \text{true} \mid \text{false} \mid \text{fix } f(x).e \mid \text{nil} \mid \text{cons}(v_1, v_2) \mid \Lambda e \mid \text{pack } v$

Figure 38: Syntax of values and expressions in RelRefU

Unary types	A	$::=$	$\text{bool} \mid \text{int} \mid A_1 \to A_2 \mid \text{list}[n]\, A \mid \forall i{::}S.\, A \mid \exists i{::}S.\, A \mid C\ \&\ A \mid C \supset A$
Relational types	τ	$::=$	$\text{bool}_r \mid \text{int}_r \mid \tau_1 \to \tau_2 \mid U\,(A_1, A_2) \mid \text{list}[n]^\alpha\, \tau \mid \forall i{::}S.\, \tau \mid$ $\exists i{::}S.\, \tau \mid \square\, \tau \mid C\ \&\ \tau \mid C \supset \tau$
Expressions	e	$::=$	$x \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fix } f(x).e \mid e_1\ e_2 \mid \mathtt{switch}\ e \mid \mathtt{NC}\ e$ $\mid \mathtt{split}\ (e_1, e_2) \text{ with } C \mid \mathtt{contra}\ e \mid \mathtt{der}\ e \mid \Lambda i.e \mid e[I] \mid \text{pack } e \text{ with } I \mid$ $\text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \mid \text{cons}_{NC}(e_1, e_2) \mid \text{cons}_C(e_1, e_2) \mid \text{fix}_{NC}\ f(x).e$ $\mid \text{let } x = e_1 \text{ in } e_2 \mid \begin{pmatrix} \text{case } e \text{ of nil } \to e_1 \\ \mid h\ ::_{NC} tl\ \to e_2 \mid h ::_C tl\ \to e_3 \end{pmatrix}$
Value	v	$:=$	$\mathtt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid\ \mid \text{nil}$ $\mid \text{cons}_{NC}(v_1, v_2) \mid \text{cons}_C(v_1, v_2) \mid \Lambda i.e \mid \text{pack } v \text{ with } I$

Figure 39: Syntax of values and expressions in RelRefU Core

$$
\begin{array}{llll}
\text{Unary types} & A & ::= & \text{bool} \mid \text{int} \mid A_1 \to A_2 \mid \text{list}[n]\, A \mid \forall i{::}S.\, A \mid \exists i{::}S.\, A \mid C \,\&\, A \mid C \supset A \\
\text{Relational types} & \tau & ::= & \text{bool}_r \mid \text{int}_r \mid \tau_1 \to \tau_2 \mid U\,(A_1, A_2) \mid \text{list}[n]^{\alpha}\, \tau \mid \forall i{::}S.\, \tau \mid \\
 & & & \exists i{::}S.\, \tau \mid \Box\, \tau \mid C \,\&\, \tau \mid C \supset \tau \\
\text{Expressions} & e & ::= & x \mid \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fix } f(x).e \mid e_1\ e_2 \mid \texttt{switch}\ e \mid \texttt{NC}\ e \\
 & & & \mid \texttt{split}\ (e_1, e_2) \text{ with } C \mid \texttt{contra}\ e \mid \texttt{der}\ e \mid \Lambda i.e \mid e[I] \mid \text{pack } e \text{ with } I \mid \\
 & & & \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \mid \text{cons}_{NC}(e_1, e_2) \mid \text{cons}_{C}(e_1, e_2) \mid \text{fix}_{NC}\ f(x).e \\
 & & & \mid (e : \tau) \mid (e : A) \mid \text{let } x = e_1 \text{ in } e_2 \mid \begin{pmatrix} \text{case } e \text{ of nil } \to e_1 \\ \mid h\ ::_{NC} tl\ \to e_2 \mid h ::_{C} tl\ \to e_3 \end{pmatrix} \\
\text{Value} & v & := & \texttt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid\ \mid \text{nil} \\
 & & & \mid \text{cons}_{NC}(v_1, v_2) \mid \text{cons}_{C}(v_1, v_2) \mid \Lambda i.e \mid \text{pack } v \text{ with } I
\end{array}
$$

Figure 40: Syntax of values and expressions in BiRelRefU

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x : \tau}\ \textbf{r-var} \qquad \frac{\mathtt{b} \in \{\text{true}, \text{false}\}}{\Delta; \Phi_a; \Gamma \vdash \mathtt{b} \backsim \mathtt{b} : \text{bool}_r}\ \textbf{r-bool}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \text{bool}_r \qquad \Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \tau}{\Delta; \Phi_a; \Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 \backsim \text{if } e' \text{ then } e_1' \text{ else } e_2' : \tau}\ \textbf{r-if}$$

$$\frac{\Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 : \tau_1 \to \tau_2}\ \textbf{r-fix}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau_1 \to \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \tau_1}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' : \tau_2}\ \textbf{r-app}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e : \tau \\ \forall x \in \mathit{dom}(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\,\Gamma(x)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \backsim e : \square\,\tau}\ \textbf{r-nochange}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau'}\ \textbf{r-}\sqsubseteq$$

$$\frac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \to \tau_2\ \mathtt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e : \tau_2 \qquad \forall x \in \mathit{dom}(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\,\Gamma(x)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e : \square\,(\tau_1 \to \tau_2)}\ \textbf{r-fixNC}$$

$$\frac{\begin{array}{c} i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau \\ i \notin \text{FIV}(\Phi_a; \Gamma)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \Lambda e \backsim \Lambda e' : \forall i{::}S.\,\tau}\ \textbf{r-iLam} \qquad \frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \forall i{::}S.\,\tau \\ \Delta \vdash I : S\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \backsim e'[\,] : \tau\{I/i\}}\ \textbf{r-iApp}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \backsim \text{pack } e' : \exists i{::}S.\,\tau}\ \textbf{r-pack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \exists i{::}S.\,\tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } x \text{ in } e_2' : \tau_2}\ \textbf{r-unpack1}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : C \supset \tau}\ \textbf{r-c-impII} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : C \supset \tau \qquad \Delta; \Phi_a \vDash C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' : \tau}\ \textbf{r-c-implE}$$

$$\frac{|\Gamma|_1 \vdash e_1 : A_1 \qquad |\Gamma|_2 \vdash e_2 : A_2}{\Gamma \vdash e_1 \backsim e_2 : U\,(A_1, A_2)}\ \textbf{r-switch}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \backsim \text{let } x = e_1' \text{ in } e_2' : \tau_2}\ \textbf{r-let}$$

Figure 41: RelRefU relational typing rules(Part 1)

$$\frac{\Delta; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{nil} \backsim \text{nil} : \text{list}[0]^\alpha\, \tau} \ \textbf{r-nil}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') : \text{list}[n+1]^{\alpha+1}\, \tau} \ \textbf{r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \square\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' : \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \backsim \text{cons}(e_1', e_2') : \text{list}[n+1]^{\alpha}\, \tau} \ \textbf{r-cons2}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \text{list}[n]^\alpha\, \tau \\ \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \backsim e_1' : \tau' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : \square\, \tau, tl : \text{list}[i]^\alpha\, \tau, \Gamma \vdash e_2 \backsim e_2' : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h : \tau, tl : \text{list}[i]^\beta\, \tau, \Gamma \vdash e_2 \backsim e_2' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \ \text{case } e \text{ of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2 \backsim \ \text{case } e' \text{ of nil} \rightarrow e_1' \mid h :: tl \rightarrow e_2' : \tau'} \ \textbf{r-caseL}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : C \ \&\ \tau} \ \textbf{r-c-andI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : C \ \&\ \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' : \tau_2} \ \textbf{r-c-andE}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta; \Phi_a \wedge \neg C; \Gamma \vdash e_1 \backsim e_2 : \tau \qquad \Delta \vdash C \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau} \ \textbf{r-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau} \ \textbf{r-contra}$$

Figure 42: RelRefU relational typing rules (Part 2)

$$\frac{}{\Delta; \Phi_a \models \text{int}_r \sqsubseteq \square\, \text{int}_r}\ \textbf{int-}\square \qquad \frac{}{\Delta; \Phi_a \models \square\, U\, (\text{int}, \text{int}) \sqsubseteq \text{int}_r}\ \square\textbf{U-int}$$

$$\frac{}{\Delta; \Phi_a \models \text{bool}_r \sqsubseteq \square \text{bool}_r}\textbf{bool-}\square \qquad \frac{\Delta; \Phi_a \models \tau_1' \sqsubseteq \tau_1 \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2'}{\Delta; \Phi_a \models \tau_1 \to \tau_2 \sqsubseteq \tau_1' \to \tau_2'} \to \qquad \frac{}{\Delta; \Phi_a \models \tau \sqsubseteq \tau}\textbf{refl}$$

$$\frac{}{\Delta; \Phi_a \models \square(\tau_1 \to \tau_2) \sqsubseteq \square\tau_1 \to \square\tau_2}\square\textbf{diff} \qquad \frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i \notin \text{FIV}(\Phi_a)}{\Delta; \Phi_a \models \forall i{::}S.\, \tau \sqsubseteq \forall i{::}S.\, \tau'}\forall\textbf{diff}$$

$$\frac{}{\Delta; \Phi_a \models \square\forall i{::}S.\, \tau \sqsubseteq \forall i{::}S.\, \square\tau}\forall\square \qquad \frac{}{\Delta; \Phi_a \models U\, (\forall i{::}S.\, A, \forall i{::}S.\, A') \sqsubseteq \forall i{::}S.\, U\, (A, A')}\forall\ \textbf{U}$$

$$\frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \leq \alpha' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \tau \sqsubseteq \text{list}[n']^{\alpha'}\, \tau'}\ \textbf{l1}$$

$$\frac{\Delta; \Phi_a \models \alpha \doteq 0}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \tau \sqsubseteq \text{list}[n]^{\alpha}\, \square\, \tau}\ \textbf{l2} \qquad \frac{}{\Delta; \Phi_a \models \text{list}[n]^{\alpha}\, \square\, \tau \sqsubseteq \square\, (\text{list}[n]^{\alpha}\, \tau)}\ \textbf{l}\square$$

$$\frac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \tau}\ \textbf{T} \qquad \frac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \square\, \square\, \tau}\ \textbf{D} \qquad \frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2}{\Delta; \Phi_a \models \square\, \tau_1 \sqsubseteq \square\, \tau_2}\ \textbf{B-}\square$$

$$\frac{}{\models \tau \sqsubseteq U\, (|\tau|_1, |\tau|_2)}\ \textbf{W} \qquad \frac{\models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\models U\, (A_1, A_2) \sqsubseteq U\, (A_1', A_2')}\ \textbf{U}$$

$$\frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \sqsubseteq \exists i{::}S.\, \tau'}\ \exists \qquad \frac{}{\Delta; \Phi_a \models \exists i{::}S.\, \square\, \tau \sqsubseteq \square\, (\exists i{::}S.\, \tau)}\ \exists\square$$

$$\frac{\Delta; \Phi_a \wedge C \models C' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C \,\&\, \tau \sqsubseteq C' \,\&\, \tau'}\textbf{c-and} \qquad \frac{}{\Delta; \Phi_a \models C \,\&\, \square\, \tau \sqsubseteq \square\, (C \,\&\, \tau)}\ \textbf{c-and-}\square$$

$$\frac{\Delta; \Phi_a \wedge C' \models C \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C \supset \tau \sqsubseteq C' \supset \tau'}\textbf{c-impl} \qquad \frac{}{\Delta; \Phi_a \models \square\, (C \supset \tau) \sqsubseteq C \supset \square\, \tau}\ \textbf{c-impl-}\square$$

$$\frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_3}{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_3}\textbf{trans}$$

Figure 43: RelRefU relational subtyping rules

$$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A_1' \to A_2'} \textbf{ u-}{\to}\textbf{ exec}$$

$$\frac{i :: S, \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models^{\mathsf{A}} \forall i{::}S.\, A \sqsubseteq \forall i{::}S.\, A'} \textbf{u-}\forall\textbf{exec} \qquad \frac{i :: S, \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models^{\mathsf{A}} \exists i{::}S.\, A \sqsubseteq \exists i{::}S.\, A'} \textbf{ u-}\exists$$

$$\frac{\Delta; \Phi_a \wedge C \models C' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} C \,\&\, A \sqsubseteq C' \,\&\, A'} \textbf{u-c-and} \qquad \frac{\Delta; \Phi_a \wedge C' \models C \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} C \supset A \sqsubseteq C' \supset A'} \textbf{u-c-impl}$$

$$\frac{}{\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A} \textbf{u-refl} \qquad \frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_3}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_3} \textbf{u-tran}$$

$$\frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} \mathrm{list}[n]\, A \sqsubseteq \mathrm{list}[n']\, A'} \textbf{ u-l}$$

Figure 44: RelRefU unary subtyping rules

$\boxed{\Delta; \Phi_a \models \tau_1 \equiv \tau_2}$ checks whether τ_1 is equivalent to τ_2

$$\frac{}{\Delta; \Phi_a \models \mathrm{bool}_r \equiv \mathrm{bool}_r} \textbf{ eq-bool} \qquad \frac{}{\Delta; \Phi_a \models \mathrm{int}_r \equiv \mathrm{int}_r} \textbf{ eq-int}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2'}{\Delta; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2'} \textbf{ eq-}\forall \qquad \frac{\begin{array}{c}\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \doteq \alpha' \\ \Delta; \Phi_a \models \tau \equiv \tau'\end{array}}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \equiv \mathrm{list}[n']^{\alpha'}\, \tau'} \textbf{ eq-list}$$

$$\frac{i, \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau'} \textbf{ eq-}\forall \qquad \frac{i, \Delta; \Phi_a \models \tau \equiv \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \equiv \exists i{::}S.\, \tau'} \textbf{ eq-}\exists$$

$$\frac{\Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \square\, \tau \equiv \square\, \tau'} \textbf{eq-B-}\square$$

$$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2}{\Delta; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1', A_2')} \textbf{eq-U}$$

$$\frac{\Delta; C \wedge \Phi_a \models C' \qquad \Delta; C' \wedge \Phi_a \models C \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \supset \tau \equiv C' \supset \tau'} \textbf{eq-c-impl}$$

$$\frac{\Delta; C' \wedge \Phi_a \models C \qquad \Delta; C \wedge \Phi_a \models C' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \,\&\, \tau \equiv C' \,\&\, \tau'} \textbf{eq-c-prod}$$

Figure 45: RelRefU Core binary type equivalence rules

$$\frac{}{\Delta; \Phi_a; \Omega \vdash \mathtt{n} :^c \text{int}}\ \textbf{c-const} \qquad \frac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash x :^c A}\ \textbf{c-var}$$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \to A_2\ \mathtt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2}{\Delta; \Phi_a; \Omega \vdash \text{fix } f(x).e :^c A_1 \to A_2}\ \textbf{c-fix}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 :^c A_1 \to A_2 \qquad \Delta; \Phi_a; \Omega \vdash e_2 :^c A_1}{\Delta; \Phi_a; \Omega \vdash e_1\ e_2 :^c A_2}\ \textbf{c-app}$$

$$\frac{i :: S, \Delta; \Phi_a; \Omega \vdash e :^c A \qquad i \notin \text{FIV}(\Phi_a; \Omega)}{\Delta; \Phi_a; \Omega \vdash \Lambda i.e :^c \forall i{::}S.\, A}\ \textbf{c-iLam}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e :^c \forall i{::}S.\, A \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Omega \vdash e[I] :^c A\{I/i\}}\ \textbf{c-iApp} \qquad \frac{\Delta; \Phi_a; \Omega \vdash e :^c A\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Omega \vdash \text{pack } e \text{ with } I :^c \exists i{::}S.\, A}\ \textbf{c-pack}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 :^c \exists i{::}S.\, A_1 \qquad i :: S, \Delta; \Phi_a; x : A_1, \Omega \vdash e_2 :^c A_2 \qquad i \notin FV(\Phi_a; \Gamma, A_2, k_2, t_2)}{\Delta; \Phi_a; \Omega \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 :^c A_2}\ \textbf{c-unpack}$$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A\ \mathtt{wf}}{\Delta; \Phi_a; \Omega \vdash \text{nil} :^c \text{list}[0]\, A}\ \textbf{c-nil} \qquad \frac{\Delta; \Phi_a; \Omega \vdash e_1 :^c A \qquad \Delta; \Phi_a; \Omega \vdash e_2 :^c \text{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash \text{cons}_C(e_1, e_2) :^c \text{list}[n+1]\, A}\ \textbf{c-cons}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 :^c A_1 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash e_2 :^c A_2}{\Delta; \Phi_a; \Omega \vdash \text{let } x = e_1 \text{ in } e_2 :^c A_2}\ \textbf{c-let}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash e :^c \text{list}[n]\, A \\ \Delta; \Phi_a \wedge n = 0; \Omega \vdash e_1 :^c A' \qquad i, \Delta; \Phi_a \wedge n = i + 1; h : A, tl : \text{list}[i]\, A, \Omega \vdash e_2 :^c A'\end{array}}{\Delta; \Phi_a; \Omega \vdash\ \text{case } e \text{ of nil} \to e_1 \mid h ::_{NC} tl \to e_2 \mid h ::_C tl \to e_3 :^c A'}\ \textbf{c-caseL}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Omega \vdash e :^c A}{\Delta; \Phi_a; \Omega \vdash e :^c C \,\&\, A}\ \textbf{c-candI}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 :^c C \,\&\, A_1 \qquad \Delta; \Phi_a \wedge C; x : A_1, \Omega \vdash e_2 :^c A_2}{\Delta; \Phi_a; \Omega \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 :^c A_2}\ \textbf{c-candE}$$

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash e :^c A}{\Delta; \Phi_a; \Omega \vdash e :^c C \supset A}\ \textbf{c-cimpI} \qquad \frac{\Delta; \Phi_a; \Omega \vdash e :^c C \supset A \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Omega \vdash \text{celim } e :^c A}\ \textbf{c-cimplE}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e :^c A \qquad \Delta; \Phi_a \models A \sqsubseteq A'}{\Delta; \Phi_a; \Omega \vdash e :^c A'}\ \textbf{c-}\sqsubseteq_{\text{exec}}$$

Figure 46: RelRefU Core unary typing rules

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x :^c \tau} \textbf{ c-r-var}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 :^c \tau_1 \to \tau_2} \textbf{ c-r-fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \square(\tau_1 \to \tau_2), \square\Gamma \vdash e \backsim e :^c \tau_2}{\Delta; \Phi_a; \square\Gamma \vdash \text{fix}_{NC} \ f(x).e \backsim \text{fix}_{NC} \ f(x).e :^c \square(\tau_1 \to \tau_2)} \textbf{ c-r-fixNC}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \text{list}[n]^\alpha \, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_1, e_2) \backsim \text{cons}_C(e_1', e_2') :^c \text{list}[n+1]^{\alpha+1} \, \tau} \textbf{ c-r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \square\tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \text{list}[n]^\alpha \, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_{NC}(e_1, e_2) \backsim \text{cons}_{NC}(e_1', e_2') :^c \text{list}[n+1]^{\alpha} \, \tau} \textbf{ c-r-cons2}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \text{list}[n]^\alpha \, \tau \\ \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \backsim e_1' :^c \tau' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : \square\tau, tl : \text{list}[i]^\alpha \, \tau, \Gamma \vdash e_2 \backsim e_2' :^c \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h' : \tau, tl : \text{list}[i]^\beta \, \tau, \Gamma \vdash e_3 \backsim e_3' :^c \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil } \to e_1 \\ \mid h \ ::_N tl \to e_2 \\ \mid h' \ ::_C tl' \to e_3\end{array} \backsim \begin{array}{l}\text{case } e \text{ of nil } \to e_1' \\ \mid h \ ::_N tl \to e_2' \\ \mid h' \ ::_C tl' \to e_3'\end{array} :^c \tau'} \textbf{ c-r-caseL}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \square\tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{der } e_1 \backsim \texttt{der } e_2 :^c \tau} \textbf{ c-der} \qquad \frac{\Delta; \Phi_a; \square\Gamma \vdash e \backsim e :^c \tau}{\Delta; \Phi_a; \square\Gamma, \Gamma' \vdash \texttt{NC } e \backsim \texttt{NC } e :^c \square\tau} \textbf{ c-nochange}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau'} \textbf{ c-r}-\sqsubseteq$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e_1 \backsim e_2 :^c \tau \qquad \Delta; \Phi_a \wedge \neg C; \Gamma \vdash e_1' \backsim e_2' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{split } (e_1, e_1') \text{ with } C \backsim \texttt{split } (e_2, e_2') \text{ with } C :^c \tau} \textbf{ c-r-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \texttt{contra } e_1 \backsim \texttt{contra } e_2 :^c \tau} \textbf{ c-r-contra}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \forall i{::}S.\, \tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[I] \backsim e'[I] :^c \tau\{I/i\}} \textbf{ c-r-iApp}$$

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad i \notin \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda i.e \backsim \Lambda i.e' :^c \forall i{::}S.\, \tau} \textbf{ c-r-iLam}$$

Figure 47: RelRefU Core binary typing rules (Part 1)

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \supset \tau} \ \textbf{c-r-cimpI} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' :^c \tau} \ \textbf{c-r-cimplE}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c C \ \& \ \tau} \ \textbf{c-r-candI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c C \ \& \ \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' :^c \tau_2} \ \textbf{c-r-candE}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \rightarrow \tau_2 \\ \Delta; \Phi_a; \Gamma \vdash e_2 \backsim e_2' :^c \tau_1\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ e_2 \backsim e_1' \ e_2' :^c \tau_2} \ \textbf{c-r-app}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \backsim \text{let } x = e_1' \text{ in } e_2' :^c \tau_2} \ \textbf{c-r-let}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \text{ with } I \backsim \text{pack } e' \text{ with } I :^c \exists i{::}S.\, \tau} \ \textbf{c-r-pack}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' :^c \exists i{::}S.\, \tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' :^c \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' :^c \tau_2} \ \textbf{c-r-unpack1}$$

$$\frac{|\Gamma|_1 \vdash e_1 :^c A_1 \qquad |\Gamma|_2 \vdash e_2 :^c A_2}{\Gamma \vdash \texttt{switch}\, e_1 \backsim \texttt{switch}\, e_2 :^c U\,(A_1, A_2)} \ \textbf{c-switch}$$

Figure 48: RelRefU Core binary typing rules (Part 2)

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \to A_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \to A_2, \Omega \vdash e \rightsquigarrow e^* : A_2}{\Delta; \Phi_a; \Omega \vdash \text{fix } f(x).e \rightsquigarrow \text{fix } f(x).e^* : A_1 \to A_2} \textbf{ e-u-fix}$$

$$\frac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash x \rightsquigarrow x : A} \textbf{ e-u-var}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : A_1 \to A_2 \qquad \Delta; \Phi_a; \Omega \vdash e_2 \rightsquigarrow {e_2}^* : A_1}{\Delta; \Phi_a; \Omega \vdash e_1 \ e_2 \rightsquigarrow e_1^* \ e_2^* : A_2} \textbf{ e-u-app}$$

$$\frac{\Delta; \Phi_a \vdash A \ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash \text{nil} \rightsquigarrow \text{nil} : \text{list}[0] \, A} \textbf{ e-u-nil}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : A \qquad \Delta; \Phi_a; \Omega \vdash e_2 \rightsquigarrow {e_2}^* : \text{list}[n] \, A}{\Delta; \Phi_a; \Omega \vdash \text{cons}(e_1, e_2) \rightsquigarrow \text{cons}_C(e_1^*, e_2^*) : \text{list}[n+1] \, A} \textbf{ e-u-cons}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : \text{list}[n] \, A \\ \Delta; \Phi_a \wedge n = 0; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : A' \qquad i, \Delta; \Phi_a \wedge n = i + 1; h : A, tl : \text{list}[i] \, A, \Omega \vdash e_2 \rightsquigarrow {e_2}^* : A'\end{array}}{\Delta; \Phi_a; \Omega \vdash \begin{array}{l}\text{case } e \text{ of nil} \to e_1 \\ \quad \mid h :: tl \to e_2\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^* \text{ of nil } \to e_1^* \\ \mid h ::_{NC} tl \to e_2^* \\ \mid h ::_C tl \to e_2^*\end{array} : A'} \textbf{ e-u-caseL}$$

$$\frac{i :: S, \Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A \qquad i \notin \text{FIV}(\Phi_a; \Omega)}{\Delta; \Phi_a; \Omega \vdash \Lambda.e \rightsquigarrow \Lambda i.e^* : \forall i{::}S. \, A} \textbf{ e-u-iLam}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : \forall i{::}S. \, A \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Omega \vdash e[\,] \rightsquigarrow e^*[I] : A\{I/i\}} \textbf{ e-u-iApp}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Omega \vdash \text{pack } e \rightsquigarrow \text{pack } e^* \text{ with } I : \exists i{::}S. \, A} \textbf{ e-u-pack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : \exists i{::}S. \, A_1 \\ i :: S, \Delta; \Phi_a; x : A_1, \Omega \vdash e_2 \rightsquigarrow {e_2}^* : A_2 \qquad i \notin FV(\Phi_a; \Omega, A_2, k_2, t_2)\end{array}}{\Delta; \Phi_a; \Omega \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \rightsquigarrow \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 : \text{unpack } e_1^* \text{ as } (x, i) \text{ in } e_2^* \ A_2} \textbf{ e-u-unpack}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : A_1 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash e_2 \rightsquigarrow {e_2}^* : A_2}{\Delta; \Phi_a; \Omega \vdash \text{let } x = e_1 \text{ in } e_2 \rightsquigarrow \text{let } x = e_1^* \text{ in } e_2^* : A_2} \textbf{ e-u-let}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Omega \vdash e \rightsquigarrow e^* : A}{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : C \ \& \ A} \textbf{ e-u-andI}$$

$$\frac{\Delta; \Phi_a; \Omega \vdash e_1 \rightsquigarrow {e_1}^* : C \ \& \ A_1 \qquad \Delta; \Phi_a \wedge C; x : A_1, \Omega \vdash e_2 \rightsquigarrow {e_2}^* : A_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \rightsquigarrow \text{clet } e_1^* \text{ as } x \text{ in } e_2^* : A_2} \textbf{ e-u-c-andE}$$

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash e \rightsquigarrow e^* : A}{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : C \supset A} \textbf{ e-u-c-impI} \qquad \frac{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : C \supset A \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Omega \vdash \text{celim } e \rightsquigarrow \text{celim } e^* : A} \textbf{ e-u-c-implE}$$

Figure 49: RelRefU unary embedding typing rules

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \backsim x \rightsquigarrow x \backsim x : \tau} \textbf{ e-r-var}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \to \tau_2, \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \backsim \text{fix } f(x).e_2 \rightsquigarrow \text{fix } f(x).{e_1}^* \backsim \text{fix } f(x).{e_2}^* : \tau_1 \to \tau_2} \textbf{ e-r-fix}$$

$$\frac{\begin{array}{c} \Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2 \qquad \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \square\,\Gamma(x_i)} \\ \forall x_i \in dom(\Gamma), \quad \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\,\Gamma(x) \qquad e^{**} = \text{let } \overline{y_i = e_i \ x_i} \text{ in fix}_{NC} \ f(x).e^*\overline{[\texttt{der} \ y_i/x_i]} \end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e \rightsquigarrow e^{**} \backsim e^{**} : \square\,(\tau_1 \to \tau_2)} \textbf{ e-r-fixNC}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \supset \tau} \textbf{ e-r-c-impI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \backsim \text{celim } e' \rightsquigarrow \text{celim } e^* \backsim \text{celim } {e'}^* : \tau} \textbf{ e-r-c-implE}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : \tau}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim {e'}^* : C \,\&\, \tau} \textbf{ e-r-andI}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : C \,\&\, \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \backsim \text{clet } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow \text{clet } e_1^* \text{ as } x \text{ in } e_2^* \backsim \text{clet } e_1'^* \text{ as } x \text{ in } e_2'^* : \tau_2} \textbf{ e-r-c-andE}$$

Figure 50: RelRefU relational embedding rules (Part 1)

$$\frac{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \tau \qquad \Delta;\Phi_a;\Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \mathrm{list}[n]^\alpha\,\tau}{\Delta;\Phi_a;\Gamma \vdash \mathrm{cons}(e_1,e_2) \backsim \mathrm{cons}(e_1',e_2') \rightsquigarrow \mathrm{cons}_C(e_1^*,e_2^*) \backsim \mathrm{cons}_C(e_1'^*,e_2'^*) : \mathrm{list}[n+1]^{\alpha+1}\,\tau}\ \textbf{e-r-cons1}$$

$$\frac{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \square\,\tau \qquad \Delta;\Phi_a;\Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \mathrm{list}[n]^\alpha\,\tau}{\Delta;\Phi_a;\Gamma \vdash \mathrm{cons}(e_1,e_2) \backsim \mathrm{cons}(e_1',e_2') \rightsquigarrow \mathrm{cons}_{NC}(e_1',e_2') \backsim \mathrm{cons}_{NC}(e_1^*,e_2^*) : \mathrm{list}[n+1]^{\alpha}\,\tau}\ \textbf{e-r-cons2}$$

$$\frac{\Delta;\Phi_a;\Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau \qquad \forall x_i \in dom(\Gamma),\quad e_i = \mathtt{coerce}_{\Gamma(x_i),\square\,\Gamma(x_i)}}{\Delta;\Phi_a;\Gamma,\Gamma' \vdash e \backsim e \rightsquigarrow \mathrm{let}\ \overline{y_i = e_i\ x_i}\ \mathrm{in}\ \mathtt{NC}\ e^*[\overline{y_i/x_i}] \backsim \mathrm{let}\ \overline{y_i = e_i\ x_i}\ \mathrm{in}\ \mathtt{NC}\ e^*[\overline{y_i/x_i}] : \square\,\tau}\ \textbf{e-nochange}$$

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \mathrm{list}[n]^\alpha\,\tau \qquad \Delta;\Phi_a \wedge n=0;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \tau' \\ i,\Delta;\Phi_a \wedge n=i+1; h:\square\,\tau, tl:\mathrm{list}[i]^\alpha\,\tau,\Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau' \\ i,\beta,\Delta;\Phi_a \wedge n=i+1 \wedge \alpha=\beta+1; h:\tau, tl:\mathrm{list}[i]^\beta\,\tau,\Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^{**} \backsim e_2'^{**} : \tau'\end{array}}{\Delta;\Phi_a;\Gamma \vdash \begin{array}{l}\mathrm{case}\ e\ \mathrm{of}\ \mathrm{nil} \to e_1 \\ \quad\mid h :: tl \to e_2\end{array} \backsim \begin{array}{l}\mathrm{case}\ e\ \mathrm{of}\ \mathrm{nil} \to e_1' \\ \quad\mid h :: tl \to e_2'\end{array} \rightsquigarrow \begin{array}{l}\mathrm{case}\ e^*\ \mathrm{of}\ \mathrm{nil} \to e_1^* \\ \mid h ::_{NC} tl \to e_2^* \\ \mid h ::_C tl \to e_2^{**}\end{array} \backsim \begin{array}{l}\mathrm{case}\ e'^*\ \mathrm{of}\ \mathrm{nil} \to e_1'^* \\ \mid h ::_{NC} tl \to e_2'^* \\ \mid h ::_C tl \to e_2'^{**}\end{array} : \tau'}\ \textbf{e-r-caseL}$$

$$\frac{i :: S,\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau \qquad i \notin \mathrm{FIV}(\Phi_a;\Gamma)}{\Delta;\Phi_a;\Gamma \vdash \Lambda.e \backsim \Lambda.e' \rightsquigarrow \Lambda i.e^* \backsim \Lambda i.e'^* : \forall i{::}S.\,\tau}\ \textbf{e-r-iLam}$$

$$\frac{\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \forall i{::}S.\,\tau \qquad \Delta \vdash I : S}{\Delta;\Phi_a;\Gamma \vdash e[\,] \backsim e'[\,] \rightsquigarrow e^*[I] \backsim e'^*[I] : \tau\{I/i\}}\ \textbf{e-r-iApp}$$

$$\frac{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta;\Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \mathtt{coerce}_{\tau,\tau'}}{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\boldsymbol{\sqsubseteq}$$

$$\frac{\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta;\Phi_a;\Gamma \vdash \mathrm{pack}\ e \backsim \mathrm{pack}\ e' \rightsquigarrow \mathrm{pack}\ e^*\ \mathrm{with}\ I \backsim \mathrm{pack}\ e'^*\ \mathrm{with}\ I : \exists i{::}S.\,\tau}\ \textbf{e-r-pack}$$

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \exists i{::}S.\,\tau_1 \qquad i :: S,\Delta;\Phi_a; x:\tau_1,\Gamma \vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2 \\ i \notin FV(\Phi_a;\Gamma,\tau_2,t_2) \qquad e_1^{**} = \mathrm{unpack}\ e_1^*\ \mathrm{as}\ (x,i)\ \mathrm{in}\ e_2^* \qquad e_2^{**} = \mathrm{unpack}\ e_1'^*\ \mathrm{as}\ (x,i)\ \mathrm{in}\ e_2'^*\end{array}}{\Delta;\Phi_a;\Gamma \vdash \mathrm{unpack}\ e_1\ \mathrm{as}\ x\ \mathrm{in}\ e_2 \backsim \mathrm{unpack}\ e_1'\ \mathrm{as}\ x\ \mathrm{in}\ e_2' \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau_2}\ \textbf{e-r-unpack}$$

$$\frac{\Delta; C \wedge \Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta;\neg C \wedge \Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau \qquad \Delta \vdash C\ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \mathtt{split}\ (e_1^*, e_1^{**})\ \mathrm{with}\ C \backsim \mathtt{split}\ (e_2^*, e_2^{**})\ \mathrm{with}\ C : \tau}\ \textbf{e-r-split}$$

$$\frac{\Delta;\Phi_a \models \bot \qquad \Delta;\Phi_a \vdash \Gamma\ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \mathtt{contra}\ e_1 \backsim \mathtt{contra}\ e_2 : \tau}\ \textbf{e-r-contra}$$

$$\frac{|\Gamma|_1 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow e_1^* : A_1 \qquad |\Gamma|_2 \vdash^{t_2}_{k_2} e_2 \rightsquigarrow e_2^* : A_2}{\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \mathtt{switch}\ e_1^* \backsim \mathtt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}$$

Figure 51: RelRefU relational embedding rules (Part 2)

$$\frac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A_1' \to A_2' \Rightarrow \Phi_1 \wedge \Phi_2} \ \textbf{alg-u-fun}$$

$$\frac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi_1 \qquad \Phi = n \doteq n' \wedge \Phi}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} \mathrm{list}[n]\, A \sqsubseteq \mathrm{list}[n']\, A' \Rightarrow \Phi} \ \textbf{alg-u-list}$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi_1}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} \forall i{::}S.\, A \sqsubseteq \forall i{::}S.\, A' \Rightarrow \forall i :: S.\Phi_1 \wedge \Phi_2} \textbf{alg-u-}\forall$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi \qquad i \notin FV(\Phi_a)}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} \exists i{::}S.\, A \sqsubseteq \exists i{::}S.\, A' \Rightarrow \forall i :: S.\Phi} \ \textbf{alg-u-}\exists$$

$$\frac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} C \supset A \sqsubseteq C' \supset A' \Rightarrow (C' \to C) \wedge \Phi} \textbf{alg-u-c-impl}$$

$$\frac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} C \ \&\ A \sqsubseteq C' \ \&\ A' \Rightarrow (C \to C') \wedge \Phi} \textbf{alg-u-c-prod}$$

Figure 52: RelRefU unary algortihmic subtyping rules

$$\frac{}{\Delta; \psi_a; \Phi_a \models \mathrm{int}_r \equiv \mathrm{int}_r \Rightarrow \top}\ \textbf{alg-r-int}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a \models \tau_1 \to \tau_2 \equiv \tau_1' \to \tau_2' \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-fun}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \equiv \mathrm{list}[n']^{\alpha'}\, \tau' \Rightarrow \Phi \wedge n \doteq n' \wedge \alpha \doteq \alpha'}\ \textbf{alg-r-list}$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \forall i{::}S.\, \tau \equiv \forall i{::}S.\, \tau' \Rightarrow \forall i :: S.\Phi}\ \forall$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad i \notin FV(\Phi_a)}{\Delta; \psi_a; \Phi_a \models \exists i{::}S.\, \tau \equiv \exists i{::}S.\, \tau' \Rightarrow \forall i :: S.\Phi}\ \textbf{alg-r-}\exists \qquad \frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \square\, \tau_1 \equiv \square\, \tau_2 \Rightarrow \Phi}\ \textbf{B-}\square$$

$$\frac{\begin{array}{ll} \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \Rightarrow \Phi_1 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1' \\ \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2 \Rightarrow \Phi_2' \end{array}}{\Delta; \psi_a; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1', A_2') \Rightarrow \Phi_1 \wedge \Phi_1' \wedge \Phi_2 \wedge \Phi_2'}\ \textbf{U}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models C \supset \tau \equiv C' \supset \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\ \textbf{c-impl}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models C \,\&\, \tau \equiv C' \,\&\, \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\ \textbf{c-prod}$$

Figure 53: RelRefU binary algorithmic equivalence rules

$$\frac{\Omega(x) = A}{\Delta; \psi; \Phi_a; \Omega \vdash x \uparrow A \Rightarrow \top}\ \textbf{alg-u-var-}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : A_1 \to A_2, x : A_1, \Omega \vdash e \downarrow A_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{fix } f(x).e \downarrow A_1 \to A_2 \Rightarrow \Phi}\ \textbf{alg-u-fix-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow A_1 \to A_2 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a; \Omega \vdash e_2 \downarrow A_1 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Omega \vdash e_1\ e_2 \uparrow A_2 \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-u-app-}\uparrow$$

$$\frac{i :: S, \Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \Lambda i.e \downarrow \forall i{::}S.\ A \Rightarrow \forall i :: S.\Phi}\ \textbf{alg-u-iLam-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow \forall i{::}S.\ A' \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash e\,[I] \uparrow A'\{I/i\} \Rightarrow \Phi}\ \textbf{alg-u-iApp-}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A\{I/i\} \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{pack } e \text{ with } I \downarrow \exists i{::}S.\ A \Rightarrow \Phi}\ \textbf{alg-u-pack-}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow \exists i{::}S.\ A_1 \Rightarrow \Phi_1 \\ i :: S, \Delta; \psi_a; \Phi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Omega, A_2) \qquad \Phi = \Phi_1 \wedge \forall i :: S.\Phi_2\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \downarrow A_2 \Rightarrow \Phi}\ \textbf{alg-u-unpack-}\downarrow$$

$$\frac{\begin{array}{c} i \in \text{fresh}(\mathbb{N}) \\ \Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \downarrow A \Rightarrow \Phi_1 \qquad \Delta; i, \psi_a; \Phi_a; \Omega \vdash e_2 \downarrow \text{list}[i]\ A \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1)\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{cons}_C(e_1, e_2) \downarrow \text{list}[n]\ A \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'}\ \textbf{alg-u-cons-}\downarrow$$

$$\frac{\Delta, \psi_a; \Phi \vdash^{\mathsf{A}} A\ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{nil} \downarrow \text{list}[n]\ A \Rightarrow n \doteq 0}\ \textbf{alg-u-nil-}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow \text{list}[n]\ A \Rightarrow \Phi_1 \qquad \Delta; \psi_a; n \doteq 0 \wedge \Phi_a; \Omega \vdash e_1 \downarrow A' \Rightarrow \Phi_2 \\ i \in \text{fresh}(\mathbb{N}) \qquad i :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \Phi_a; h : A, tl : \text{list}[i]\ A, \Omega \vdash e_2 \downarrow A' \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_2) \wedge (\forall i :: \mathbb{N}.(n \doteq i+1) \to \Phi_3)\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \begin{array}{l}\text{case } e \text{ of nil } \to e_1 \\ \mid h ::_{NC} tl \to e_2 \\ \mid h ::_C tl \to e_3\end{array} \downarrow A' \Rightarrow (\Phi_1 \wedge \Phi_{body})}\ \textbf{alg-u-caseL-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow A_1 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{let } x = e_1 \text{ in } e_2 \downarrow A_2 \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2}\ \textbf{alg-u-let-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash^{\mathsf{A}} A\ \texttt{wf} \qquad \text{FIV}(A, k, t) \in \Delta}{\Delta; \psi_a; \Phi_a; \Omega \vdash (e : A, k, t) \uparrow A \Rightarrow \Phi}\ \textbf{alg-u-anno-}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-}\uparrow\downarrow$$

Figure 54: RelRefU unary algorithmic typing rules (Part 1)

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Omega \vdash e \downarrow A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow k \Rightarrow C \wedge (C \to \Phi)} \ \textbf{alg-u-c-andI}\downarrow$$

$$\frac{\Delta; \Phi \wedge C; \Omega \vdash e \downarrow A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow C \supset A \Rightarrow C \to \Phi} \ \textbf{alg-u-c-impI}\ \downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow C \supset A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{celim } e \uparrow A \Rightarrow C \wedge \Phi} \ \textbf{alg-u-c-implE}\uparrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow C \mathbin{\&} A_1 \Rightarrow \Phi_1 \qquad k_2, t_2 \in \text{fresh}(\mathbb{R}) \\ \Delta; k_2, t_2, \psi, \psi_a; \Phi \wedge C; x : A_1, \Omega \vdash e_2 \downarrow A_2 \Rightarrow \Phi_2 \qquad \Phi_2' = C \to \Phi_2 \wedge k \doteq (k_1 + k_2) \wedge (t_1 + t_2) \doteq t\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \downarrow A_2 \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2')} \ \textbf{alg-u-c-andE}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; C \wedge \Phi_a; \Omega \vdash e_1 \downarrow A \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Omega \vdash e_2 \downarrow A \Rightarrow \Phi_2 \qquad \Delta \vdash C \ \texttt{wf}\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{split}\ (e_1, e_2) \text{ with } C \downarrow A \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2} \ \textbf{alg-u-split}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a \models \bot}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{contra}\ e \downarrow A \Rightarrow \top} \ \textbf{alg-u-contra}\downarrow$$

Figure 55: RelRefU unary algorithmic typing rules (Part 2)

$$\frac{\Delta; \psi_a; \Phi_a; \square\Gamma \vdash e \ominus e \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\Gamma \vdash \texttt{NC}\ e \ominus \texttt{NC}\ e \downarrow \square\tau \Rightarrow \Phi} \textbf{ alg-r-nochange-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \square\tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \ominus \texttt{der}\ e_2 \uparrow \tau \Rightarrow \Phi} \textbf{ alg-r-der-}\uparrow \qquad \frac{\Gamma(x) = \tau}{\Delta; \psi_a; \Phi_a; \Gamma \vdash x \ominus x \uparrow \tau \Rightarrow \top} \textbf{ alg-r-var-}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \tau_1 \to \tau_2, x : \tau_1, \Gamma \vdash e \ominus e' \downarrow \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{fix } f(x).e \ominus \text{fix } f(x).e' \downarrow \tau_1 \to \tau_2 \Rightarrow \Phi} \textbf{ alg-r-fix-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \square(\tau_1 \to \tau_2), x : \tau_1, \square\Gamma \vdash e \ominus e' \downarrow \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\Gamma \vdash \text{fix}_{NC}\ f(x).e \ominus \text{fix}_{NC}\ f(x).e \downarrow \square(\tau_1 \to \tau_2) \Rightarrow \Phi} \textbf{ alg-r-fix-}\downarrow\square$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \tau_1 \to \tau_2 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_1 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \uparrow \tau_2 \Rightarrow \Phi_1 \wedge \Phi_2} \textbf{ alg-r-app-}\uparrow$$

$$\frac{\Delta, \psi_a; \Phi \vdash \tau\ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{nil} \ominus \text{nil} \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow n \doteq 0} \textbf{ alg-r-nil-}\downarrow$$

$$\frac{\begin{array}{c} i, \beta \in \text{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \Rightarrow \Phi_1 \\ \Delta; i, \beta, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^\beta\,\tau \Rightarrow \Phi_2 \qquad \Phi_2' = n \doteq (i+1) \wedge \exists\beta :: \mathbb{N}.\Phi_2 \wedge \alpha \doteq \beta + 1 \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{cons}_C(e_1, e_2) \ominus \text{cons}_C(e_1', e_2') \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'} \textbf{ alg-r-consC-}\downarrow$$

$$\frac{\begin{array}{c} i \in \text{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \square\tau \Rightarrow \Phi_1 \\ \Delta; i, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^\alpha\,\tau \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{cons}_{NC}(e_1, e_2) \ominus \text{cons}_{NC}(e_1', e_2') \downarrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'} \textbf{ alg-r-consNC-}\downarrow$$

$$\frac{\begin{array}{c} \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \text{list}[n]^\alpha\,\tau \Rightarrow \Phi_e \qquad \Delta; \psi_a; n \doteq 0 \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau' \Rightarrow \Phi_1 \\ i :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \Phi_a; h : \square\tau, tl : \text{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau' \Rightarrow \Phi_2 \\ i :: \mathbb{N}, \beta :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \alpha \doteq \beta + 1 \wedge \Phi_a; h : \tau, tl : \text{list}[i]^\beta\,\tau, \Gamma \vdash e_3 \ominus e_3' \downarrow \tau' \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_1) \wedge (\forall i :: \mathbb{N}.(n \doteq i+1) \to (\Phi_2 \wedge \forall\beta :: \mathbb{N}.(\alpha \doteq \beta+1) \to \Phi_3)) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \begin{array}{l} \text{case } e \text{ of nil } \to e_1 \\ \mid h\ ::_{NC}\ tl \to e_2 \\ \mid h\ ::_C\ tl \quad \to e_3 \end{array} \ominus \begin{array}{l} \text{case } e' \text{ of nil } \to e_1' \\ \mid h\ ::_{NC}\ tl \to e_2' \\ \mid h\ ::_C\ tl \quad \to e_3' \end{array} \downarrow \tau' \Rightarrow \Phi_e \wedge \Phi_{body}} \textbf{ alg-r-caseL-}\downarrow$$

Figure 56: RelRefU binary algorithmic typing rules (Part 1)

$$\frac{i :: S, \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \Lambda i.e \ominus \Lambda i.e' \downarrow \forall i{::}S.\,\tau \Rightarrow (\forall i :: S.\Phi)} \ \textbf{alg-r-iLam-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \forall i{::}S.\,\tau' \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e\,[I] \ominus e'\,[I] \uparrow \tau'\{I/i\} \Rightarrow \Phi} \ \textbf{alg-r-iApp-}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau\{I/i\} \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{pack } e \text{ with } I \ominus \text{pack } e' \text{ with } I \downarrow \exists i{::}S.\,\tau \Rightarrow \Phi} \ \textbf{alg-r-pack-}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \exists i{::}S.\,\tau_1 \Rightarrow \Phi_1 \\ i :: S, \Delta; \psi_a; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2) \qquad \Phi = (\Phi_1 \wedge \forall i :: S.\Phi_2)\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' \downarrow \tau_2 \Rightarrow \Phi} \ \textbf{alg-r-unpack-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Gamma \vdash e_1 \ominus e_1 \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow C \,\&\, \tau \Rightarrow C \wedge (C \rightarrow \Phi)} \ \textbf{alg-r-c-andI}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow C \,\&\, \tau_1 \Rightarrow \Phi_1 \\ \Delta; \psi_a; \Phi \wedge C; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2 \Rightarrow \Phi_2 \qquad \Phi_2' = C \rightarrow \Phi_2\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \ominus \text{clet } e_1' \text{ as } x \text{ in } e_2' \downarrow \tau_2 \Rightarrow \Phi_1 \wedge \Phi_2'} \ \textbf{alg-r-c-andE}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \tau' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau' \equiv \tau \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi_1 \wedge \Phi_2} \ \textbf{alg-r-}\uparrow\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash \tau \ \mathtt{wf} \qquad \mathrm{FIV}(\tau) \in \Delta}{\Delta; \psi_a; \Phi_a; \Gamma \vdash (e : \tau) \ominus (e' : \tau) \uparrow \tau \Rightarrow \Phi} \ \textbf{alg-r-anno-}\uparrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; C \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \tau \Rightarrow \Phi_2 \qquad \Delta \vdash C \ \mathtt{wf}\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathtt{split}\ (e_1, e_2) \text{ with } C \ominus \mathtt{split}\ (e_1', e_2') \text{ with } C \downarrow \tau \Rightarrow C \rightarrow \Phi_1 \wedge \neg C \rightarrow \Phi_2} \ \textbf{alg-r-split}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a \models \bot}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathtt{contra}\ e \ominus \mathtt{contra}\ e' \downarrow \tau \Rightarrow \top} \ \textbf{alg-r-contra}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi; |\Gamma|_1 \vdash e_1 \uparrow A_1 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi; |\Gamma|_2 \vdash e_2 \uparrow A_2 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathtt{switch}\ e_1 \ominus \mathtt{switch}\ e_2 \uparrow U\,(A_1, A_2) \Rightarrow \Phi_1 \wedge \Phi_2} \ \textbf{alg-r-switch}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi; |\Gamma|_1 \vdash e_1 \downarrow A_1 \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi; |\Gamma|_2 \vdash e_2 \downarrow A_2 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathtt{switch}\ e_1 \ominus \mathtt{switch}\ e_2 \downarrow U\,(A_1, A_2) \Rightarrow (\Phi_1 \wedge \Phi_2)} \ \textbf{alg-r-switch}\downarrow$$

Figure 57: RelRefU binary algorithmic typing rules (Part 2)

4.2 RelRefU Lemmas

Lemma 22 **(Substitution of RelRefU unary Core)**
Assume that

1. $\Delta; \Psi_a; \Omega \vdash e_1 :^c A \quad (1)$.
2. $\Delta; \Psi_a; \Omega, x : A \vdash e_1' :^c A' \quad (2)$.

then $\Delta; \Psi_a; \Omega \vdash e_1'[e_1/x] :^c A'$.

Proof. By induction on the typing derivation of the second assumption (2).

$$\frac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash x :^c A}\ \textbf{c-var}$$

Subcase: $z = x$.
TS: $\Delta; \Psi_a; \Omega \vdash z[e_1/x] :^c A'$.
Because $x = z$,$z[e_1/x] = e_1$, STS: $\Delta; \Phi_a; \Gamma \vdash e_1 :^c A'$.
From $(\Omega, x : A)(z) = A'$ and $x = z$, we know $A = A'$.
By the above statements, STS: $\Delta; \Psi_a; \Omega \vdash e_1 :^c A$, which is proved by the assumption.
Subcase: $z \neq x$.
we know: $\Omega(z) = A' \quad (\star)$.
TS: $\Delta; \Psi_a; \Omega \vdash z[e_1/x] :^c A'$.
Because $x \neq z$, $z[e_1/x] = z$, STS: $\Delta; \Psi_a; \Omega \vdash z :^c A'$.
By the core rule **c-r-vars** and $(\star)$, we construct the derivation :

$$\frac{\Omega(z) = A'}{\Delta; \Psi_a; \Omega \vdash z :^c A'}\ \textbf{c-r-var}$$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \to A_2\ \texttt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \to A_2, \Omega \vdash e :^c A_2}{\Delta; \Phi_a; \Omega \vdash \textbf{fix}\ f(x).e :^c A_1 \to A_2}\ \textbf{c-fix}$$

$e_1' = \text{fix } f(z).e_1$, $A' = A_1 \to A_2$.
TS: $\Delta; \Phi_a; \Omega \vdash \text{fix } f(z).(e_1''[e_1/x]) :^c A'$.
By IH on (1) and (3), $\Delta; \Psi_a; \Omega \vdash e_1''[e_1/x] :^c A_2 \quad (4)$.
By core rule **c-fix** and (4), we can derive :
$\Delta; \Phi_a; \Omega \vdash \text{fix } f(z).(e_1''[e_1/x]) :^c A'$. □

Lemma 23 **(Substitution of RelRefU Core)**
Assume that

1. $\Delta; \Psi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau \quad (1)$.
2. $\Delta; \Psi_a; \Gamma, x : \tau \vdash e_1' \backsim e_2' :^c \tau' \quad (2)$.

then $\Delta; \Psi_a; \Gamma \vdash e_1'[e_1/x] \backsim e_2'[e_2/x] :^c \tau'$.

Proof. By induction on the typing derivation of the second assumption (2).

Case $$\frac{(\Gamma, x : \tau)(z) = \tau'}{\Delta; \Phi_a; \Gamma, x : \tau \vdash z \backsim z :^c \tau'}\ \textbf{c-r-var}$$

Subcase: $z = x$.
TS: $\Delta; \Phi_a; \Gamma \vdash z[e_1/x] \backsim z[e_2/x] :^c \tau'$.
Because $x = z$,$z[e_1/x] = e_1$, $z[e_2/x] = e_1$, STS: $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau'$.
From $(\Gamma, x : \tau)(z) = \tau'$ and $x = z$, we know $\tau = \tau'$.
By the above statements, STS: $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau$, which is proved by the assumption.

Subcase: $z \neq x$.
we know: $\Gamma(z) = \tau'$ $(\star)$.
TS: $\Delta; \Phi_a; \Gamma \vdash z[e_1/x] \backsim z[e_2/x] :^c \tau'$.
Because $x \neq z$, $z[e_1/x] = z$, $z[e_2/x] = z$, STS: $\Delta; \Phi_a; \Gamma \vdash z \backsim z :^c \tau'$.
By the core rule **c-r-vars** and $(\star)$, we construct the derivation :

$$\frac{\Gamma(z) = \tau'}{\Delta; \Phi_a; \Gamma \vdash z \backsim z :^c \tau'} \ \textbf{c-r-var}$$

Case $\dfrac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e_1'' \backsim e_2'' :^c \square\, \tau' \quad (3)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \texttt{der}\ e_1'' \backsim \texttt{der}\ e_2'' :^c \tau'}$ **c-der**
$e_1' = \texttt{der}\ e_1''$, $e_2' = \texttt{der}\ e_2''$.
TS: $\Delta; \Phi_a; \Gamma \vdash \texttt{der}\ (e_1''[e_1/x]) \backsim \texttt{der}\ (e_2''[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Psi_a; \Gamma \vdash e_1''[e_1/x] \backsim e_2''[e_2/x] :^c \square\tau'$ (4).
By core rule **c-der** and (4), we can derive

$$\frac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e_1''[e_1/x] \backsim e_2''[e_2/x] :^c \square\, \tau' \quad (4)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \texttt{der}\ (e_1''[e_1/x]) \backsim \texttt{der}\ (e_2''[e_2/x]) :^c \tau'} \ \textbf{c-der}.$$

Case $\dfrac{\Delta; \Phi_a; z : \tau_1, f : \tau_1 \to \tau_2, \Gamma, x : \tau \vdash e_1'' \backsim e_2'' :^c \tau_2 \quad (3)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \textbf{fix}\ f(z).e_1'' \backsim \textbf{fix}\ f(z).e_2'' :^c \tau_1 \to \tau_2}$ **c-r-fix**
$e_1' = \text{fix}\ f(z).e_1$, $e_2' = \text{fix}\ f(z).e_2'', \tau' = \tau_1 \to \tau_2$.
TS: $\Delta; \Phi_a; \Gamma \vdash \text{fix}\ f(z).(e_1''[e_1/x]) \backsim \text{fix}\ f(z).(e_2''[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Psi_a; \Gamma \vdash e_1''[e_1/x] \backsim e_2''[e_2/x] :^c \tau_2$ (4).
By core rule **c-r-fix** and (4), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \text{fix}\ f(z).(e_1''[e_1/x]) \backsim \text{fix}\ f(z).(e_2''[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; z : \tau_1, f : \square\,(\tau_1 \to \tau_2), \square\,\Gamma, x : \tau \vdash e \backsim e :^c \tau_2 \quad (3)}{\Delta; \Phi_a; \square\,\Gamma, \Gamma', x : \tau \vdash \textbf{fix}_{NC}\ f(z).e \backsim \textbf{fix}_{NC}\ f(z).e :^c \square\,(\tau_1 \to \tau_2)}$ **c-r-fixNC**
$e_1' = \text{fix}\ f(z).e$, $e_2' = \text{fix}\ f(z).e, \tau' = \square\,(\tau_1 \to \tau_2)$.
TS: $\Delta; \Phi_a; \square\,\Gamma, \Gamma' \vdash \text{fix}\ f(z).(e[e_1/x]) \backsim \text{fix}\ f(z).(e[e_2/x]) :^c \tau'$.
By IH on (1) and (3), $\Delta; \Phi_a; z : \tau_1, f : \square\,(\tau_1 \to \tau_2), \square\,\Gamma \vdash e_1''[e_1/x] \backsim e_2''[e_2/x] :^c \tau_2$ (4).
By core rule **c-r-fixNC** and (4), we can derive :
$\Delta; \Phi_a; \square\,\Gamma, \Gamma' \vdash \text{fix}\ f(z).(e[e_1/x]) \backsim \text{fix}\ f(z).(e[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; \Gamma, x : \tau \vdash e_3 \backsim e_3' :^c \tau'' \quad (3) \qquad \Delta; \Phi_a; \Gamma, x : \tau \vdash e_4 \backsim e_4' :^c \textbf{list}[n]^\alpha\, \tau'' \quad (4)}{\Delta; \Phi_a; \Gamma, x : \tau \vdash \textbf{cons}_C(e_3, e_4) \backsim \textbf{cons}_C(e_3', e_4') :^c \textbf{list}[n+1]^{\alpha+1}\, \tau''}$ **c-r-cons1**
$e_1' = \text{cons}_C(e_3, e_4)$, $e_2' = \text{cons}_C(e_3', e_4'), \tau' = \text{list}[n+1]^{\alpha+1}\, \tau''$.

TS: $\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_3[e_1/x], e_4[e_1/x]) \backsim \text{cons}_C(e_3'[e_2/x], e_4'[e_2/x]) :^c \tau'$.
By IH on (1) and (3),$\Delta; \Phi_a; \Gamma \vdash e_3[e_1/x] \backsim e_3'[e_2/x] :^c \tau''$ (5).
By IH on (1) and (4),$\Delta; \Phi_a; \Gamma \vdash e_4[e_1/x] \backsim e_4'[e_2/x] :^c \tau''$ (6).
By core rule **c-r-fixNC** and (5),(6), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_3[e_1/x], e_4[e_1/x]) \backsim \text{cons}_C(e_3'[e_2/x], e_4'[e_2/x]) :^c \tau'$.

Case $\dfrac{\Delta; \Phi_a; \square\,\Gamma, x : \tau \vdash e \backsim e :^c \tau'' \quad (3)}{\Delta; \Phi_a; \square\,\Gamma, \Gamma', x : \tau \vdash \texttt{NC}\ e \backsim \texttt{NC}\ e :^c \square\,\tau''}$ **c-nochange**
$e_1' = \texttt{NC}\ e$, $e_2' = \texttt{NC}\ e, \tau' = \square\,\tau''$.
TS: $\Delta; \Phi_a; \Gamma \vdash \texttt{NC}\ e[e_1/x] \backsim \texttt{NC}\ e[e_2/x]) :^c \tau'$.
By IH on (1) and (3),$\Delta; \Phi_a; \Gamma \vdash e[e_1/x] \backsim e[e_2/x] :^c \tau''$ (4).
By core rule **c-nochange** and (4), we can derive :
$\Delta; \Phi_a; \Gamma \vdash \texttt{NC}\ e[e_1/x] \backsim \texttt{NC}\ e[e_2/x]) :^c \tau'$

Case $\dfrac{\Delta; \Phi_a; |\Gamma|_1 \vdash e_1'' :^c A_1 \quad (3) \qquad \Delta; \Phi_a; |\Gamma|_2 \vdash e_2'' :^c A_2 \quad (4)}{\Delta; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1'' \backsim \texttt{switch}\ e_2'' :^c U\,(A_1, A_2)}$ **c-switch**

$e_1' = \texttt{switch}\ e_1''$, $e_2' = \texttt{switch}\ e_2'', \tau' = U\,(A_1, A_2)$.
TS: $\Delta; \Phi_a; \Gamma \vdash e_1' \backsim e_2' :^c \tau'$.
By Lemma 22 on (1) and (3),$\Delta; \Phi_a; |\Gamma|_1 \vdash e_1''[e_1/x] :^c A_1 \quad (5)$.
By Lemma 22 on (1) and (4),$\Delta; \Phi_a; |\Gamma|_2 \vdash e_2''[e_2/x] :^c A_2 \quad (6)$.
By **c-switch** and (5),(6), we conclude that $\Delta; \Phi_a; \Gamma \vdash e_1' \backsim e_2' :^c \tau'$.

□

Lemma 24 **(Reflexivity of Unary Algorithmic Subtyping in RelRefU)**
$\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the unary type. □

Lemma 25 **(Reflexivity of Algorithmic Binary Type Equivalence in RelRefU)**
$\forall \Delta, \psi_a, \Phi_a$, there exists Φ s.t $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$.

Proof. By induction on the binary type. Most cases are the same as RelRef's corresponding proof.

Case $U\,(A_1, A_2)$
By algorithmic equivalence rule **U**.

$$\dfrac{\begin{array}{ll}\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1 \Rightarrow \Phi_1 \quad (\star) & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1 \Rightarrow \Phi_1 \\ \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2 \Rightarrow \Phi_2 \quad (\diamond) & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2 \Rightarrow \Phi_2\end{array}}{\Delta; \psi_a; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1, A_2) \Rightarrow \Phi_1 \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_2}\,\mathbf{U}$$

By Lemma 24 on $(\star)$ and $(\diamond)$, we conclude
$\Delta; \psi_a; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1, A_2) \Rightarrow \Phi_1 \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_2$ and $\Delta; \psi_a; \Phi_a \models \Phi_1 \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_2$.

□

Lemma 26 **(Transitivity of Unary Algorithmic Subtyping in RelRefU)**
If $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_3 \Rightarrow \Phi_2$ and $\Delta; \Phi_a \models \Phi_1 \wedge \Phi_2$, then $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_3 \Rightarrow \Phi_3$ for some Φ_3 such that $\Delta; \Phi_a \models \Phi_3$.

Proof. By induction on the first two algorithmic subtyping derivation.

Case $\dfrac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1 \quad (1) \qquad \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2 \quad (2)}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A_1' \to A_2' \Rightarrow \Phi_1 \wedge \Phi_2}$ **alg-u-fun,** $\dfrac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1'' \sqsubseteq A_1' \Rightarrow \Phi_1' \quad (3}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1' \to}$

u-fun
By IH on (1) and (3), we get
$\models^{\mathsf{A}} A_1'' \sqsubseteq A_1 \Rightarrow \Phi_1 \wedge \Phi_1'$ and $\Delta; \Phi_a \models \Phi_1 \wedge \Phi_1'$.
By IH on (2) and (4), we get
$\models^{\mathsf{A}} A_2 \sqsubseteq A_2'' \Rightarrow \Phi_2 \wedge \Phi_2'$ and $\Delta; \Phi_a \models \Phi_2 \wedge \Phi_2'$.
By rule **alg-u-fun** and above statements, we conclude
$\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A_1'' \to A_2'' \Rightarrow \Phi_1' \wedge \Phi_2' \wedge \Phi_1 \wedge \Phi_2$.

□

Theorem 27 **(Soundness of the Algorithmic Unary Subtyping in RelRefU)**
Assume that

1. $\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi$
2. $\mathrm{FIV}(\Phi_a, A, A') \subseteq \Delta, \psi_a$

3. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable s.t $\Delta \ \triangleright \ \theta_a : \psi_a$ is derivable.

Then $\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A[\theta_a] \sqsubseteq A'[\theta_a]$.

Proof. By induction on the algorithmic unary subtyping derivation.

Case $$\frac{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A'_1 \sqsubseteq A_1 \Rightarrow \Phi_1 \quad (1) \qquad \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A'_2 \Rightarrow \Phi_2 \quad (2)}{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A'_1 \to A'_2 \Rightarrow \Phi_1 \wedge \Phi_2} \ \textbf{alg-u-fun}$$
By IH on (1) and assumption 2,3, we get $\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A'_1[\theta_a] \sqsubseteq A_1[\theta_a]$.
By IH on (2) and assumption 2,3, we get $\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A_2[\theta_a] \sqsubseteq A'_2[\theta_a]$.
By subtyping rule $\to$ **exec**, then we conclude
$\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A_1 \to A_2[\theta_a] \sqsubseteq A'_1 \to A'_2[\theta_a]$.

□

Theorem 28 (Completeness of the Unary Algorithmic Subtyping)
Assume that $\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'$. Then $\exists \Phi$. such that $\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the unary subtyping derivation.

Case $$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A'_1 \sqsubseteq A_1 \quad (1) \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A'_2 \quad (2)}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A'_1 \to A'_2} \to \textbf{exec}$$
By IH on (1), there exists Φ_1 s.t $\Delta; \Phi_a \models^{\mathsf{A}} A'_1 \sqsubseteq A_1 \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models \Phi_1$.
By IH on (2), there exists Φ_2 s.t $\Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A'_2 \Rightarrow \Phi_2$ and $\Delta; \Phi_a \models \Phi_2$.
By **alg-u-fun** and the above statements, we conclude there eixsts $\Phi = \Phi_1 \wedge \Phi_2$, s.t $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \to A_2 \sqsubseteq A'_1 \to A'_2 \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

□

Theorem 29 (Soundness of the Algorithmic Binary Type Equality in RelRefU)
Assume that

1. $\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$
2. $\text{FIV}(\Phi_a, \tau, \tau') \subseteq \Delta, \psi_a$
3. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable s.t $\Delta \ \triangleright \ \theta_a : \psi_a$ is derivable.

Then $\Delta; \Phi_a[\theta_a] \models \tau[\theta_a] \equiv \tau'[\theta_a]$.

Proof. By induction on the algorithmic binary type equivalence derivation. Same as RelRef.

Case $$\frac{\begin{array}{ll} \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A'_1 \Rightarrow \Phi_1 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A'_1 \sqsubseteq A_1 \Rightarrow \Phi'_1 \\ \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A'_2 \Rightarrow \Phi_2 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A'_2 \sqsubseteq A_2 \Rightarrow \Phi'_2 \end{array}}{\Delta; \psi_a; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A'_1, A'_2) \Rightarrow \Phi_1 \wedge \Phi'_1 \wedge \Phi_2 \wedge \Phi'_2} \ \textbf{U}$$
From assumption 3, $\Delta; \Phi_a[\theta_a] \models (\Phi_1 \wedge \Phi'_1 \wedge \Phi_2 \wedge \Phi'_2)[\theta_a]$
we know $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a]$ (a)
$\Delta; \Phi_a[\theta_a] \models \Phi'_1[\theta_a]$ (b)
$\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a]$ (c)
$\Delta; \Phi_a[\theta_a] \models \Phi'_2[\theta_a]$ (d)
By IH on the first premise and (a), the second and (b), the third and (c) and the fourth and (d) and the rule **eq-U**, we can conclude that $\Delta; \Phi_a[\theta_a] \models U(A_1, A_2)[\theta_a] \equiv U(A'_1, A'_2)[\theta_a]$.

□

Theorem 30 (Completeness of the Binary Algorithmic Type Equivalence in RelRefU) Assume that $\Delta; \Phi_a \models \tau \equiv \tau'$. Then $\exists\, \Phi$. such that $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the binary subtyping derivation. Same as RelRef.

Case $\dfrac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2}{\Delta; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1', A_2')}$ **eq-U**

By IH on the first premise, we know $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \Rightarrow \Phi_1$ and $\Delta; \Phi_a \vDash \Phi_1$.
Similarly by IH on the other premises respectively, we get
$\Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1'$ and $\Delta; \Phi_a \vDash \Phi_1'$
$\Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2$ and $\Delta; \Phi_a \vDash \Phi_2$
$\Delta; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2 \Rightarrow \Phi_2'$ and $\Delta; \Phi_a \vDash \Phi_2'$
By the above statements and the relational algorithmic type equivalence rule **U**, we conclude where $\Phi = \Phi_1 \wedge \Phi_1' \wedge \Phi_2 \wedge \Phi_2'$:
$\Delta; \Phi_a \models U(A_1, A_2) \equiv U(A_1', A_2') \Rightarrow \Phi$ and $\Delta; \Phi_a \vDash \Phi$

□

Lemma 31 (Existence of coercions for relational subtyping in RelRefU) If $\Delta; \Phi_a \models \tau \sqsubseteq \tau'$ then there exists $\mathtt{coerce}_{\tau,\tau'} \in$ Core s.t. $\Delta; \Phi; \cdot \vdash \mathtt{coerce}_{\tau,\tau'} \backsim \mathtt{coerce}_{\tau,\tau'} :^{c} \tau \to \tau'$.

Proof. Proof is by induction on the subtyping derivation. We denote the witness e of type $\tau \to \tau'$ as $\mathtt{coerce}_{\tau,\tau'}$ for clarity.

Case $\dfrac{}{\Delta; \Phi_a \models \square\, U\,(\mathbf{int}, \mathbf{int}) \sqsubseteq \mathbf{int}_r}$ **□U-int**
Then, we can construct the derivation using the primitive function $\mathrm{box}_U : \square\, U\,(\mathrm{int}, \mathrm{int}) \to \mathrm{int}_r$

$$\dfrac{}{\Delta; \Phi; \cdot \vdash \mathrm{fix}\ f(x).\mathrm{box}_U x \backsim \mathrm{fix}\ f(x).\mathrm{box}_U x :^{c} \square\, U\,(\mathrm{int}, \mathrm{int}) \to \mathrm{int}_r}$$

Case $\dfrac{}{\Delta; \Phi_a \models U\,(\forall i{::}S.\, A, \forall i{::}S.\, A') \sqsubseteq \forall i{::}S.\, U\,(A, A')}$ **∀U**
Then, we can immediately construct the following derivation where $e = \mathrm{fix}\ f(x).\Lambda i.\mathtt{switch}\ (x\,[i])$ using the **c-switch** and **c-iApp** rules.
$\Delta; \Phi; \cdot \vdash e \backsim e :^{c} (U\,(\forall i{::}S.\, A, \forall i{::}S.\, A')) \to \forall i{::}S.\, U\,(A, A')$

□

Theorem 32 (Types are preserved by embedding in RelRefU)

1. If $\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A$, then $\Delta; \Phi_a; \Omega \vdash e^* :^{c} A$ and $\Delta; \Phi_a; \Omega \vdash e : A$.
2. If $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$, then $\Delta; \Phi_a; \Gamma \vdash e_1^* \backsim e_2^* :^{c} \tau$ and $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau$.

Proof. By simultaneous induction on the given derivations.
Proof of Theorem 32.1:

Case $\dfrac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \to A_2\ \mathtt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \to A_2, \Omega \vdash e \rightsquigarrow e^* : A_2}{\Delta; \Phi_a; \Omega \vdash \mathbf{fix}\ f(x).e \rightsquigarrow \mathbf{fix}\ f(x).e^* : A_1 \to A_2}$ **e-u-fix**
By Theorem 18.1 on the premise, we get $x : A_1, f : A_1 \to A_2, \Omega \vdash e :^{c} A_2$. Then, by unary core rule **c-fix**,we conclude: $\dfrac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \to A_2\ \mathtt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \to A_2, \Omega \vdash e :^{c} A_2}{\Delta; \Phi_a; \Omega \vdash \mathrm{fix}\ f(x).e :^{c} A_1 \to A_2}$ **c-fix**.

Proof of Theorem 32.2:

Case $$\dfrac{\Delta; \Phi_a; |\Gamma|_1 \vdash e_1 \rightsquigarrow e_1^* : A_1 \quad (\star) \qquad \Delta; \Phi_a; |\Gamma|_2 \vdash e_2 \rightsquigarrow e_2^* : A_2 \quad (\diamond)}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{switch}\ e_1^* \backsim \texttt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}$$

By Theorem 32.1 on $(\star)$, we get $\Delta; \Phi; \Omega \vdash e_1^* :^c A_1 \quad (\star\star)$.

By Theorem 32.1 on $(\diamond)$, we get $\Delta; \Phi; \Omega \vdash e_2^* :^c A_2 \quad (\diamond\diamond)$.

Then, we conclude as follows: $\dfrac{\Delta; \Phi_a; |\Gamma|_1 \vdash e_1 :^c A_1 \qquad \Delta; \Phi_a; |\Gamma|_2 \vdash e_2 :^c A_2}{\Delta; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1 \backsim \texttt{switch}\ e_2 :^c U\,(A_1, A_2)}\ \textbf{c-switch}$

Case $$\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond) \qquad e' = \texttt{coerce}_{\tau,\tau'}}{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\sqsubseteq$$

By Theorem 32.2 on $(\star)$, $\Delta; \Phi; \Gamma \vdash e_1^* \backsim e_2^* :^c \tau \quad (\star\star)$.

By Lemma 31 using $(\diamond)$, we know that $\Delta; \Phi; \cdot \vdash e' \backsim e' :^c \tau \to \tau' \quad (\diamond\diamond)$.

By applying **c-r-app** rule on $(\star\star)$ and $(\diamond\diamond)$, we get $\Delta; \Phi; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau' \quad (\spadesuit)$.

By reflexivity of binary type equivalence, we know $\Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit)$.

Then, we conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau' \quad (\spadesuit) \qquad \Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit)}{\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \backsim e'\ e_2^* :^c \tau'}\ \textbf{c-r}-\sqsubseteq$$

□

Theorem 33 **(Completeness of embedding in RelRefU)**

1. If $\Delta; \Phi_a; \Omega \vdash e : A$, then there exists an e^* such that $\Delta; \Phi; \Omega \vdash e \rightsquigarrow e^* : A$.
2. If $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 : \tau$, then there exist e_1^*, e_2^* such that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 \rightsquigarrow e_1^* \backsim e_2^* : \tau$.

Proof. By simultaneous induction on the given RelCost derivations.

Proof of Theorem 33.1:

Case $$\dfrac{\Delta; \Phi_a; \Omega \vdash_k^t e : A \quad (\star) \qquad \Delta; \Phi_a; \models^{\mathsf{A}} A \sqsubseteq A' \quad (\diamond)}{\Delta; \Phi_a; \Omega \vdash e : A'}\ \sqsubseteq$$

By Theorem 33.1 on $(\star)$, we get $\exists e^*$ such that $\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A \quad (\star\star)$.

By **e-u-**$\sqsubseteq$ rule using $(\star\star)$, $(\diamond)$, we conclude as follows

$$\dfrac{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a; \Omega \vdash e \rightsquigarrow e^* : A'}\ \textbf{e-u-}\sqsubseteq$$

Proof of Theorem 33.2:

Case $\dfrac{\Delta;\Phi;|\Gamma|_1 \vdash e_1 : A_1 \quad (\star) \qquad \Delta;\Phi;|\Gamma|_2 \vdash e_2 : A_2 \quad (\diamond)}{\Delta;\Phi;\Gamma \vdash e_1 \backsim e_2 : U\,(A_1, A_2)}$ **r-switch**

By Theorem 33.1 on $(\star)$, we get $\exists\, e_1^*$ such that $\Delta;\Phi;\Omega \vdash^{t_1}_{k_1} e_1 \rightsquigarrow e_1{}^* : A_1 \quad (\star\star)$.
By Theorem 33.1 on $(\diamond)$, we get $\exists\, e_2^*$ such that $\Delta;\Phi;\Omega \vdash^{t_2}_{k_2} e_2 \rightsquigarrow e_2{}^* : A_2 \quad (\diamond\diamond)$.
By **e-switch** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\dfrac{\Delta;\Phi_a;|\Gamma|_1 \vdash e_1 \rightsquigarrow e_1^* : A_1 \quad (\star\star) \qquad \Delta;\Phi_a;|\Gamma|_2 \vdash e_2 \rightsquigarrow e_2^* : A_2 \quad (\diamond\diamond)}{\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_2 \rightsquigarrow \texttt{switch}\ e_1^* \backsim \texttt{switch}\ e_2^* : U\,(A_1, A_2)}\ \textbf{e-switch}.$$

Case $\dfrac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e \backsim e : \tau \\ \forall x \in dom(\Gamma).\ \ (\star)\Delta;\Phi_a \models \Gamma(x) \sqsubseteq \Box\,\Gamma(x) \quad (\diamond)\end{array}}{\Delta;\Phi_a;\Gamma,\Gamma' \vdash e \backsim e : \Box\,\tau}$ **rr-nochange**

By IH2 on $(\star)$, we get $\exists\, e^*$ such that $\Delta;\Phi_a;\Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau \quad (\dagger)$.
By Lemma 11 on $(\diamond)$, we get $\exists e_i = \texttt{coerce}_{\Gamma(x_i),\Box\,(\Gamma(x_i)}$ for all $x_i \in dom(\Gamma) \quad (\dagger\dagger)$.
By **e-nochange** embedding rule using $(\dagger)$ and $(\dagger\dagger)$, we can conclude as follows:

$$\dfrac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau \quad (\dagger) \\ \forall x_i \in dom(\Gamma),\quad e_i = \texttt{coerce}_{\Gamma(x_i),\Box\,\Gamma(x_i)} \quad (\dagger\dagger) \qquad \forall x_i \in dom(\Gamma),\quad \Delta;\Phi_a \models \Gamma(x_i) \sqsubseteq \Box\,\Gamma(x_i)\end{array}}{\Delta;\Phi_a;\Gamma,\Gamma' \vdash e \backsim e \rightsquigarrow \text{let } \overline{y_i = e_i\ x_i} \text{ in NC } e^*[\overline{\texttt{der}\ y_i/x_i}] \backsim \text{let } \overline{y_i = e_i\ x_i} \text{ in NC } e^*[\overline{\texttt{der}\ y_i/x_i}] : \Box\,\tau}\ \textbf{e-nochange}.$$

Case $\dfrac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e \backsim e' : \textbf{list}[n]^\alpha\,\tau \quad (\star) \qquad \Delta;\Phi_a \wedge n = 0;\Gamma \vdash e_1 \backsim e_1' : \tau' \quad (\diamond) \\ i,\Delta;\Phi_a \wedge n = i+1; h : \Box\,\tau, tl : \textbf{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \backsim e_2' : \tau' \quad (\dagger) \\ i,\beta,\Delta;\Phi_a \wedge n = i+1 \wedge \alpha = \beta+1; h : \tau, tl : \textbf{list}[i]^\beta\,\tau, \Gamma \vdash e_2 \backsim e_2' : \tau' \quad (\spadesuit)\end{array}}{\Delta;\Phi_a;\Gamma \vdash\ \textbf{case}\ e\ \textbf{of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2 \backsim\ \textbf{case}\ e'\ \textbf{of nil} \rightarrow e_1' \mid h :: tl \rightarrow e_2' : \tau'}$ **rr-caseL**

By IH2 on $(\star)$, we get $\exists\, e^*$ and $\exists\, e'^*$ s.t. $\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \text{list}[n]^\alpha\,\tau \quad (\star\star)$.
By IH2 on $(\diamond)$, we get $\exists\, e_1^*$ and $\exists\, e_1'^*$ s.t. $\Delta; n \doteq 0 \wedge \Phi_a;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow e_1^* \backsim e_1'^* : \tau' \quad (\diamond\diamond)$.
By IH2 on $(\dagger)$, we get $\exists\, e_2^*$ and $\exists\, e_2'^*$ s.t.
$i :: S, \Delta; n \doteq i+1 \wedge \Phi_a; h : \Box\,\tau, tl : \text{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^* \backsim e_2'^* : \tau' \quad (\dagger\dagger)$.
By IH2 on $(\spadesuit)$, we get $\exists\, e_2^{**}$ and $\exists\, e_2'^{**}$ s.t.
$i :: S, \beta :: S, \Delta; n \doteq i+1 \wedge \alpha \doteq \beta+1 \wedge \Phi_a; h : \tau, tl : \text{list}[i]^\beta\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^{**} \backsim e_2'^{**} : \tau' \quad (\spadesuit\spadesuit)$.

By **e-caseL** embedding rule using $(\star\star)$, $(\diamond\diamond)$, and $(\spadesuit\spadesuit)$, we can conclude as follows

$$\dfrac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \text{list}[n]^\alpha\,\tau \qquad \Delta;\Phi_a \wedge n = 0;\Gamma \vdash e_1 \backsim e_1' \rightsquigarrow e_1{}^* \backsim e_1'{}^* : \tau' \\ i,\Delta;\Phi_a \wedge n = i+1; h : \Box\,\tau, tl : \text{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2{}^* \backsim e_2'{}^* : \tau' \\ i,\beta,\Delta;\Phi_a \wedge n = i+1 \wedge \alpha = \beta+1; h : \tau, tl : \text{list}[i]^\beta\,\tau, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^{**} \backsim e_2'^{**} : \tau'\end{array}}{\Delta;\Phi_a;\Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil} \rightarrow e_1 \\ \quad \mid h :: tl \rightarrow e_2\end{array} \backsim \begin{array}{l}\text{case } e \text{ of nil} \rightarrow e_1' \\ \quad \mid h :: tl \rightarrow e_2'\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^* \text{ of nil } \rightarrow e_1^* \\ \mid h ::_{NC} tl \rightarrow e_2^* \backsim \\ \mid h ::_C tl \quad \rightarrow e_2^{**}\end{array} \begin{array}{l}\text{case } e'^* \text{ of nil } \rightarrow e_1'^* \\ \mid h ::_{NC} tl \rightarrow e_2'^* \\ \mid h ::_C tl \quad \rightarrow e_2'^{**}\end{array} : \tau'}\ \textbf{e-r-caseL}$$

Case $\dfrac{\begin{array}{c}\Delta;\Phi_a \vdash \tau_1 \rightarrow \tau_2\ \texttt{wf} \\ \Delta;\Phi_a; x : \tau_1, f : \Box\,(\tau_1 \rightarrow \tau_2), \Gamma \vdash e \backsim e : \tau_2 \quad (\star) \qquad \forall x \in dom(\Gamma).\Delta;\Phi_a \models \Gamma(x) \sqsubseteq \Box\,\Gamma(x) \quad (\diamond)\end{array}}{\Delta;\Phi_a;\Gamma \vdash \textbf{fix}\ f(x).e \backsim \textbf{fix}\ f(x).e : \Box\,(\tau_1 \rightarrow \tau_2)}$ **rr-**

fixNC
By IH2 on ($\star$), we get $\exists\, e^*$ such that $\Delta; \Phi_a; x : \tau_1, f : \Box\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2 \quad (\star\star)$.
By Lemma 31 on ($\diamond$), we get $\exists e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,(\Gamma(x_i)}$ for all $x_i \in dom(\Gamma) \quad (\diamond\diamond)$.

By **e-fixNC** embedding rule using ($\star\star$) and ($\diamond\diamond$), we can conclude as follows:

$$\dfrac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \to \tau_2 \ \texttt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \Box\,(\tau_1 \to \tau_2), \Gamma \vdash e \backsim e \rightsquigarrow e^* \backsim e^* : \tau_2 \qquad \forall x_i \in dom(\Gamma),\ \ e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,\Gamma(x_i)} \\ \forall x_i \in dom(\Gamma),\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \Box\,\Gamma(x) \qquad e^{**} = \text{let } \overline{y_i = e_i\ x_i} \text{ in } \text{fix}_{NC}\ f(x).e^*[\overline{\texttt{der}\ y_i / x_i}]\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \backsim \text{fix } f(x).e \rightsquigarrow e^{**} \backsim e^{**} : \Box\,(\tau_1 \to \tau_2)}\ \textbf{e-r-fixNC}.$$

Case $\dfrac{\begin{array}{c} i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau \quad (\star) \\ i \notin \textbf{FIV}(\Phi_a; \Gamma)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \Lambda e \backsim \Lambda e' : \forall i{::}S.\,\tau}\ \textbf{rr-iLam}$

By IH2 on ($\star$), we get $\exists\, e^*$ and $\exists\, e'^*$ such that $i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau \quad (\star\star)$.
By **e-iLam** embedding rule using ($\star\star$) , we can conclude as follows:

$$\dfrac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau \qquad i \notin \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda.e \backsim \Lambda.e' \rightsquigarrow \Lambda i.e^* \backsim \Lambda i.e'^* : \forall i{::}S.\,\tau}\ \textbf{e-r-iLam}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \forall i{::}S.\,\tau \quad (\star) \\ \Delta \vdash I : S \quad (\diamond)\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \backsim e'[\,] : \tau\{I/i\}}\ \textbf{rr-iApp}$

By IH2 on ($\star$), we get $\exists\, e^*$ such that $\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \forall i{::}S.\,\tau \quad (\star\star)$.
By **e-iApp** embedding rule using ($\star\star$) and ($\diamond$), we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \forall i{::}S.\,\tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[] \backsim e'[] \rightsquigarrow e^*[I] \backsim e'^*[I] : \tau\{I/i\}}\ \textbf{e-r-iApp}.$$

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' : \tau\{I/i\} \quad (\star) \qquad \Delta \vdash I :: S \quad (\diamond)}{\Delta; \Phi_a; \Gamma \vdash \textbf{pack}\ e \backsim \textbf{pack}\ e' : \exists i{::}S.\,\tau}\ \textbf{rr-pack}$

By IH2 on ($\star$), we get $\exists\, e^*$ and $\exists\, e'^*$ such that $\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau\{I/i\} \quad (\star\star)$.
By **e-pack** embedding rule using ($\star\star$) and ($\diamond$) , we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' \rightsquigarrow e^* \backsim e'^* : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \backsim \text{pack } e' \rightsquigarrow \text{pack } e^* \text{ with } I \backsim \text{pack } e'^* \text{ with } I : \exists i{::}S.\,\tau}\ \textbf{e-r-pack}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' : \exists i{::}S.\,\tau_1 \quad (\star) \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' : \tau_2 \quad (\diamond) \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \textbf{unpack}\ e_1\ \textbf{as}\ x\ \textbf{in}\ e_2 \backsim \textbf{unpack}\ e_1'\ \textbf{as}\ x\ \textbf{in}\ e_2' : \tau_2}\ \textbf{rr-unpack1}$

By IH2 on ($\star$), we get $\exists\, e_1^*$ and $\exists\, e_1'^*$ such that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_1' \rightsquigarrow e_1^* \backsim e_1'^* : \exists i{::}S.\,\tau_1 \quad (\star\star)$.
By IH2 on ($\diamond$), we get $\exists\, e_2^*$ and $\exists\, e_2'^*$ such that $i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \backsim e_2' \rightsquigarrow e_2^* \backsim e_2'^* : \tau_2 \quad (\diamond\diamond)$.
By **e-unpack** embedding rule using ($\star\star$) and ($\diamond\diamond$) , we can conclude as follows:

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_1' \rightsquigarrow {e_1}^* \backsim {e_1'}^* : \exists i{::}S.\,\tau_1 \qquad i :: S,\Delta;\Phi_a;x:\tau_1,\Gamma\vdash e_2 \backsim e_2' \rightsquigarrow {e_2}^* \backsim {e_2'}^* : \tau_2 \\ i\notin FV(\Phi_a;\Gamma,\tau_2,t_2) \qquad e_1^{**} = \text{unpack } e_1^* \text{ as } (x,i) \text{ in } e_2^* \qquad e_2^{**} = \text{unpack } e_1'^* \text{ as } (x,i) \text{ in } e_2'^*\end{array}}{\Delta;\Phi_a;\Gamma\vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \backsim \text{unpack } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow e_1^{**} \backsim e_2^{**} : \tau_2}\ \textbf{e-r-unpack}.$$

Case $\dfrac{\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_2 : \tau \quad (\star) \qquad \Delta;\Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond)}{\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_2 : \tau'}\ \textbf{rr-}\sqsubseteq$

By IH2 on $(\star)$, we get $\exists\, e_1^*, e_2^*$ such that $\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \quad (\star\star)$.

By Lemma 31 on $(\diamond)$, we can show that $\exists e' = \mathtt{coerce}_{\tau,\tau'} \quad (\diamond\diamond)$.

By **e-r-**$\sqsubseteq$ rule using $(\star\star)$, $(\diamond\diamond)$ and $(\diamond)$, we conclude as follows

$$\frac{\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_2 \rightsquigarrow {e_1}^* \backsim {e_2}^* : \tau \qquad \Delta;\Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \mathtt{coerce}_{\tau,\tau'}}{\Delta;\Phi_a;\Gamma\vdash e_1 \backsim e_2 \rightsquigarrow e'\ e_1^* \backsim e'\ e_2^* : \tau'}\ \textbf{e-r-}\sqsubseteq$$

□

Theorem 34 (Soundness of algorithmic typechecking in RelRefU)

1. Assume that $\Delta;\psi_a;\Phi_a;\Omega\vdash e \downarrow A \to \Phi$ and $\mathrm{FIV}(\Phi_a,\Omega,A) \subseteq dom(\Delta,\psi_a)$ and θ_a is a valid substitution for ψ_a s.t. $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A[\theta_a]$.
2. Assume that $\Delta;\psi_a;\Phi_a;\Omega\vdash e \uparrow A \Rightarrow \Phi$ and $\mathrm{FIV}(\Phi_a,\Omega) \subseteq dom(\Delta,\psi_a)$ and θ_a is the valid substitutions for ψ_a such that $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A[\theta_a]$.
3. Assume that $\Delta;\psi_a;\Phi_a;\Gamma\vdash e \ominus e' \downarrow \tau \Rightarrow \Phi$ and $\mathrm{FIV}(\Phi_a,\Gamma,\tau) \subseteq dom(\Delta,\psi_a)$ and θ_a is a valid substitution for ψ_a such that $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta;\Phi_a[\theta_a];\Gamma[\theta_a]\vdash |e| \backsim |e'| :^c \tau[\theta_a]$.
4. Assume that $\Delta;\psi_a;\Phi_a;\Gamma\vdash e \ominus e' \uparrow \tau \Rightarrow \Phi$ and $\mathrm{FIV}(\Phi_a,\Gamma) \subseteq dom(\Delta,\psi_a)$ and θ_a is the valid substitution for ψ_a such that $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ holds. Then, $\Delta;\Phi_a[\theta_a];\Gamma[\theta_a]\vdash |e| \backsim |e'| :^c \tau[\theta_a]$.

Proof. By simultaneous induction on the given algorithmic typing derivations.

Proof of Theorem 34.1:

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega\vdash e \uparrow A' \Rightarrow \Phi_1 \qquad \Delta;\psi_a;\Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi_2}{\Delta;\psi_a;\Phi_a;\Omega\vdash e \downarrow A \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-}\uparrow\downarrow$

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A[\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega,A) \subseteq dom(\Delta,\psi_a) \quad (\star)$ and

$\Delta;\Phi_a[\theta_a] \models \Phi_1 \wedge \Phi_2[\theta_a] \quad (\star\star)$

Using $(\star)$ and $(\diamond)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta \ \triangleright\ \theta_a : \psi_a$
b) $\Delta;\Phi_a[\theta_a] \models \Phi_1[\theta_a]$
c) $\Delta;\Phi_a[\theta_a] \models \Phi_2[\theta_a]$

By IH2 on the first premise using $(\star)$, a) and b), we can show that

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A'[\theta_a] \tag{4.1}$$

By Theorem 27 using the second premise and c), we obtain

$$\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A'[\theta_a] \sqsubseteq A[\theta_a] \tag{4.2}$$

Note that due to ($\star$), we can conclude by the **c-$\sqsubseteq$** exec rule using eqs. (5.6) and (5.7) that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c A[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; f : A_1 \to A_2, x : A_1, \Omega \vdash e \downarrow A_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \textbf{fix}\ f(x).e \downarrow A_1 \to A_2 \Rightarrow \Phi}$ **alg-u-fix-$\downarrow$**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{fix}\ f(x).|e| :^c A_1[\theta_a] \to A_2[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A_1 \to A_2) \subseteq dom(\Delta, \psi_a)$ ($\star$) and

$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ ($\star\star$)

Using ($\star$), we can show that

a) $\text{FIV}(\Phi_a, \Omega, A_1, A_1 \to A_2) \subseteq dom(\Delta, \psi_a)$.

We also can show that ($\star\star$)'s derivation must be in a form such that we have

b) $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$

By IH1 on the first premise using a) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; x : A_1[\theta_a], f : A_1[\theta_a] \to A_2[\theta_a], \Omega[\theta_a] \vdash |e| :^c A_2[\theta_a] \tag{4.3}$$

By the **c-fix** rule using eq. (5.8), we obtain

$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{fix}\ f(x).|e| :^c A_1[\theta_a] \to A_2[\theta_a]$.

Case $\dfrac{i :: S, \Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \Lambda i.e \downarrow \forall i{::}S.\ A \Rightarrow \forall i :: S.\Phi}$ **alg-u-iLam-$\downarrow$**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \Lambda i.|e| :^c \forall i{::}S.\ A[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, \forall i{::}S.\ A) \subseteq dom(\Delta, \psi_a)$ ($\star$) and

$\Delta; \Phi_a[\theta_a] \models \forall i :: S.\Phi[\theta_a]$ ($\star\star$)

Using ($\star$), we can show that

a) $\text{FIV}(\Phi_a, \Omega, A) \subseteq i, dom(\Delta, \psi_a)$.

We can also show that ($\star\star$)'s derivation must be in a form such that we have

b) $i :: S, \Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$

By IH1 on the premise using a) and b), we can show that

$$i :: S, \Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c A[\theta_a] \tag{4.4}$$

By the **c-iLam** rule using eq. (5.9), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \Lambda i.|e| :^c \forall i{::}S.\, A[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A\{I/i\} \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash \textbf{pack } e \textbf{ with } I \downarrow \exists i{::}S.\, A \Rightarrow \Phi}$ **alg-u-pack-↓**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{pack } |e| \text{ with } I :^c \exists i{::}S.\, A[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, \exists i{::}S.\, A) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$

Using $(\star)$ and the second premise, we can show that

a) $\text{FIV}(\Phi_a, \Omega, A\{I/i\}) \subseteq dom(\Delta, \psi_a)$.

By IH1 on the premise using a) and $(\star\star)$, we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c A[\theta_a]\{I/i\} \tag{4.5}$$

By the **c-pack** rule using eq. (5.10) and the second premise, we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{pack } |e| \text{ with } I :^c \exists i{::}S.\, A[\theta_a]$.

Case $\dfrac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow \exists i{::}S.\, A_1 \Rightarrow \Phi_1 \\ i :: S, \Delta; \psi_a; \Phi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Omega, A_2) \qquad \Phi = \Phi_1 \wedge \forall i :: S.\Phi_2\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \textbf{unpack } e_1 \textbf{ as } (x, i) \textbf{ in } e_2 \downarrow A_2 \Rightarrow \Phi}$ **alg-u-unpack-↓**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{unpack } |e_1| \text{ with } (x, i) \text{ in } |e_2| :^c A_2[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A_2) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models \Phi_1 \wedge \forall i :: S.\Phi_2[\theta_a]$ $(\star\star)$

Using $(\star)$, $(\diamond)$ and the 3rd premise, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a]$

b) $i :: S, \Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a]$

By IH2 on the first premise using $(\star)$, a) , we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e_1| :^c \exists i{::}S.\, A_1[\theta_a] \tag{4.6}$$

From $(\star)$, we can show that

c) $\text{FIV}(\Phi_a, A_1, \Omega, A_2) \subseteq i, dom(\Delta, \psi, \psi_a)$

By IH1 on the second premise using b), c), $(\star)$, we obtain

$$i :: S, \Delta; \Phi_a[\theta_a]; x : A_1[\theta\, \theta_a], \Omega[\theta_a] \vdash |e_2| :^c A_2[\theta_a] \tag{4.7}$$

Then by the **c-unpack** rule using eqs. (5.11) and (5.12), we can show that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \text{unpack } |e_1| \text{ with } (x, i) \text{ in } |e_2$

Case $$\dfrac{\begin{array}{c}\Delta; \psi_a; C \wedge \Phi_a; \Omega \vdash e_1 \downarrow A \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Omega \vdash e_2 \downarrow A \Rightarrow \Phi_2 \qquad \Delta \vdash C \ \texttt{wf}\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{split}\ (e_1, e_2)\ \textbf{with}\ C \downarrow A \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2}\ \textbf{alg-u-split}\!\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \texttt{split}\ (|e_1|, |e_2|)$ with $C :^c A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A) \subseteq \mathit{dom}(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models (C \to \Phi_1 \wedge \neg C \to \Phi_2)[\theta_a]$ $(\star\star)$
Using $(\star)$ and the third premise, we can show that

a) $\text{FIV}(C \wedge \Phi_a, \Omega, A) \subseteq \mathit{dom}(\Delta, \psi_a)$.
b) $\text{FIV}(\neg C \wedge \Phi_a, \Omega, A) \subseteq \mathit{dom}(\Delta, \psi_a)$.

Using $(\star\star)$ and the third premise, we can show that

c) $\Delta; C \wedge \Phi_a[\theta_a] \models \Phi_1[\theta_a]$
d) $\Delta; \neg C \wedge \Phi_a[\theta_a] \models \Phi_2[\theta_a]$

By IH1 on the first premise using $(\star)$ and c), we can show that

$$\Delta; C \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e_1| :^c A[\theta_a]\{I[\theta_a]/i\} \tag{4.8}$$

By IH1 on the second premise using $(\star)$ and d), we can show that

$$\Delta; \neg C \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e_2| :^c A[\theta_a]\{I[\theta_a]/i\} \tag{4.9}$$

By the **c-split** rule using eqs. (5.13) and (5.14) and the third premise, we obtain
$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash \texttt{split}\ (|e_1|, |e_2|)$ with $C :^c A[\theta_a]$.

Case $$\dfrac{\Delta; \Phi \wedge C; \Omega \vdash e \downarrow A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow C \supset A \Rightarrow C \to \Phi}\ \textbf{alg-u-c-impI}\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c C[\theta_a] \supset A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, C \supset A, k, t) \subseteq \mathit{dom}(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models (C \to \Phi)[\theta_a]$ $(\star\star)$
Using $(\star)$, we can show that

a) $\text{FIV}(C \wedge \Phi_a, \Omega, A) \subseteq \mathit{dom}(\Delta, \psi_a)$.

By IH1 on the premise using $(\star)$ and a), we can show that

$$\Delta; C[\theta_a] \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c A[\theta_a] \tag{4.10}$$

By the **c-cimpI** rule using eq. (5.15), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c C[\theta_a] \supset A[\theta_a]$.

Proof of Theorem 34.2:

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega\vdash e\downarrow A\Rightarrow \Phi \qquad \Delta;\Phi_a\vdash^{\mathsf{A}} A\ \mathtt{wf} \qquad \mathbf{FIV}(A,k,t)\in\Delta}{\Delta;\psi_a;\Phi_a;\Omega\vdash (e:A,k,t)\uparrow A\Rightarrow\Phi}$ **alg-u-anno-↑**

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |(e:A,k,t)| :^c A[\theta_a]$.

Since by definition, $\forall e.\ |(e:\_)| = |e|$, STS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A[\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega)\subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a]\models\Phi[\theta_a]$ $(\star\star)$

Using the third premise, we can show that

a) $\mathrm{FIV}(\Phi_a,\Omega,A)\subseteq dom(\Delta,\psi_a)$.

By IH1 on the first premise using $(\star\star)$ and a), we can conclude that $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e| :^c A[\theta_a]$.

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega\vdash e_1\uparrow A_1\to A_2\Rightarrow\Phi_1 \qquad \Delta;\psi_a;\Phi_a;\Omega\vdash e_2\downarrow A_1\Rightarrow\Phi_2}{\Delta;\psi_a;\Phi_a;\Omega\vdash e_1\ e_2\uparrow A_2\Rightarrow\Phi_1\wedge\Phi_2}$ **alg-u-app-↑**

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e_1|\ |e_2| :^c A_2[\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega)\subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a]\models(\Phi_1\wedge\Phi_2)[\theta_a]$ $(\star\star)$ such that $\Delta\ \rhd\ \theta_a:\psi_a$ are derivable.

From $(\star\star)$, we know

a) $\Delta;\Phi_a[\theta_a]\models\Phi_1[\theta_a]$

b) $\Delta;\Phi_a[\theta_a]\models\Phi_2[\theta_a]$

By IH on the first premise using $(\star)$ and (a), we obtain

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e_1| :^c A_1[\theta_a]\to A_2[\theta_a] \tag{4.11}$$

By IH2 on the third premise using $(\star)$ and (b), we obtain

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e_2| :^c A_1[\theta\,\theta_a] \tag{4.12}$$

Then, by using **c-app** rule using eqs. (5.16) and (5.17), we can show that $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e_1|\ |e_2| :^c A_2[\theta_a]$.

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega\vdash e\uparrow \forall i{::}S.\ A'\Rightarrow\Phi \qquad \Delta\vdash I::S}{\Delta;\psi_a;\Phi_a;\Omega\vdash e\,[I]\uparrow A'\{I/i\}\Rightarrow\Phi}$ **alg-u-iApp-↑**

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a]\vdash |e|\ [I] :^c (A'\{I/i\})[\theta\,\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega)\subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a]\models\Phi_2[\theta_a]$ $(\star\star)$ such that $\Delta\ \rhd\ \theta_a:\psi_a$ are derivable.

By IH2 on the first premise using ($\star$), we obtain

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e| :^c \forall i{::}S.\, A'[\theta_a] \tag{4.13}$$

Then, by **c-iApp** rule using eq. (5.18) and the second premise, we can conclude that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e|\ [I] :^c A'[\theta_a]\{I/i\}$.

Proof of Theorem 34.3:

Case $\dfrac{\Delta; \psi_a; \Phi_a; \square\,\Gamma \vdash e \ominus e \downarrow \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\,\Gamma \vdash \texttt{NC}\ e \ominus \texttt{NC}\ e \downarrow \square\,\tau \Rightarrow \Phi}$ **alg-r-nochange-**$\downarrow$

TS: $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\,\Gamma[\theta_a] \vdash \texttt{NC}\ |e| \ominus \texttt{NC}\ |e| : \square\,\tau[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma', \square\,\Gamma, \tau) \subseteq dom(\Delta, \psi_a)$ ($\star$) and

$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ ($\star\star$)

Using ($\star$), we can show that

a) $\text{FIV}(\Phi_a, \square\,\Gamma, \tau) \subseteq dom(\Delta, \psi_a)$.

By IH3 on the premise using a) and ($\star\star$), we can show that

$$\Delta; \Phi_a[\theta_a]; \square\,\Gamma[\theta_a] \vdash |e| \ominus |e| : \tau[\theta_a] \tag{4.14}$$

By the **c-nochange** rule using eq. (4.14), we obtain $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\,\Gamma[\theta_a] \vdash \texttt{NC}\ |e| \ominus \texttt{NC}\ |e| : \square\,\tau[\theta_a]$.

Case

$$\dfrac{\begin{array}{c} \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \mathbf{list}[n]^\alpha\,\tau \Rightarrow \Phi_e \qquad \Delta; \psi_a; n \doteq 0 \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau' \Rightarrow \Phi_1 \\ i :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \Phi_a; h : \square\,\tau, tl : \mathbf{list}[i]^\alpha\,\tau, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau' \Rightarrow \Phi_2 \\ i :: \mathbb{N}, \beta :: \mathbb{N}, \Delta; \psi_a; n \doteq i+1 \wedge \alpha \doteq \beta + 1 \wedge \Phi_a; h : \tau, tl : \mathbf{list}[i]^\beta\,\tau, \Gamma \vdash e_3 \ominus e_3' \downarrow \tau' \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_1) \wedge (\forall i :: \mathbb{N}.(n \doteq i+1) \to (\Phi_2 \wedge \forall \beta :: \mathbb{N}.(\alpha \doteq \beta+1) \to \Phi_3)) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \begin{array}{l} \mathbf{case}\ e\ \mathbf{of\ nil} \to e_1 \\ \mid h ::_{NC} tl \to e_2 \\ \mid h ::_C tl \to e_3 \end{array} \ominus \begin{array}{l} \mathbf{case}\ e'\ \mathbf{of\ nil} \to e_1' \\ \mid h ::_{NC} tl \to e_2' \\ \mid h ::_C tl \to e_3' \end{array} \downarrow \tau' \Rightarrow (\Phi_e \wedge \Phi_{body})}$$

alg-r-caseL-$\downarrow$

TS:

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \begin{array}{l} \text{case}\ e\ \text{of nil} \to |e_1| \\ \mid h ::_N tl \to |e_2| \\ \mid h ::_C tl \to |e_3| \end{array} \ominus \begin{array}{l} \text{case}\ e'\ \text{of nil} \to |e_1'| \\ \mid h ::_N tl \to |e_2'| \\ \mid h ::_C tl \to |e_3'| \end{array} : \tau'[\theta_a]$$

By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, \tau') \subseteq dom(\Delta, \psi_a)$ ($\star$) and

$\Delta; \Phi_a[\theta_a] \models (\Phi_e \wedge \Phi_{body})[\theta_a]$ ($\star\star$)

Using ($\star$), ($\star\star$)'s derivation must be in a form such that we have

a) $\Delta; \Phi_a[\theta_a] \models \Phi_e[\theta_a]$

b) $\Delta; n[\theta_a] \doteq 0 \wedge \Phi_a[\theta_a] \models \Phi_1[\theta_a]$

c) $i :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \Phi_a[\theta_a] \models \Phi_2[\theta_a]$

d) $i :: S, \beta :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \alpha[\theta_a] \doteq \beta + 1 \wedge \Phi_a[\theta_a] \models \Phi_3[\theta_a]$

By IH4 on the first premise using a) and ($\star$), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \text{list}[n[\theta_a]]^{\alpha[\theta_a]}\, \tau[\theta_a] \tag{4.15}$$

By IH3 on the second premise using b) and ($\star$), we can show that

$$\Delta; n[\theta_a] \doteq 0 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1| \ominus |e_1'| : \tau'[\theta_a] \tag{4.16}$$

By IH3 on the third premise using c) and ($\star$), we can show that

$$i :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_2| \ominus |e_2'| : \tau'[\theta_a] \tag{4.17}$$

By IH3 on the fourth premise using d) and ($\star$), we can show that

$$i :: S, \beta :: S, \Delta; n[\theta_a] \doteq i + 1 \wedge \alpha[\theta_a] \doteq \beta + 1 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_3| \ominus |e_3'| : \tau'[\theta_a] \tag{4.18}$$

Then by **c-r-caseL** rule using eqs. (4.15) to (4.18), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \begin{array}{l} \text{case } e \text{ of nil } \rightarrow |e_1| \\ \mid h ::_N tl \rightarrow |e_2| \\ \mid h ::_C tl \rightarrow |e_3| \end{array} \ominus \begin{array}{l} \text{case } e' \text{ of nil } \rightarrow |e_1'| \\ \mid h ::_N tl \rightarrow |e_2'| \\ \mid h ::_C tl \rightarrow |e_3'| \end{array} : \tau'[\theta_a]$$

Case $$\frac{\begin{array}{c} i \in \textbf{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \square\,\tau \Rightarrow \Phi_1 \\ \Delta; i, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \textbf{list}[i]^\alpha\, \tau \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \textbf{cons}_{NC}(e_1, e_2) \ominus \textbf{cons}_{NC}(e_1', e_2') \downarrow \textbf{list}[n]^\alpha\, \tau \Rightarrow \Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2'}\ \textbf{alg-r-consNC-}\!\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \text{cons}_{NC}(|e_1|, |e_2|) \ominus \text{cons}_{NC}(|e_1'|, |e_2'|) : \text{list}[n[\theta_a]]^{\alpha[\theta_a]}\, \tau[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, \text{list}[n]^\alpha\, \tau) \subseteq dom(\Delta, \psi_a)$ ($\star$) and
$\Delta; \Phi_a[\theta_a] \models (\Phi_1 \wedge \exists i :: \mathbb{N}.\Phi_2')[\theta_a]$ ($\star\star$)
Using ($\star$), ($\star\star$)'s derivation must be in a form such that we have

a) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a]$

b) $\Delta \vdash I :: \mathbb{N}$

c) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, i \mapsto I]$

d) $\Delta; \Phi_a[\theta_a] \models (I + 1) \doteq n[\theta_a]$

By IH3 on the second premise using ($\star$) and a), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1| \ominus |e_1'| : \square\, \tau[\theta_a] \tag{4.19}$$

By IH3 on the third premise using ($\star$) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_2| \ominus |e_2'| : \mathrm{list}[I]^{\alpha[\theta_a]}\, \tau[\theta_a] \tag{4.20}$$

By **c-r-cons2** typing rule using eqs. (4.19) and (4.20), we obtain
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \mathrm{cons}_{NC}(|e_1|, |e_2|) \ominus \mathrm{cons}_{NC}(|e_1'|, |e_2'|) : \mathrm{list}[I+1]^{\alpha[\theta_a]}\, \tau[\theta_a]$.
We conclude by applying **c-r-$\sqsubseteq$** rule to this using d) .

Proof of Theorem 34.4:

Case $$\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash \tau\ \texttt{wf} \qquad \mathbf{FIV}(\tau) \in \Delta}{\Delta; \psi_a; \Phi_a; \Gamma \vdash (e : \tau) \ominus (e' : \tau) \uparrow \tau \Rightarrow \Phi}\ \textbf{alg-r-anno-}\uparrow$$
TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |(e : \tau, t)| \ominus |(e' : \tau, t)| : \tau[\theta_a]$.
Since by definition, $\forall e.\ |(e : \_, \_)| = |e|$, STS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \tau[\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ ($\star$) and $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ ($\star\star$)
Using the third premise, we can show that

a) $\mathrm{FIV}(\Phi_a, \Gamma, \tau) \subseteq dom(\Delta, \psi_a)$.

By IH4 on the first premise using ($\star\star$) and a), we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \tau[\theta_a]$.

Case $$\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \square\, \tau \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \ominus \texttt{der}\ e_2 \uparrow \tau \Rightarrow \Phi}\ \textbf{alg-r-der-}\uparrow$$
TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \texttt{der}\ |e| \backsim \texttt{der}\ |e'| :^c \tau[\theta\, \theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ ($\star$) and
$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ ($\star\star$) such that $\Delta \ \triangleright\ \theta_a : \psi_a$ are derivable.
By IH4 on the first premise using ($\star$) and $\star\star$, we obtain

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| : \square\, \tau[\theta_a] \tag{4.21}$$

Then, by **c-der** rule using eq. (4.21), we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \texttt{der}\ |e| \ominus \texttt{der}\ |e'| : \tau[\theta\, \theta_a]$.

$\square$

Theorem 35 (Completeness of algorithmic typechecking in RelRefU)

1. Assume that $\Delta; \Phi_a; \Omega \vdash e :^c A$. Then, there exists e' such that $\Delta; \cdot; \Phi_a; \Omega \vdash e' \downarrow A$ and $\Delta; \Phi_a \models \Phi$ and $|e'| = e$.

2. Assume that $\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \tau$. Then, there exist e_1', e_2' such that $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$ and $|e_1'| = e_1$ and $|e_2'| = e_2$.

Proof. By simultaneous induction on the given Core typing derivations.

Proof of Theorem 35.1:

Case $\dfrac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash x :^c A}$ **c-var**

We can conclude as follows

$$\dfrac{\dfrac{\Omega(x) = A}{\Delta; \cdot; \Phi_a; \Omega \vdash x \uparrow A \Rightarrow \top}\textbf{alg-u-var-}\uparrow \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash x \downarrow A \Rightarrow \top}\textbf{alg-r-}\uparrow\downarrow$$

Case $\dfrac{\Delta; \Phi_a; \Omega \vdash e_1 :^c A \qquad \Delta; \Phi_a; \Omega \vdash e_2 :^c \textbf{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash \textbf{cons}_C(e_1, e_2) :^c \textbf{list}[n+1]\, A}$ **c-cons**

By IH1 on the first premise, $\exists e_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Omega \vdash e_1' \downarrow A \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|e_1'| = e_1$

By IH1 on the second premise, $\exists e_2'$ such that

d) $\Delta; \cdot; \Phi_a; \Omega \vdash e_2' \downarrow \mathrm{list}[n]\, A \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$

f) $|e_2'| = e_2$

Then, we can conclude as follows

1.

$$\dfrac{\begin{array}{c} i \in \mathrm{fresh}(\mathbb{N}) \qquad \Delta; \psi_a; \Phi_a; \Omega \vdash e_1' \downarrow A \Rightarrow \Phi_1' \\ \Delta; i, k_2', t_2', \psi_a; \Phi_a; \Omega \vdash e_2' \downarrow \mathrm{list}[i]\, A \Rightarrow \Phi_2' \qquad \Phi_2'' = (\Phi_2 \wedge n+1 \doteq (i+1) \wedge) \end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \mathrm{cons}_C(e_1', e_2') \downarrow \mathrm{list}[n+1]\, A \Rightarrow \Phi_1' \wedge \exists i :: \mathbb{N}.\Phi_2''}\textbf{alg-u-cons-}\downarrow$$

2. Using c) and f), $|\mathrm{cons}_C(e_1', e_2')| = \mathrm{cons}_C(e_1, e_2)$.

Proof of Theorem 35.2:

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \backsim e_2 :^c \square\tau}{\Delta; \Phi_a; \Gamma \vdash \mathtt{der}\ e_1 \backsim \mathtt{der}\ e_2 :^c \tau}$ **c-der**

By IH2 on the premise, $\exists e_1', e_2'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \tau, t \Rightarrow \Phi$

b) $\Delta; \Phi_a \models \Phi$

c) $|e'_1| = e_1$ and $|e'_2| = e_2$

Then, we can conclude by using a), b) and c) as follows:

$$\frac{\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \square\tau \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der}\ e'_1 \ominus \texttt{der}\ e'_2 \downarrow \tau \Rightarrow \Phi}\ \textbf{alg-r-der-}\downarrow$$ and

1.

$$\frac{\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \square\tau \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der}\ e'_1 \ominus \texttt{der}\ e'_2 \downarrow \tau \Rightarrow \Phi}\ \textbf{alg-r-der-}\downarrow}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e'_1 : \tau) \ominus (\texttt{der}\ e'_2 : \tau) \uparrow \tau \Rightarrow \Phi}\ \textbf{alg-r-anno-}\uparrow \quad \Delta; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi' \text{ by } \textit{Lemma } 8}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e'_1 : \tau) \ominus (\texttt{der}\ e'_2 : \tau) \downarrow \tau \Rightarrow \Phi \land \Phi'}\ \textbf{alg-r-}\uparrow\downarrow$$

2. By c), $|(\texttt{der}\ e'_i : \tau)| = \texttt{der}\ |e'_i|$.

3. By b) and Lemma 25.

Case $$\frac{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a; \Gamma \vdash e \backsim e' :^c \tau'}\ \textbf{c-r-}\equiv$$

By IH2 on the first premise, $\exists e'_1, e'_2$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \tau \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|e'_1| = e$ and $|e'_2| = e'$

By IH2 on the second premise,

d) $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$.

Then, we can conclude as follows

1. By using a) and d)

$$\frac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \tau \Rightarrow \Phi_1}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e'_1 : \tau) \ominus (e'_2 : \tau) \uparrow \tau \Rightarrow \Phi_1}\ \textbf{alg-r-anno-}\uparrow \qquad \Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e'_1 : \tau) \ominus (e'_2 : \tau) \downarrow \tau' \Rightarrow \Phi_1 \land \Phi_2}\ \textbf{alg-r-}\uparrow\downarrow$$

2. By using b), e) ,we can show that $\Delta; \Phi_a \models \Phi_1 \land \Phi_2$

3. By c), $|(e'_1 : \tau)| = e'_1$ and $(e'_2 : \tau) = e'_2$

$$\textbf{Case}\quad \frac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_1' :^c \tau_1 \to \tau_2 \\ \Delta;\Phi_a;\Gamma \vdash e_2 \backsim e_2' :^c \tau_1\end{array}}{\Delta;\Phi_a;\Gamma \vdash e_1\ e_2 \backsim e_1'\ e_2' :^c \tau_2}\ \textbf{c-r-app}$$

By IH2 on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta;\cdot;\Phi_a;\Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \to \tau_2 \Rightarrow \Phi_1$

b) $\Delta;\Phi_a \models \Phi_1$

c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By IH2 on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

d) $\Delta;\cdot;\Phi_a;\Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1 \Rightarrow \Phi_2$

e) $\Delta;\Phi_a \models \Phi_2$

f) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

Then, we can conclude as follows

1.

$$\frac{\dfrac{\dfrac{\Delta;\cdot;\Phi_a;\Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \to \tau_2 \Rightarrow \Phi_1}{\Delta;\cdot;\Phi_a;\Gamma \vdash (\overline{e}_1 : \tau_1 \to \tau_2) \ominus \overline{e}_1' : \tau_1 \to \tau_2) \uparrow \tau_1 \to \tau_2 \Rightarrow \Phi_1}\ \textbf{alg-r-anno-}\uparrow \quad \Delta;\cdot;\Phi_a;\Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1 \Rightarrow \Phi_2}{\Delta;\cdot;\Phi_a;\Gamma \vdash E_1 \ominus E_2 \uparrow \tau_2 \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{c-r-app}}{\Delta;\cdot;\Phi_a;\Gamma \vdash E_1 \ominus E_2 \downarrow \tau_2 \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-}\uparrow\downarrow$$

where $E_1 = (\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2$ and $E_2 = (\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'$.

2. By using b) and e) .

3. Using c) and f), $|(\overline{e}_1 : \tau_1 \to \tau_2)\ \overline{e}_2| = e_1\ e_2$ and $|(\overline{e}_1' : \tau_1 \to \tau_2)\ \overline{e}_2'| = e_1'\ e_2'$.

$$\textbf{Case}\quad \frac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e_1 \backsim e_1' :^c \exists i{::}S.\,\tau_1 \\ i :: S, \Delta;\Phi_a;x:\tau_1,\Gamma \vdash e_2 \backsim e_2' :^c \tau_2 \qquad i \notin FV(\Phi_a;\Gamma,\tau_2,t_2)\end{array}}{\Delta;\Phi_a;\Gamma \vdash \textbf{unpack}\ e_1\ \textbf{as}\ (x,i)\ \textbf{in}\ e_2 \backsim \textbf{unpack}\ e_1'\ \textbf{as}\ (x,i)\ \textbf{in}\ e_2' :^c \tau_2}\ \textbf{c-r-unpack1}$$

By IH2 on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta;\cdot;\Phi_a;\Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \exists i{::}S.\,\tau_1 \Rightarrow \Phi_1$

b) $\Delta;\Phi_a \models \Phi_1$

c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By IH2 on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

d) $i :: S, \Delta;\cdot;\Phi_a;x:\tau_1,\Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_2 \Rightarrow \Phi_2$

e) $i :: S, \Delta;\Phi_a \models \Phi_2$

f) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

Then, we can conclude as follows

1.

$$\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}'_1 \downarrow \exists i{::}S.\, \tau_1 \Rightarrow \Phi_1 \quad (a)}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E'_1 \uparrow \exists i{::}S.\, \tau_1 \Rightarrow \Phi_1}\ \textbf{alg-r-anno-}\!\uparrow \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}'_2 \downarrow \tau_2 \Rightarrow \Phi_2 \qquad \Phi' = \Phi_1 \wedge \Phi'_2}{\Delta; \cdot; \Phi_a; \Gamma \vdash \text{unpack } E_1 \text{ as } (x, i) \text{ in } \overline{e}_2 \ominus \text{unpack } E'_1 \text{ as } (x, i) \text{ in } \overline{e}'_2 \downarrow \tau_2 \Rightarrow \Phi'}\ \textbf{alg-r-unpack-}\!\downarrow$$

   where $E_1 = (\overline{e}_1 : \exists i{::}S.\, \tau_1)$ and $E_2 = (\overline{e}'_1 : \exists i{::}S.\, \tau_1)$

2. By using b) and e).

3. Using c) and f), $|\text{unpack } (\overline{e}_1 : \exists i{::}S.\, \tau_1) \text{ as } (x, i) \text{ in } \overline{e}_2| = \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2$ and $|\text{unpack } (\overline{e}'_1 : \exists i{::}S.\, \tau_1) \text{ as } (x, i) \text{ in } \overline{e}'_2| = \text{unpack } e'_1 \text{ as } (x, i) \text{ in } e'_2$.

$\square$

5 RelCost

This appendix considers the following additions to the main paper.

- The constrained type $C \supset \tau$, which is eliminated with the "celim e" construct.
- The type $U\,A$ is generalized to $U\,(A_1, A_2)$ (like in RelCost's appendix), allowing us to relate two expressions of two different unary types A_1 and A_2, respectively. As a result of this change, **switch** rule, $\rightarrow$ **exec** subtyping rule, and some of the asynchronous rules are also generalized.

We first present RelCost's syntax, typing and subtyping rules. Then, we introduce RelCostCore and the embedding of RelCost into RelCostCore. Finally, the bidirectional system BiRelCost is introduced.

In Section 6, we present our benchmark programs along with the results of the experimental evaluation.

We use some abbreviations throughout. STS stands for "suffices to show", TS stands for "to show", and RTS stands for "remains to show".

Relational types	τ	::=	$\text{unit} \mid \text{int} \mid \tau_1 \times \tau_2 \mid \tau_1 + \tau_2 \mid \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \mid \text{list}[n]^\alpha\, \tau \mid$ $\forall i \overset{\text{diff}(t)}{::} S.\, \tau \mid \exists i{::}S.\, \tau \mid U\,(A_1, A_2) \mid \square\, \tau \mid C \,\&\, \tau \mid C \supset \tau$
Unary types	A	::=	$\text{unit}_r \mid \text{int}_r \mid A_1 \times A_2 \mid A_1 + A_2 \mid A_1 \xrightarrow{\text{exec}(k,t)} A_2 \mid \text{list}[n]\, A \mid$ $\forall i \overset{\text{exec}(k,t)}{::} S.\, A \mid \exists i{::}S.\, A \mid C \,\&\, A \mid C \supset A$
Sorts	S	::=	$\mathbb{N} \mid \mathbb{R}$
Index terms	I, k, t, α	::=	$i \mid 0 \mid \infty \mid I + 1 \mid I_1 + I_2 \mid I_1 - I_2 \mid \frac{I_1}{I_2} \mid I_1 \cdot I_2 \mid \lceil I \rceil \mid \lfloor I \rfloor \mid$ $\texttt{log}_2(I) \mid I_1^{I_2} \mid \texttt{min}(I_1, I_2) \mid \texttt{max}(I_1, I_2) \mid \sum_{i=I_1}^{I_n} I$
Constraints	C	::=	$I_1 \doteq I_2 \mid I_1 < I_2 \mid \neg C \mid$
Constraint env.	Φ	::=	$\top \mid C \wedge \Phi$
Sort env.	Δ	::=	$\emptyset \mid \Delta, i :: S$
Unary type env.	Ω	::=	$\emptyset \mid \Omega, x : A$
Relational type env.	Γ	::=	$\emptyset \mid \Gamma, x : \tau$
Primitive env.	Υ	::=	$\emptyset \mid \Upsilon, \zeta : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \mid \Upsilon, \zeta : A_1 \xrightarrow{\text{exec}(k,t)} A_2$
RelCost typing judg.			$\Omega \vdash_k^t e : A$ $\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau$
RelCostCore typing judg.			$\Omega \vdash_k^t e :^c A$ $\Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \tau$

Figure 58: Syntax of types and contexts

Terms	e	::=	$x \mid \texttt{n} \mid \text{fix } f(x).e \mid e_1\ e_2 \mid \zeta\ e \mid \langle e_1, e_2 \rangle \mid \pi_1(e) \mid \pi_2(e) \mid$ $\text{inl } e \mid \text{inr } e \mid \text{case } (e, x.e_1, y.e_2) \mid \text{nil} \mid \text{cons}(e_1, e_2) \mid$ $\text{case } e \text{ of nil} \to e_1 \mid h :: tl \to e_2 \mid \Lambda.e \mid e[\,] \mid$ $\text{pack } e \mid \text{unpack } e_1 \text{ as } x \text{ in } e_2 \mid \text{let } x = e_1 \text{ in } e_2 \mid () \mid$ $\text{clet } e_1 \text{ as } x \text{ in } e_2 \mid \text{celim } e$
Values	v	::=	$\texttt{n} \mid \text{fix } f(x).v \mid \langle v_1, v_2 \rangle \mid \text{inl } v \mid \text{inr } v \mid \text{nil} \mid \text{cons}(v_1, v_2) \mid \Lambda e \mid \text{pack } v \mid ()$

Figure 59: Syntax of values and expressions in RelCost

$$
\begin{array}{llll}
\text{Terms} & e & ::= & x \mid \mathtt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid e_1\ e_2 \mid \zeta\ e \mid \langle e_1, e_2\rangle \mid \pi_1(e) \mid \pi_2(e) \mid \\
 & & & \text{inl } e \mid \text{inr } e \mid \ \text{case } (e, x.e_1, y.e_2) \mid \text{nil} \mid \text{cons}_{NC}(e_1, e_2) \mid \text{cons}_{C}(e_1, e_2) \mid \\
 & & & \ \text{case } e \text{ of nil} \to e_1 \mid h ::_{NC} tl \to e_2 \mid h ::_{C} tl \to e_3 \mid \Lambda i.e \mid e\,[I] \mid \\
 & & & \text{pack } e \text{ with } I \mid \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \mid \text{let } x = e_1 \text{ in } e_2 \mid () \mid \\
 & & & \text{clet } e_1 \text{ as } x \text{ in } e_2 \mid \text{celim } e \mid \mathtt{der}\ e \mid \mathtt{switch}\ e \mid \mathtt{NC}\ e \mid \mathtt{split}\ (e_1, e_2) \text{ with } C \mid \mathtt{contra}\ e \\
 & & & \\
\text{Values} & v & ::= & \mathtt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid \langle v_1, v_2\rangle \mid \text{inl } v \mid \text{inr } v \mid \text{nil} \\
 & & & \mid \text{cons}_{NC}(v_1, v_2) \mid \text{cons}_{C}(v_1, v_2) \mid \Lambda i.e \mid \text{pack } v \text{ with } I \mid ()
\end{array}
$$

Figure 60: Syntax of values and expressions in RelCostCore

$$
\begin{array}{llll}
\text{Terms} & e & ::= & x \mid \mathtt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid e_1\ e_2 \mid \zeta\ e \mid \langle e_1, e_2\rangle \mid \pi_1(e) \mid \pi_2(e) \mid \\
 & & & \text{inl } e \mid \text{inr } e \mid \ \text{case } (e, x.e_1, y.e_2) \mid \text{nil} \mid \text{cons}_{NC}(e_1, e_2) \mid \text{cons}_{C}(e_1, e_2) \mid \\
 & & & \ \text{case } e \text{ of nil} \to e_1 \mid h ::_{NC} tl \to e_2 \mid h ::_{C} tl \to e_3 \mid \Lambda i.e \mid e\,[I] \mid \\
 & & & \text{pack } e \text{ with } I \mid \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \mid \text{let } x = e_1 \text{ in } e_2 \mid () \mid \\
 & & & \text{clet } e_1 \text{ as } x \text{ in } e_2 \mid \text{celim } e \mid \mathtt{der}\ \mid \mathtt{switch}\ e \mid \mathtt{NC}\ e \mid \mathtt{split}\ (e_1, e_2) \text{ with } C \mid \\
 & & & \mathtt{contra}\ e \mid (e : \tau, t) \mid (e : A, k, t) \\
 & & & \\
\text{Values} & v & ::= & \mathtt{n} \mid \text{fix } f(x).e \mid \text{fix}_{NC}\ f(x).e \mid \langle v_1, v_2\rangle \mid \text{inl } v \mid \text{inr } v \mid \text{nil} \\
 & & & \mid \text{cons}_{NC}(v_1, v_2) \mid \text{cons}_{C}(v_1, v_2) \mid \Lambda i.e \mid \text{pack } v \text{ with } I \mid ()
\end{array}
$$

Figure 61: Syntax of values and expressions in BiRelCost

$\boxed{\Delta; \Phi_a \vdash^A A \ \texttt{wf}}$ checks well-formedness of the unary type A

$\boxed{\Delta; \Phi_a \vdash \tau \ \texttt{wf}}$ checks well-formedness of the binary type τ

$$\frac{}{\Delta; \Phi_a \vdash \text{unit}_r \ \texttt{wf}} \ \textbf{wf-unit} \qquad \frac{}{\Delta; \psi; \Phi_a \vdash \text{int}_r \ \texttt{wf}} \ \textbf{wf-int}$$

$$\frac{\Delta; \psi; \Phi_a \vdash \tau_1 \ \texttt{wf} \qquad \Delta; \psi; \Phi_a \vdash \tau_2 \ \texttt{wf}}{\Delta; \psi; \Phi_a \vdash \tau_1 \times \tau_2 \ \texttt{wf}} \ \textbf{wf-prod} \qquad \frac{\Delta; \psi; \Phi_a \vdash \tau_1 \ \texttt{wf} \qquad \Delta; \psi; \Phi_a \vdash \tau_2 \ \texttt{wf}}{\Delta; \psi; \Phi_a \vdash \tau_1 + \tau_2 \ \texttt{wf}} \ \textbf{wf-sum}$$

$$\frac{\Delta; \psi; \Phi_a \vdash \tau_1 \ \texttt{wf} \qquad \Delta; \psi; \Phi_a \vdash \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a \vdash t :: \mathbb{R}}{\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf}} \ \textbf{wf-fun}$$

$$\frac{\Delta, \Phi_a \vdash n :: \mathbb{N} \qquad \Delta, \Phi_a \vdash \alpha :: \mathbb{N} \qquad \Delta; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a \vdash \text{list}[n]^\alpha \, \tau \ \texttt{wf}} \ \textbf{wf-list}$$

$$\frac{i :: S, \Delta; \Phi_a \vdash \tau \ \texttt{wf} \qquad i :: S, \Delta; \Phi_a \vdash t :: \mathbb{R}}{\Delta; \Phi_a \vdash \forall i \overset{\text{diff}(t)}{::} S.\, \tau \ \texttt{wf}} \ \textbf{wf-}\forall \qquad \frac{i :: S, \Delta; \Phi_a \vdash \tau \ \texttt{wf}\Phi}{\Delta; \Phi_a \vdash \exists i{::}S.\, \tau \ \texttt{wf}\Phi} \ \textbf{wf-}\exists$$

$$\frac{\Delta; \psi; \Phi_a \vdash^A A_1 \ \texttt{wf} \qquad \Delta; \psi; \Phi_a \vdash^A A_2 \ \texttt{wf}}{\Delta; \psi; \Phi_a \vdash U\,(A_1, A_2) \ \texttt{wf}} \ \textbf{wf-}U \qquad \frac{\Delta; \psi; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \psi; \Phi_a \vdash \Box\, \tau \ \texttt{wf}} \ \textbf{wf-box}$$

$$\frac{\Delta; \Phi_a \vdash_\epsilon C \ \texttt{wf} \qquad \Delta; C \wedge \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a \vdash C \supset \tau \ \texttt{wf}} \ \textbf{wf-C}\supset \qquad \frac{\Delta; \Phi_a \vdash_\epsilon C \ \texttt{wf} \qquad \Delta; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a \vdash C \ \& \ \tau \ \texttt{wf}} \ \textbf{wf-C}\&$$

Figure 62: Well-formedness of binary types

$\boxed{\Delta \vdash^A A \ \mathtt{wf}}$ checks well-formedness of the unary type A Φ

$$\frac{}{\Delta; \Phi_a \vdash^A \text{unit } \mathtt{wf}} \ \textbf{wf-u-unit} \qquad \frac{}{\Delta; \psi; \Phi_a \vdash^A \text{int } \mathtt{wf}} \ \textbf{wf-u-int}$$

$$\frac{\Delta; \psi; \Phi_a \vdash^A A_1 \ \mathtt{wf} \qquad \Delta; \psi; \Phi_a \vdash^A A_2 \ \mathtt{wf}}{\Delta; \psi; \Phi_a \vdash^A A_1 \times A_2 \ \mathtt{wf}} \ \textbf{wf-u-prod}$$

$$\frac{\Delta; \psi; \Phi_a \vdash^A A_1 \ \mathtt{wf} \qquad \Delta; \psi; \Phi_a \vdash^A A_2 \ \mathtt{wf}}{\Delta; \psi; \Phi_a \vdash^A A_1 + A_2 \ \mathtt{wf}} \ \textbf{wf-u-sum}$$

$$\frac{\Delta; \psi; \Phi_a \vdash^A A_1 \ \mathtt{wf} \qquad \Delta; \psi; \Phi_a \vdash^A A_2 \ \mathtt{wf} \qquad \Delta; \Phi_a \vdash k :: \mathbb{R} \qquad \Delta; \Phi_a \vdash t :: \mathbb{R}}{\Delta; \Phi_a \vdash^A A_1 \xrightarrow{\text{exec}(k,t)} A_2 \ \mathtt{wf}} \ \textbf{wf-u-fun}$$

$$\frac{\Delta, \Phi_a \vdash n :: \mathbb{N} \qquad \Delta; \Phi_a \vdash^A A \ \mathtt{wf}}{\Delta; \Phi_a \vdash^A \text{list}[n] \, A \ \mathtt{wf}} \ \textbf{wf-u-list}$$

$$\frac{i :: S, \Delta; \Phi_a \vdash^A A \ \mathtt{wf} \qquad i :: S, \Delta; \Phi_a \vdash k :: \mathbb{R} \qquad i :: S, \Delta; \Phi_a \vdash t :: \mathbb{R}}{\Delta; \Phi_a \vdash^A \forall i \overset{\text{exec}(k,t)}{::} S. \, A \ \mathtt{wf}} \ \textbf{wf-u-}\forall$$

$$\frac{i :: S, \Delta; \Phi_a \vdash^A A \ \mathtt{wf}}{\Delta; \Phi_a \vdash^A \exists i{::}S. \, A \ \mathtt{wf}} \ \textbf{wf-u-}\exists \qquad \frac{\Delta; \Phi_a \vdash_\epsilon C \ \mathtt{wf} \qquad \Delta; C \wedge \Phi_a \vdash^A A \ \mathtt{wf}}{\Delta; \Phi_a \vdash^A C \supset A \ \mathtt{wf}} \ \textbf{wf-u-C}\rangle$$

$$\frac{\Delta; \Phi_a \vdash_\epsilon C \ \mathtt{wf} \qquad \Delta; \Phi_a \vdash^A A \ \mathtt{wf}}{\Delta; \Phi_a \vdash^A C \ \& \ A \ \mathtt{wf}} \ \textbf{wf-u-C}\wedge$$

Figure 63: Well-formedness of unary types

$$\frac{}{\Delta \ \triangleright \ [] : \cdot} \qquad \frac{\Delta \vdash I :: S \qquad \Delta \ \triangleright \ \theta : \psi}{\Delta \ \triangleright \ \theta[M \mapsto I] : M :: S, \psi}$$

Figure 64: Sorting of Substitutions

$\boxed{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2}$ Binary type τ_1 is a subtype of relational type τ_2

$\boxed{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2}$ Unary type A_1 is a subtype of type A_2

$$\frac{}{\Delta; \Phi_a \models \mathrm{int}_r \sqsubseteq \square\, \mathrm{int}_r}\ \textbf{int-}\square \qquad \frac{}{\Delta; \Phi_a \models \square\, U\,(\mathrm{int}, \mathrm{int}) \sqsubseteq \mathrm{int}_r}\ \square\textbf{U-int}$$

$$\frac{}{\Delta; \Phi_a \models \mathrm{unit}_r \sqsubseteq \square\, \mathrm{unit}_r}\ \textbf{unit} \qquad \frac{\Delta; \Phi_a \models \tau_1' \sqsubseteq \tau_1 \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2' \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a \models \tau_1 \xrightarrow{\mathrm{diff}(t)} \tau_2 \sqsubseteq \tau_1' \xrightarrow{\mathrm{diff}(t')} \tau_2'}\ \rightarrow \textbf{diff}$$

$$\frac{}{\Delta; \Phi_a \models \square\,(\tau_1 \xrightarrow{\mathrm{diff}(t)} \tau_2) \sqsubseteq \square\, \tau_1 \xrightarrow{\mathrm{diff}(0)} \square\, \tau_2}\ \rightarrow \square\textbf{diff}$$

$$\frac{}{\Delta; \Phi_a \models U\,(A_1 \xrightarrow{\mathrm{exec}(k,t)} A_2) \sqsubseteq U\, A_1 \xrightarrow{\mathrm{diff}(t-k)} U\, A_2}\ \rightarrow \textbf{execdiff}$$

$$\frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i :: S, \Delta; \Phi_a \models t \leq t' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \forall i \overset{\mathrm{diff}(t)}{::} S.\, \tau \sqsubseteq \forall i \overset{\mathrm{diff}(t')}{::} S.\, \tau'}\ \forall\textbf{diff}$$

$$\frac{}{\Delta; \Phi_a \models \square\,(\forall i \overset{\mathrm{diff}(t)}{::} S.\, \tau) \sqsubseteq \forall i \overset{\mathrm{diff}(0)}{::} S.\, \square\, \tau}\ \forall\ \square$$

$$\frac{}{\Delta; \Phi_a \models U\,(\forall i \overset{\mathrm{exec}(k,t)}{::} S.\, A, \forall i \overset{\mathrm{exec}('k,t')}{::} S.\, A') \sqsubseteq \forall i \overset{\mathrm{diff}(t-k')}{::} S.\, U\,(A, A')}\ \forall\textbf{U}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2'}{\Delta; \Phi_a \models \tau_1 \times \tau_2 \sqsubseteq \tau_1' \times \tau_2'}\ \times \qquad \frac{}{\Delta; \Phi_a \models \square\, \tau_1 \times \square\, \tau_2 \equiv \square\,(\tau_1 \times \tau_2)}\ \times\square$$

$$\frac{}{\Delta; \Phi_a \models U\,(A_1 \times A_2, A_1' \times A_2') \sqsubseteq U\,(A_1, A_1') \times U\,(A_2, A_2')}\ \times\textbf{U}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2'}{\Delta; \Phi_a \models \tau_1 + \tau_2 \sqsubseteq \tau_1' + \tau_2'}\ + \qquad \frac{}{\Delta; \Phi_a \models \square\, \tau_1 + \square\, \tau_2 \sqsubseteq \square\,(\tau_1 + \tau_2)}\ +\square$$

$$\frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \leq \alpha' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \sqsubseteq \mathrm{list}[n']^{\alpha'}\, \tau'}\ \textbf{l1}$$

$$\frac{\Delta; \Phi_a \models \alpha \doteq 0}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \sqsubseteq \mathrm{list}[n]^{\alpha}\, \square\, \tau}\ \textbf{l2} \qquad \frac{}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \square\, \tau \sqsubseteq \square\,(\mathrm{list}[n]^{\alpha}\, \tau)}\ \textbf{l}\square$$

Figure 65: RelCost subtyping rules (part 1)

$\boxed{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2}$ Binary type τ_1 is a subtype of type τ_2

$$\frac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \sqsubseteq \exists i{::}S.\, \tau'}\ \exists \qquad \frac{}{\Delta; \Phi_a \models \exists i{::}S.\, \Box\,\tau \sqsubseteq \Box\,(\exists i{::}S.\, \tau)}\ \exists\Box$$

$$\frac{\Delta; \Phi_a \wedge C \models C' \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C \,\&\, \tau \sqsubseteq C' \,\&\, \tau'}\ \textbf{c-and} \qquad \frac{}{\Delta; \Phi_a \models C \,\&\, \Box\,\tau \sqsubseteq \Box\,(C \,\&\, \tau)}\ \textbf{c-and-}\Box$$

$$\frac{\Delta; \Phi_a \wedge C' \models C \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau'}{\Delta; \Phi_a \models C \supset \tau \sqsubseteq C' \supset \tau'}\ \textbf{c-impl} \qquad \frac{}{\Delta; \Phi_a \models \Box\,(C \supset \tau) \sqsubseteq C \supset \Box\,\tau}\ \textbf{c-impl-}\Box$$

$$\frac{}{\Delta; \Phi_a \models \Box\,\tau \sqsubseteq \tau}\ \textbf{T} \qquad \frac{}{\Delta; \Phi_a \models \Box\,\tau \sqsubseteq \Box\,\Box\,\tau}\ \textbf{D} \qquad \frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2}{\Delta; \Phi_a \models \Box\,\tau_1 \sqsubseteq \Box\,\tau_2}\ \textbf{B-}\Box$$

$$\frac{}{\Delta; \Phi_a \models \tau \sqsubseteq U\,|\tau|}\ \textbf{W} \qquad \frac{\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models U\,A \sqsubseteq U\,A'}\ \textbf{U} \qquad \frac{}{\Delta; \Phi_a \models \tau \sqsubseteq \tau}\ \textbf{refl}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_3}{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_3}\ \textbf{trans}$$

Figure 66: $\mathsf{RelCost}$ subtyping rules (Part 2)

$\boxed{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2}$ Unary type A_1 is a subtype of type A_2

$$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \qquad \Delta; \Phi_a \models k' \leq k \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \xrightarrow{\text{exec}(k,t)} A_2 \sqsubseteq A_1' \xrightarrow{\text{exec}(k',t')} A_2'} \rightarrow \textbf{exec}$$

$$\frac{i :: S, \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad i :: S, \Delta; \Phi_a \models k' \leq k \qquad i :: S, \Delta; \Phi_a \models t \leq t' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models^{\mathsf{A}} \forall i \overset{\text{exec}(k,t)}{::} S.\, A \sqsubseteq \forall i \overset{\text{exec}(k',t')}{::} S.\, A} \textbf{u-}\forall\textbf{exec}$$

$$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \times A_2 \sqsubseteq A_1' \times A_2'} \textbf{u-}\times \qquad \frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2'}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 + A_2 \sqsubseteq A_1' + A_2'} \textbf{u-+}$$

$$\frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} \text{list}[n]\, A \sqsubseteq \text{list}[n']\, A'} \textbf{u-l} \qquad \frac{i :: S, \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models^{\mathsf{A}} \exists i{::}S.\, A \sqsubseteq \exists i{::}S.\, A'} \textbf{u-}\exists$$

$$\frac{\Delta; \Phi_a \wedge C \models C' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} C \,\&\, A \sqsubseteq C' \,\&\, A'} \textbf{u-c-and} \qquad \frac{\Delta; \Phi_a \wedge C' \models C \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A'}{\Delta; \Phi_a \models^{\mathsf{A}} C \supset A \sqsubseteq C' \supset A'} \textbf{u-c-impl}$$

$$\frac{}{\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A} \textbf{u-refl} \qquad \frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_3}{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_3} \textbf{u-tran}$$

Figure 67: RelCost unary subtyping rules

$\boxed{\text{General rules}}$

$$\frac{\Delta;\Phi_a;|\Gamma|\vdash_{k_1}^{t_1} e_1 : A \qquad \Delta;\Phi_a;|\Gamma|\vdash_{k_2}^{t_2} e_2 : A}{\Delta;\Phi_a;\Gamma\vdash e_1\ominus e_2 \lesssim t_1-k_2 : U\,A}\ \textbf{switch}
\qquad
\frac{\Delta;\Phi_a;\Gamma\vdash e\ominus e\lesssim t:\tau \qquad \forall x\in dom(\Gamma).\ \ \Delta;\Phi_a\models\Gamma(x)\sqsubseteq\Box\Gamma(x)}{\Delta;\Phi_a;\Gamma,\Gamma';\Omega\vdash e\ominus e\lesssim 0:\Box\tau}\ \textbf{nochange}$$

$\boxed{\text{Constant integers and unit}}$

$$\frac{}{\Delta;\Phi_a;\Omega\vdash_0^0 \mathtt{n}:\mathrm{int}}\ \textbf{const}
\qquad
\frac{}{\Delta;\Phi_a;\Gamma\vdash \mathtt{n}\ominus\mathtt{n}\lesssim 0:\mathrm{int}_r}\ \textbf{r-const}
\qquad
\frac{}{\Delta;\Phi_a;\Omega\vdash_0^0 ():\mathrm{unit}}\ \textbf{unit}$$

$$\frac{}{\Delta;\Phi_a;\Gamma\vdash ()\ominus()\lesssim 0:\mathrm{unit}_r}\ \textbf{r-unit}$$

$\boxed{\text{Variables } x}$

$$\frac{\Omega(x)=A}{\Delta;\Phi_a;\Omega\vdash_0^0 x:A}\ \textbf{var}
\qquad
\frac{\Gamma(x)=\tau}{\Delta;\Phi_a;\Gamma\vdash x\ominus x\lesssim 0:\tau}\ \textbf{r-var}$$

$\boxed{\mathrm{inl}\ e}$

$$\frac{\Delta;\Phi_a;\Omega\vdash_k^t e:A_1 \qquad \Delta;\Phi_a\vdash^{\mathsf{A}} A_2\ \mathtt{wf}}{\Delta;\Phi_a;\Omega\vdash_k^t \mathrm{inl}\ e:A_1+A_2}\ \textbf{inl}
\qquad
\frac{\Delta;\Phi_a;\Gamma\vdash e\ominus e'\lesssim t:\tau_1 \qquad \Delta;\Phi_a\vdash\tau_2\ \mathtt{wf}}{\Delta;\Phi_a;\Gamma\vdash \mathrm{inl}\ e\ominus\mathrm{inl}\ e'\lesssim t:\tau_1+\tau_2}\ \textbf{r-inl}$$

$\boxed{\mathrm{inr}\ e}$

$$\frac{\Delta;\Phi_a;\Omega\vdash_k^t e:A_2 \qquad \Delta;\Phi_a\vdash^{\mathsf{A}} A_1\ \mathtt{wf}}{\Delta;\Phi_a;\Omega\vdash_k^t \mathrm{inr}\ e:A_1+A_2}\ \textbf{inr}
\qquad
\frac{\Delta;\Phi_a;\Gamma\vdash e\ominus e'\lesssim t:\tau_2 \qquad \Delta;\Phi_a\vdash\tau_1\ \mathtt{wf}}{\Delta;\Phi_a;\Gamma\vdash \mathrm{inr}\ e\ominus\mathrm{inr}\ e'\lesssim t:\tau_1+\tau_2}\ \textbf{r-inr}$$

$\boxed{\mathrm{case}\ (e, x.e_1, y.e_2)}$

$$\frac{\Delta;\Phi_a;\Omega\vdash_k^t e:A_1+A_2 \qquad \Delta;\Phi_a;x:A_1,\Omega\vdash_{k'}^{t'} e_1:A \qquad \Delta;\Phi_a;y:A_2,\Omega\vdash_{k'}^{t'} e_2:A}{\Delta;\Phi_a;\Omega\vdash_{k+k'+c_{case}}^{t+t'+c_{case}}\ \mathrm{case}\ (e,x.e_1,y.e_2):A}\ \textbf{case}$$

$$\frac{\Delta;\Phi_a;\Gamma\vdash e\ominus e'\lesssim t:\tau_1+\tau_2 \qquad \Delta;\Phi_a;x:\tau_1,\Gamma\vdash e_1\ominus e_1'\lesssim t':\tau \qquad \Delta;\Phi_a;y:\tau_2,\Gamma\vdash e_2\ominus e_2'\lesssim t':\tau}{\Delta;\Phi_a;\Gamma\vdash\ \mathrm{case}\ (e,x.e_1,y.e_2)\ominus\ \mathrm{case}\ (e',x.e_1',y.e_2')\lesssim t+t':\tau}\ \textbf{r-case}$$

Figure 68: RelCost typing rules (Part 1)

$\boxed{\text{fix } f(x).e}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \xrightarrow{\text{exec}(k,t)} A_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \xrightarrow{\text{exec}(k,t)} A_2, \Omega \vdash_k^t e : A_2}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{fix } f(x).e : A_1 \xrightarrow{\text{exec}(k,t)} A_2} \ \textbf{fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, \Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \ominus \text{fix } f(x).e_2 \lesssim 0 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2} \ \textbf{r-fix}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \\ \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2), \Gamma \vdash e \ominus e \lesssim t : \tau_2 \qquad \forall x \in dom(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\,\Gamma(x)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \ominus \text{fix } f(x).e \lesssim 0 : \square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2)} \ \textbf{r-fixNC}$$

$\boxed{e_1\ e_2}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 : A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 : A_1}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2+k+c_{app}}^{t_1+t_2+t+c_{app}} e_1\ e_2 : A_2} \ \textbf{app}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \\ \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_1\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \lesssim t_1 + t_2 + t : \tau_2} \ \textbf{r-app}$$

$\boxed{\langle e_1, e_2 \rangle}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 : A_1 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 : A_2}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \langle e_1, e_2 \rangle : A_1 \times A_2} \ \textbf{prod}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \tau_1 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \langle e_1, e_2 \rangle \ominus \langle e_1', e_2' \rangle \lesssim t_1 + t_2 : \tau_1 \times \tau_2} \ \textbf{r-prod}$$

$\boxed{\pi_1(e)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e : A_1 \times A_2}{\Delta; \Phi_a; \Omega \vdash_{k+c_{proj}}^{t+c_{proj}} \pi_1(e) : A_1} \ \textbf{proj1} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau_1 \times \tau_2}{\Delta; \Phi_a; \Gamma \vdash \pi_1(e) \ominus \pi_1(e') \lesssim t : \tau_1} \ \textbf{r-proj1}$$

$\boxed{\pi_2(e)}$

Symmetric rules.

Figure 69: RelCost typing rules (Part 2)

$\boxed{\text{nil}}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A \ \mathtt{wf}}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{nil} : \text{list}[0]\, A}\ \textbf{nil} \qquad \frac{\Delta; \Phi_a \vdash \tau \ \mathtt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{nil} \ominus \text{nil} \lesssim 0 : \text{list}[0]^{\alpha}\, \tau}\ \textbf{r-nil}$$

$\boxed{\text{cons}(e_1, e_2)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 : A \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 : \text{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \text{cons}(e_1, e_2) : \text{list}[n+1]\, A}\ \textbf{cons}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \text{list}[n]^{\alpha}\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \ominus \text{cons}(e_1', e_2') \lesssim t_1 + t_2 : \text{list}[n+1]^{\alpha+1}\, \tau}\ \textbf{r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \square\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \text{list}[n]^{\alpha}\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \ominus \text{cons}(e_1', e_2') \lesssim t_1 + t_2 : \text{list}[n+1]^{\alpha}\, \tau}\ \textbf{r-cons2}$$

$\boxed{\text{case } e \text{ of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2}$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash_k^t e : \text{list}[n]\, A \\ \Delta; \Phi_a \wedge n = 0; \Omega \vdash_{k'}^{t'} e_1 : A' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : A, tl : \text{list}[i]\, A, \Omega \vdash_{k'}^{t'} e_2 : A'\end{array}}{\Delta; \Phi_a; \Omega \vdash_{k+k'+c_{caseL}}^{t+t'+c_{caseL}} \text{case } e \text{ of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2 : A'}\ \textbf{caseL}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \text{list}[n]^{\alpha}\, \tau \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \lesssim t' : \tau' \\ i, \Delta; \Phi_a \wedge n = i+1; h : \square\, \tau, tl : \text{list}[i]^{\alpha}\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h : \tau, tl : \text{list}[i]^{\beta}\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{ case } e \text{ of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2 \ominus \text{ case } e' \text{ of nil} \rightarrow e_1' \mid h :: tl \rightarrow e_2' \lesssim t + t' : \tau'}\ \textbf{r-caseL}$$

$\boxed{\text{leaf}}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A \ \mathtt{wf}}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{leaf} : \text{tree}[0]\, A}\ \textbf{leaf} \qquad \frac{\Delta; \Phi_a \vdash \tau \ \mathtt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{leaf} \ominus \text{leaf} \lesssim 0 : \text{tree}[0]^{\alpha}\, \tau}\ \textbf{r-leaf}$$

$\boxed{\text{node}(e_l, e, e_r)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e : A \qquad \Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_l : \text{tree}[i]\, A \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_r : \text{tree}[j]\, A}{\Delta; \Phi_a; \Omega \vdash_{k+k_1+k_2}^{t+t_1+t_2} \text{node}(e_l, e, e_r) : \text{tree}[i+j+1]\, A}\ \textbf{node}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau \\ \Delta; \Phi_a; \Gamma \vdash e_l \ominus e_l' \lesssim t_1 : \text{tree}[i]^{\alpha}\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_r \ominus e_r' \lesssim t_2 : \text{tree}[j]^{\beta}\, \tau\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{node}(e_l, e, e_r) \ominus \text{node}(e_l', e', e_r') \lesssim t + t_1 + t_2 : \text{tree}[i+j+1]^{\alpha+\beta+1}\, \tau}\ \textbf{r-node1}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \square\, \tau \\ \Delta; \Phi_a; \Gamma \vdash e_l \ominus e_l' \lesssim t_1 : \text{tree}[i]^{\alpha}\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_r \ominus e_r' \lesssim t_2 : \text{tree}[j]^{\beta}\, \tau\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{node}(e_l, e, e_r) \ominus \text{node}(e_l', e', e_r') \lesssim t + t_1 + t_2 : \text{tree}[i+j+1]^{\alpha+\beta}\, \tau}\ \textbf{r-node2}$$

Figure 70: RelCost typing rules (Part 3)

$\boxed{\text{case } e \text{ of leaf} \to e_1 \mid \text{node}(l,x,r) \to e_2}$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash_k^t e : \text{tree}[n]\, A \qquad \Delta; \Phi_a \wedge n = 0; \Omega \vdash_{k'}^{t'} e_1 : A' \\ i, j, \Delta; \Phi_a \wedge n = i + j + 1; x : A, l : \text{tree}[i]\, A, r : \text{tree}[j]\, A, \Omega \vdash_{k'}^{t'} e_2 : A'\end{array}}{\Delta; \Phi_a; \Omega \vdash_{k+k'+c_{caseT}}^{t+t'+c_{caseT}} \text{ case } e \text{ of leaf} \to e_1 \mid \text{node}(l,x,r) \to e_2 : A'} \textbf{ caseT}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \text{tree}[n]^{\alpha}\, \tau \qquad \Delta; \Phi_a \wedge n = 0 \wedge; \Gamma \vdash e_1 \ominus e_1' \lesssim t' : \tau' \\ i, j, \beta, \theta, \Delta; \Phi_a \wedge n = i + j + 1 \wedge \alpha = \beta + \theta; x : \square\, \tau, l : \text{tree}[i]^{\beta}\, \tau, r : \text{tree}[j]^{\theta}\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau' \\ i, j, \beta, \theta, \Delta; \Phi_a \wedge n = i + j + 1 \wedge \alpha = \beta + \theta + 1; x : \tau, l : \text{tree}[i]^{\beta}\, \tau, r : \text{tree}[j]^{\theta}\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{ case } e \text{ of leaf} \to e_1 \mid \text{node}(l,x,r) \to e_2 \ominus \text{ case } e' \text{ of leaf} \to e_1' \mid \text{node}(l,x,r) \to \lesssim t + t' : e_2'\, \tau'} \textbf{ r-caseT}$$

$\boxed{\Lambda e}$

$$\frac{i :: S, \Delta; \Phi_a; \Omega \vdash_k^t e : A \qquad i \notin \text{FIV}(\Phi_a; \Omega)}{\Delta; \Phi_a; \Omega \vdash_0^0 \Lambda e : \forall i \overset{\text{exec}(k,t)}{::} S.\, A} \textbf{ iLam}$$

$$\frac{\begin{array}{c} i :: S, \Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau \\ i \notin \text{FIV}(\Phi_a; \Gamma)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \Lambda e \ominus \Lambda e' \lesssim 0 : \forall i \overset{\text{diff}(t)}{::} S.\, \tau} \textbf{ r-iLam}$$

$\boxed{e[\,]}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e : \forall i \overset{\text{exec}(k',t')}{::} S.\, A \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Omega \vdash_{k+k'[I/i]}^{t+t'[I/i]} e[\,]\ : A\{I/i\}} \textbf{ iApp}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \forall i \overset{\text{diff}(t')}{::} S.\, \tau \\ \Delta \vdash I : S\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \ominus e'[\,] \lesssim t + t'[I/i] : \tau\{I/i\}} \textbf{ r-iApp}$$

$\boxed{\text{pack } e}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e : A\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Omega \vdash_k^t \text{pack } e : \exists i{::}S.\, A} \textbf{ pack}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \ominus \text{pack } e' \lesssim t : \exists i{::}S.\, \tau} \textbf{ r-pack}$$

$\boxed{\text{unpack } e \text{ as } x \text{ in } e'}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 : \exists i{::}S.\, A_1 \qquad i :: S, \Delta; \Phi_a; x : A_1, \Omega \vdash_{k_2}^{t_2} e_2 : A_2 \qquad i \notin FV(\Phi_a; \Gamma, A_2, k_2, t_2)}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \text{unpack } e_1 \text{ as } x \text{ in } e_2 : A_2} \textbf{ unpack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \exists i{::}S.\, \tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } x \text{ in } e_2' \lesssim t_1 + t_2 : \tau_2} \textbf{ r-unpack1}$$

Figure 71: Typing rules (Part 4)

Primitive application

$$\frac{\Upsilon(\zeta) = A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e : A_1}{\Delta; \Phi_a; \Omega \vdash^{t+t'+c_{app}}_{k+k'+c_{app}} \zeta\ e : A_2}\ \textbf{primapp}$$

$$\frac{\Upsilon(\zeta) = \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t' : \tau_1}{\Delta; \Phi_a; \Gamma \vdash \zeta\ e \ominus \zeta\ e' \lesssim t + t' : \tau_2}\ \textbf{r-primapp}$$

$C \,\&\, \tau$ intro. rules

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Omega \vdash^{t}_{k} e : A}{\Delta; \Phi_a; \Omega \vdash^{t}_{k} e : C \,\&\, A}\ \textbf{c-andI}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \lesssim t : \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : C \,\&\, \tau}\ \textbf{c-andI}$$

$C \,\&\, \tau$ elim. rules

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t_1}_{k_1} e_1 : C \,\&\, A_1 \qquad \Delta; \Phi_a \wedge C; x : A_1, \Omega \vdash^{t_2}_{k_2} e_2 : A_2}{\Delta; \Phi_a; \Omega \vdash^{t_1+t_2}_{k_1+k_2} \text{clet}\ e_1\ \text{as}\ x\ \text{in}\ e_2 : A_2}\ \textbf{c-andE}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : C \,\&\, \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet}\ e_1\ \text{as}\ x\ \text{in}\ e_2 \ominus \text{clet}\ e_1'\ \text{as}\ x\ \text{in}\ e_2' \lesssim t_1 + t_2 : \tau_2}\ \textbf{r-c-andE}$$

$C \supset \tau$ intro. rules

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash^{t}_{k} e : A}{\Delta; \Phi_a; \Omega \vdash^{t}_{k} e : C \supset A}\ \textbf{c-impI}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \lesssim t : \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : C \supset \tau}\ \textbf{r-c-impI}$$

$C \supset \tau$ elim. rules

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t}_{k} e : C \supset A \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Omega \vdash^{t}_{k} \text{celim}\ e : A}\ \textbf{c-implE}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim}\ e \ominus \text{celim}\ e' \lesssim t : \tau}\ \textbf{r-c-implE}$$

let $x = e_1$ in e_2

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t_1}_{k_1} e_1 : A_1 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash^{t_2}_{k_2} e_2 : A_2}{\Delta; \Phi_a; \Omega \vdash^{t_1+t_2+c_{let}}_{k_1+k_2+c_{let}} \text{let}\ x = e_1\ \text{in}\ e_2 : A_2}\ \textbf{let}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let}\ x = e_1\ \text{in}\ e_2 \ominus \text{let}\ x = e_1'\ \text{in}\ e_2' \lesssim t_1 + t_2 : \tau_2}\ \textbf{r-let1}$$

$\boxed{\text{Subtyping}}$

$$\frac{\Delta;\Phi_a;\Omega \vdash^{t}_{k} e : A \qquad \Delta;\Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad \Delta;\Phi_a \models k' \leq k \qquad \Delta;\Phi_a \models t \leq t'}{\Delta;\Phi_a;\Omega \vdash^{t'}_{k'} e : A'}\ \sqsubseteq_{\text{exec}}$$

$$\frac{\begin{array}{c}\Delta;\Phi_a;\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau \qquad \Delta;\Phi_a \models \tau \sqsubseteq \tau' \\ \Delta;\Phi_a \models t \leq t'\end{array}}{\Delta;\Phi_a;\Gamma \vdash e_1 \ominus e_2 \lesssim t' : \tau'}\ \textbf{r-}\sqsubseteq$$

$\boxed{\text{Constraint dependent typing}}$

$$\frac{\Delta;\Phi_a \wedge C;\Gamma \vdash^{t}_{k} e : A \qquad \Delta;\Phi_a \wedge \neg C;\Gamma \vdash^{t}_{k} e : A \qquad \Delta \vdash C \ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash^{t}_{k} e : A}\ \textbf{split}$$

$$\frac{\Delta;\Phi_a \wedge C;\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau \qquad \Delta;\Phi_a \wedge \neg C;\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau \qquad \Delta \vdash C \ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau}\ \textbf{r-split}$$

$$\frac{\Delta;\Phi_a \models \bot \qquad \Delta;\Phi_a \vdash \Omega \ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash^{t}_{k} e : A}\ \textbf{contra} \qquad \frac{\Delta;\Phi_a \models \bot \qquad \Delta;\Phi_a \vdash \Gamma \ \mathtt{wf}}{\Delta;\Phi_a;\Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau}\ \textbf{r-contra}$$

$\boxed{\text{Asynchronous typing}}$

$$\frac{\begin{array}{c}\Delta;\Phi_a;|\Gamma| \vdash^{t_1}_{k_1} e_1 : A_1 \\ \Delta;\Phi_a;x : U\,A_1,\Gamma \vdash e_2 \ominus e \lesssim t_2 : \tau_2\end{array}}{\Delta;\Phi_a;\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus e \lesssim t_1 + t_2 + c_{let} : \tau_2}\ \textbf{r-let-e}$$

$$\frac{\Delta;\Phi_a;|\Gamma|_2 \vdash^{t_1}_{k_1} e_1 : A_1 \qquad \Delta;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e \ominus e_2 \lesssim t_2 : \tau_2}{\Delta;\Phi_a;\Gamma \vdash e \ominus \text{let } x = e_1 \text{ in } e_2 \lesssim t_2 - k_1 - c_{let} : \tau_2}\ \textbf{r-e-let}$$

$$\frac{\Delta;\Phi_a;|\Gamma|_1 \vdash^{t_1}_{k_1} e_1 : A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : U\,(A_1,A_2')}{\Delta;\Phi_a;\Gamma \vdash e_1\ e_2 \ominus e_2' \lesssim t_1 + t_2 + t + c_{app} : U\,(A_2,A_2')}\ \textbf{r-app-e}$$

$$\frac{\Delta;\Phi_a;|\Gamma|_2 \vdash^{t_1}_{k_1} e_1' : A_1' \xrightarrow{\text{exec}(k,t)} A_2' \qquad \Delta;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : U\,(A_2,A_1')}{\Delta;\Phi_a;\Gamma \vdash e_2 \ominus e_1'\ e_2' \lesssim t_2 - k_1 - k - c_{app} : U\,(A_2,A_2')}\ \textbf{r-e-app}$$

$$\frac{\begin{array}{c}\Delta;\Phi_a;|\Gamma|_1 \vdash^{t}_{-} e : A_1 + A_2 \\ \Delta;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e_1 \ominus e' \lesssim t' : \tau \qquad \Delta;\Phi_a;y : U\,(A_2,A_2),\Gamma \vdash e_2 \ominus e' \lesssim t' : \tau\end{array}}{\Delta;\Phi_a;\Gamma \vdash \text{ case } (e, x.e_1, y.e_2) \ominus e' \lesssim t' + t + c_{case} : \tau}\ \textbf{r-case-e}$$

$$\frac{\begin{array}{c}\Delta;\Phi_a;|\Gamma|_2 \vdash^{-}_{k'} e' : A_1 + A_2 \\ \Delta;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e \ominus e_1' \lesssim t : \tau \qquad \Delta;\Phi_a;y : U\,(A_2,A_2),\Gamma \vdash e \ominus e_2' \lesssim t : \tau\end{array}}{\Delta;\Phi_a;\Gamma \vdash e \ominus \text{ case } (e', x.e_1', y.e_2') \lesssim t - k' - c_{case} : \tau}\ \textbf{r-e-case}$$

Figure 72: RelCost typing rules (Part 6)

$\boxed{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e_1^* \ominus e_2^* \lesssim t : \tau}$ Expressions $e_1 \ominus e_2$ are embedded into $e_1^* \ominus e_1^*$ with the relational type τ and the relational cost t.

$\boxed{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : A}$ Expression e is embedded into $e*$ with the unary type A and the minimum and maximum execution costs k and t, respectively.

General rules

$$\frac{\Delta; \Phi_a; |\Gamma| \vdash_{k_1}^{t_1} e_1 \rightsquigarrow e_1^* : A \qquad \Delta; \Phi_a; |\Gamma| \vdash_{k_2}^{t_2} e_2 \rightsquigarrow e_2^* : A}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow \texttt{switch}\ e_1^* \ominus \texttt{switch}\ e_2^* \lesssim t_1 - k_2 : U\,A}\ \textbf{e-switch}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau \qquad \forall x_i \in dom(\Gamma),\quad e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,\Gamma(x_i)}}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \ominus e \rightsquigarrow \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in}\ \texttt{NC}\ e^*[\overline{y_i/x_i}] \ominus \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in}\ \texttt{NC}\ e^*[\overline{y_i/x_i}] \lesssim 0 : \Box\,\tau}\ \textbf{e-nochange}$$

Constant integers and unit

$$\frac{}{\Delta; \Phi_a; \Omega \vdash_0^0 \texttt{n} \rightsquigarrow \texttt{n} : \text{int}}\ \textbf{e-u-const} \qquad \frac{}{\Delta; \Phi_a; \Gamma \vdash \texttt{n} \ominus \texttt{n} \rightsquigarrow \texttt{n} \ominus \texttt{n} \lesssim 0 : \text{int}_r}\ \textbf{e-r-const}$$

$$\frac{}{\Delta; \Phi_a; \Omega \vdash_0^0 () \rightsquigarrow () : \text{unit}}\ \textbf{e-u-unit} \qquad \frac{}{\Delta; \Phi_a; \Gamma \vdash () \ominus () \rightsquigarrow () \ominus () \lesssim 0 : \text{unit}_r}\ \textbf{e-r-unit}$$

Variables x

$$\frac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash_0^0 x \rightsquigarrow x : A}\ \textbf{e-u-var} \qquad \frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \ominus x \rightsquigarrow x \ominus x \lesssim 0 : \tau}\ \textbf{e-r-var}$$

inl e

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : A_1 \qquad \Delta; \Phi_a \vdash A_2\ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash_k^t \text{inl}\ e \rightsquigarrow \text{inl}\ e^* : A_1 + A_2}\ \textbf{e-u-inl}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau_1 \qquad \Delta; \Phi_a \vdash \tau_2\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{inl}\ e_1 \ominus \text{inl}\ e_2 \rightsquigarrow \text{inl}\ {e_1}^* \ominus \text{inl}\ {e_2}^* \lesssim t : \tau_1 + \tau_2}\ \textbf{e-r-inl}$$

inr e

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : A_2 \qquad \Delta; \Phi_a \vdash A_1\ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash_k^t \text{inr}\ e \rightsquigarrow \text{inr}\ e^* : A_1 + A_2}\ \textbf{e-u-inr}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau_2 \qquad \Delta; \Phi_a \vdash \tau_1\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{inr}\ e_1 \ominus \text{inl}\ e_2 \rightsquigarrow \text{inr}\ {e_1}^* \ominus \text{inl}\ {e_2}^* \lesssim t : \tau_1 + \tau_2}\ \textbf{e-r-inr}$$

case $(e, x.e_1, y.e_2)$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : A_1 + A_2 \\ \Delta; \Phi_a; x : A_1, \Omega \vdash_{k'}^{t'} e_1 \rightsquigarrow {e_1}^* : A \qquad \Delta; \Phi_a; y : A_2, \Omega \vdash_{k'}^{t'} e_2 \rightsquigarrow {e_2}^* : A\end{array}}{\Delta; \Phi_a; \Omega \vdash_{k+k'+c_{case}}^{t+t'+c_{case}} \text{case}\ (e, x.e_1, y.e_2) \rightsquigarrow \text{case}\ (e^*, x.e_1^*, y.e_2^*) : A}\ \textbf{e-u-case}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau_1 + \tau_2 \\ \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t' : \tau \qquad \Delta; \Phi_a; y : \tau_2, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t' : \tau \\ e_1^{**} = \text{case}\ (e^*, x.e_1^*, y.e_2^*) \qquad e_2^{**} = \text{case}\ (e', x.e_1'^*, y.e_2'^*)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{case}\ (e, x.e_1, y.e_2) \ominus \text{case}\ (e', x.e_1', y.e_2') \rightsquigarrow e_1^{**} \ominus e_2^{**} \lesssim t + t' : \tau}\ \textbf{e-r-case}$$

Figure 73: Embedding Rules (Part 1)

$\boxed{\text{fix } f(x).e}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \xrightarrow{\text{exec}(k,t)} A_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \xrightarrow{\text{exec}(k,t)} A_2, \Omega \vdash_k^t e \rightsquigarrow e^* : A_2}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{fix } f(x).e \rightsquigarrow \text{fix } f(x).e^* : A_1 \xrightarrow{\text{exec}(k,t)} A_2} \ \textbf{e-u-fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \ominus \text{fix } f(x).e_2 \rightsquigarrow \text{fix } f(x).{e_1}^* \ominus \text{fix } f(x).{e_2}^* \lesssim 0 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2} \ \textbf{e-r-fix}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \Box\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2), \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau_2 \\ \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \Box\,\Gamma(x_i)} \qquad e^{**} = \text{let } \overline{y_i = e_i\ x_i} \text{ in fix}_{NC}\ f(x).e^*\overline{[y_i/x_i]}\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \ominus \text{fix } f(x).e \rightsquigarrow e^{**} \ominus e^{**} \lesssim 0 : \Box\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2)} \ \textbf{e-r-fixNC}$$

$\boxed{e_1\ e_2}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 \rightsquigarrow {e_1}^* : A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 \rightsquigarrow {e_2}^* : A_1}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2+k+c_{app}}^{t_1+t_2+t+c_{app}} e_1\ e_2 \rightsquigarrow e_1^*\ e_2^* : A_2} \ \textbf{e-u-app}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_1}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \rightsquigarrow e_1^*\ e_2^* \ominus e_1'^*\ e_2'^* \lesssim t_1 + t_2 + t : \tau_2} \ \textbf{e-r-app}$$

$\boxed{\langle e_1, e_2 \rangle}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 \rightsquigarrow {e_1}^* : A_1 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 \rightsquigarrow {e_2}^* : A_2}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \langle e_1, e_2 \rangle \rightsquigarrow \langle e_1^*, e_2^* \rangle : A_1 \times A_2} \ \textbf{e-u-prod}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : \tau_1 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \langle e_1, e_2 \rangle \ominus \langle e_1', e_2' \rangle \rightsquigarrow \langle e_1^*, e_2^* \rangle \ominus \langle e_1'^*, e_2'^* \rangle \lesssim t_1 + t_2 : \tau_1 \times \tau_2} \ \textbf{e-r-prod}$$

Figure 74: Embedding Rules (Part 2)

$\boxed{\pi_1(e)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash t \ominus e \rightsquigarrow t^* \ominus e^* \lesssim k : A_1 \times A_2}{\Delta; \Phi_a; \Omega \vdash_k^t \pi_1(e) \rightsquigarrow \pi_1(e^*) : A_1} \ \textbf{e-u-proj1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau_1 \times \tau_2}{\Delta; \Phi_a; \Gamma \vdash \pi_1(e) \ominus \pi_1(e') \rightsquigarrow \pi_1(e^*) \ominus \pi_1(e'^*) \lesssim t : \tau_1} \ \textbf{e-r-proj1}$$

$\boxed{\text{nil}}$

$$\frac{\Delta; \Phi_a \vdash A \ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash^{0}_{0} \text{nil} \rightsquigarrow \text{nil} : \text{list}[0]\, A}\ \textbf{e-u-nil} \qquad \frac{\Delta; \Phi_a \vdash \tau \ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{nil} \ominus \text{nil} \rightsquigarrow \text{nil} \ominus \text{nil} \lesssim 0 : \text{list}[0]^{\alpha}\, \tau}\ \textbf{e-r-nil}$$

$\boxed{\text{cons}(e_1, e_2)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash^{k_1}_{t_1} e_1 \rightsquigarrow {e_1}^{*} : A \qquad \Delta; \Phi_a; \Omega \vdash^{k_2}_{t_2} e_2 \rightsquigarrow {e_2}^{*} : \text{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash^{t_1+t_2}_{k_1+k_2} \text{cons}(e_1, e_2) \rightsquigarrow \text{cons}_C(e_1^{*}, e_2^{*}) : \text{list}[n+1]\, A}\ \textbf{e-u-cons}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^{*} \ominus {e_1'}^{*} \lesssim t_1 : \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^{*} \ominus {e_2'}^{*} \lesssim t_2 : \text{list}[n]^{\alpha}\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \ominus \text{cons}(e_1', e_2') \rightsquigarrow \text{cons}_C(e_1^{*}, e_2^{*}) \ominus \text{cons}_C(e_1'^{*}, e_2'^{*}) \lesssim t_1 + t_2 : \text{list}[n+1]^{\alpha+1}\, \tau}\ \textbf{e-r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^{*} \ominus {e_1'}^{*} \lesssim t_1 : \Box\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^{*} \ominus {e_2'}^{*} \lesssim t_2 : \text{list}[n]^{\alpha}\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}(e_1, e_2) \ominus \text{cons}(e_1', e_2') \rightsquigarrow \text{cons}_{NC}(e_1', e_2') \ominus \text{cons}_{NC}(e_1^{*}, e_2^{*}) \lesssim t_1 + t_2 : \text{list}[n+1]^{\alpha}\, \tau}\ \textbf{e-r-cons2}$$

$\boxed{\text{case } e \text{ of nil} \to e_1 \mid h :: tl \to e_2}$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash^{t}_{k} e \rightsquigarrow e^{*} : \text{list}[n]\, A \\ \Delta; \Phi_a \wedge n = 0; \Omega \vdash^{t'}_{k'} e_1 \rightsquigarrow {e_1}^{*} : A' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : A, tl : \text{list}[i]\, A, \Omega \vdash^{t'}_{k'} e_2 \rightsquigarrow {e_2}^{*} : A'\end{array}}{\Delta; \Phi_a; \Omega \vdash^{t+t'+c_{caseL}}_{k+k'+c_{caseL}} \begin{array}{l}\text{case } e \text{ of nil} \to e_1 \\ \mid h :: tl \to e_2\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^{*} \text{ of nil } \to e_1^{*} \\ \mid h ::_{NC} tl \to e_2^{*} \\ \mid h ::_{C} tl \to e_2^{*}\end{array} A'}\ \textbf{e-u-caseL}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^{*} \ominus e'^{*} \lesssim t : \text{list}[n]^{\alpha}\, \tau \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^{*} \ominus {e_1'}^{*} \lesssim t' : \tau' \\ i, \Delta; \Phi_a \wedge n = i+1; h : \Box\, \tau, tl : \text{list}[i]^{\alpha}\, \tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^{*} \ominus {e_2'}^{*} \lesssim t' : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h : \tau, tl : \text{list}[i]^{\beta}\, \tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow e_3^{*} \ominus e_3'^{*} \lesssim t' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil } \to e_1 \\ \mid h :: tl \to e_2\end{array} \ominus \begin{array}{l}\text{case } e \text{ of nil } \to e_1' \\ \mid h :: tl \to e_2'\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^{*} \text{ of nil } \to e_1^{*} \\ \mid h ::_{NC} tl \to e_2^{*} \\ \mid h ::_{C} tl \to e_3^{*}\end{array} \ominus \begin{array}{l}\text{case } e'^{*} \text{ of nil } \to e_1'^{*} \\ \mid h ::_{NC} tl \to e_2'^{*} \\ \mid h ::_{C} tl \to e_3'^{*}\end{array} \lesssim t + t' : \tau'}\ \textbf{e-r-ca}$$

$\boxed{\Lambda e}$

$$\frac{i :: S, \Delta; \Phi_a; \Omega \vdash^{t}_{k} e \rightsquigarrow e^{*} : A \qquad i \notin \text{FIV}(\Phi_a; \Omega)}{\Delta; \Phi_a; \Omega \vdash^{0}_{0} \Lambda.e \rightsquigarrow \Lambda i.e^{*} : \forall i \overset{\text{exec}(k,t)}{::} S.\, A}\ \textbf{e-u-iLam}$$

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^{*} \ominus e'^{*} \lesssim t : \tau \qquad i \notin \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda.e \ominus \Lambda.e' \rightsquigarrow \Lambda i.e^{*} \ominus \Lambda i.e'^{*} \lesssim 0 : \forall i \overset{\text{diff}(t)}{::} S.\, \tau}\ \textbf{e-r-iLam}$$

$\boxed{e[\,]}$

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t}_{k} e \rightsquigarrow e^{*} : \forall i \overset{\text{exec}(k',t')}{::} S.\, A \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Omega \vdash^{t+t'[I/i]+c_{iApp}}_{k+k'[I/i]+c_{iApp}} e[] \rightsquigarrow e^{*}[I] : A\{I/i\}}\ \textbf{e-u-iApp}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^{*} \ominus e'^{*} \lesssim t : \forall i \overset{\text{diff}(t')}{::} S.\, \tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[] \ominus e'[] \rightsquigarrow e^{*}[I] \ominus e'^{*}[I] \lesssim t + t'[I/i] : \tau\{I/i\}}\ \textbf{e-r-iApp}$$

Figure 75: Embedding Rules (Part 3)

$\boxed{\text{pack } e}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : A\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Omega \vdash_k^t \text{pack } e \rightsquigarrow \text{pack } e^* \text{ with } I : \exists i{::}S.\, A} \textbf{ e-u-pack}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \ominus \text{pack } e' \rightsquigarrow \text{pack } e^* \text{ with } I \ominus \text{pack } e'^* \text{ with } I \lesssim t : \exists i{::}S.\, \tau} \textbf{ e-r-pack}$$

$\boxed{\text{unpack } e \text{ as } x \text{ in } e'}$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 \rightsquigarrow {e_1}^* : \exists i{::}S.\, A_1 \\ i :: S, \Delta; \Phi_a; x : A_1, \Omega \vdash_{k_2}^{t_2} e_2 \rightsquigarrow {e_2}^* : A_2 \qquad i \notin FV(\Phi_a; \Omega, A_2, k_2, t_2)\end{array}}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2+c_{unpack}}^{t_1+t_2} \text{unpack } e_1 \text{ as } x \text{ in } e_2 \rightsquigarrow \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 : \text{unpack } e_1^* \text{ as } (x, i) \text{ in } e_2^*\ A_2} \textbf{ e-u-unpack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : \exists i{::}S.\, \tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_2 \\ i \notin FV(\Phi_a; \Gamma, \tau_2, t_2) \qquad e_1^{**} = \text{unpack } e_1^* \text{ as } (x, i) \text{ in } e_2^* \qquad e_2^{**} = \text{unpack } e_1'^* \text{ as } (x, i) \text{ in } e_2'^*\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow e_1^{**} \ominus e_2^{**} \lesssim t_1 + t_2 : \tau_2} \textbf{ e-r-unpack}$$

$\boxed{\text{let } x = e_1 \text{ in } e_2}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 \rightsquigarrow {e_1}^* : A_1 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash_{k_2}^{t_2} e_2 \rightsquigarrow {e_2}^* : A_2}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2+c_{let}}^{t_1+t_2+c_{let}} \text{let } x = e_1 \text{ in } e_2 \rightsquigarrow \text{let } x = e_1^* \text{ in } e_2^* : A_2} \textbf{ e-u-let}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus \text{let } x = e_1' \text{ in } e_2' \rightsquigarrow \text{let } x = e_1^* \text{ in } e_2^* \ominus \text{let } x = e_1'^* \text{ in } e_2'^* \lesssim t_1 + t_2 : \tau_2} \textbf{ e-r-let}$$

$\boxed{C \,\&\, \tau \text{ intro. rules}}$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Omega \vdash_k^t e \rightsquigarrow e^* : A}{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : C \,\&\, A} \textbf{ e-u-andI}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : C \,\&\, \tau} \textbf{ e-r-andI}$$

$\boxed{C \,\&\, \tau \text{ elim. rules}}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 \rightsquigarrow {e_1}^* : C \,\&\, A_1 \qquad \Delta; \Phi_a \wedge C; x : A_1, \Omega \vdash_{k_u}^{t_2} e_2 \rightsquigarrow {e_2}^* : A_2}{\Delta; \Phi_a; \Gamma \vdash_{k_1+k_2}^{t_1+t_2} \text{clet } e_1 \text{ as } x \text{ in } e_2 \rightsquigarrow \text{clet } e_1^* \text{ as } x \text{ in } e_2^* : A_2} \textbf{ e-u-c-andE}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : C \,\&\, \tau_1 \\ \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_2\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \ominus \text{clet } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow \text{clet } e_1^* \text{ as } x \text{ in } e_2^* \ominus \text{clet } e_1'^* \text{ as } x \text{ in } e_2'^* \lesssim t_1 + t_2 : \tau_2} \textbf{ e-r-c-andE}$$

$\boxed{C \supset \tau \text{ intro. rules}}$

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash_k^t e \rightsquigarrow e^* : A}{\Delta; \Phi_a; \Omega \vdash_k^t e \rightsquigarrow e^* : C \supset A} \textbf{ e-u-c-impI} \qquad \frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : C \supset \tau} \textbf{ e-r-c-impI}$$

Figure 76: Embedding Rules (Part 4)

$\boxed{C \,\&\, \tau \text{ elim. rules}}$

$$\frac{\Delta; \Phi_a; \Omega \vdash^t_k e \rightsquigarrow e^* : C \supset A \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Omega \vdash^t_k \text{celim } e \rightsquigarrow \text{celim } e^* : A} \textbf{ e-u-c-implE}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \ominus \text{celim } e' \rightsquigarrow \text{celim } e^* \ominus \text{celim } e'^* \lesssim t : \tau} \textbf{ e-r-c-implE}$$

$\boxed{\text{Primitive application}}$

$$\frac{\Upsilon(\zeta) = A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e \rightsquigarrow e^* : A_1}{\Delta; \Phi_a; \Omega \vdash^{t+t'}_{k+k'} \zeta\ e \rightsquigarrow \zeta\ e^* : A_2} \textbf{ e-u-primapp}$$

$$\frac{\Upsilon(\zeta) = \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t' : \tau_1}{\Delta; \Phi_a; \Gamma \vdash \zeta\ e \ominus \zeta\ e' \rightsquigarrow \zeta\ e^* \ominus \zeta\ e'^* \lesssim t + t' : \tau_2} \textbf{ e-r-primapp}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Omega\ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash^t_k e \rightsquigarrow \texttt{contra}\ e : \tau} \textbf{ e-u-contra}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow \texttt{contra}\ e_1 \ominus \texttt{contra}\ e_2 \lesssim t : \tau} \textbf{ e-r-contra}$$

$\boxed{\text{Subsumption}}$

$$\frac{\Delta; \Phi_a; \Omega \vdash^t_k e \rightsquigarrow e^* : A \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad \Delta; \Phi_a \models k' \leq k \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e \rightsquigarrow e^* : A'} \textbf{ e-u-}\sqsubseteq$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \texttt{coerce}_{\tau,\tau'} \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e'\ e_1^* \ominus e'\ e_2^* \lesssim t' : \tau'} \textbf{ e-r-}\sqsubseteq$$

$\boxed{\text{Constraint-based rules}}$

$$\frac{\Delta; C \wedge \Phi_a; \Omega \vdash^t_k e \rightsquigarrow e^* : A \qquad \Delta; \neg C \wedge \Phi_a; \Omega \vdash^t_k e \rightsquigarrow e^{**} : A \qquad \Delta \vdash C\ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash^t_k e \rightsquigarrow \texttt{split}\ (e^*, e^{**}) \text{ with } C : A} \textbf{ e-u-split}$$

$$\frac{\begin{array}{c}\Delta; C \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau \\ \Delta; \neg C \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e_1^{**} \ominus e_2^{**} \lesssim t : \tau \qquad \Delta \vdash C\ \texttt{wf}\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow \texttt{split}\ (e_1^*, e_1^{**}) \text{ with } C \ominus \texttt{split}\ (e_2^*, e_2^{**}) \text{ with } C \lesssim t : \tau} \textbf{ e-r-split}$$

Figure 77: Embedding Rules (Part 5)

Asynchronous typing rules

$$\frac{\Delta; \Phi_a; |\Gamma|_1 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow {e_1}^* : A_1 \qquad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e_2 \ominus e \rightsquigarrow {e_2}^* \ominus e^* \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus e \rightsquigarrow \text{let } x = e_1^* \text{ in } e_2^* \ominus e^* \lesssim t_1 + t_2 + c_{let} : \tau_2} \textbf{ e-r-let-e}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_2 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow {e_1}^* : A_1 \qquad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e \ominus e_2 \rightsquigarrow e^* \ominus {e_2}^* \lesssim t_2 : \tau_2}{\Delta; \Phi_a; \Gamma \vdash e \ominus \text{let } x = e_1 \text{ in } e_2 \rightsquigarrow e^* \ominus \text{let } x = e_1^* \text{ in } e_2^* \lesssim t_2 - k_1 - c_{let} : \tau_2} \textbf{ e-r-e-r-let}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_1 \vdash^{t_1}_{k_1} e_1 \rightsquigarrow {e_1}^* : A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : U\,(A_1, A_2')}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_2' \rightsquigarrow e_1^*\ e_2^* \ominus e_2'^* \lesssim t_1 + t_2 + t + c_{app} : U\,(A_2, A_2')} \textbf{ e-r-app-e}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_2 \vdash^{t_1}_{k_1} e_1' \rightsquigarrow {e_1'}^* : A_1' \xrightarrow{\text{exec}(k,t)} A_2' \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : U\,(A_2, A_1')}{\Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_1'\ e_2' \rightsquigarrow e_2^* \ominus e_1'^*\ e_2'^* \lesssim t_2 - k_1 - k - c_{app} : U\,(A_2, A_2')} \textbf{ e-r-e-r-app}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; |\Gamma|_1 \vdash^{t}_{-} e \rightsquigarrow e^* : A_1 + A_2 \qquad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e_1 \ominus e' \rightsquigarrow {e_1}^* \ominus {e'}^* \lesssim t' : \tau \\ \Delta; \Phi_a; y : U\,(A_2, A_2), \Gamma \vdash e_2 \ominus e' \rightsquigarrow {e_2}^* \ominus {e'}^* \lesssim t' : \tau\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{ case } (e, x.e_1, y.e_2) \ominus e' \rightsquigarrow \text{ case } (e^*, x.e_1^*, y.e_2^*) \ominus e'^* \lesssim t' + t + c_{case} : \tau} \textbf{ e-r-case-r-e}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; |\Gamma|_2 \vdash^{-}_{k'} e' \rightsquigarrow {e'}^* : A_1 + A_2 \qquad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e \ominus e_1' \rightsquigarrow e^* \ominus {e_1'}^* \lesssim t : \tau \\ \Delta; \Phi_a; y : U\,(A_2, A_2), \Gamma \vdash e \ominus e_2' \rightsquigarrow e^* \ominus {e_2'}^* \lesssim t : \tau\end{array}}{\Delta; \Phi_a; \Gamma \vdash e \ominus \text{ case } (e', x.e_1', y.e_2') \rightsquigarrow e^* \ominus \text{ case } (e'^*, x.e_1'^*, y.e_2'^*) \lesssim t - k' - c_{case} : \tau} \textbf{ e-r-e-r-case}$$

Figure 78: Embedding Rules (Part 6)

$\boxed{\Delta; \Phi_a \models \tau_1 \equiv \tau_2}$ checks whether τ_1 is equivalent to τ_2

$$\frac{}{\Delta; \Phi_a \models \mathrm{int}_r \equiv \mathrm{int}_r}\ \textbf{eq-int} \qquad \frac{}{\Delta; \Phi_a \models \mathrm{unit}_r \equiv \mathrm{unit}_r}\ \textbf{eq-unit}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2' \qquad \Delta; \Phi_a \models t \doteq t'}{\Delta; \Phi_a \models \tau_1 \xrightarrow{\mathrm{diff}(t)} \tau_2 \equiv \tau_1' \xrightarrow{\mathrm{diff}(t')} \tau_2'}\ \textbf{eq-fun} \qquad \frac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2'}{\Delta; \Phi_a \models \tau_1 \times \tau_2 \equiv \tau_1' \times \tau_2'}\ \textbf{eq-prod}$$

$$\frac{\Delta; \Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta; \Phi_a \models \tau_2 \equiv \tau_2'}{\Delta; \Phi_a \models \tau_1 + \tau_2 \equiv \tau_1' + \tau_2'}\ \textbf{eq-sum} \qquad \frac{\Delta; \Phi_a \models n \doteq n' \qquad \Delta; \Phi_a \models \alpha \doteq \alpha' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \equiv \mathrm{list}[n']^{\alpha'}\, \tau'}\ \textbf{eq-list}$$

$$\frac{i, \Delta; \Phi_a \models \tau \equiv \tau' \qquad i, \Delta; \Phi_a \models t \doteq t'}{\Delta; \Phi_a \models \forall i \overset{\mathrm{diff}(t)}{::} S.\, \tau \equiv \forall i \overset{\mathrm{diff}(t')}{::} S.\, \tau'}\textbf{eq-}\forall \qquad \frac{i, \Delta; \Phi_a \models \tau \equiv \tau' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \equiv \exists i{::}S.\, \tau'}\ \textbf{eq-}\exists$$

$$\frac{\Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models \Box\, \tau \equiv \Box\, \tau'}\textbf{eq-B-}\Box$$

$$\frac{\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \qquad \Delta; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2}{\Delta; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1', A_2')}\textbf{eq-U}$$

$$\frac{\Delta; C \wedge \Phi_a \models C' \qquad \Delta; C' \wedge \Phi_a \models C \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C \supset \tau \equiv C' \supset \tau'}\textbf{eq-c-impl}$$

$$\frac{\Delta; C' \wedge \Phi_a \models C \qquad \Delta; C \wedge \Phi_a \models C' \qquad \Delta; \Phi_a \models \tau \equiv \tau'}{\Delta; \Phi_a \models C\ \&\ \tau \equiv C'\ \&\ \tau'}\textbf{eq-c-prod}$$

Figure 79: RelCostCore binary type equivalence rules

General rules

$$\frac{\Delta; \Phi_a; |\Gamma| \vdash_{k_1}^{t_1} e_1 :^c A \qquad \Delta; \Phi_a; |\Gamma| \vdash_{k_2}^{t_2} e_2 :^c A}{\Delta; \Phi_a; \Gamma \vdash \texttt{switch}\, e_1 \ominus \texttt{switch}\, e_2 \lesssim t_1 - k_2 :^c U\, A} \textbf{ c-switch}$$

$$\frac{\Delta; \Phi_a; \Box\,\Gamma \vdash e \ominus e \lesssim t :^c \tau}{\Delta; \Phi_a; \Box\,\Gamma, \Gamma' \vdash \texttt{NC}\, e \ominus \texttt{NC}\, e \lesssim 0 :^c \Box\,\tau} \textbf{ c-nochange} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \Box\,\tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{der}\, e_1 \ominus \texttt{der}\, e_2 \lesssim t :^c \tau} \textbf{ c-der}$$

Constant integers and unit

$$\frac{}{\Delta; \Phi_a; \Omega \vdash_0^0 \texttt{n} :^c \text{int}} \textbf{ c-const} \qquad \frac{}{\Delta; \Phi_a; \Gamma \vdash \texttt{n} \ominus \texttt{n} \lesssim 0 :^c \text{int}_r} \textbf{ c-r-const}$$

$$\frac{}{\Delta; \Phi_a; \Omega \vdash_0^0 () :^c \text{unit}} \textbf{ c-unit} \qquad \frac{}{\Delta; \Phi_a; \Gamma \vdash () \ominus () \lesssim 0 :^c \text{unit}_r} \textbf{ c-r-unit}$$

Variables x

$$\frac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash_0^0 x :^c A} \textbf{ c-var} \qquad \frac{\Gamma(x) = \tau}{\Delta; \Phi_a; \Gamma \vdash x \ominus x \lesssim 0 :^c \tau} \textbf{ c-r-var}$$

inl e

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A_1 \qquad \Delta; \Phi_a \vdash^{\mathsf{A}} A_2 \;\texttt{wf}}{\Delta; \Phi_a; \Omega \vdash_k^t \text{inl}\, e :^c A_1 + A_2} \textbf{ c-inl}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau_1 \qquad \Delta; \Phi_a \vdash \tau_2 \;\texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{inl}\, e \ominus \text{inl}\, e' \lesssim t :^c \tau_1 + \tau_2} \textbf{ c-r-inl}$$

inr e

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A_2 \qquad \Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \;\texttt{wf}}{\Delta; \Phi_a; \Omega \vdash_k^t \text{inr}\, e :^c A_1 + A_2} \textbf{ c-inr}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau_2 \qquad \Delta; \Phi_a \vdash \tau_1 \;\texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{inr}\, e \ominus \text{inr}\, e' \lesssim t :^c \tau_1 + \tau_2} \textbf{ c-r-inr}$$

case $(e, x.e_1, y.e_2)$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A_1 + A_2 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash_{k'}^{t'} e_1 :^c A \qquad \Delta; \Phi_a; y : A_2, \Omega \vdash_{k'}^{t'} e_2 :^c A}{\Delta; \Phi_a; \Omega \vdash_{k+k'+c_{case}}^{t+t'+c_{case}} \text{case}\, (e, x.e_1, y.e_2) :^c A} \textbf{ c-case}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau_1 + \tau_2 \\ \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_1 \ominus e_1' \lesssim t' :^c \tau \qquad \Delta; \Phi_a; y : \tau_2, \Gamma \vdash e_2 \ominus e_2' \lesssim t' :^c \tau\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{case}\, (e, x.e_1, y.e_2) \ominus \text{case}\, (e', x.e_1', y.e_2') \lesssim t + t' :^c \tau} \textbf{ c-r-case}$$

Figure 80: RelCostCore typing rules (Part 1)

$\boxed{\text{fix } f(x).e}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A_1 \xrightarrow{\text{exec}(k,t)} A_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : A_1, f : A_1 \xrightarrow{\text{exec}(k,t)} A_2, \Omega \vdash_k^t e :^c A_2}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{fix } f(x).e :^c A_1 \xrightarrow{\text{exec}(k,t)} A_2} \ \textbf{c-fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, \Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e_1 \ominus \text{fix } f(x).e_2 \lesssim 0 :^c \tau_1 \xrightarrow{\text{diff}(t)} \tau_2} \ \textbf{c-r-fix}$$

$$\frac{\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \ \texttt{wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \Box(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2), \Box\Gamma \vdash e \ominus e \lesssim t :^c \tau_2}{\Delta; \Phi_a; \Box\Gamma \vdash \text{fix}_{NC}\ f(x).e \ominus \text{fix}_{NC}\ f(x).e \lesssim 0 :^c \Box(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2)} \ \textbf{c-r-fixNC}$$

$\boxed{e_1\ e_2}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 :^c A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 :^c A_1}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2+k+c_{app}}^{t_1+t_2+t+c_{app}} e_1\ e_2 :^c A_2} \ \textbf{c-app}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \\ \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_1\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \lesssim t_1 + t_2 + t :^c \tau_2} \ \textbf{c-r-app}$$

$\boxed{\langle e_1, e_2 \rangle}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 :^c A_1 \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 :^c A_2}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \langle e_1, e_2 \rangle :^c A_1 \times A_2} \ \textbf{c-prod}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \tau_1 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \langle e_1, e_2 \rangle \ominus \langle e_1', e_2' \rangle \lesssim t_1 + t_2 :^c \tau_1 \times \tau_2} \ \textbf{c-r-prod}$$

$\boxed{\pi_1(e)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A_1 \times A_2}{\Delta; \Phi_a; \Omega \vdash_{k+c_{proj}}^{t+c_{proj}} \pi_1(e) :^c A_1} \ \textbf{c-proj1} \qquad \frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau_1 \times \tau_2}{\Delta; \Phi_a; \Gamma \vdash \pi_1(e) \ominus \pi_1(e') \lesssim t :^c \tau_1} \ \textbf{c-r-proj1}$$

$\boxed{\pi_2(e)}$

Symmetric rules.

Figure 81: RelCostCore typing rules (Part 2)

$\boxed{\text{nil}}$

$$\frac{\Delta; \Phi_a \vdash^{\mathsf{A}} A \ \mathtt{wf}}{\Delta; \Phi_a; \Omega \vdash_0^0 \text{nil} :^c \text{list}[0]\, A} \ \textbf{c-nil} \qquad \frac{\Delta; \Phi_a \vdash \tau \ \mathtt{wf}}{\Delta; \Phi_a; \Gamma \vdash \text{nil} \ominus \text{nil} \lesssim 0 :^c \text{list}[0]^\alpha\, \tau} \ \textbf{c-r-nil}$$

$\boxed{\text{cons}(e_1, e_2)}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 :^c A \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 :^c \text{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \text{cons}_C(e_1, e_2) :^c \text{list}[n+1]\, A} \ \textbf{c-cons}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_C(e_1, e_2) \ominus \text{cons}_C(e_1', e_2') \lesssim t_1 + t_2 :^c \text{list}[n+1]^{\alpha+1}\, \tau} \ \textbf{c-r-cons1}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \square\, \tau \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \text{list}[n]^\alpha\, \tau}{\Delta; \Phi_a; \Gamma \vdash \text{cons}_{NC}(e_1, e_2) \ominus \text{cons}_{NC}(e_1', e_2') \lesssim t_1 + t_2 :^c \text{list}[n+1]^{\alpha}\, \tau} \ \textbf{c-r-cons2}$$

$\boxed{\text{case } e \text{ of nil} \to e_1 \mid h :: tl \to e_2}$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Omega \vdash_k^t e :^c \text{list}[n]\, A \\ \Delta; \Phi_a \wedge n = 0; \Omega \vdash_{k'}^{t'} e_1 :^c A' \qquad i, \Delta; \Phi_a \wedge n = i+1; h : A, tl : \text{list}[i]\, A, \Omega \vdash_{k'}^{t'} e_2 :^c A'\end{array}}{\Delta; \Phi_a; \Omega \vdash_{k+k'+c_{caseL}}^{t+t'+c_{caseL}} \text{case } e \text{ of nil} \to e_1 \mid h ::_{NC} tl \to e_2 \mid h ::_C tl \to e_3 :^c A'} \ \textbf{c-caseL}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \text{list}[n]^\alpha\, \tau \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \lesssim t' :^c \tau' \\ i, \Delta; \Phi_a \wedge n = i+1; h : \square\, \tau, tl : \text{list}[i]^\alpha\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' :^c \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h' : \tau, tl : \text{list}[i]^\beta\, \tau, \Gamma \vdash e_3 \ominus e_3' \lesssim t' :^c \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil} \to e_1 \\ \mid h ::_N tl \to e_2 \\ \mid h' ::_C tl' \to e_3\end{array} \ominus \begin{array}{l}\text{case } e \text{ of nil} \to e_1' \\ \mid h ::_N tl \to e_2' \\ \mid h' ::_C tl' \to e_3'\end{array} \lesssim t + t' :^c \tau'} \ \textbf{c-r-caseL}$$

Figure 82: RelCostCore typing rules (Part 3)

$\boxed{\Lambda e}$

$$\frac{i :: S, \Delta; \Phi_a; \Omega \vdash_k^t e :^c A \qquad i \notin \mathrm{FIV}(\Phi_a; \Omega)}{\Delta; \Phi_a; \Omega \vdash_0^0 \Lambda i.e :^c \forall i \overset{\mathrm{exec}(k,t)}{::} S.\, A} \ \textbf{c-iLam}$$

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau \qquad i \notin \mathrm{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda i.e \ominus \Lambda i.e' \lesssim 0 :^c \forall i \overset{\mathrm{diff}(t)}{::} S.\, \tau} \ \textbf{c-r-iLam}$$

$\boxed{e[\,]}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c \forall i \overset{\mathrm{exec}(k',t')}{::} S.\, A \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Omega \vdash_{k+k'[I/i]}^{t+t'[I/i]} e[I] :^c A\{I/i\}} \ \textbf{c-iApp}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \forall i \overset{\mathrm{diff}(t')}{::} S.\, \tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[I] \ominus e'[I] \lesssim t + t'[I/i] :^c \tau\{I/i\}} \ \textbf{c-r-iApp}$$

$\boxed{\text{pack } e}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Omega \vdash_k^t \text{pack } e \text{ with } I :^c \exists i{::}S.\, A} \ \textbf{c-pack}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \text{ with } I \ominus \text{pack } e' \text{ with } I \lesssim t :^c \exists i{::}S.\, \tau} \ \textbf{c-r-pack}$$

$\boxed{\text{unpack } e \text{ as } x \text{ in } e'}$

$$\frac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 :^c \exists i{::}S.\, A_1 \qquad i :: S, \Delta; \Phi_a; x : A_1, \Omega \vdash_{k_2}^{t_2} e_2 :^c A_2 \qquad i \notin FV(\Phi_a; \Gamma, A_2, k_2, t_2)}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 :^c A_2} \ \textbf{c-unpack}$$

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \exists i{::}S.\, \tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' \lesssim t_1 + t_2 :^c \tau_2} \ \textbf{c-r-unpack1}$$

Figure 83: RelCostCore typing rules (Part 4)

Primitive application

$$\frac{\Upsilon(\zeta) = A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e :^c A_1}{\Delta; \Phi_a; \Omega \vdash^{t+t'+c_{app}}_{k+k'+c_{app}} \zeta\ e :^c A_2}\ \textbf{c-primapp}$$

$$\frac{\Upsilon(\zeta) = \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t' :^c \tau_1}{\Delta; \Phi_a; \Gamma \vdash \zeta\ e \ominus \zeta\ e' \lesssim t + t' :^c \tau_2}\ \textbf{c-r-primapp}$$

$C \ \&\ \tau$ intro. rules

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Omega \vdash^t_k e :^c A}{\Delta; \Phi_a; \Omega \vdash^t_k e :^c C \ \&\ A}\ \textbf{c-c-andI}$$

$$\frac{\Delta; \Phi_a \models C \qquad \Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \lesssim t :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c C \ \&\ \tau}\ \textbf{c-c-andI}$$

$C \ \&\ \tau$ elim. rules

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t_1}_{k_1} e_1 :^c C \ \&\ A_1 \qquad \Delta; \Phi_a \wedge C; x : A_1, \Omega \vdash^{t_2}_{k_2} e_2 :^c A_2}{\Delta; \Phi_a; \Omega \vdash^{t_1+t_2}_{k_1+k_2} \text{clet } e_1 \text{ as } x \text{ in } e_2 :^c A_2}\ \textbf{c-c-andE}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c C \ \&\ \tau_1 \qquad \Delta; \Phi_a \wedge C; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \ominus \text{clet } e_1' \text{ as } x \text{ in } e_2' \lesssim t_1 + t_2 :^c \tau_2}\ \textbf{c-r-c-andE}$$

$C \supset \tau$ intro. rules

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash^t_k e :^c A}{\Delta; \Phi_a; \Omega \vdash^t_k e :^c C \supset A}\ \textbf{c-c-impI} \qquad \frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e \ominus e' \lesssim t :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c C \supset \tau}\ \textbf{c-r-c-impI}$$

$C \supset \tau$ elim. rules

$$\frac{\Delta; \Phi_a; \Omega \vdash^t_k e :^c C \supset A \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Omega \vdash^t_k \text{celim } e :^c A}\ \textbf{c-c-implE}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c C \supset \tau \qquad \Delta; \Phi_a \models C}{\Delta; \Phi_a; \Gamma \vdash \text{celim } e \ominus \text{celim } e' \lesssim t :^c \tau}\ \textbf{c-r-c-implE}$$

let $x = e_1$ in e_2

$$\frac{\Delta; \Phi_a; \Omega \vdash^{t_1}_{k_1} e_1 :^c A_1 \qquad \Delta; \Phi_a; x : A_1, \Omega \vdash^{t_2}_{k_2} e_2 :^c A_2}{\Delta; \Phi_a; \Omega \vdash^{t_1+t_2+c_{let}}_{k_1+k_2+c_{let}} \text{let } x = e_1 \text{ in } e_2 :^c A_2}\ \textbf{c-let}$$

$$\frac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \tau_1 \qquad \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus \text{let } x = e_1' \text{ in } e_2' \lesssim t_1 + t_2 :^c \tau_2}\ \textbf{c-r-let1}$$

Figure 84: RelCostCore typing rules (Part 5)

Unary Subtyping

$$\frac{\Delta; \Phi_a; \Omega \vdash_k^t e :^c A \qquad \Delta; \Phi_a \models A \sqsubseteq A' \quad \Delta; \Phi_a \models k' \leq k \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Omega \vdash_{k'}^{t'} e :^c A'} \ \textbf{c-}\sqsubseteq$$

Binary Subeffecting

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau' \quad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t' :^c \tau'} \ \textbf{c-r-}\equiv$$

Constraint dependent typing

$$\frac{\Delta; \Phi_a \wedge C; \Omega \vdash_k^t e_1 :^c A \qquad \Delta; \Phi_a \wedge \neg C; \Omega \vdash_k^t e_2 :^c A}{\Delta; \Phi_a; \Omega \vdash_k^t \texttt{split}\ (e_1, e_2) \text{ with } C :^c A} \ \textbf{c-split}$$

$$\frac{\Delta; \Phi_a \wedge C; \Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \tau \qquad \Delta; \Phi_a \wedge \neg C; \Gamma \vdash e_1' \ominus e_2' \lesssim t :^c \tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{split}\ (e_1, e_1') \text{ with } C \ominus \texttt{split}\ (e_2, e_2') \text{ with } C \lesssim t :^c \tau} \ \textbf{c-r-split}$$

$$\frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Omega\ \texttt{wf}}{\Delta; \Phi_a; \Omega \vdash_k^t \texttt{contra}\ e :^c A} \ \textbf{c-contra} \qquad \frac{\Delta; \Phi_a \models \bot \qquad \Delta; \Phi_a \vdash \Gamma\ \texttt{wf}}{\Delta; \Phi_a; \Gamma \vdash \texttt{contra}\ e_1 \ominus \texttt{contra}\ e_2 \lesssim t :^c \tau} \ \textbf{c-r-contra}$$

Asynchronous typing

$$\frac{\Delta; \Phi_a; |\Gamma| \vdash_{k_1}^{t_1} e_1 :^c A_1 \quad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e_2 \ominus e \lesssim t_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus e \lesssim t_1 + t_2 + c_{let} :^c \tau_2} \ \textbf{c-r-let-e}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_2 \vdash_{k_1}^{t_1} e_1 :^c A_1 \qquad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e \ominus e_2 \lesssim t_2 :^c \tau_2}{\Delta; \Phi_a; \Gamma \vdash e \ominus \text{let } x = e_1 \text{ in } e_2 \lesssim t_2 - k_1 - c_{let} :^c \tau_2} \ \textbf{c-r-e-let}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_1 \vdash_{k_1}^{t_1} e_1 :^c A_1 \xrightarrow{\text{exec}(k,t)} A_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c U\,(A_1, A_2')}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_2' \lesssim t_1 + t_2 + t + c_{app} :^c U\,(A_2, A_2')} \ \textbf{c-r-app-e}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_2 \vdash_{k_1}^{t_1} e_1' :^c A_1' \xrightarrow{\text{exec}(k,t)} A_2' \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c U\,(A_2, A_1')}{\Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_1'\ e_2' \lesssim t_2 - k_1 - k - c_{app} :^c U\,(A_2, A_2')} \ \textbf{c-r-e-app}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_1 \vdash_{-}^{t} e :^c A_1 + A_2 \quad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e_1 \ominus e' \lesssim t' :^c \tau \qquad \Delta; \Phi_a; y : U\,(A_2, A_2), \Gamma \vdash e_2 \ominus e' \lesssim t' :^c \tau}{\Delta; \Phi_a; \Gamma \vdash \text{ case } (e, x.e_1, y.e_2) \ominus e' \lesssim t' + t + c_{case} :^c \tau} \ \textbf{c-r-case-e}$$

$$\frac{\Delta; \Phi_a; |\Gamma|_2 \vdash_{k'}^{-} e' :^c A_1 + A_2 \quad \Delta; \Phi_a; x : U\,(A_1, A_1), \Gamma \vdash e \ominus e_1' \lesssim t :^c \tau \qquad \Delta; \Phi_a; y : U\,(A_2, A_2), \Gamma \vdash e \ominus e_2' \lesssim t :^c \tau}{\Delta; \Phi_a; \Gamma \vdash e \ominus \text{ case } (e', x.e_1', y.e_2') \lesssim t - k' - c_{case} :^c \tau} \ \textbf{c-r-e-case}$$

Figure 85: RelCostCore typing rules (Part 6)

$\boxed{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow \tau, t \Rightarrow \Phi}$ Under the existential variable context ψ_a and the assumption Φ_a, $e_1 \ominus e_2$ checks against the input type τ and the difference cost t. Finally, it generates the constraint Φ.

$\boxed{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \tau \Rightarrow [\psi], t, \Phi}$ Under the existential variable context ψ_a and the assumption Φ_a, $e_1 \ominus e_2$ synthesizes the output type τ and the relative cost t where all the newly generated existential variables are defined in ψ. Finally, it generates the constraint Φ.

$\boxed{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi}$ Under the existential variable context ψ_a and the assumption Φ_a, e checks against the unary input type A and the minimum execution cost k and maximum execution cost t. Finally, it generates the constraint Φ.

$\boxed{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A \Rightarrow [\psi], k, t, \Phi}$ Under the existential variable context ψ_a and the assumption Φ_a, e synthesizes the unary output type A, the minimum execution cost k and maximum execution cost t, where all the newly generated existential variables are defined in ψ. Finally, it generates the constraint Φ.

$\boxed{\texttt{switch}\ e}$

$$\frac{\Delta; \psi_a; \Phi; |\Gamma| \vdash e_1 \uparrow A \Rightarrow [\psi_1], \_, t_1, \Phi_1 \qquad \Delta; \psi_a; \Phi; |\Gamma| \vdash e_2 \uparrow A \Rightarrow [\psi_2], k_2, \_, \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1 \ominus \texttt{switch}\ e_2 \uparrow U\ A \Rightarrow [\psi_1, \psi_2], t_1 - k_2, \Phi_1 \wedge \Phi_2} \textbf{ alg-r-switch}\uparrow$$

$$\frac{\begin{array}{c} k_1, t_1, k_2, t_2 \in \text{fresh}(\mathbb{R}) \\ \Delta; k_1, t_1, \psi_a; \Phi; |\Gamma| \vdash e_1 \downarrow A, k_1, t_1 \Rightarrow \Phi_1 \qquad \Delta; k_2, t_2, \psi_a; \Phi; |\Gamma| \vdash e_2 \downarrow A, k_2, t_2 \Rightarrow \Phi_2 \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1 \ominus \texttt{switch}\ e_2 \downarrow t, U\ A \Rightarrow \exists k_1, t_1 :: \mathbb{R}.(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 - k_2 \doteq t)} \textbf{ alg-r-switch}\downarrow$$

$\boxed{\texttt{NC}\ e}$

$$\frac{t' \in \text{fresh}(\mathbb{R}) \qquad \Delta; t', \psi_a; \Phi_a; \square\Gamma \vdash e \ominus e \downarrow \tau, t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\Gamma \vdash \texttt{NC}\ e \ominus \texttt{NC}\ e \downarrow \square\tau, t \Rightarrow 0 \doteq t \wedge (\exists t' :: \mathbb{R}.\Phi)} \textbf{ alg-r-nochange-}\downarrow$$

$\boxed{\texttt{der}\ e}$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \square\tau \Rightarrow [\psi], t, \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \ominus \texttt{der}\ e_2 \uparrow \tau \Rightarrow [\psi], t, \Phi} \textbf{ alg-r-der-}\uparrow$$

$\boxed{\text{Constant Integers } \texttt{n} \text{ and unit}}$

$$\frac{}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{n} \uparrow \text{int} \Rightarrow [.], 0, 0, \top} \textbf{ alg-u-n-}\uparrow \qquad \frac{}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{n} \ominus \texttt{n} \uparrow \text{int}_r \Rightarrow [.], 0, \top} \textbf{ alg-r-n-}\uparrow$$

$$\frac{}{\Delta; \psi_a; \Phi_a; \Omega \vdash () \uparrow \text{unit} \Rightarrow [.], 0, 0, \top} \textbf{ alg-u-unit-}\uparrow$$

$$\frac{}{\Delta; \psi_a; \Phi_a; \Gamma \vdash () \ominus \texttt{n} \uparrow \text{unit}_r \Rightarrow [.], 0, \top} \textbf{ alg-r-unit-}\uparrow$$

$\boxed{\text{Variables } x}$

$$\frac{\Omega(x) = A}{\Delta; \psi; \Phi_a; \Omega \vdash x \uparrow A \Rightarrow [.], 0, 0, \top} \textbf{ alg-u-var-}\uparrow \qquad \frac{\Gamma(x) = \tau}{\Delta; \psi_a; \Phi_a; \Gamma \vdash x \ominus x \uparrow \tau \Rightarrow [.], 0, \top} \textbf{ alg-r-var-}\uparrow$$

Figure 86: Algorithmic typing rules (part 1)

$\boxed{\text{inl } e}$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A_1, k, t \Rightarrow \Phi \qquad \Delta, \psi_a; \Phi_a \vdash^{\mathsf{A}} A_2 \ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{inl } e \downarrow A_1 + A_2, k, t \Rightarrow \Phi} \ \textbf{alg-u-inl-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow \tau_1, t \Rightarrow \Phi \qquad \Delta, \psi_a; \Phi_a \vdash \tau_2 \ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{inl } e_1 \ominus \text{inl } e_2 \downarrow \tau_1 + \tau_2, t \Rightarrow \Phi} \ \textbf{alg-r-inl-}\downarrow$$

$\boxed{\text{inr } e}$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A_2, k, t \Rightarrow \Phi \qquad \Delta, \psi_a; \Phi_a \vdash^{\mathsf{A}} A_1 \ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{inr } e \downarrow A_1 + A_2, k, t \Rightarrow \Phi} \ \textbf{alg-u-inr-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow \tau_2, t \Rightarrow \Phi \qquad \Delta, \psi_a; \Phi_a \vdash \tau_1 \ \texttt{wf}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{inr } e_1 \ominus \text{inr } e_2 \downarrow \tau_1 + \tau_2, t \Rightarrow \Phi} \ \textbf{alg-r-inr-}\downarrow$$

$\boxed{\text{case } (e, x.e_1, y.e_2)}$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A_1 + A_2 \Rightarrow [\psi], k_e, t_e, \Phi_1 \\ k', t' \in \text{fresh}(\mathbb{R}) \qquad \Delta; k', t', \psi, \psi_a; \Phi_a; \Omega, x : A_1 \vdash e_1 \downarrow A, k', t' \Rightarrow \Phi_2 \\ \Delta; k', t', \psi, \psi_a; \Phi_a; \Omega, y : A_2 \vdash e_2 \downarrow A, k', t' \Rightarrow \Phi_3 \\ \Phi = \exists(\psi).\Phi_1 \wedge (\exists k', t' :: \mathbb{R}.\Phi_2 \wedge \Phi_3 \wedge k \doteq k' + k_e + c_{case} \wedge t' + t_e + c_{case} \doteq t)\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{ case } (e, x.e_1, y.e_2) \downarrow A, k, t \Rightarrow \Phi} \ \textbf{alg-u-case-}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \tau_1 + \tau_2 \Rightarrow [\psi], t_e, \Phi_1 \\ t' \in \text{fresh}(\mathbb{R}) \qquad \Delta; t', \psi, \psi_a; \Phi_a; \Gamma, x : \tau_1 \vdash e_1 \ominus e_1' \downarrow \tau, t' \Rightarrow \Phi_2 \\ \Delta; t', \psi, \psi_a; \Phi_a; \Gamma, y : \tau_2 \vdash e_2 \ominus e_2' \downarrow \tau, t' \Rightarrow \Phi_3 \qquad \Phi = \exists(\psi).\Phi_1 \wedge (\exists t' :: \mathbb{R}.\Phi_2 \wedge \Phi_3 \wedge (t' + t_e \doteq t))\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{ case } (e, x.e_1, y.e_2) \ominus \text{ case } (e', x.e_1', y.e_2') \downarrow \tau, t \Rightarrow \Phi} \ \textbf{alg-r-case-}\downarrow$$

$\boxed{\text{fix } f(x).e}$

$$\frac{\Delta; \psi_a; \Phi_a; f : A_1 \xrightarrow{\text{exec}(k',t')} A_2, x : A_1, \Omega \vdash e \downarrow A_2, k', t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{fix } f(x).e \downarrow A_1 \xrightarrow{\text{exec}(k',t')} A_2, k, t \Rightarrow \Phi \wedge k \doteq 0 \wedge 0 \doteq t} \ \textbf{alg-u-fix-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \tau_1 \xrightarrow{\text{diff}(t')} \tau_2, x : \tau_1, \Gamma \vdash e \ominus e' \downarrow \tau_2, t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{fix } f(x).e \ominus \text{fix } f(x).e' \downarrow \tau_1 \xrightarrow{\text{diff}(t')} \tau_2, t \Rightarrow \Phi \wedge 0 \doteq t} \ \textbf{alg-r-fix-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; f : \square\,(\tau_1 \xrightarrow{\text{diff}(t')} \tau_2), x : \tau_1, \square\,\Gamma \vdash e \ominus e' \downarrow \tau_2, t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\,\Gamma \vdash \text{fix}_{NC}\ f(x).e \ominus \text{fix}_{NC}\ f(x).e \downarrow \square\,(\tau_1 \xrightarrow{\text{diff}(t')} \tau_2), t \Rightarrow \Phi \wedge 0 \doteq t} \ \textbf{alg-r-fix-}\downarrow\square$$

Figure 87: Algorithmic typing rules (part 2)

$\boxed{e_1\ e_2}$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;\Omega \vdash e_1 \uparrow A_1 \xrightarrow{\text{exec}(k_e,t_e)} A_2 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ k_2, t_2 \in \text{fresh}(\mathbb{R}) \qquad \Delta;k_2,t_2,\psi,\psi_a;\Phi_a;\Omega \vdash e_2 \downarrow A_1, k_2, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Omega \vdash e_1\ e_2 \uparrow A_2 \Rightarrow [k_2,t_2,\psi], k_1+k_2+k_e+c_{app}, t_1+t_2+t_e+c_{app}, \Phi_1 \wedge \Phi_2}\ \textbf{alg-u-app-}\uparrow$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;\Gamma \vdash e_1 \ominus e_1' \uparrow \tau_1 \xrightarrow{\text{diff}(t_e)} \tau_2 \Rightarrow [\psi], t_1, \Phi_1 \\ t_2 \in \text{fresh}(\mathbb{R}) \qquad \Delta;t_2,\psi,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow \tau_1, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \uparrow \tau_2 \Rightarrow [t_2,\psi], t_1+t_2+t_e, \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-app-}\uparrow$$

$\boxed{(e_1, e_2)}$

$$\frac{\begin{array}{c}k_1,t_1,k_2,t_2 \in \text{fresh}(\mathbb{R}) \\ \Delta;k_1,t_1,\psi_a;\Phi_a;\Omega \vdash e_1 \downarrow A_1, k_1, t_1 \Rightarrow \Phi_1 \qquad \Delta;k_2,t_2,\psi_a;\Phi_a;\Omega \vdash e_2 \downarrow A_1, k_2, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Omega \vdash \langle e_1, e_2\rangle \downarrow A_1 \times A_2, k, t \Rightarrow \exists k_1,t_1 :: \mathbb{R}.\Phi_1 \wedge \exists k_2,t_2 :: \mathbb{R}.\Phi_2 \wedge t_1+t_2 \doteq t \wedge k \doteq k_1+k_2}\ \textbf{alg-u-prod-}\downarrow$$

$$\frac{\begin{array}{c}t_1,t_2 \in \text{fresh}(\mathbb{R}) \\ \Delta;t_1,\psi_a;\Phi_a;\Gamma \vdash e_1 \ominus e_1' \downarrow \tau_1, t_1 \Rightarrow \Phi_1 \qquad \Delta;t_1,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow \tau_1, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Omega \vdash \langle e_1, e_2\rangle \ominus \langle e_1', e_2'\rangle \downarrow \tau_1 \times \tau_2, t \Rightarrow \exists t_1 :: \mathbb{R}.\Phi_1 \wedge \exists t_1 :: \mathbb{R}.\Phi_2 \wedge t_1+t_2 \doteq t}\ \textbf{alg-r-prod-}\downarrow$$

$\boxed{\pi_1(e)}$

$$\frac{\Delta;\psi_a;\Phi_a;\Omega \vdash e \uparrow A_1 \times A_2 \Rightarrow [\psi], k, t, \Phi}{\Delta;\psi_a;\Phi_a;\Omega \vdash \pi_1(e) \uparrow A_1 \Rightarrow [\psi], k, t, \Phi}\ \textbf{alg-u-proj1-}\uparrow$$

$$\frac{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus e' \uparrow \tau_1 \times \tau_2 \Rightarrow [\psi], t, \Phi}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \pi_1(e) \ominus \pi_1(e') \uparrow \tau_1 \Rightarrow [\psi], t, \Phi}\ \textbf{alg-r-proj1-}\uparrow$$

$\boxed{\text{nil}}$

$$\frac{\Delta,\psi_a;\Phi \vdash^{\mathsf{A}} A\ \texttt{wf}}{\Delta;\psi_a;\Phi_a;\Omega \vdash \text{nil} \downarrow \text{list}[n]\, A, k, t \Rightarrow n \doteq 0 \wedge k \doteq 0 \wedge 0 \doteq t}\ \textbf{alg-u-nil-}\downarrow$$

$$\frac{\Delta,\psi_a;\Phi \vdash \tau\ \texttt{wf}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \text{nil} \ominus \text{nil} \downarrow \text{list}[n]^{\alpha}\, \tau, t \Rightarrow n \doteq 0 \wedge 0 \doteq t}\ \textbf{alg-r-nil-}\downarrow$$

$\boxed{\text{cons}(e_1, e_2)}$

$$\frac{\begin{array}{c}k_1,t_1,k_2,t_2 \in \text{fresh}(\mathbb{R}) \qquad i \in \text{fresh}(\mathbb{N}) \\ \Delta;k_1,t_1,\psi_a;\Phi_a;\Omega \vdash e_1 \downarrow A, k_1, t_1 \Rightarrow \Phi_1 \qquad \Delta;i,k_2,t_2,\psi_a;\Phi_a;\Omega \vdash e_2 \downarrow \text{list}[i]\, A, k_2, t_2 \Rightarrow \Phi_2 \\ \Phi_2' = (\Phi_2 \wedge n \doteq (i+1) \wedge k \doteq k_1+k_2 \wedge t_1+t_2 \doteq t)\end{array}}{\Delta;\psi_a;\Phi_a;\Omega \vdash \text{cons}_C(e_1,e_2) \downarrow \text{list}[n]\, A, k, t \Rightarrow \exists k_1,t_1 :: \mathbb{R}.(\Phi_1 \wedge \exists k_2,t_2 :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2')}\ \textbf{alg-u-cons-}\downarrow$$

$$\frac{\begin{array}{c}t_1,t_2 \in \text{fresh}(\mathbb{R}) \qquad i,\beta \in \text{fresh}(\mathbb{N}) \\ \Delta;t_1,\psi_a;\Phi_a;\Gamma \vdash e_1 \ominus e_1' \downarrow \tau, t_1 \Rightarrow \Phi_1 \qquad \Delta;i,\beta,t_2,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^{\beta}\, \tau, t_2 \Rightarrow \Phi_2 \\ \Phi_2' = n \doteq (i+1) \wedge \exists \beta :: \mathbb{N}.\Phi_2 \wedge \alpha \doteq \beta+1 \wedge t_1+t_2 \doteq t\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \text{cons}_C(e_1,e_2) \ominus \text{cons}_C(e_1',e_2') \downarrow \text{list}[n]^{\alpha}\, \tau, t \Rightarrow \exists t_1 :: \mathbb{R}.(\Phi_1 \wedge \exists t_1,t_2 :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2')}\ \textbf{alg-r-consC-}\downarrow$$

$$\frac{\begin{array}{c}t_1,t_2 \in \text{fresh}(\mathbb{R}) \qquad i \in \text{fresh}(\mathbb{N}) \qquad \Delta;t_1,\psi_a;\Phi_a;\Gamma \vdash e_1 \ominus e_1' \downarrow \Box\tau, t_1 \Rightarrow \Phi_1 \\ \Delta;i,t_2,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow \text{list}[i]^{\alpha}\, \tau, t_2 \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \wedge t_1+t_2 \doteq t\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \text{cons}_{NC}(e_1,e_2) \ominus \text{cons}_{NC}(e_1',e_2') \downarrow \text{list}[n]^{\alpha}\, \tau, t \Rightarrow \exists t_1 :: \mathbb{R}.(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2')}\ \textbf{alg-r-consNC-}\downarrow$$

Figure 88: Algorithmic typing rules (part 3)

$\boxed{\mathrm{case}\ e\ \mathrm{of\ nil} \to e_1 \mid h :: tl \to e_2}$

$$\dfrac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow \mathrm{list}[n]\, A \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ k_2, t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta; k_2, t_2, \psi, \psi_a; n \doteq 0 \wedge \Phi_a; \Omega \vdash e_1 \downarrow A', k_2, t_2 \Rightarrow \Phi_2 \\ i \in \mathrm{fresh}(\mathbb{N}) \qquad i :: \mathbb{N}, \Delta; k_2, t_2, \psi, \psi_a; n \doteq i + 1 \wedge \Phi_a; h : A, tl : \mathrm{list}[i]\, A, \Omega \vdash e_2 \downarrow A', k_2, t_2 \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_2) \wedge (\forall i :: \mathbb{N}.(n \doteq i + 1) \to \Phi_3) \wedge k \doteq (k_1 + k_2 + c_{caseL}) \wedge (t_1 + t_2 + c_{caseL}) \doteq t\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \begin{array}{l}\mathrm{case}\ e\ \mathrm{of\ nil} \to e_1 \\ \mid h ::_{NC} tl \to e_2 \\ \mid h ::_C tl \to e_3\end{array} \downarrow A', k, t \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_{body})}\ \textbf{alg-u-caseL-}\!\downarrow$$

$$\dfrac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \mathrm{list}[n]^{\alpha}\, \tau \Rightarrow [\psi], t_1, \Phi_e \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta; t_2, \psi, \psi_a; n \doteq 0 \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau', t_2 \Rightarrow \Phi_1 \\ i :: \mathbb{N}, \Delta; t_2, \psi, \psi_a; n \doteq i + 1 \wedge \Phi_a; h : \square\tau, tl : \mathrm{list}[i]^{\alpha}\, \tau, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau', t_2 \Rightarrow \Phi_2 \\ i :: \mathbb{N}, \beta :: \mathbb{N}, \Delta; t_2, \psi, \psi_a; n \doteq i + 1 \wedge \alpha \doteq \beta + 1 \wedge \Phi_a; h : \tau, tl : \mathrm{list}[i]^{\beta}\, \tau, \Gamma \vdash e_3 \ominus e_3' \downarrow \tau', t_2 \Rightarrow \Phi_3 \\ \Phi_{body} = (n \doteq 0 \to \Phi_1) \wedge (\forall i :: \mathbb{N}.(n \doteq i + 1) \to (\Phi_2 \wedge \forall \beta :: \mathbb{N}.(\alpha \doteq \beta + 1) \to \Phi_3)) \wedge t_1 + t_2 \doteq t\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \begin{array}{l}\mathrm{case}\ e\ \mathrm{of\ nil} \to e_1 \\ \mid h ::_{NC} tl \to e_2 \\ \mid h ::_C tl \to e_3\end{array} \ominus \begin{array}{l}\mathrm{case}\ e'\ \mathrm{of\ nil} \to e_1' \\ \mid h ::_{NC} tl \to e_2' \\ \mid h ::_C tl \to e_3'\end{array} \downarrow \tau', t \Rightarrow \exists(\psi).(\Phi_e \wedge \exists t_2 :: \mathbb{R}.\Phi_{body})}\ \textbf{alg-r-caseL-}\!\downarrow$$

$\boxed{\Lambda i.e}$

$$\dfrac{i :: S, \Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k_e, t_e \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \Lambda i.e \downarrow \forall i \overset{\mathrm{exec}(k_e, t_e)}{::} S.\, A, k, t \Rightarrow (\forall i :: S.\Phi) \wedge k \doteq 0 \wedge 0 \doteq t}\ \textbf{alg-u-iLam-}\!\downarrow$$

$$\dfrac{i :: S, \Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau, t_e \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \Lambda i.e \ominus \Lambda i.e' \downarrow \forall i \overset{\mathrm{diff}(t_e)}{::} S.\, \tau, t \Rightarrow (\forall i :: S.\Phi) \wedge 0 \doteq t}\ \textbf{alg-r-iLam-}\!\downarrow$$

$\boxed{e\,[I]}$

$$\dfrac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow \forall i \overset{\mathrm{exec}(k_e, t_e)}{::} S.\, A' \Rightarrow [\psi], k, t, \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash e\,[I] \uparrow A'\{I/i\} \Rightarrow [\psi], k + k_e[I/i], t + t_e[I/i], \Phi}\ \textbf{alg-u-iApp-}\!\uparrow$$

$$\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \forall i \overset{\mathrm{diff}(t_e)}{::} S.\, \tau' \Rightarrow [\psi], t, \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e\,[I] \ominus e'\,[I] \uparrow \tau'\{I/i\} \Rightarrow [\psi], t + t_e[I/i], \Phi}\ \textbf{alg-r-iApp-}\!\uparrow$$

$\boxed{\mathrm{pack}\ e\ \mathrm{with}\ I}$

$$\dfrac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A\{I/i\}, k, t \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash \mathrm{pack}\ e\ \mathrm{with}\ I \downarrow \exists i{::}S.\, A, k, t \Rightarrow \Phi}\ \textbf{alg-u-pack-}\!\downarrow$$

$$\dfrac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau\{I/i\}, t \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathrm{pack}\ e\ \mathrm{with}\ I \ominus \mathrm{pack}\ e'\ \mathrm{with}\ I \downarrow \exists i{::}S.\, \tau, t \Rightarrow \Phi}\ \textbf{alg-r-pack-}\!\downarrow$$

Figure 89: Algorithmic typing rules (part 4)

$\boxed{\text{unpack } e \text{ as } (x, i) \text{ in } e'}$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow \exists i{::}S.\, A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \qquad k_2, t_2 \in \text{fresh}(\mathbb{R}) \\ i :: S, \Delta; k_2, t_2, \psi, \psi_a; \Phi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2, k_2, t_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Omega, A_2, k_2, t_2) \\ \Phi = \exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\forall i :: S.\Phi_2 \wedge k \doteq k_1 + k_2 + c_{unpack} \wedge t_1 + t_2 + c_{unpack} \doteq t)\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \downarrow A_2, k, t \Rightarrow \Phi} \textbf{ alg-u-unpack-}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \exists i{::}S.\, \tau_1 \Rightarrow [\psi], t_1, \Phi_1 \\ t_2 \in \text{fresh}(\mathbb{R}) \qquad i :: S, \Delta; t_2, \psi, \psi_a; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2, t_2 \Rightarrow \Phi_2 \\ i \notin FV(\Phi_a; \Gamma, \tau_2, t_2) \qquad \Phi = \exists(\psi).(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\forall i :: S.\Phi_2 \wedge t_1 + t_2 \doteq t)\end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } (x, i) \text{ in } e_2' \downarrow \tau_2, t \Rightarrow \Phi} \textbf{ alg-r-unpack-}\downarrow$$

$\boxed{\zeta\ e}$

$$\frac{\Upsilon(\zeta) : A_1 \xrightarrow{\text{exec}(k_e, t_e)} A_2 \qquad k, t \in \text{fresh}(\mathbb{R}) \qquad \Delta; k, t, \psi_a; \Phi_a; \Omega \vdash e \downarrow A_1, k, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \zeta\ e \uparrow A_2 \Rightarrow [k, t, \psi], k + k_e, t + t_e, \Phi} \textbf{ alg-u-primapp-}\uparrow$$

$$\frac{\Upsilon(\zeta) : \tau_1 \xrightarrow{\text{diff}(t_e)} \tau_2 \qquad t \in \text{fresh}(\mathbb{R}) \qquad \Delta; t, \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \downarrow \tau_1, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \zeta e_1 \ominus \zeta\ e_2 \uparrow \tau_2 \Rightarrow [t, \psi], t + t_e, \Phi} \textbf{ alg-r-primapp-}\uparrow$$

$\boxed{C\ \&\ \cdot\ \text{intro}}$

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow k, t, C\ \&\ A \Rightarrow C \wedge (C \to \Phi)} \textbf{ alg-u-c-andI}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi \wedge C; \Gamma \vdash e_1 \ominus e_1 \downarrow \tau, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \ominus e_2 \downarrow C\ \&\ \tau, t \Rightarrow C \wedge (C \to \Phi)} \textbf{ alg-r-c-andI}\downarrow$$

$\boxed{\text{clet } e_1 \text{ as } x \text{ in } e_2}$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow C\ \&\ A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ k_2, t_2 \in \text{fresh}(\mathbb{R}) \qquad \Delta; k_2, t_2, \psi, \psi_a; \Phi \wedge C; x : A_1, \Omega \vdash e_2 \downarrow A_2, k_2, t_2 \Rightarrow \Phi_2 \\ \Phi_2' = C \to \Phi_2 \wedge k \doteq (k_1 + k_2) \wedge (t_1 + t_2) \doteq t\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \downarrow A_2, k, t \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2')} \textbf{ alg-u-c-andE}\downarrow$$

$$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \ominus e_1' \uparrow C\ \&\ \tau_1 \Rightarrow [\psi], t_1, \Phi_1 \qquad t_2 \in \text{fresh}(\mathbb{R}) \\ \Delta; t_2, \psi, \psi_a; \Phi \wedge C; x : \tau_1, \Omega \vdash e_2 \ominus e_2' \downarrow \tau_2, t_2 \Rightarrow \Phi_2 \qquad \Phi_2' = C \to \Phi_2 \wedge (t_1 + t_2) \doteq t\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{clet } e_1 \text{ as } x \text{ in } e_2 \ominus \text{clet } e_1' \text{ as } x \text{ in } e_2' \downarrow \tau_2, t \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2')} \textbf{ alg-r-c-andE}\downarrow$$

Figure 90: Algorithmic typing rules (part 5)

$\boxed{C \supset \cdot \text{ intro}}$

$$\frac{\Delta; \Phi \wedge C; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow C \supset A, k, t \Rightarrow C \rightarrow \Phi} \textbf{ alg-u-c-implI} \downarrow$$

$$\frac{\Delta; \Phi \wedge C; \Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow C \supset \tau, t \Rightarrow C \rightarrow \Phi} \textbf{ alg-r-c-implI}\downarrow$$

$\boxed{\text{celim } e}$

$$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow C \supset A \Rightarrow [\psi], k, t, \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{celim } e \uparrow A \Rightarrow [\psi], k, t, C \wedge \Phi} \textbf{ alg-u-c-implE}\uparrow$$

$$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow C \supset \tau \Rightarrow [\psi], t, \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{celim } e \ominus \text{celim } e' \uparrow \tau \Rightarrow [\psi], t, C \wedge \Phi} \textbf{ alg-r-c-implE}\uparrow$$

$\boxed{\text{let } x = e_1 \text{ in } e_2}$

$$\frac{\begin{array}{c} \Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ k_2, t_2 \in \text{fresh}(\mathbb{R}) \qquad \Delta; k_2, t_2, \psi, \psi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2, k_2, t_2 \Rightarrow \Phi_2 \\ \Phi_2' = \Phi_2 \wedge k \doteq (k_1 + k_2 + c_{let}) \wedge (t_1 + t_2 + c_{let}) \doteq t \end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{let } x = e_1 \text{ in } e_2 \downarrow A_2, k, t \Rightarrow \exists(\psi).\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2'} \textbf{ alg-u-let-}\downarrow$$

$$\frac{\begin{array}{c} \Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \uparrow \tau_1 \Rightarrow [\psi], t_1, \Phi_1 \\ t_2 \in \text{fresh}(\mathbb{R}) \qquad \Delta; t_2, \psi, \psi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \downarrow \tau_2, t_2 \Rightarrow \Phi_2 \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \ominus \text{let } x = e_1' \text{ in } e_2' \downarrow \tau_2, t \Rightarrow \exists(\psi).\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 \doteq t} \textbf{ alg-r-let-}\downarrow$$

$\boxed{\texttt{split } e}$

$$\frac{\begin{array}{c} \Delta; \psi_a; C \wedge \Phi_a; \Omega \vdash e_1 \downarrow A, k, t \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Omega \vdash e_2 \downarrow A, k, t \Rightarrow \Phi_2 \qquad \Delta \vdash C \texttt{ wf} \end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{split } (e_1, e_2) \text{ with } C \downarrow A, k, t \Rightarrow C \rightarrow \Phi_1 \wedge \neg C \rightarrow \Phi_2} \textbf{ alg-u-split}\downarrow$$

$$\frac{\begin{array}{c} \Delta; \psi_a; C \wedge \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \tau, t \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \tau, t \Rightarrow \Phi_2 \qquad \Delta \vdash C \texttt{ wf} \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{split } (e_1, e_2) \text{ with } C \ominus \texttt{split } (e_1', e_2') \text{ with } C \downarrow \tau, t \Rightarrow C \rightarrow \Phi_1 \wedge \neg C \rightarrow \Phi_2} \textbf{ alg-r-split}\downarrow$$

$\boxed{\texttt{contra } e}$

$$\frac{\Delta; \psi_a; \Phi_a \models \bot}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{contra } e \downarrow A, k, t \Rightarrow \top} \textbf{ alg-u-contra}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a \models \bot}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{contra } e \ominus \texttt{contra } e' \downarrow \tau, t \Rightarrow \top} \textbf{ alg-r-contra}\downarrow$$

Figure 91: Algorithmic typing rules (part 6)

$\boxed{\uparrow\downarrow}$

$$\frac{\Delta;\psi_a;\Phi_a;\Omega \vdash e \uparrow A' \Rightarrow [\psi], k', t', \Phi_1 \qquad \Delta;\psi,\psi_a;\Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi_2}{\Delta;\psi_a;\Phi_a;\Omega \vdash e \downarrow A, k, t \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2 \wedge t' \leq t \wedge k \leq k'} \textbf{ alg-}\uparrow\downarrow$$

$$\frac{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus e' \uparrow \tau' \Rightarrow [\psi], t', \Phi_1 \qquad \Delta;\psi,\psi_a;\Phi_a \models \tau' \equiv \tau \Rightarrow \Phi_2}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2 \wedge t' \leq t} \textbf{ alg-r-}\uparrow\downarrow$$

$\boxed{(e : A, k, t)}$

$$\frac{\Delta;\psi_a;\Phi_a;\Omega \vdash e \downarrow A, k, t \Rightarrow \Phi \qquad \Delta;\Phi_a \vdash^{\mathsf{A}} A\ \mathtt{wf} \qquad \mathrm{FIV}(A,k,t) \in \Delta}{\Delta;\psi_a;\Phi_a;\Omega \vdash (e : A, k, t) \uparrow A \Rightarrow [\cdot], k, t, \Phi} \textbf{ alg-u-anno-}\uparrow$$

$\boxed{(e : \tau, t)}$

$$\frac{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \Phi \qquad \Delta;\Phi_a \vdash \tau\ \mathtt{wf} \qquad \mathrm{FIV}(\tau,t) \in \Delta}{\Delta;\psi_a;\Phi_a;\Gamma \vdash (e : \tau, t) \ominus (e' : \tau, t) \uparrow \tau \Rightarrow [\cdot], t, \Phi} \textbf{ alg-r-anno-}\uparrow$$

$\boxed{\text{Asynchronous rules}}$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;|\Gamma|_1 \vdash e_1 \uparrow A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;t_2,\psi,\psi_a;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e_2 \ominus e \downarrow \tau_2, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathrm{let}\ x = e_1\ \mathrm{in}\ e_2 \ominus e \downarrow \tau_2, t \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 + c_{let} \doteq t)} \textbf{ alg-r-let-e}\downarrow$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;|\Gamma|_2 \vdash e_1 \uparrow A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;t_2,\psi,\psi_a;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e \ominus e_2 \downarrow \tau_2, t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus \mathrm{let}\ x = e_1\ \mathrm{in}\ e_2 \downarrow \tau_2, t \Rightarrow \exists(\psi).(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_2 - k_1 - c_{let} \doteq t)} \textbf{ alg-r-e-let}\downarrow$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi;|\Gamma|_1 \vdash e_1 \uparrow A_1 \xrightarrow{\mathrm{exec}(k_e,t_e)} A_2 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;t_2,\psi,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow U\,(A_1,A_2'), t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e_1\ e_2 \ominus e_2' \downarrow U\,(A_2,A_2'), t \Rightarrow \exists(\psi).\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 + t_e + c_{app} \doteq t} \textbf{ alg-r-app-e}\downarrow$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi;|\Gamma|_2 \vdash e_1' \uparrow A_1' \xrightarrow{\mathrm{exec}(k_e,t_e)} A_2' \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;t_2,\psi,\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_2' \downarrow U\,(A_2,A_1'), t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e_2 \ominus e_1'\ e_2' \downarrow U\,(A_2,A_2'), t \Rightarrow \exists(\psi).\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_2 - k_1 - k_e + c_{app} \doteq t} \textbf{ alg-r-e-app}\downarrow$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;|\Gamma|_2 \vdash e \uparrow A_1 + A_2 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;\psi_a;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e_1 \ominus e' \downarrow \tau, t_2 \Rightarrow \Phi_2 \\ \Delta;\psi_a;\Phi_a;y : U\,(A_2,A_2),\Gamma \vdash e_2 \ominus e' \downarrow \tau, t_2 \Rightarrow \Phi_3\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \ \mathrm{case}\ (e, x.e_1, y.e_2) \ominus e' \downarrow \tau, t \Rightarrow \exists(\phi).\Phi_1 \wedge (\exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 + c_{case} \doteq t)} \textbf{ alg-r-case-e}\downarrow^{-}$$

$$\frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;|\Gamma|_2 \vdash e' \uparrow A_1 + A_2 \Rightarrow [\psi], k_1, t_1, \Phi_1 \\ t_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Delta;\psi_a;\Phi_a;x : U\,(A_1,A_1),\Gamma \vdash e \ominus e_1' \downarrow \tau, t_2 \Rightarrow \Phi_2 \\ \Delta;\psi_a;\Phi_a;y : U\,(A_2,A_2),\Gamma \vdash e \ominus e_2' \downarrow \tau, t_2 \Rightarrow \Phi_3\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e \ominus\ \mathrm{case}\ (e', x.e_1', y.e_2') \downarrow \tau, t \Rightarrow \exists(\phi).\Phi_1 \wedge (\exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_2 - k_1 - c_{case} \doteq t)} \textbf{ alg-r-e-case}\downarrow^{-}$$

Figure 92: Algorithmic typing rules (part 7)

$\boxed{\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \Rightarrow \Phi}$ checks whether A_1 is subtype of A_2 and generates constraints Φ

$\boxed{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_2 \Rightarrow \Phi}$ checks whether τ_1 is equivalent to τ_2 and generates constraints Φ

$$\frac{}{\Delta; \psi_a; \Phi_a \models \mathrm{int}_r \equiv \mathrm{int}_r \Rightarrow \top}\ \textbf{alg-r-int} \qquad \frac{}{\Delta; \psi_a; \Phi_a \models \mathrm{unit}_r \equiv \mathrm{unit}_r \Rightarrow \top}\ \textbf{alg-r-unit}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a \models \tau_1 \xrightarrow{\mathrm{diff}(t)} \tau_2 \equiv \tau_1' \xrightarrow{\mathrm{diff}(t')} \tau_2' \Rightarrow \Phi_1 \wedge \Phi_2 \wedge t \doteq t'}\ \textbf{alg-r-fun}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a \models \tau_1 \times \tau_2 \equiv \tau_1' \times \tau_2' \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-prod}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a \models \tau_1 + \tau_2 \equiv \tau_1' + \tau_2' \Rightarrow \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-sum}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \mathrm{list}[n]^{\alpha}\, \tau \equiv \mathrm{list}[n']^{\alpha'}\, \tau' \Rightarrow \Phi \wedge n \doteq n' \wedge \alpha \doteq \alpha'}\ \textbf{alg-r-list}$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \forall i \overset{\mathrm{diff}(t)}{::} S.\, \tau \equiv \forall i \overset{\mathrm{diff}(t')}{::} S.\, \tau' \Rightarrow \forall i :: S.\Phi \wedge t \doteq t'}\ \forall$$

$$\frac{i, \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad i \notin FV(\Phi_a)}{\Delta; \psi_a; \Phi_a \models \exists i{::}S.\, \tau \equiv \exists i{::}S.\, \tau' \Rightarrow \forall i :: S.\Phi}\ \textbf{alg-r-}\exists \qquad \frac{\Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau_2 \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models \Box\, \tau_1 \equiv \Box\, \tau_2 \Rightarrow \Phi}\ \textbf{B-}\Box$$

$$\frac{\begin{array}{ll} \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_1' \Rightarrow \Phi_1 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1' \\ \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2 & \Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A_2' \sqsubseteq A_2 \Rightarrow \Phi_2' \end{array}}{\Delta; \psi_a; \Phi_a \models U\,(A_1, A_2) \equiv U\,(A_1', A_2') \Rightarrow \Phi_1 \wedge \Phi_1' \wedge \Phi_2 \wedge \Phi_2'}\ \textbf{U}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models C \supset \tau \equiv C' \supset \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\ \textbf{c-impl}$$

$$\frac{\Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a \models C \;\&\; \tau \equiv C' \;\&\; \tau' \Rightarrow C \leftrightarrow C' \wedge \Phi}\ \textbf{c-prod}$$

Figure 93: Algortihmic type equivalence rules

$$
\begin{array}{lcl}
|.| & : & \text{Expression} \rightarrow \text{Expression} \\
\\
|\mathtt{n}| & = & \mathtt{n} \\
|()| & = & () \\
|x| & = & x \\
|\text{fix } f(x).e| & = & \text{fix } f(x).|e| \\
|\text{fix}_{NC}\ f(x).e| & = & \text{fix}_{NC}\ f(x).|e| \\
|e_1\ e_2| & = & |e_1|\ |e_2| \\
\vdots & & \\
|(e : A, k, t)| & = & |e| \\
|(e : \tau, t)| & = & |e|
\end{array}
$$

Figure 94: Annotation erasure

5.1 Metatheory

Lemma 36 (Embedding of Binary Subtyping in RelCost)
If $\Delta; \Phi \models \tau \sqsubseteq \tau'$ then $\exists e \in \mathsf{RelCostCore}$ such that $\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c \tau \xrightarrow{\text{diff}(0)} \tau'$.

Proof. Proof is by induction on the subtyping derivation. We denote the witness e of type $\tau \xrightarrow{\text{diff}(0)} \tau'$ as $\mathtt{coerce}_{\tau,\tau'}$ for clarity.

Case $$\frac{\Delta; \Phi_a \models \tau_1' \sqsubseteq \tau_1 \quad (\star) \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2' \quad (\diamond) \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a \models \tau_1 \xrightarrow{\mathbf{diff}(t)} \tau_2 \sqsubseteq \tau_1' \xrightarrow{\mathbf{diff}(t')} \tau_2'} \rightarrow \mathbf{diff}$$

By IH on $(\star)$, $\exists \mathtt{coerce}_{\tau_1',\tau_1}$. $\Delta; \Phi; \cdot \vdash \mathtt{coerce}_{\tau_1',\tau_1} \ominus \mathtt{coerce}_{\tau_1',\tau_1} \lesssim 0 :^c \tau_1' \xrightarrow{\text{diff}(0)} \tau_1$
By IH on $(\diamond)$, $\exists \mathtt{coerce}_{\tau_2,\tau_2'}$. $\Delta; \Phi; \cdot \vdash \mathtt{coerce}_{\tau_2,\tau_2'} \ominus \mathtt{coerce}_{\tau_2,\tau_2'} \lesssim 0 :^c \tau_2 \xrightarrow{\text{diff}(0)} \tau_2'$

Then, using these two statements and $\Delta; \Phi \models t \leq t'$ with binary subeffecting rule (rule **c-r-$\sqsubseteq$** in Figure 84), we can construct the following derivation where $e = \lambda x.\lambda y.\mathtt{coerce}_{\tau_2,\tau_2'}\ (x\ (\mathtt{coerce}_{\tau_1',\tau_1}\ \ y))$

$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c (\tau_1 \xrightarrow{\text{diff}(t)} \tau_2) \xrightarrow{\text{diff}(0)} \tau_1' \xrightarrow{\text{diff}(t')} \tau_2'$

Case $$\frac{}{\Delta; \Phi_a \models \mathbf{unit}_r \sqsubseteq \square\, \mathbf{unit}_r} \mathbf{unit}$$
Then, we can immediately construct the derivation using the rule **c-nochange** in Figure 80.

$$\frac{}{\Delta; \Phi; \cdot \vdash \lambda x.\mathtt{NC}\ () \ominus \lambda x.\mathtt{NC}\ () \lesssim 0 :^c \text{unit}_r \xrightarrow{\text{diff}(0)} \square\, \text{unit}_r}$$

Case $$\frac{}{\Delta; \Phi_a \models \mathbf{int}_r \sqsubseteq \square\, \mathbf{int}_r} \mathbf{int}\text{-}\square$$
Then, we can construct the derivation using the primitive function $\text{box}_{\text{int}} : \text{int}_r \xrightarrow{\text{diff}(0)} \square\, \text{int}_r$

$$\frac{}{\Delta; \Phi; \cdot \vdash \lambda x.\text{box}_{\text{int}} x \ominus \lambda x.\text{box}_{\text{int}} x \lesssim 0 :^c \text{int}_r \xrightarrow{\text{diff}(0)} \square\, \text{int}_r}$$

Case $$\frac{}{\Delta; \Phi_a \models \square\, U\, (\mathbf{int}, \mathbf{int}) \sqsubseteq \mathbf{int}_r} \square\mathbf{U\text{-}int}$$
Then, we can construct the derivation using the primitive function $\text{box}_U : \square\, U\, (\text{int}, \text{int}) \xrightarrow{\text{diff}(0)} \text{int}_r$

$$\frac{}{\Delta; \Phi; \cdot \vdash \lambda x.\text{box}_U x \ominus \lambda x.\text{box}_U x \lesssim 0 :^c \square\, U\, (\text{int}, \text{int}) \xrightarrow{\text{diff}(0)} \text{int}_r}$$

Case $$\frac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \tau} \mathbf{T}$$
Then, we can immediately construct the derivation using the rule **c-der** in Figure 80.

$$\frac{}{\Delta; \Phi; \cdot \vdash \lambda x.\mathtt{der}\ x \ominus \lambda x.\mathtt{der}\ x \lesssim 0 :^c \square\, \tau \xrightarrow{\text{diff}(0)} \tau}$$

Case $$\frac{}{\Delta; \Phi_a \models \square\, \tau \sqsubseteq \square\, \square\, \tau} \mathbf{D}$$
Then, we can immediately construct the derivation using the rule **c-nochange** in Figure 80.

$$\frac{}{\Delta; \Phi; \cdot \vdash \lambda x.\mathtt{NC}\ x \ominus \lambda x.\mathtt{NC}\ x \lesssim 0 :^c \square\, \tau \xrightarrow{\text{diff}(0)} \square\, \square\, \tau}$$

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 (\star)}{\Delta; \Phi_a \models \square\, \tau_1 \sqsubseteq \square\, \tau_2}$ **B-□**

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1,\tau_2}$. $\Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau_1,\tau_2} \ominus \texttt{coerce}_{\tau_1,\tau_2} \lesssim 0 :^c \tau_1 \xrightarrow{\text{diff}(0)} \tau_2$

Then, using $(\star)$ and the rules **c-der** and **c-nochange** in Figure 80, we can construct the derivation

$$\dfrac{}{\Delta; \Phi; \cdot \vdash \lambda x.\texttt{NC}\,(\texttt{coerce}_{\tau_1,\tau_2}\ (\texttt{der}\ x)) \ominus \lambda x.\texttt{NC}\,(\texttt{coerce}_{\tau_1,\tau_2}\ (\texttt{der}\ x)) \lesssim 0 :^c \square\, \tau_1 \xrightarrow{\text{diff}(0)} \square\, \tau_2}$$

Case $\dfrac{}{\Delta; \Phi_a \models \tau \sqsubseteq U\,|\tau|}$ **W**

Then, we can immediately construct the derivation using the rule **c-switch** in Figure 80.

$$\dfrac{}{\Delta; \Phi; \cdot \vdash \lambda x.\texttt{switch}\ x \ominus \lambda x.\texttt{switch}\ x \lesssim 0 :^c \tau \xrightarrow{\text{diff}(0)} U\,(|\tau|_1, |\tau|_2)}$$

Case $\dfrac{}{\Delta; \Phi_a \models \tau \sqsubseteq \tau}$**refl**

Then, we can immediately construct the derivation

$$\dfrac{}{\Delta; \Phi; \cdot \vdash \lambda x.x \ominus \lambda x.x \lesssim 0 :^c \tau \xrightarrow{\text{diff}(0)} \tau}$$

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_2 \quad (\star) \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_3 \quad (\diamond)}{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_3}$**trans**

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau_1,\tau_2}$. $i :: S, \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau_1,\tau_2} \ominus \texttt{coerce}_{\tau_1,\tau_2} \lesssim 0 :^c \tau_1 \xrightarrow{\text{diff}(0)} \tau_2$
By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau_2,\tau_3}$. $i :: S, \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau_2,\tau_3} \ominus \texttt{coerce}_{\tau_2,\tau_3} \lesssim 0 :^c \tau_2 \xrightarrow{\text{diff}(0)} \tau_3$

Then, using $(\star)$ and $(\diamond)$, we can construct the derivation simply by function composition

$$\dfrac{}{\Delta; \Phi; \cdot \vdash \lambda x.\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) \ominus \lambda x.\texttt{coerce}_{\tau_2,\tau_3}\ (\texttt{coerce}_{\tau_1,\tau_2}\ x) \lesssim 0 :^c \tau_1 \xrightarrow{\text{diff}(0)} \tau_3}$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,(\tau_1 \xrightarrow{\mathbf{diff}(t)} \tau_2) \sqsubseteq \square\, \tau_1 \xrightarrow{\mathbf{diff}(0)} \square\, \tau_2} \rightarrow \square\mathbf{diff}$

Then, we can immediately construct the derivation where $e = \lambda x.\lambda y.\texttt{NC}\ (\texttt{der}\ x)\ (\texttt{der}\ y))$
$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c \square\,(\tau_1 \xrightarrow{\text{diff}(k)} \tau_2) \xrightarrow{\text{diff}(0)} \square\, \tau_1 \xrightarrow{\text{diff}(0)} \square\, \tau_2$

Case $\dfrac{}{\Delta; \Phi_a \models U\,(A_1 \xrightarrow{\mathbf{exec}(k,t)} A_2) \sqsubseteq U\, A_1 \xrightarrow{\mathbf{diff}(t-k)} U\, A_2} \rightarrow \mathbf{execdiff}$

Then, we can immediately construct the following derivation where $e = \lambda x.\lambda y.\texttt{switch}\ (x\ y)$ using the **c-switch** and **c-app** rules.
$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c (U\,(A_1 \xrightarrow{\text{exec}(k,t)} A_2, A_1' \xrightarrow{\text{exec}(k',t')} A_2')) \xrightarrow{\text{diff}(0)} U\,(A_1, A_1') \xrightarrow{\text{diff}(t-k')} U\,(A_2, A_2')$

Case $\dfrac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\star) \qquad i :: S, \Delta; \Phi_a \models t \leq t' \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \forall i \overset{\mathbf{diff}(t)}{::} S.\, \tau \sqsubseteq \forall i \overset{\mathbf{diff}(t')}{::} S.\, \tau'}$ $\forall\mathbf{diff}$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau,\tau'}$. $i :: S, \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \ominus \texttt{coerce}_{\tau,\tau'} \lesssim 0 :^c \tau \xrightarrow{\text{diff}(0)} \tau'$

Then, using this, the second premise and the **c-r-iLam** and **c-r-iApp** rules in RelCostCore, we can construct the following derivation:

$$\Delta; \Phi; \cdot \vdash \lambda x.\Lambda i.\mathtt{coerce}_{\tau,\tau'}\ (x\ [i]) \ominus \lambda x.\Lambda i.\mathtt{coerce}_{\tau,\tau'}\ (x\ [i]) \lesssim 0 :^{c} (\forall i \overset{\text{diff}(t)}{::} S.\, \tau) \xrightarrow{\text{diff}(0)} \forall i \overset{\text{diff}(t')}{::} S.\, \tau'$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,(\forall i \overset{\textbf{diff}(t)}{::} S.\, \tau) \sqsubseteq \forall i \overset{\textbf{diff}(0)}{::} S.\, \square\,\tau}\ \forall\,\square$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-iLam** and **c-r-iApp** rules in Figures 80 and 83.

$$\Delta; \Phi; \cdot \vdash \lambda x.\Lambda i.\mathtt{NC}\ ((\mathtt{der}\ x)\ [i]) \ominus \lambda x.\Lambda i.\mathtt{NC}\ ((\mathtt{der}\ x)\ [i]) \lesssim 0 :^{c} \square\,(\forall i \overset{\text{diff}(t)}{::} S.\, \tau) \xrightarrow{\text{diff}(0)} \forall i \overset{\text{diff}(t')}{::} S.\, \square\,\tau$$

Case $\dfrac{}{\Delta; \Phi_a \models U\,(\forall i \overset{\textbf{exec}(k,t)}{::} S.\, A, \forall i \overset{\textbf{exec}('k,t')}{::} S.\, A') \sqsubseteq \forall i \overset{\textbf{diff}(t-k')}{::} S.\, U\,(A, A')}\ \forall\mathbf{U}$

Then, we can immediately construct the following derivation where $e = \lambda x.\Lambda i.\mathtt{switch}\ (x\,[i])$ using the **c-switch** and **c-iApp** rules in Figures 80 and 83.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (U\,(\forall i \overset{\text{exec}(k,t)}{::} S.\, A, \forall i \overset{\text{exec}(k',t')}{::} S.\, A')) \xrightarrow{\text{diff}(0)} \forall i \overset{\text{diff}(t-k')}{::} S.\, U\,(A, A')$$

Case $\dfrac{\Delta; \Phi_a \models \tau_1 \sqsubseteq \tau_1'\ \ (\star) \qquad \Delta; \Phi_a \models \tau_2 \sqsubseteq \tau_2'\ \ (\diamond)}{\Delta; \Phi_a \models \tau_1 \times \tau_2 \sqsubseteq \tau_1' \times \tau_2'}\ \times$

By IH on $(\star)$, $\exists \mathtt{coerce}_{\tau_1,\tau_1'}\ .\ \ \Delta; \Phi; \cdot \vdash \mathtt{coerce}_{\tau_1,\tau_1'} \ominus \mathtt{coerce}_{\tau_1,\tau_1'} \lesssim 0 :^{c} \tau_1 \xrightarrow{\text{diff}(0)} \tau_1'$

By IH on $(\diamond)$, $\exists \mathtt{coerce}_{\tau_2,\tau_2'}\ .\ \ \Delta; \Phi; \cdot \vdash \mathtt{coerce}_{\tau_2,\tau_2'} \ominus \mathtt{coerce}_{\tau_2,\tau_2'} \lesssim 0 :^{c} \tau_2 \xrightarrow{\text{diff}(0)} \tau_2'$

Then, using these two statements and the rules **c-prod** and **c-proj** in Figure 81, we can show the following derivation where $e = \lambda x.\langle \mathtt{coerce}_{\tau_1,\tau_1'}\ (\pi_1 x), \mathtt{coerce}_{\tau_2,\tau_2'}\ (\pi_2 x)\rangle$

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (\tau_1 \times \tau_2) \xrightarrow{\text{diff}(0)} \tau_1' \times \tau_2'$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,\tau_1 \times \square\,\tau_2 \equiv \square\,(\tau_1 \times \tau_2)}\ \times\square$

We show the direction from right-to-left using the rules **c-der**, **c-nochange**, **c-r-proj**, **c-r-let** and **c-r-prod** in Figures 80, 81 and 83 where the expression $e = \lambda x.\text{let } a = \pi_1 x \text{ in}$
$\text{let } b = \pi_2 x \text{ in } \mathtt{NC}\ (\langle \mathtt{der}\ a, \mathtt{der}\ b\rangle)$.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} \square\,\tau_1 \times \square\,\tau_2 \xrightarrow{\text{diff}(0)} \square\,(\tau_1 \times \tau_2)$$

Case $\dfrac{}{\Delta; \Phi_a \models U\,(A_1 \times A_2, A_1' \times A_2') \sqsubseteq U\,(A_1, A_1') \times U\,(A_2, A_2')}\ \times\mathbf{U}$

Then, we can immediately construct the following derivation where $e = \lambda x.(\langle \mathtt{switch}\ \pi_1 x, \mathtt{switch}\ \pi_2 x\rangle)$ using the **c-switch**, **c-r-prod** and **c-r-proj** rules in Figures 80 and 81.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (U\,(A_1 \times A_2, A_1' \times A_2')) \xrightarrow{\text{diff}(0)} U\,(A_1, A_1') \times U\,(A_2, A_2')$$

Case $\dfrac{}{\Delta; \Phi_a \models \square\,\tau_1 + \square\,\tau_2 \sqsubseteq \square\,(\tau_1 + \tau_2)}\ +\square$

We can construct the following derivation by using the rules **c-der**, **c-nochange**, **c-r-case**, **c-r-inl** and **c-r-inr** in Figure 80 where the expression $e = \lambda x.\ \text{case}\ (x, a.\mathtt{NC}\ (\text{inl}\ \mathtt{der}\ a), b.\mathtt{NC}\ (\text{inr}\ \mathtt{der}\ b))$.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} \square\,\tau_1 + \square\,\tau_2 \xrightarrow{\text{diff}(0)} \square\,(\tau_1 + \tau_2)$$

Case $\dfrac{\Delta; \Phi_a \models n \doteq n' \quad (\star) \qquad \Delta; \Phi_a \models \alpha \leq \alpha' \quad (\diamond) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\dagger)}{\Delta; \Phi_a \models \textbf{list}[n]^{\alpha}\, \tau \sqsubseteq \textbf{list}[n']^{\alpha'}\, \tau'}$ **l1**

By IH on ($\dagger$), $\exists \texttt{coerce}_{\tau,\tau'}$. $\Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \ominus \texttt{coerce}_{\tau,\tau'} \lesssim 0 :^{c} \tau \xrightarrow{\text{diff}(0)} \tau'$
We first construct the more generic term for type:

$$\text{unit}_r \xrightarrow{\text{diff}(0)} \forall n{::}\mathbb{N}.\, \forall n'{::}\mathbb{N}.\, \forall \alpha{::}\mathbb{N}.\, \forall \alpha'{::}\mathbb{N}.\, ((n = n' \wedge \alpha \leq \alpha') \,\&\, \text{list}[n]^{\alpha}\, \tau) \xrightarrow{\text{diff}(0)} \text{list}[n']^{\alpha'}\, \tau' \tag{5.1}$$

and then instantiate the term for eq. (5.1) later.
It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda n'.\Lambda \alpha.\Lambda \alpha'.\lambda x.\text{clet } x \text{ as } e \text{ in}$
$\quad \text{case } e \text{ of}$
$\quad\ \ \text{nil} \to \text{nil}$
$\quad \mid h\ ::_N tl\ \to \text{let } r = \text{fList } ()\, [n-1]\, [n'-1]\, [\alpha]\, [\alpha']\, tl \text{ in cons}_{NC}(\texttt{NC}\, (\texttt{coerce}_{\tau,\tau'}\ \texttt{der}\ h), r)$
$\quad \mid h ::_C tl\ \to \text{let } r = \text{fList } ()\, [n-1]\, [n'-1]\, [\alpha-1]\, [\alpha'-1]\, tl \text{ in cons}_{C}(\texttt{coerce}_{\tau,\tau'}\ \ h, r)$

Then, we can instantiate fList using ($\star$) and ($\diamond$) as follows where
$e'' = \lambda x.\text{fList } ()\ [n][n'][\alpha][\alpha']\ x$

$$\Delta; \Phi; \cdot \vdash e'' \ominus e'' \lesssim 0 :^{c} \text{list}[n]^{\alpha}\, \tau \xrightarrow{\text{diff}(0)} \text{list}[n']^{\alpha'}\, \tau'$$

Case $\dfrac{}{\Delta; \Phi_a \models \textbf{list}[n]^{\alpha}\, \square\, \tau \sqsubseteq \square\, (\textbf{list}[n]^{\alpha}\, \tau)}$ **l□**

We first construct the more generic term for type

$$\text{unit}_r \xrightarrow{\text{diff}(0)} \forall n{::}\mathbb{N}.\, \forall \alpha{::}\mathbb{N}.\, \text{list}[n]^{\alpha}\, \square\, \tau \xrightarrow{\text{diff}(0)} \square\, (\text{list}[n]^{\alpha}\, \tau) \tag{5.2}$$

and then instantiate the term for eq. (5.2) later. It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda \alpha.\lambda x.$
$\quad \text{case } e \text{ of}$
$\quad\ \ \text{nil} \to \texttt{NC}\ (\text{nil})$
$\quad \mid h\ ::_N tl\ \to \text{let } r = \text{fList } ()\, [n-1]\, [\alpha]\, tl \text{ in}$
$\qquad\qquad\qquad \texttt{NC}\ (\text{cons}_{NC}(\texttt{der}\ h, \texttt{der}\ r))$
$\quad \mid h ::_C tl\ \to \text{let } r = \text{fList } ()\, [n-1]\, [\alpha-1]\, tl \text{ in}$
$\qquad\qquad\qquad \texttt{NC}\ (\text{cons}_{C}(\texttt{der}\ h, \texttt{der}\ r))$

Then, we can instantiate fList with a concrete n and α as follows where $e'' = \lambda x.\text{fList } ()\ [n][\alpha]\ x$

$$\Delta; \Phi; \cdot \vdash e'' \ominus e'' \lesssim 0 :^{c} \text{list}[n]^{\alpha}\, \square\, \tau \xrightarrow{\text{diff}(0)} \square\, (\text{list}[n]^{\alpha}\, \tau)$$

Case $\dfrac{\Delta; \Phi_a \models \alpha \doteq 0}{\Delta; \Phi_a \models \textbf{list}[n]^{\alpha}\, \tau \sqsubseteq \textbf{list}[n]^{\alpha}\, \square\, \tau}$ **l2**

We first construct the more generic term for type

$$\text{unit}_r \xrightarrow{\text{diff}(0)} \forall n{::}\mathbb{N}.\, \forall \alpha{::}\mathbb{N}.\, (\alpha = 0 \,\&\, \text{list}[n]^{\alpha}\, \tau) \xrightarrow{\text{diff}(0)} \text{list}[n]^{\alpha}\, \square\, \tau \tag{5.3}$$

and then instantiate the term for eq. (5.3) later. It can be shown that such a derivation can be constructed for expression
$e' = \text{fix fList}(\_).\Lambda n.\Lambda \alpha.\lambda x.\text{clet } x \text{ as } e \text{ in}$
$\quad \text{case } e \text{ of}$
$\quad\ \ \text{nil} \to \text{nil}$
$\quad \mid h\ ::_N tl\ \to \text{let } r = \text{fList } ()\, [n-1][\alpha]\, tl \text{ in cons}_{NC}(\texttt{NC}\ h, r)$
$\quad \mid h ::_C tl\ \to \texttt{contra}$

Then, we can instantiate fList with a concrete n and α (note the premise $\alpha = 0$) as follows where $e'' = \lambda x.\text{fList}\ ()\ [n][\alpha]\ x$

$$\Delta; \Phi; \cdot \vdash e'' \ominus e'' \lesssim 0 :^{c} \text{list}[n]^{0}\, \tau \xrightarrow{\text{diff}(0)} \text{list}[n]^{0}\, \Box\, \tau$$

Case $\dfrac{i :: S, \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\star) \qquad i \notin FV(\Phi_a)}{\Delta; \Phi_a \models \exists i{::}S.\, \tau \sqsubseteq \exists i{::}S.\, \tau'}\ \exists$

By IH on $(\star)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ i :: S, \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \ominus \texttt{coerce}_{\tau,\tau'} \lesssim 0 :^{c} \tau \xrightarrow{\text{diff}(0)} \tau'$

Then, using this and the **c-r-pack** and **c-r-unpack** rules in RelCostCore in Figure 83, we can construct the following derivation where $e = \lambda x.\text{unpack}\ x\ \text{as}\ (y, i)\ \text{in pack}\ (\texttt{coerce}_{\tau,\tau'}\ \ y)\ \text{with}\ i$

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (\exists i{::}S.\, \tau) \xrightarrow{\text{diff}(0)} \exists i{::}S.\, \tau'$$

Case $\dfrac{}{\Delta; \Phi_a \models \exists i{::}S.\, \Box\, \tau \sqsubseteq \Box\, (\exists i{::}S.\, \tau)}\ \exists\Box$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-pack** and **c-r-unpack** rules in in Figures 80 and 83 where $e = \lambda x.\text{unpack}\ x\ \text{as}\ (y, i)\ \text{in}\ \texttt{NC}\ (\text{pack}\ \texttt{der}\ y\ \text{with}\ i)$.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (\exists i{::}S.\, \Box\, \tau) \xrightarrow{\text{diff}(0)} \Box\, (\exists i{::}S.\, \tau)$$

Case $\dfrac{\Delta; \Phi_a \wedge C' \models C \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond)}{\Delta; \Phi_a \models C \supset \tau \sqsubseteq C' \supset \tau'}\ \textbf{c-impl}$

By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \ominus \texttt{coerce}_{\tau,\tau'} \lesssim 0 :^{c} \tau \xrightarrow{\text{diff}(0)} \tau'$

Then, using this and the premise $(\star)$ along with the **c-r-c-implI** and **c-r-c-implE** rules in Figure 84, we can construct the following derivation where $e = \lambda x.\texttt{coerce}_{\tau,\tau'}\ \ (\text{celim}\ x)$

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} (C \supset \tau) \xrightarrow{\text{diff}(0)} C' \supset \tau'$$

Case $\dfrac{}{\Delta; \Phi_a \models \Box\, (C \supset \tau) \sqsubseteq C \supset \Box\, \tau}\ \textbf{c-impl-}\Box$

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange** and **c-r-c-implE** rules in RelCostCore where $e = \lambda x.\texttt{NC}\ (\text{celim}\ \texttt{der}\ x)$.

$$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^{c} \Box\, (C \supset \tau) \xrightarrow{\text{diff}(0)} (C \supset \Box\, \tau)$$

Case $\dfrac{\Delta; \Phi_a \wedge C \models C' \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond)}{\Delta; \Phi_a \models C\ \&\ \tau \sqsubseteq C'\ \&\ \tau'}\ \textbf{c-and}$

By IH on $(\diamond)$, $\exists \texttt{coerce}_{\tau,\tau'}\ .\ \ \Delta; \Phi; \cdot \vdash \texttt{coerce}_{\tau,\tau'} \ominus \texttt{coerce}_{\tau,\tau'} \lesssim 0 :^{c} \tau \xrightarrow{\text{diff}(0)} \tau'$

Then, using this and the premise $(\star)$ along with the **c-r-c-prodI** and **c-r-c-prodE** rules in Figure 81, we can construct the following derivation where $e = \lambda x.\text{clet}\ x\ \text{as}\ y\ \text{in}\ \texttt{coerce}_{\tau,\tau'}\ \ y$

$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c (C \mathbin{\&} \tau) \xrightarrow{\text{diff}(0)} C' \mathbin{\&} \tau'$

Case $\dfrac{}{\Delta; \Phi_a \models C \mathbin{\&} \square \tau \sqsubseteq \square (C \mathbin{\&} \tau)}$ **c-and-□**

Then, we can immediately construct the following derivation using the **c-der**, **c-nochange**, **c-r-c-prodI** and **c-r-c-prodE** rules in Figures 80, 81 and 83 where $e = \lambda x.\text{clet } x \text{ as } y \text{ in } \texttt{NC}\ (\texttt{der}\ y)$.

$\Delta; \Phi; \cdot \vdash e \ominus e \lesssim 0 :^c (C \mathbin{\&} \square \tau) \xrightarrow{\text{diff}(0)} \square (C \mathbin{\&} \tau)$

□

Lemma 37 (Reflexivity of Algorithmic Binary Type Equivalence in RelCost)
$\Delta; \psi_a; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi$ and $\Delta; \psi_a; \Phi_a \models \Phi$.

Proof. By induction on the binary type. □

Lemma 38 (Reflexivity of Unary Algorithmic Subtyping in RelCost)
$\Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the unary type. □

Lemma 39 (Transitivity of Unary Algorithmic Subtyping in RelCost)
If $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \Rightarrow \Phi_1$ and $\Delta; \Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_3 \Rightarrow \Phi_2$ and $\Delta; \Phi_a \models \Phi_1 \wedge \Phi_2$, then $\Delta; \Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_3 \Rightarrow \Phi_3$ for some Φ_3 such that $\Delta; \Phi_a \models \Phi_3$.

Proof. By induction on the first subtyping derivation. □

Theorem 40 (Soundness of the Algorithmic Unary Subtyping in RelCost)
Assume that

1. $\Delta; \psi_a; \Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi$
2. $\text{FIV}(\Phi_a, A, A') \subseteq \Delta, \psi_a$
3. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable s.t $\Delta \ \triangleright \ \theta_a : \psi_a$ is derivable.

Then $\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A'[\theta_a] \sqsubseteq A[\theta_a]$.

Proof. By induction on the algorithmic unary subtyping derivation. □

Theorem 41 (Completeness of the Unary Algorithmic Subtyping in RelCost)
Assume that $\Delta; \Phi_a \models^{\mathsf{A}} A' \sqsubseteq A$. Then $\exists \Phi$. such that $\Delta; \Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the unary subtyping derivation. □

Theorem 42 (Soundness of the Algorithmic Binary Type Equality in RelCost)
Assume that

1. $\Delta; \psi_a; \Phi_a \models \tau' \equiv \tau \Rightarrow \Phi$
2. $\text{FIV}(\Phi_a, \tau, \tau') \subseteq \Delta, \psi_a$
3. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable s.t $\Delta \ \triangleright \ \theta_a : \psi_a$ is derivable.

Then $\Delta; \Phi_a[\theta_a] \models \tau'[\theta_a] \equiv \tau[\theta_a]$.

Proof. By induction on the algorithmic binary type equivalence derivation. □

Theorem 43 (Completeness of the Binary Algorithmic Type Equivalence in RelCost) Assume that $\Delta; \Phi_a \models \tau' \equiv \tau$. Then $\exists \Phi$. such that $\Delta; \Phi_a \models \tau' \equiv \tau \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the binary subtyping derivation. □

Theorem 44 (Soundness of RelCostCore & Type Preservation of Embedding) The following holds.

1. If $\Delta; \Phi; \Omega \vdash_k^t e \rightsquigarrow e^* : A$, then $\Delta; \Phi; \Omega \vdash_k^t e^* :^c A$ and $\Delta; \Phi; \Omega \vdash_k^t e : A$.
2. If $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e_1^* \ominus e_2^* \lesssim t : \tau$, then $\Delta; \Phi; \Gamma \vdash e_1^* \ominus e_2^* \lesssim t :^c \tau$ and $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau$.

Proof. Proof is by simultaneous induction on the embedding derivations. The proof follows from the embedding rules presented in Figures 80 to 85. We show a few representative cases.

Proof of Theorem 44.2:

Case $$\dfrac{\Delta; \Phi_a; |\Gamma| \vdash_{k_1}^{t_1} e_1 \rightsquigarrow e_1^* : A \quad (\star) \qquad \Delta; \Phi_a; |\Gamma| \vdash_{k_2}^{t_2} e_2 \rightsquigarrow e_2^* : A \quad (\diamond)}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow \texttt{switch}\ e_1^* \ominus \texttt{switch}\ e_2^* \lesssim t_1 - k_2 : U\,A}\ \textbf{e-switch}$$

By Theorem 44.1 on $(\star)$, we get $\Delta; \Phi; \Omega \vdash_{k_1}^{t_1} e_1^* :^c A_1 \quad (\star\star)$.

By Theorem 44.1 on $(\diamond)$, we get $\Delta; \Phi; \Omega \vdash_{k_2}^{t_2} e_2^* :^c A_2 \quad (\diamond\diamond)$.

Then, we conclude as follows: $$\dfrac{\Delta; \Phi_a; |\Gamma| \vdash_{k_1}^{t_1} e_1 :^c A \qquad \Delta; \Phi_a; |\Gamma| \vdash_{k_2}^{t_2} e_2 :^c A}{\Delta; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1 \ominus \texttt{switch}\ e_2 \lesssim t_1 - k_2 :^c U\,A}\ \textbf{c-switch}$$

Case $$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau \\ \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \texttt{coerce}_{\tau,\tau'} \qquad \Delta; \Phi_a \models t \leq t'\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e'\ e_1^* \ominus e'\ e_2^* \lesssim t' : \tau'}\ \textbf{e-r-}\sqsubseteq[(\star)][(\diamond)][(\dagger)]$$

By Theorem 44.2 on $(\star)$, $\Delta; \Phi; \Gamma \vdash e_1^* \ominus e_2^* \lesssim t :^c \tau \quad (\star\star)$.

By Lemma 36 using $(\diamond)$, we know that $\Delta; \Phi; \cdot \vdash e' \ominus e' \lesssim 0 :^c \tau \xrightarrow{\text{diff}(0)} \tau' \quad (\diamond\diamond)$.

By applying **c-r-app** rule in Figure 82 tp $(\star\star)$ and $(\diamond\diamond)$, we get $\Delta; \Phi; \Gamma \vdash e'\ e_1^* \ominus e'\ e_2^* \lesssim t :^c \tau' \quad (\spadesuit)$.

By reflexivity of binary type equivalence, we know $\Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit)$.

Then, we conclude as follows:

$$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \ominus e'\ e_2^* \lesssim t :^c \tau' \quad (\spadesuit) \qquad \Delta; \Phi_a \models \tau' \equiv \tau' \quad (\spadesuit\spadesuit) \\ \Delta; \Phi_a \models t \leq t' \quad (\dagger)\end{array}}{\Delta; \Phi_a; \Gamma \vdash e'\ e_1^* \ominus e'\ e_2^* \lesssim t' :^c \tau'}\ \textbf{c-r}-\sqsubseteq$$

□

Theorem 45 (Completeness of RelCostCore) The following holds.

1. If $\Delta; \Phi; \Omega \vdash_k^t e : A$ then, $\exists e^*$ such that $\Delta; \Phi; \Omega \vdash_k^t e \rightsquigarrow e^* : A$.
2. If $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau$ then, $\exists e_1^*$ and $\exists e_2^*$ such that $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e_1^* \ominus e_2^* \lesssim t : \tau$.

Proof. Proof is by simultaneous induction on the typing derivations. The proof follows from the embedding rules presented in Figures 73-75. We show a few representative cases.

Proof of Theorem 45.1:

Case $\dfrac{\Delta; \Phi_a; \Omega \vdash^t_k e : A \quad (\star) \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \quad (\diamond) \qquad \Delta; \Phi_a \models k' \leq k \quad (\dagger) \qquad \Delta; \Phi_a \models t \leq t' \quad \dagger\dagger}{\Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e : A'} \sqsubseteq \textbf{exec}$

By Theorem 45.1 on $(\star)$, we get $\exists e^*$ such that $\Delta; \Phi; \Omega \vdash^t_k e \rightsquigarrow e^* : A \quad (\star\star)$.

By **e-u-**$\sqsubseteq$ rule using $(\star\star)$, $(\diamond)$, $(\dagger)$ and $(\dagger\dagger)$, we conclude as follows

$$\dfrac{\Delta; \Phi_a; \Omega \vdash^t_k e \rightsquigarrow e^* : A \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \qquad \Delta; \Phi_a \models k' \leq k \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Omega \vdash^{t'}_{k'} e \rightsquigarrow e^* : A'} \textbf{e-u-}\sqsubseteq$$

Proof of Theorem 45.2:

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; |\Gamma| \vdash^{t_1}_{k_1} e_1 : A \quad (\star) \\ \Delta; \Phi_a; |\Gamma| \vdash^{t_2}_{k_2} e_2 : A \quad (\diamond)\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t_1 - k_2 : U\, A} \textbf{switch}$

By Theorem 45.1 on $(\star)$, we get $\exists e_1^*$ such that $\Delta; \Phi; \Omega \vdash^{t_1}_{k_1} e_1 \rightsquigarrow {e_1}^* : A_1 \quad (\star\star)$.

By Theorem 45.1 on $(\diamond)$, we get $\exists e_2^*$ such that $\Delta; \Phi; \Omega \vdash^{t_2}_{k_2} e_2 \rightsquigarrow {e_2}^* : A_2 \quad (\diamond\diamond)$.

By **e-switch** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; |\Gamma| \vdash^{t_1}_{k_1} e_1 \rightsquigarrow e_1^* : A \quad (\star\star) \qquad \Delta; \Phi_a; |\Gamma| \vdash^{t_2}_{k_2} e_2 \rightsquigarrow e_2^* : A \quad (\diamond\diamond)}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow \texttt{switch}\ e_1^* \ominus \texttt{switch}\ e_2^* \lesssim t_1 - k_2 : U\, A} \textbf{e-switch}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e \lesssim t : \tau \quad (\star) \\ \forall x \in dom(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\, \Gamma(x) \quad (\diamond)\end{array}}{\Delta; \Phi_a; \Gamma, \Gamma'; \Omega \vdash e \ominus e \lesssim 0 : \square\, \tau} \textbf{nochange}$

By Theorem 45.2 on $(\star)$, we get $\exists e^*$ such that $\Delta; \Phi; \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau \quad (\dagger)$.

By Lemma 36 on $(\diamond)$, we get $\exists e_i = \texttt{coerce}_{\Gamma(x_i), \square\,(\Gamma(x_i))}$ for all $x_i \in dom(\Gamma) \quad (\dagger\dagger)$.

By **e-nochange** embedding rule using $(\dagger)$ and $(\dagger\dagger)$, we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau \quad (\dagger) \qquad \forall x_i \in dom(\Gamma),\quad e_i = \texttt{coerce}_{\Gamma(x_i), \square\,\Gamma(x_i)} \quad (\dagger\dagger)}{\Delta; \Phi_a; \Gamma, \Gamma' \vdash e \ominus e \rightsquigarrow \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in}\ \texttt{NC}\ e^*[\overline{y_i/x_i}] \ominus \text{let}\ \overline{y_i = e_i\ x_i}\ \text{in}\ \texttt{NC}\ e^*[\overline{y_i/x_i}] \lesssim 0 : \square\, \tau} \textbf{e-nochange}.$$

Case $\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \textbf{list}[n]^\alpha\, \tau \quad (\star) \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \lesssim t' : \tau' \quad (\diamond) \\ i, \Delta; \Phi_a \wedge n = i+1; h : \square\, \tau, tl : \textbf{list}[i]^\alpha\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau' \quad (\dagger) \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta + 1; h : \tau, tl : \textbf{list}[i]^\beta\, \tau, \Gamma \vdash e_2 \ominus e_2' \lesssim t' : \tau' \quad (\spadesuit)\end{array}}{\Delta; \Phi_a; \Gamma \vdash\ \ \textbf{case}\ e\ \textbf{of nil} \rightarrow e_1 \mid h :: tl \rightarrow e_2 \ominus\ \ \textbf{case}\ e'\ \textbf{of nil} \rightarrow e_1' \mid h :: tl \rightarrow e_2' \lesssim t + t' : \tau'} \textbf{r-caseL}$

By Theorem 45.2 on $(\star)$, we get $\exists e^*$ and $\exists e'^*$ s.t. $\Delta; \Phi; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \text{list}[n]^\alpha\, \tau \quad (\star\star)$.

By Theorem 45.2 on $(\diamond)$, we get $\exists e_1^*$ and $\exists e_1'^*$ s.t. $\Delta; n \doteq 0 \wedge \Phi; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow e_1^* \ominus e_1'^* \lesssim t : \tau' \quad (\diamond\diamond)$.

By Theorem 45.2 on $(\dagger)$, we get $\exists e_2^*$ and $\exists e_2'^*$ s.t.

$i :: S, \Delta; n \doteq i + 1 \wedge \Phi; h : \square\, \tau, tl : \text{list}[i]^\alpha\, \tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow e_2^* \ominus e_2'^* \lesssim t : \tau' \quad (\dagger\dagger)$.

By Theorem 45.2 on $(\spadesuit)$, we get $\exists e_2^{**}$ and $\exists e_2'^{**}$ s.t.

$i :: S, \beta :: S, \Delta; n \doteq i+1 \wedge \alpha \doteq \beta+1 \wedge \Phi; h : \tau, tl : \text{list}[i]^{\beta}\,\tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow e_2^{**} \ominus e_2'^{**} \lesssim t : \tau' \quad (\spadesuit\spadesuit).$

By **e-caseL** embedding rule using $(\star\star)$, $(\diamond\diamond)$, and $(\spadesuit\spadesuit)$, we can conclude as follows

$$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \text{list}[n]^{\alpha}\,\tau \qquad \Delta; \Phi_a \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t' : \tau' \\ i, \Delta; \Phi_a \wedge n = i+1; h : \square\,\tau, tl : \text{list}[i]^{\alpha}\,\tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t' : \tau' \\ i, \beta, \Delta; \Phi_a \wedge n = i+1 \wedge \alpha = \beta+1; h : \tau, tl : \text{list}[i]^{\beta}\,\tau, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow e_3^* \ominus e_3'^* \lesssim t' : \tau'\end{array}}{\Delta; \Phi_a; \Gamma \vdash \begin{array}{l}\text{case } e \text{ of nil } \rightarrow e_1 \\ \mid h \,::\, tl \rightarrow e_2\end{array} \ominus \begin{array}{l}\text{case } e \text{ of nil } \rightarrow e_1' \\ \mid h \,::\, tl \rightarrow e_2'\end{array} \rightsquigarrow \begin{array}{l}\text{case } e^* \text{ of nil } \rightarrow e_1^* \\ \mid h \,::_{NC} tl \rightarrow e_2^* \ominus \\ \mid h \,::_{C} tl \rightarrow e_3^*\end{array} \begin{array}{l}\text{case } e'^* \text{ of nil } \rightarrow e_1'^* \\ \mid h \,::_{NC} tl \rightarrow e_2'^* \lesssim t+t' : \tau' \\ \mid h \,::_{C} tl \rightarrow e_3'^*\end{array}} \textbf{ e-r-ca}$$

Case $$\frac{\begin{array}{c}\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\textbf{diff}(t)} \tau_2 \texttt{ wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \xrightarrow{\textbf{diff}(t)} \tau_2), \Gamma \vdash e \ominus e \lesssim t : \tau_2 \quad (\star) \\ \forall x \in dom(\Gamma).\ \ \Delta; \Phi_a \models \Gamma(x) \sqsubseteq \square\,\Gamma(x) \quad (\diamond)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \textbf{fix } f(x).e \ominus \textbf{fix } f(x).e \lesssim 0 : \square\,(\tau_1 \xrightarrow{\textbf{diff}(t)} \tau_2)} \textbf{ r-fixNC}$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e^*$ such that $\Delta; \Phi; x : \tau_1, f : \square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2), \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau_2 \quad (\star\star)$.
By Lemma 36 on $(\diamond)$, we get $\exists e_i = \texttt{coerce}_{\Gamma(x_i), \square\,(\Gamma(x_i))}$ for all $x_i \in dom(\Gamma) \quad (\diamond\diamond)$.

By **e-fixNC** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\frac{\begin{array}{l}\Delta; \Phi_a \vdash \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \texttt{ wf} \qquad \Delta; \Phi_a; x : \tau_1, f : \square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2), \Gamma \vdash e \ominus e \rightsquigarrow e^* \ominus e^* \lesssim t : \tau_2 \\ \forall x_i \in dom(\Gamma), \quad e_i = \texttt{coerce}_{\Gamma(x_i), \square\,\Gamma(x_i)} \qquad e^{**} = \text{let } \overline{y_i = e_i\ x_i} \text{ in fix}_{NC}\ f(x).e^*[\overline{y_i/x_i}]\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{fix } f(x).e \ominus \text{fix } f(x).e \rightsquigarrow e^{**} \ominus e^{**} \lesssim 0 : \square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2)} \textbf{ e-r-fixNC}.$$

Case $$\frac{\begin{array}{c}i :: S, \Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau \quad (\star) \\ i \not\in \textbf{FIV}(\Phi_a; \Gamma)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \Lambda e \ominus \Lambda e' \lesssim 0 : \forall i \overset{\textbf{diff}(t)}{::} S.\,\tau} \textbf{ r-iLam}$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e^*$ and $\exists\, e'^*$ such that $i :: S, \Delta; \Phi; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau \quad (\star\star)$.
By **e-iLam** embedding rule using $(\star\star)$, we can conclude as follows:

$$\frac{i :: S, \Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau \qquad i \not\in \text{FIV}(\Phi_a; \Gamma)}{\Delta; \Phi_a; \Gamma \vdash \Lambda.e \ominus \Lambda.e' \rightsquigarrow \Lambda i.e^* \ominus \Lambda i.e'^* \lesssim 0 : \forall i \overset{\text{diff}(t)}{::} S.\,\tau} \textbf{ e-r-iLam}.$$

Case $$\frac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \forall i \overset{\textbf{diff}(t')}{::} S.\,\tau \quad (\star) \\ \Delta \vdash I : S \quad (\diamond)\end{array}}{\Delta; \Phi_a; \Gamma \vdash e[\,] \ominus e'[\,] \lesssim t + t'[I/i] : \tau\{I/i\}} \textbf{ r-iApp}$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e^*$ such that $\Delta; \Phi; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \forall i \overset{\text{exec}(t',\tau)}{::} S. \quad (\star\star)$.
By **e-iApp** embedding rule using $(\star\star)$ and $(\diamond)$, we can conclude as follows:

$$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \forall i \overset{\text{diff}(t')}{::} S.\,\tau \qquad \Delta \vdash I : S}{\Delta; \Phi_a; \Gamma \vdash e[\,] \ominus e'[\,] \rightsquigarrow e^*[I] \ominus e'^*[I] \lesssim t + t'[I/i] : \tau\{I/i\}} \textbf{ e-r-iApp}.$$

Case $$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t : \tau\{I/i\} \quad (\star) \qquad \Delta \vdash I :: S \quad (\diamond)}{\Delta; \Phi_a; \Gamma \vdash \textbf{pack } e \ominus \textbf{pack } e' \lesssim t : \exists i{::}S.\,\tau} \textbf{ r-pack}$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e^*$ and $\exists\, e'^*$ such that $\Delta; \Phi; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau\{I/i\}$ $(\star\star)$.

By **e-pack** embedding rule using $(\star\star)$ and $(\diamond)$, we can conclude as follows:

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \rightsquigarrow e^* \ominus e'^* \lesssim t : \tau\{I/i\} \qquad \Delta \vdash I :: S}{\Delta; \Phi_a; \Gamma \vdash \text{pack } e \ominus \text{pack } e' \rightsquigarrow \text{pack } e^* \text{ with } I \ominus \text{pack } e'^* \text{ with } I \lesssim t : \exists i{::}S.\, \tau}\ \textbf{e-r-pack}.$$

Case $$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 : \exists i{::}S.\, \tau_1 \quad (\star) \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 : \tau_2 \quad (\diamond) \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)\end{array}}{\Delta; \Phi_a; \Gamma \vdash \textbf{unpack } e_1 \textbf{ as } x \textbf{ in } e_2 \ominus \textbf{unpack } e_1' \textbf{ as } x \textbf{ in } e_2' \lesssim t_1 + t_2 : \tau_2}\ \textbf{r-unpack1}$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e_1^*$ and $\exists\, e_1'^*$ such that $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow e_1^* \ominus e_1'^* \lesssim t : \exists i{::}S.\, \tau_1$ $(\star\star)$.

By Theorem 45.2 on $(\diamond)$, we get $\exists\, e_2^*$ and $\exists\, e_2'^*$ such that $i :: S, \Delta; \Phi; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow e_2^* \ominus e_2'^* \lesssim t : \tau_2$ $(\diamond\diamond)$.

By **e-unpack** embedding rule using $(\star\star)$ and $(\diamond\diamond)$, we can conclude as follows:

$$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \rightsquigarrow {e_1}^* \ominus {e_1'}^* \lesssim t_1 : \exists i{::}S.\, \tau_1 \\ i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \rightsquigarrow {e_2}^* \ominus {e_2'}^* \lesssim t_2 : \tau_2 \\ i \notin FV(\Phi_a; \Gamma, \tau_2, t_2) \qquad e_1^{**} = \text{unpack } e_1^* \text{ as } (x,i) \text{ in } e_2^* \qquad e_2^{**} = \text{unpack } e_1'^* \text{ as } (x,i) \text{ in } e_2'^*\end{array}}{\Delta; \Phi_a; \Gamma \vdash \text{unpack } e_1 \text{ as } x \text{ in } e_2 \ominus \text{unpack } e_1' \text{ as } x \text{ in } e_2' \rightsquigarrow e_1^{**} \ominus e_2^{**} \lesssim t_1 + t_2 : \tau_2}\ \textbf{e-r-unpack}.$$

Case $$\dfrac{\begin{array}{c}\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t : \tau \quad (\star) \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \quad (\diamond) \\ \Delta; \Phi_a \models t \leq t' \quad \dagger\end{array}}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t' : \tau'}\ \textbf{r-}\sqsubseteq$$

By Theorem 45.2 on $(\star)$, we get $\exists\, e_1^*, e_2^*$ such that $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau$ $(\star\star)$.

By Lemma 36 on $(\diamond)$, we can show that $\exists e' = \texttt{coerce}_{\tau,\tau'}$ $(\diamond\diamond)$.

By **e-r-**$\sqsubseteq$ rule using $(\star\star)$, $(\diamond\diamond)$ and $(\dagger)$, we conclude as follows

$$\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow {e_1}^* \ominus {e_2}^* \lesssim t : \tau \qquad \Delta; \Phi_a \models \tau \sqsubseteq \tau' \qquad e' = \texttt{coerce}_{\tau,\tau'} \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \rightsquigarrow e'\ e_1^* \ominus e'\ e_2^* \lesssim t' : \tau'}\ \textbf{e-r-}\sqsubseteq$$

$\square$

Theorem 46 (Invariant of the Algorithmic Typechecking)
We have the following.

1. Assume that $\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi$ and $\text{FIV}(\Phi_a, \Omega; A, k, t) \subseteq dom(\Delta, \psi_a)$. Then $\text{FIV}(\Phi) \subseteq dom(\Delta; \psi_a)$.
2. Assume that $\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A \Rightarrow [\psi], k, t, \Phi$ and $\text{FIV}(\Phi_a, \Omega) \subseteq dom(\Delta, \psi_a)$. Then $\text{FIV}(A, k, t, \Phi) \subseteq dom(\Delta, \psi; \psi_a)$.
3. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \Phi$ and $\text{FIV}(\Phi_a, \Gamma, \tau, t) \subseteq dom(\Delta, \psi_a)$.Then $\text{FIV}(\Phi) \subseteq dom(\Delta; \psi_a)$.
4. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \tau \Rightarrow [\psi], t, \Phi$ and $\text{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$.Then $\text{FIV}(\tau, t, \Phi) \subseteq dom(\Delta; \psi; \psi_a)$.

Theorem 47 (Soundness of the Algorithmic Typechecking in RelCost)
We have the following.

1. Assume that $\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi$ and
    1.1. $\text{FIV}(\Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$

1.2. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable for some θ_a such that $\Delta \rhd \theta_a : \psi_a$ is derivable

Then $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]$.

2. Assume that $\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A \Rightarrow [\psi], k, t, \Phi$ and

2.1. $\text{FIV}(\Phi_a, \Omega) \subseteq dom(\Delta, \psi_a)$

2.2. $\forall\theta\ \forall\theta_a.\ \Delta; \Phi_a[\theta_a] \models \Phi[\theta\,\theta_a]$ is provable s.t $\Delta \rhd \theta : \psi$ and $\Delta \rhd \theta_a : \psi_a$ are derivable

Then $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta\,\theta_a]}_{k[\theta\,\theta_a]} |e| :^c A[\theta\,\theta_a]$.

3. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \Phi$ and

3.1. $\text{FIV}(\Phi_a, \Gamma, \tau, t) \subseteq dom(\Delta, \psi_a)$

3.2. $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable for some θ_a such that $\Delta \rhd \theta_a : \psi_a$ is derivable

Then $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta_a] :^c \tau[\theta_a]$.

4. Assume that $\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \uparrow \tau \Rightarrow [\psi], t, \Phi$ and

4.1. $\text{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$

4.2. $\forall\theta\ \forall\theta_a.\ \Delta; \Phi_a[\theta_a] \models \Phi[\theta\,\theta_a]$ is provable s.t $\Delta \rhd \theta : \psi$ and $\Delta \rhd \theta_a : \psi_a$ are derivable

Then $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta\,\theta_a] :^c \tau[\theta\,\theta_a]$.

Proof. Statements (1—4) follow from simultaneous structural induction on the algorithmic typing derivations. We present several cases below.

Proof of Theorem 47.1:

Case

$$\frac{k_1, t_1, k_2, t_2 \in \mathbf{fresh}(\mathbb{R}) \qquad \Delta; k_1, t_1, \psi_a; \Phi_a; \Omega \vdash e_1 \downarrow A_1, k_1, t_1 \Rightarrow \Phi_1 \qquad \Delta; k_2, t_2, \psi_a; \Phi_a; \Omega \vdash e_2 \downarrow A_1, k_2, t_2 \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Omega \vdash \langle e_1, e_2\rangle \downarrow A_1 \times A_2, k, t \Rightarrow \exists k_1, t_1 :: \mathbb{R}.\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 \doteq t \wedge k \doteq k_1 + k_2}\ \textbf{alg-u-prod-}\!\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \langle |e_1|, |e_2|\rangle :^c A_1[\theta_a] \times A_2[\theta_a]$.

By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$

$\Delta; \Phi_a[\theta_a] \models (\exists k_1, t_1 :: \mathbb{R}.\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\Phi_2 \wedge (t_1 + t_2) \doteq t \wedge (k_1 + k_2) \doteq k)[\theta_a]$ $(\star\star)$

Using $(\star)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta \vdash K_1 :: \mathbb{R}$ and $\Delta \vdash T_1 :: \mathbb{R}$

b) $\Delta \vdash K_2 :: \mathbb{R}$ and $\Delta \vdash T_2 :: \mathbb{R}$

c) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a, k_1 \mapsto K_1, t_1 \mapsto T_1]$

d) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, k_2 \mapsto K_2, t_2 \mapsto T_2]$

e) $\Delta; \Phi_a[\theta_a] \models (T_1 + T_2) \doteq t[\theta_a] \wedge (K_1 + K_2) \doteq k[\theta_a]$

By Theorem 47.1 on the first premise using $(\star)$ and c), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{T_1}_{K_1} |e_1| :^c A_1[\theta_a] \tag{5.4}$$

By Theorem 47.1 on the second premise using ($\star$) and d), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{T_2}_{K_2} |e_2| :^c A_2[\theta_a] \tag{5.5}$$

Combining eqs. (5.4) and (5.5) with **c-prod** rule, we get
$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{T_1+T_2}_{K_1+K_2} \langle |e_1|, |e_2| \rangle :^c A_1[\theta_a] \times A_2[\theta_a]$.
Then, by using e) with the **c-$\sqsubseteq$** exec rule , we can conclude that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \langle |e_1|, |e_2| \rangle :^c A_1[\theta_a] \times A_2[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow A' \Rightarrow [\psi], k', t', \Phi_1 \qquad \Delta; \psi, \psi_a; \Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi_2}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k, t \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2 \wedge t' \leq t \wedge k \leq k'}$ **alg-↑↓**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$ ($\star$) and
$\Delta; \Phi_a[\theta_a] \models (\exists(\psi).\Phi_1 \wedge \Phi_2 \wedge t' \leq t \wedge k \leq k')[\theta_a]$ ($\star\star$)
By Theorem 46 using ($\star$) and the first premise, we get $\mathrm{FIV}(A', k', t', \Phi_1) \subseteq dom(\Delta, \psi; \psi_a)$ ($\diamond$).
Using ($\star$) and ($\diamond$), ($\star\star$)'s derivation must be in a form such that we have

a) $\Delta \ \triangleright \ \theta_a : \psi_a$
b) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a, \theta_a]$
c) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, \theta_a]$
d) $\Delta; \Phi_a[\theta_a] \models t'[\theta\, \theta_a] \leq t[\theta_a] \wedge k[\theta_a] \leq k'[\theta\, \theta_a]$

By Theorem 47.2 on the first premise using ($\star$), a) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t'[\theta\, \theta_a]}_{k'[\theta\, \theta_a]} |e| :^c A'[\theta\, \theta_a] \tag{5.6}$$

By Theorem 40 using the second premise and c), we obtain

$$\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A'[\theta\, \theta_a] \sqsubseteq A[\theta\, \theta_a] \tag{5.7}$$

Note that due to ($\star$), we have $A[\theta\, \theta_a] = A[\theta_a]$. Then we can conclude by the **c-$\sqsubseteq$** exec rule using eqs. (5.6) and (5.7) and (d) that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; f : A_1 \xrightarrow{\mathbf{exec}(k', t')} A_2, x : A_1, \Omega \vdash e \downarrow A_2, k', t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \mathbf{fix}\ f(x).e \downarrow A_1 \xrightarrow{\mathbf{exec}(k', t')} A_2, k, t \Rightarrow \Phi \wedge k \doteq 0 \wedge 0 \doteq t}$ **alg-u-fix-↓**

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \mathrm{fix}\ f(x).|e| :^c A_1[\theta_a] \xrightarrow{\mathrm{exec}(k'[\theta_a], t'[\theta_a])} A_2[\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Omega, A_1 \xrightarrow{\mathrm{exec}(k', t')}, k, t) \subseteq dom(\Delta, \psi_a)$ ($\star$) and
$\Delta; \Phi_a[\theta_a] \models \Phi \wedge 0 \doteq k \wedge 0 \doteq t'[\theta_a]$ ($\star\star$)
Using ($\star$), we can show that

a) $\mathrm{FIV}(\Phi_a, \Omega, A_1, A_1 \xrightarrow{\mathrm{exec}(k', t')}, A_2, k', t') \subseteq dom(\Delta, \psi_a)$.

We also can show that $(\star\star)$'s derivation must be in a form such that we have

b) $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$
c) $\Delta; \Phi_a[\theta_a] \models 0 \doteq k[\theta_a]$
d) $\Delta; \Phi_a[\theta_a] \models 0 \doteq t[\theta_a]$

By Theorem 47.1 on the first premise using a) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; x : A_1[\theta_a], f : A_1[\theta_a] \xrightarrow{\text{exec}(k'[\theta_a], t'[\theta_a])} A_2[\theta_a], \Omega[\theta_a] \vdash^{t'[\theta_a]}_{k'[\theta_a]} |e| :^c A_2[\theta_a] \tag{5.8}$$

By the **c-fix** rule using eq. (5.8), we obtain
$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^0_0 \text{fix } f(x).|e| :^c A_1[\theta_a] \xrightarrow{\text{exec}(k'[\theta_a], t'[\theta_a])} A_2[\theta_a]$.
By **c-**$\sqsubseteq$ exec rule using (c) and (d), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \text{fix } f(x).|e| :^c A_1[\theta_a] \xrightarrow{\text{exec}(k'[\theta_a], t'[\theta_a])} A_2[\theta_a]$.

Case $\dfrac{i :: S, \Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A, k_e, t_e \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash \Lambda i.e \downarrow \forall i \overset{\textbf{exec}(k_e, t_e)}{::} S.\, A, k, t \Rightarrow (\forall i :: S.\Phi) \wedge k \doteq 0 \wedge 0 \doteq t}$ **alg-u-iLam-**$\downarrow$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \Lambda i.|e| :^c \forall i \overset{\text{exec}(k_e[\theta_a], t_e[\theta_a])}{::} S.\, A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, \forall i \overset{\text{exec}(k_e, t_e)}{::} S.\, A, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models ((\forall i :: S.\Phi) \wedge 0 \doteq k \wedge 0 \doteq t)[\theta_a]$ $(\star\star)$
Using $(\star)$, we can show that

a) $\text{FIV}(\Phi_a, \Omega, A, k_e, t_e) \subseteq i, dom(\Delta, \psi_a)$.

We can also show that $(\star\star)$'s derivation must be in a form such that we have

b) $i :: S, \Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$
c) $\Delta; \Phi_a[\theta_a] \models 0 \doteq k[\theta_a]$
d) $\Delta; \Phi_a[\theta_a] \models 0 \doteq t[\theta_a]$

By Theorem 47.1 on the premise using a) and b), we can show that

$$i :: S, \Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t_e[\theta_a]}_{k_e[\theta_a]} |e| :^c A[\theta_a] \tag{5.9}$$

By the **c-iLam** rule using eq. (5.9), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^0_0 \Lambda i.|e| :^c \forall i \overset{\text{exec}(k_e[\theta_a], t_e[\theta_a])}{::} S.\, A[\theta_a]$.
By **c-**$\sqsubseteq$ exec rule using (c) and (d), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \Lambda i.|e| :^c \forall i \overset{\text{exec}(k_e[\theta_a], t_e[\theta_a])}{::} S.\, A[\theta_a]$.

Case $\dfrac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow A\{I/i\}, k, t \Rightarrow \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash \textbf{pack } e \textbf{ with } I \downarrow \exists i{::}S.\, A, k, t \Rightarrow \Phi}$ **alg-u-pack-**$\downarrow$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \text{pack } |e| \text{ with } I :^c \exists i{::}S.\, A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, \exists i{::}S.\, A, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$

Using $(\star)$ and the second premise, we can show that

a) $\text{FIV}(\Phi_a, \Omega, A\{I/i\}, k, t) \subseteq dom(\Delta, \psi_a)$.

By Theorem 47.1 on the premise using a) and $(\star\star)$, we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]\{I/i\} \tag{5.10}$$

By the **c-pack** rule using eq. (5.10) and the second premise, we obtain
$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]}$ pack $|e|$ with $I :^c \exists i{::}S.\, A[\theta_a]$.

Case $$\frac{\begin{array}{c}\Delta; \psi_a; \Phi_a; \Omega \vdash e_1 \uparrow \exists i{::}S.\, A_1 \Rightarrow [\psi], k_1, t_1, \Phi_1 \qquad k_2, t_2 \in \textbf{fresh}(\mathbb{R}) \\ i :: S, \Delta; k_2, t_2, \psi, \psi_a; \Phi_a; x : A_1, \Omega \vdash e_2 \downarrow A_2, k_2, t_2 \Rightarrow \Phi_2 \qquad i \notin FV(\Phi_a; \Omega, A_2, k_2, t_2) \\ \Phi = \exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\forall i :: S.\Phi_2 \wedge k \doteq k_1 + k_2 + c_{unpack} \wedge t_1 + t_2 + c_{unpack} \doteq t)\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \textbf{unpack}\ e_1\ \textbf{as}\ (x, i)\ \textbf{in}\ e_2 \downarrow A_2, k, t \Rightarrow \Phi} \textbf{ alg-u-unpack-}\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]}$ unpack $|e_1|$ with (x, i) in $|e_2| :^c A_2[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A_2, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models$
$(\exists(\psi).(\Phi_1 \wedge \exists k_2, t_2 :: \mathbb{R}.\forall i :: S.\Phi_2 \wedge (t_1 + t_2 + c_{unpack}) \doteq t \wedge (k_1 + k_2 + c_{unpack}) \doteq k))[\theta_a]$ $(\star\star)$
By Theorem 46 using the first premise and $(\star)$, we get $\text{FIV}(A_1, k_1, t_1, \Phi_1) \subseteq dom(\Delta, \psi; \psi_a)$ $(\diamond)$.
Using $(\star)$, $(\diamond)$ and the 4th premise, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta \ \triangleright\ \theta : \psi$
b) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta\,\theta_a]$
c) $i :: S, \Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, \theta, k_2 \mapsto K_2, t_2 \mapsto T_2]$
d) $\Delta; \Phi_a[\theta_a] \models t_1[\theta\,\theta_a] + T_2 + c_{unpack} \doteq t[\theta_a]$
e) $\Delta; \Phi_a[\theta_a] \models k[\theta_a] \doteq k_1[\theta\,\theta_a] + K_2 + c_{unpack}$

By Theorem 47.2 on the first premise using $(\star)$, a) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t_1[\theta\,\theta_a]}_{k_1[\theta\,\theta_a]} |e_1| :^c \exists i{::}S.\, A_1[\theta\,\theta_a] \tag{5.11}$$

From $(\star)$ and $(\diamond)$, we can show that

f) $\text{FIV}(\Phi_a, A_1, \Omega, A_2, k_2, t_2) \subseteq i, k_2, t_2, dom(\Delta, \psi, \psi_a)$

By Theorem 47.1 on the second premise using c), f), $(\star)$ and $(\diamond)$, we obtain

$$i :: S, \Delta; \Phi_a[\theta_a]; x : A_1[\theta\,\theta_a], \Omega[\theta_a] \vdash^{T_2}_{K_2} |e_2| :^c A_2[\theta\,\theta_a] \tag{5.12}$$

Note that due to $(\star)$, we have $A_2[\theta\,\theta_a] = A_2[\theta_a]$. Then by the **c-unpack** rule using eqs. (5.11) and (5.12), we can show that $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t_1[\theta\,\theta_a]+T_2+c_{unpack}}_{k_1[\theta\,\theta_a]+K_1+c_{unpack}}$ unpack $|e_1|$ with (x, i) in $|e_2| :^c A_2[\theta_a]$.

By $\sqsubseteq_{\text{exec}}$ rule using (d) and (e), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \text{unpack } |e_1| \text{ with } (x, i) \text{ in } |e_2| :^c A_2[\theta_a]$.

Case $$\frac{\begin{array}{c}\Delta; \psi_a; C \wedge \Phi_a; \Omega \vdash e_1 \downarrow A, k, t \Rightarrow \Phi_1 \\ \Delta; \psi_a; \neg C \wedge \Phi_a; \Omega \vdash e_2 \downarrow A, k, t \Rightarrow \Phi_2 \qquad \Delta \vdash C \texttt{ wf}\end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \texttt{split } (e_1, e_2) \textbf{ with } C \downarrow A, k, t \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2} \ \textbf{alg-u-split}\!\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \texttt{split } (|e_1|, |e_2|) \text{ with } C :^c A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models (C \to \Phi_1 \wedge \neg C \to \Phi_2)[\theta_a]$ $(\star\star)$
Using $(\star)$ and the third premise, we can show that

a) $\text{FIV}(C \wedge \Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$.
b) $\text{FIV}(\neg C \wedge \Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$.

Using $(\star\star)$ and the third premise, we can show that

c) $\Delta; C \wedge \Phi_a[\theta_a] \models \Phi_1[\theta_a]$
d) $\Delta; \neg C \wedge \Phi_a[\theta_a] \models \Phi_2[\theta_a]$

By Theorem 47.1 on the first premise using $(\star)$ and c), we can show that

$$\Delta; C \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e_1| :^c A[\theta_a]\{I[\theta_a]/i\} \tag{5.13}$$

By Theorem 47.1 on the second premise using $(\star)$ and d), we can show that

$$\Delta; \neg C \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e_2| :^c A[\theta_a]\{I[\theta_a]/i\} \tag{5.14}$$

By the **c-split** rule using eqs. (5.13) and (5.14) and the third premise, we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} \texttt{split } (|e_1|, |e_2|) \text{ with } C :^c A[\theta_a]$.

Case $$\frac{\Delta; \Phi \wedge C; \Omega \vdash e \downarrow A, k, t \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Omega \vdash e \downarrow C \supset A, k, t \Rightarrow C \to \Phi} \ \textbf{alg-u-c-impI}\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c C[\theta_a] \supset A[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Omega, C \supset A, k, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models (C \to \Phi)[\theta_a]$ $(\star\star)$
Using $(\star)$, we can show that

a) $\text{FIV}(C \wedge \Phi_a, \Omega, A, k, t) \subseteq dom(\Delta, \psi_a)$.

By Theorem 47.1 on the premise using $(\star)$ and a), we can show that

$$\Delta; C[\theta_a] \wedge \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a] \tag{5.15}$$

By the **c-cimpI** rule using eq. (5.15), we obtain $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c C[\theta_a] \supset A[\theta_a]$.

Proof of Theorem 47.2:

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega \vdash e \downarrow A,k,t \Rightarrow \Phi \qquad \Delta;\Phi_a \vdash^{\mathsf{A}} A \ \mathtt{wf} \qquad \mathbf{FIV}(A,k,t) \in \Delta}{\Delta;\psi_a;\Phi_a;\Omega \vdash (e : A,k,t) \uparrow A \Rightarrow [\cdot],k,t,\Phi}$ **alg-u-anno-↑**

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |(e : A,k,t)| :^c A[\theta_a]$.

Since by definition, $\forall e.\ |(e : \_,\_,\_)| = |e|$, STS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega) \subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$

Using the third premise, we can show that

a) $\mathrm{FIV}(\Phi_a,\Omega,A,k,t) \subseteq dom(\Delta,\psi_a)$.

By Theorem 47.1 on the first premise using $(\star\star)$ and a), we can conclude that $\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{t[\theta_a]}_{k[\theta_a]} |e| :^c A[\theta_a]$.

Case $\dfrac{\begin{array}{c}\Delta;\psi_a;\Phi_a;\Omega \vdash e_1 \uparrow A_1 \xrightarrow{\mathbf{exec}(k_e,t_e)} A_2 \Rightarrow [\psi],k_1,t_1,\Phi_1 \\ k_2,t_2 \in \mathbf{fresh}(\mathbb{R}) \qquad \Delta;k_2,t_2,\psi,\psi_a;\Phi_a;\Omega \vdash e_2 \downarrow A_1,k_2,t_2 \Rightarrow \Phi_2\end{array}}{\Delta;\psi_a;\Phi_a;\Omega \vdash e_1\ e_2 \uparrow A_2 \Rightarrow [k_2,t_2,\psi],k_1+k_2+k_e+c_{app},t_1+t_2+t_e+c_{app},\Phi_1 \wedge \Phi_2}$ **alg-u-app-↑**

TS: $\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{(t_1+t_2+t_e+c_{app})[\theta_a,\theta_2]}_{(k_1+k_2+k_e+c_{app})[\theta_a,\theta_2]} |e_1|\ |e_2| :^c A_2[\theta_a,\theta_2]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a,\Omega) \subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a] \models (\Phi_1 \wedge \Phi_2)[\theta_a,\theta_2]$ $(\star\star)$ such that $\Delta \ \triangleright\ \theta_2 : k_2,t_2,\psi$ $(\diamond)$ and $\Delta \ \triangleright\ \theta_a : \psi_a$ are derivable.

By $(\diamond)$, we can show that $\theta_2 = k_2,t_2,\theta$ such that

a) $\theta(k_2) = K_2$ and $\theta(t_2) = T_2$ for some K_2 and T_2.

b) $\Delta \ \triangleright\ \theta : \psi$

By Theorem 47.2 on the first premise using $(\star)$ and (b), we obtain

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{t_1[\theta\,\theta_a]}_{k_1[\theta\,\theta_a]} |e_1| :^c A_1[\theta\,\theta_a] \xrightarrow{\mathrm{exec}(k_e[\theta\,\theta_a],t_e[\theta\,\theta_a])} A_2[\theta\,\theta_a] \tag{5.16}$$

By Theorem 46.2 on the first premise and $(\star)$, we get

c) $\mathrm{FIV}(A_1 \xrightarrow{\mathrm{exec}(k_e,t_e)} A_2,k_1,t_1,\Phi_1) \subseteq dom(\Delta,\psi,\psi_a)$.

By $(\star)$ and c), we get

d) $\mathrm{FIV}(\Phi_a,\Omega,A_2,k_2,t_2) \subseteq k_2,t_2,dom(\Delta,\psi,\psi_a)$.

By Theorem 47.2 on the third premise using (c), (d) and $(\star\star)$, we obtain

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{T_2}_{K_2} |e_2| :^c A_1[\theta\,\theta_a] \tag{5.17}$$

Then, by using **c-app** rule using eqs. (5.16) and (5.17), we can show that $\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{t_1[\theta\,\theta_a]+t_e[\theta\,\theta_a]+T_2}_{k_1[\theta\,\theta_a]+k_e[\theta\,\theta_a]+K_2} |e_1|\ |e_2| :^c A_2[\theta_a,\theta_2]$.

Note that we have $k_2[\theta_a, \theta_2] = K_2$ and $k_2[\theta_a, \theta_2] = K_2$. Moreover, $t_1[\theta_a, \theta_2] = t_1[\theta\, \theta_a]$ and $k_1[\theta_a, \theta_2] = k_1[\theta\, \theta_a]$ (similarly for k_e and t_e) since k_2, t_2 are fresh variables.

Case $$\frac{\Delta; \psi_a; \Phi_a; \Omega \vdash e \uparrow \forall i \overset{\mathbf{exec}(k_e, t_e)}{::} S.\, A' \Rightarrow [\psi], k, t, \Phi \qquad \Delta \vdash I :: S}{\Delta; \psi_a; \Phi_a; \Omega \vdash e\, [I] \uparrow A'\{I/i\} \Rightarrow [\psi], k + k_e[I/i], t + t_e[I/i], \Phi} \ \textbf{alg-u-iApp-}\uparrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{(t+t_e[I/i])[\theta\, \theta_a]}_{(k+k_e[I/i])[\theta\, \theta_a]} |e|\ [I] :^c (A'\{I/i\})[\theta\, \theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Omega) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta\, \theta_a]$ $(\star\star)$ such that $\Delta \ \triangleright\ \theta : \psi$ $(\diamond)$ and $\Delta \ \triangleright\ \theta_a : \psi_a$ are derivable.
By Theorem 47.2 on the first premise using $(\star)$ and $(\diamond)$, we obtain

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta\, \theta_a]}_{k[\theta\, \theta_a]} |e| :^c \forall i \overset{\mathrm{exec}(k_e[\theta\, \theta_a], t_e[\theta\, \theta_a])}{::} S.\, A'[\theta\, \theta_a] \tag{5.18}$$

Then, by **c-iApp** rule using eq. (5.18) and the second premise, we can conclude that
$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{t[\theta\, \theta_a]+t_e[\theta\, \theta_a][I/i]}_{k[\theta\, \theta_a]+k_e[\theta\, \theta_a][I/i]} |e|\ [I] :^c A'[\theta\, \theta_a]\{I/i\}$.

Proof of Theorem 47.3:

Case $$\frac{t' \in \mathbf{fresh}(\mathbb{R}) \qquad \Delta; t', \psi_a; \Phi_a; \square\Gamma \vdash e \ominus e \downarrow \tau, t' \Rightarrow \Phi}{\Delta; \psi_a; \Phi_a; \Gamma', \square\Gamma \vdash \mathtt{NC}\ e \ominus \mathtt{NC}\ e \downarrow \square\tau, t \Rightarrow 0 \doteq t \wedge (\exists t' :: \mathbb{R}.\Phi)} \ \textbf{alg-r-nochange-}\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\Gamma[\theta_a] \vdash \mathtt{NC}\ e \ominus \mathtt{NC}\ e \lesssim t[\theta_a] : \square\tau[\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma', \square\Gamma, \tau, t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models (0 \doteq t \wedge \exists t' :: \mathbb{R}.\Phi)[\theta_a]$ $(\star\star)$
Using $(\star)$ and the first premise, we can show that

a) $\mathrm{FIV}(\Phi_a, \Gamma, \tau, t') \subseteq t', dom(\Delta, \psi_a)$.

Using $(\star)$, $(\star\star)$'s derivation must be in a form such that we have

b) $\Delta; \Phi_a[\theta_a] \models 0 \doteq t[\theta_a]$
c) $\Delta \vdash T' :: \mathbb{R}$ for some T'
d) $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a, t' \mapsto T']$

By Theorem 47.3 on the premise using a), d) and $(\star)$, we can show that

$$\Delta; \Phi_a[\theta_a]; \square\Gamma[\theta_a] \vdash |e| \ominus |e| \lesssim T' : \tau[\theta_a] \tag{5.19}$$

By the **c-nochange** rule using eq. (5.19), we obtain $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\Gamma[\theta_a] \vdash \mathtt{NC}\ |e| \ominus \mathtt{NC}\ |e| \lesssim 0 : \square\tau[\theta_a]$.
By the **c-r-$\sqsubseteq$** rule using this and b), we obtain $\Delta; \Phi_a[\theta_a]; \Gamma'[\theta_a], \square\Gamma[\theta_a] \vdash \mathtt{NC}\ e \ominus \mathtt{NC}\ e \lesssim t[\theta_a] : \square\tau[\theta_a]$.

$$\textbf{Case}\ \frac{\begin{array}{c}\Delta;\psi_a;\Phi_a;\Gamma\vdash e\ominus e'\uparrow \textbf{list}[n]^{\alpha}\,\tau\Rightarrow[\psi],t_1,\Phi_e\\ t_2\in\textbf{fresh}(\mathbb{R})\qquad \Delta;t_2,\psi,\psi_a;n\doteq 0\wedge\Phi_a;\Gamma\vdash e_1\ominus e_1'\downarrow\tau',t_2\Rightarrow\Phi_1\\ i::\mathbb{N},\Delta;t_2,\psi,\psi_a;n\doteq i+1\wedge\Phi_a;h:\square\,\tau,tl:\textbf{list}[i]^{\alpha}\,\tau,\Gamma\vdash e_2\ominus e_2'\downarrow\tau',t_2\Rightarrow\Phi_2\\ i::\mathbb{N},\beta::\mathbb{N},\Delta;t_2,\psi,\psi_a;n\doteq i+1\wedge\alpha\doteq\beta+1\wedge\Phi_a;h:\tau,tl:\textbf{list}[i]^{\beta}\,\tau,\Gamma\vdash e_3\ominus e_3'\downarrow\tau',t_2\Rightarrow\Phi_3\\ \Phi_{body}=(n\doteq 0\to\Phi_1)\wedge(\forall i::\mathbb{N}.(n\doteq i+1)\to(\Phi_2\wedge\forall\beta::\mathbb{N}.(\alpha\doteq\beta+1)\to\Phi_3))\wedge t_1+t_2\doteq t\end{array}}{\Delta;\psi_a;\Phi_a;\Gamma\vdash\begin{array}{l}\textbf{case}\ e\ \textbf{of nil}\ \to e_1\\ \mid h::_{NC}tl\ \to e_2\\ \mid h::_C tl\ \to e_3\end{array}\ominus\begin{array}{l}\textbf{case}\ e'\ \textbf{of nil}\ \to e_1'\\ \mid h::_{NC}tl\ \to e_2'\\ \mid h::_C tl\ \to e_3'\end{array}\downarrow\tau',t\Rightarrow\exists(\psi).(\Phi_e\wedge\exists t_2::\mathbb{R}.\Phi_{body})}\ \textbf{alg-}$$

r-caseL-$\downarrow$

TS:

$$\Delta;\Phi_a[\theta_a];\Gamma[\theta_a]\vdash\begin{array}{l}\text{case}\ e\ \text{of nil}\ \to|e_1|\\ \mid h::_N tl\ \to|e_2|\\ \mid h::_C tl\to|e_3|\end{array}\ominus\begin{array}{l}\text{case}\ e'\ \text{of nil}\ \to|e_1'|\\ \mid h::_N tl\ \to|e_2'|\\ \mid h::_C tl\to|e_3'|\end{array}\lesssim t[\theta_a]:\tau'[\theta_a]$$

By the main assumptions, we have $\text{FIV}(\Phi_a,\Gamma,\tau',t)\subseteq dom(\Delta,\psi_a)$ $(\star)$ and $\Delta;\Phi_a[\theta_a]\models(\exists(\psi).\Phi_e\wedge\exists t_2::\mathbb{R}.\Phi_{body})[\theta_a]$ $(\star\star)$

By Theorem 46 using the first premise and $(\star)$, we get $\text{FIV}(\text{list}[n]^{\alpha}\,\tau,t_1,\Phi_e)\subseteq dom(\Delta,\psi;\psi_a)$ $(\diamond)$.

Using $(\star)$ and $(\diamond)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta\ \triangleright\ \theta:\psi$
b) $\Delta;\Phi_a[\theta_a]\models\Phi_e[\theta\,\theta_a]$
c) $\Delta\vdash T_2::\mathbb{R}$
d) $\Delta;n[\theta\,\theta_a]\doteq 0\wedge\Phi_a[\theta_a]\models\Phi_1[\theta\,\theta_a]$
e) $i::S,\Delta;n[\theta\,\theta_a]\doteq i+1\wedge\Phi_a[\theta_a]\models\Phi_2[\theta_a,\theta,t_2\mapsto T_2]$
f) $i::S,\beta::S,\Delta;n[\theta\,\theta_a]\doteq i+1\wedge\alpha[\theta\,\theta_a]\doteq\beta+1\wedge\Phi_a[\theta_a]\models\Phi_3[\theta_a,\theta,t_2\mapsto T_2]$
g) $\Delta;\Phi_a[\theta_a]\models t_1[\theta\,\theta_a]+T_2\doteq t[\theta_a]$

By Theorem 47.4 on the first premise using b) and $(\star)$, we can show that

$$\Delta;\Phi_a[\theta_a];\Gamma[\theta_a]\vdash|e|\ominus|e'|\lesssim t_1[\theta\,\theta_a]:\text{list}[n[\theta\,\theta_a]]^{\alpha[\theta\,\theta_a]}\,\tau[\theta\,\theta_a]\tag{5.20}$$

By Theorem 47.3 on the second premise using d) and $(\star)$, we can show that

$$\Delta;n[\theta\,\theta_a]\doteq 0\wedge\Phi_a[\theta_a];\Gamma[\theta_a]\vdash|e_1|\ominus|e_1'|\lesssim T_2:\tau'[\theta\,\theta_a]\tag{5.21}$$

By Theorem 47.3 on the third premise using e) and $(\star)$, we can show that

$$i::S,\Delta;n[\theta\,\theta_a]\doteq i+1\wedge\Phi_a[\theta_a];\Gamma[\theta_a]\vdash|e_2|\ominus|e_2'|\lesssim T_2:\tau'[\theta\,\theta_a]\tag{5.22}$$

By Theorem 47.3 on the fourth premise using f) and ($\star$), we can show that

$$i :: S, \beta :: S, \Delta; n[\theta\,\theta_a] \doteq i+1 \wedge \alpha[\theta\,\theta_a] \doteq \beta+1 \wedge \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_3| \ominus |e_3'| \lesssim T_2 : \tau'[\theta\,\theta_a] \quad (5.23)$$

Then by **c-r-caseL** rule using eqs. (5.20) to (5.23), we can show that

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \begin{array}{l} \text{case } e \text{ of nil } \to |e_1| \\ \quad \mid h ::_N tl \to |e_2| \\ \quad \mid h ::_C tl \to |e_3| \end{array} \ominus \begin{array}{l} \text{case } e' \text{ of nil } \to |e_1'| \\ \quad \mid h ::_N tl \to |e_2'| \\ \quad \mid h ::_C tl \to |e_3'| \end{array} \lesssim t_1[\theta\,\theta_a] + T_2 : \tau'[\theta_a]$$

We conclude by applying **c-r-$\sqsubseteq$** rule to this using g).

Case

$$\frac{\begin{array}{c} t_1, t_2 \in \mathbf{fresh}(\mathbb{R}) \qquad i \in \mathbf{fresh}(\mathbb{N}) \qquad \Delta; t_1, \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \downarrow \square\,\tau, t_1 \Rightarrow \Phi_1 \\ \Delta; i, t_2, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow \mathbf{list}[i]^{\alpha}\,\tau, t_2 \Rightarrow \Phi_2 \qquad \Phi_2' = \Phi_2 \wedge n \doteq (i+1) \wedge t_1 + t_2 \doteq t \end{array}}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathbf{cons}_{NC}(e_1, e_2) \ominus \mathbf{cons}_{NC}(e_1', e_2') \downarrow \mathbf{list}[n]^{\alpha}\,\tau, t \Rightarrow \exists t_1 :: \mathbb{R}.(\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2')}\ \textbf{alg-r-consNC-}\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \text{cons}_{NC}(|e_1|, |e_2|) \ominus \text{cons}_{NC}(|e_1'|, |e_2'|) \lesssim t[\theta_a] : \text{list}[n[\theta_a]]^{\alpha[\theta_a]}\,\tau[\theta_a]$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, \text{list}[n]^{\alpha}\,\tau, t) \subseteq dom(\Delta, \psi_a)$ ($\star$) and
$\Delta; \Phi_a[\theta_a] \models (\exists t_1 :: \mathbb{R}.\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2')[\theta_a]$ ($\star\star$)
Using ($\star$), ($\star\star$)'s derivation must be in a form such that we have

a) $\Delta \vdash T_1 :: \mathbb{R}$
b) $\Delta \vdash T_2 :: \mathbb{R}$
c) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta_a, t_1 \mapsto T_1]$
d) $\Delta \vdash I :: \mathbb{N}$
e) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, t_2 \mapsto T_2, i \mapsto I]$
f) $\Delta; \Phi_a[\theta_a] \models (I+1) \doteq n[\theta_a]$
g) $\Delta; \Phi_a[\theta_a] \models (T_1 + T_2) \doteq t[\theta_a]$

By Theorem 47.3 on the third premise using ($\star$) and c), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e_1| \ominus |e_1'| \lesssim T_1 : \square\,\tau[\theta_a] \quad (5.24)$$

By Theorem 47.3 on the fourth premise using ($\star$) and e), we can show that

$$\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |e_2| \ominus |e_2'| \lesssim T_2 : \text{list}[I]^{\alpha[\theta_a]}\,\tau[\theta_a] \quad (5.25)$$

By **c-r-cons1** typing rule using eqs. (5.24) and (5.25), we obtain
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash \text{cons}_{NC}(|e_1|, |e_2|) \ominus \text{cons}_{NC}(|e_1'|, |e_2'|) \lesssim T_1 + T_2 : \text{list}[I+1]^{\alpha[\theta_a]}\,\tau[\theta_a]$.
We conclude by applying **c-r-$\sqsubseteq$** rule to this using f) and g).

Case $$\frac{\Delta;\psi_a;\Phi;|\Gamma|_1 \vdash e_1 \uparrow A_1 \xrightarrow{\textbf{exec}(k_e,t_e)} A_2 \Rightarrow [\psi], k_1, t_1, \Phi_1 \qquad t_2 \in \textbf{fresh}(\mathbb{R}) \qquad \Delta; t_2, \psi, \psi_a; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \downarrow U\,(A_1, A_2'), t_2 \Rightarrow \Phi_2}{\Delta;\psi_a;\Phi_a;\Gamma \vdash e_1\ e_2 \ominus e_2' \downarrow U\,(A_2, A_2'), t \Rightarrow \exists(\psi).\Phi_1 \wedge \exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 + t_e + c_{app} \doteq t}\ \textbf{alg-r-app-e}\!\downarrow$$

TS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1|\ |e_2| \ominus |e_2'| \lesssim t[\theta_a] : U\,(A[\theta_a], A'[\theta_a])$.
By the main assumptions, we have $\text{FIV}(\Phi_a, \Gamma, U\,(A, A'), t) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models (\exists(\psi).\Phi_1 \wedge (\exists t_2 :: \mathbb{R}.\Phi_2 \wedge t_1 + t_2 + t_e + c_{app} \doteq t) \wedge \Phi_3)[\theta_a]$ $(\star\star)$ such that $\Delta \ \triangleright\ \theta_a : \psi_a$ is derivable.
By Theorem 46 using $(\star)$ and the first premise, we get
$\text{FIV}(A_1 \xrightarrow{\text{exec}(k_e,t_e)} A_2, k_1, t_1, \Phi_1) \subseteq dom(\Delta, \psi; \psi_a)$ $(\diamond)$.
Using $(\star)$ and $(\diamond)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta \ \triangleright\ \theta : \psi$
b) $\Delta; \Phi_a[\theta_a] \models \Phi_1[\theta\,\theta_a]$
c) $\Delta; \Phi_a[\theta_a] \models \Phi_2[\theta_a, \theta, t_2 \mapsto T_2]$
d) $\Delta; \Phi_a[\theta_a] \models \Phi_3[\theta\,\theta_a]$
e) $\Delta; \Phi_a[\theta_a] \models t_1[\theta\,\theta_a] + t_e[\theta\,\theta_a] + T_2 + c_{app} \doteq t[\theta_a]$

By Theorem 47.2 on the first premise using $(\star)$, a) and b), we can show that

$$\Delta; \Phi_a[\theta_a]; |\Gamma[\theta_a]|_1 \vdash^{t_1[\theta\,\theta_a]}_{k_1[\theta\,\theta_a]} |e_1| :^c A_1[\theta\,\theta_a] \xrightarrow{\text{exec}(k_e[\theta\,\theta_a], t_e[\theta\,\theta_a])} A_2[\theta\,\theta_a] \tag{5.26}$$

From $(\star)$ and $(\diamond)$, we can show that

f) $\text{FIV}(\Phi_a, \Gamma, U\,(A_1, A'), t_2) \subseteq t_2, dom(\Delta, \psi; \psi_a)$

By Theorem 47.3 on the third premise using c), $(\star)$ and $(\diamond)$, we obtain

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_2| \ominus |e_2'| \lesssim T_2 : U\,(A_1[\theta\,\theta_a], A'[\theta\,\theta_a]) \tag{5.27}$$

By Theorem 40 on the fourth premise using $(\diamond)$, $(\star)$ and (d), we obtain

$$\Delta; \Phi_a[\theta_a] \models^{\mathsf{A}} A_2[\theta\,\theta_a] \sqsubseteq A[\theta_a] \tag{5.28}$$

By **U** subtyping rule using eq. (5.28) and **u-refl** subtyping rule, we obtain

$$\Delta; \Phi_a[\theta_a] \models U\,(A_2[\theta\,\theta_a], A'[\theta_a]) \sqsubseteq U\,(A[\theta_a], A'[\theta_a]) \tag{5.29}$$

Then by the **c-r-app-e** rule using eqs. (5.26) and (5.27), we can show that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1|\ |e_2| \ominus |e_2'| \lesssim t_1[\theta\,\theta_a] + t_e[\theta\,\theta_a] + T_2 + c_{app} : U\,(A_2[\theta_a], A'[\theta_a])$.
Applying **c-r-$\sqsubseteq$** rule to this using (e) and eq. (5.29), we can conclude as

$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e_1|\ |e_2| \ominus |e_2'| \lesssim t[\theta_a] : U\,(A[\theta_a], A'[\theta_a])$.

Proof of Theorem 47.4:

Case $$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e \ominus e' \downarrow \tau, t \Rightarrow \Phi \qquad \Delta; \Phi_a \vdash \tau\ \mathtt{wf} \qquad \mathbf{FIV}(\tau, t) \in \Delta}{\Delta; \psi_a; \Phi_a; \Gamma \vdash (e : \tau, t) \ominus (e' : \tau, t) \uparrow \tau \Rightarrow [\cdot], t, \Phi}\ \textbf{alg-r-anno-}\uparrow$$
TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash |(e : \tau, t)| \ominus |(e' : \tau, t)| \lesssim t[\theta_a] : \tau[\theta_a]$.
Since by definition, $\forall e.\ |(e : \_, \_)| = |e|$, STS: $\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta_a] : \tau[\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and $\Delta; \Phi_a[\theta_a] \models \Phi[\theta_a]$ $(\star\star)$
Using the third premise, we can show that

a) $\mathrm{FIV}(\Phi_a, \Gamma, \tau, k, t) \subseteq dom(\Delta, \psi_a)$.

By Theorem 47.4 on the first premise using $(\star\star)$ and a), we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta_a] : \tau[\theta_a]$.

Case $$\frac{\Delta; \psi_a; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \uparrow \Box\,\tau \Rightarrow [\psi], t, \Phi}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \mathtt{der}\ e_1 \ominus \mathtt{der}\ e_2 \uparrow \tau \Rightarrow [\psi], t, \Phi}\ \textbf{alg-r-der-}\uparrow$$
TS: $\Delta; \Phi_a[\theta_a]; \Omega[\theta_a] \vdash^{|e|}_{t[\theta\,\theta_a]} |e'| :^c \tau[\theta\,\theta_a]$.
By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma) \subseteq dom(\Delta, \psi_a)$ $(\star)$ and
$\Delta; \Phi_a[\theta_a] \models \Phi[\theta\,\theta_a]$ $(\star\star)$ such that $\Delta \ \triangleright\ \theta : \psi$ $(\diamond)$ and $\Delta \ \triangleright\ \theta_a : \psi_a$ are derivable.
By Theorem 47.4 on the first premise using $(\star)$ and $(\diamond)$, we obtain

$$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta\,\theta_a] : \Box\,\tau[\theta\,\theta_a] \tag{5.30}$$

Then, by **c-der** rule using eq. (5.30) and the second premise, we can conclude that
$\Delta; \Phi_a[\theta_a]; \Gamma[\theta_a] \vdash |e| \ominus |e'| \lesssim t[\theta\,\theta_a] : \tau[\theta\,\theta_a]$.

$\square$

Theorem 48 (Completeness of the Algorithmic Typechecking in RelCost)
We have the following.

1. Assume that $\Delta; \Phi_a; \Omega \vdash^t_k e :^c A$. Then, $\exists e'$ such that
   1.1. $\Delta; \cdot; \Phi_a; \Omega \vdash e' \downarrow A, k, t \Rightarrow \Phi$
   1.2. $\Delta; \Phi_a \models \Phi$
   1.3. $|e'| = e$
2. Assume that $\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \tau$. Then, $\exists e_1', e_2'$ such that
   2.1. $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau, t \Rightarrow \Phi$
   2.2. $\Delta; \Phi_a \models \Phi$
   2.3. $|e_1'| = e_1$ and $|e_2'| = e_2$

Proof. Proof is by simultaneous induction on the RelCostCore typing derivations.

Proof of Theorem 48.1:

Case $\dfrac{\Omega(x) = A}{\Delta; \Phi_a; \Omega \vdash_0^0 x :^c A}$ **c-var**

We can conclude as follows

$$\dfrac{\dfrac{\Omega(x) = A}{\Delta; \cdot; \Phi_a; \Omega \vdash x \uparrow A \Rightarrow [.], 0, 0, \top}\ \textbf{alg-u-var-}\uparrow \qquad \Delta; \Phi_a \models^{\mathsf{A}} A \sqsubseteq A \Rightarrow \Phi \ \text{ by } \textit{lemma } 38}{\Delta; \psi_a; \Phi_a; \Omega \vdash x \downarrow A, 0, 0 \Rightarrow \top}\ \textbf{alg-r-}\uparrow\downarrow$$

Case $\dfrac{\Delta; \Phi_a; \Omega \vdash_{k_1}^{t_1} e_1 :^c A \qquad \Delta; \Phi_a; \Omega \vdash_{k_2}^{t_2} e_2 :^c \mathbf{list}[n]\, A}{\Delta; \Phi_a; \Omega \vdash_{k_1+k_2}^{t_1+t_2} \mathbf{cons}_C(e_1, e_2) :^c \mathbf{list}[n+1]\, A}$ **c-cons**

By Theorem 48.2 on the first premise, $\exists e_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Omega \vdash e_1' \downarrow A, k_1, t_1 \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|e_1'| = e_1$

By a), we can show that for $k_1', t_1' \in \text{fresh}(\mathbb{R})$ where $\Phi_1' = \Phi_1 \wedge k_1 \doteq k_1' \wedge t_1 \doteq t_1'$

$$\Delta; k_1', t_1'; \Phi_a; \Omega \vdash e_1' \downarrow A, k_1', t_1' \Rightarrow \Phi_1' \tag{5.31}$$

By Theorem 48.2 on the second premise, $\exists e_2'$ such that

d) $\Delta; \cdot; \Phi_a; \Omega \vdash e_2' \downarrow \text{list}[n]\, A, k_2, t_2 \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$

f) $|e_2'| = e_2$

By a), we can show that for $i, k_2', t_2' \in \text{fresh}(\mathbb{R})$ where $\Phi_2' = \Phi_2 \wedge k_2 \doteq k_2' \wedge t_2 \doteq t_2' \wedge i \doteq n$

$$\Delta; i, k_2', t_2'; \Phi_a; \Omega \vdash e_2' \downarrow \text{list}[i]\, A, k_2', t_2' \Rightarrow \Phi_2' \tag{5.32}$$

Then, we can conclude as follows

1.

$$\dfrac{\begin{array}{c} k_1', t_1', k_2', t_2' \in \text{fresh}(\mathbb{R}) \qquad i \in \text{fresh}(\mathbb{N}) \qquad \Delta; k_1', t_1', \psi_a; \Phi_a; \Omega \vdash e_1' \downarrow A, k_1', t_1' \Rightarrow \Phi_1' \ \ \textit{eq.}\ (5.31) \\ \Delta; i, k_2', t_2', \psi_a; \Phi_a; \Omega \vdash e_2' \downarrow \text{list}[i]\, A, k_2', t_2' \Rightarrow \Phi_2' \ \ \textit{eq.}\ (5.32) \\ \Phi_2'' = (\Phi_2' \wedge n + 1 \doteq (i+1) \wedge k_1 + k_2 \doteq k_1' + k_2' \wedge t_1' + t_2' \doteq t_1 + t_2) \end{array}}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{cons}_C(e_1', e_2') \downarrow \text{list}[n+1]\, A, k_1 + k_2, t_1 + t_2 \Rightarrow \exists k_1', t_1' :: \mathbb{R}.(\Phi_1' \wedge \exists k_2', t_2' :: \mathbb{R}.\exists i :: \mathbb{N}.\Phi_2'')}\ \textbf{alg-u-cons-}\downarrow$$

2. Using b) and e) for the substitutions $k'_i = k_i$ and $t'_i = t_i$ for the fresh costs and $i = n$ for the size of the tail.
3. Using c) and f), $|\text{cons}_C(e'_1, e'_2)| = \text{cons}_C(e_1, e_2)$

Proof of Theorem 48.2:

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_2 \lesssim t :^c \square\, \tau}{\Delta; \Phi_a; \Gamma \vdash \texttt{der}\ e_1 \ominus \texttt{der}\ e_2 \lesssim t :^c \tau}$ **c-der**

By Theorem 48.2 on the premise, $\exists e'_1, e'_2$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \tau, t \Rightarrow \Phi$

b) $\Delta; \Phi_a \models \Phi$

c) $|e'_1| = e_1$ and $|e'_2| = e_2$

Then, we can conclude by using a), b) and c) as follows:

$\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \square\, \tau, t \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der}\ e'_1 \ominus \texttt{der}\ e'_2 \downarrow \tau, t \Rightarrow \Phi}$ **alg-r-der-↓** and

1.

$$\dfrac{\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e'_1 \ominus e'_2 \downarrow \square\, \tau, t \Rightarrow \Phi}{\Delta; \cdot; \Phi_a; \Gamma \vdash \texttt{der}\ e'_1 \ominus \texttt{der}\ e'_2 \downarrow \tau, t \Rightarrow \Phi}\ \textbf{alg-r-der-}\downarrow}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e'_1 : \tau, t) \ominus (\texttt{der}\ e'_2 : \tau, t) \uparrow \tau \Rightarrow [\cdot], t, \Phi}\ \textbf{alg-r-anno-}\uparrow \quad \Delta; \Phi_a \models \tau \equiv \tau \Rightarrow \Phi' \text{ by } \textit{Lemma } 37}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\texttt{der}\ e'_1 : \tau, t) \ominus (\texttt{der}\ e'_2 : \tau, t) \downarrow \tau, t \Rightarrow \Phi \wedge \Phi' \wedge t \leq t}\ \textbf{alg-r-}\uparrow\downarrow$$

2. By c), $|(\texttt{der}\ e'_i : \tau, t)| = \texttt{der}\ |e'_i|$.
3. By b) and Lemma 37.

Case $\dfrac{\Delta; \Phi_a; |\Gamma| \vdash^{t_1}_{k_1} e_1 :^c A \qquad \Delta; \Phi_a; |\Gamma| \vdash^{t_2}_{k_2} e_2 :^c A}{\Delta; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1 \ominus \texttt{switch}\ e_2 \lesssim t_1 - k_2 :^c U\, A}$ **c-switch**

By Theorem 48.1 on the first premise , $\exists e'_1$ such that

a) $\Delta; \cdot; \Phi_a; |\Gamma|_1 \vdash e'_1 \downarrow A_1, k_1, t_1 \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|e'_1| = e_1$

By a), we can show that for $k'_1, t'_1 \in \text{fresh}(\mathbb{R})$ where $\Phi'_1 = \Phi_1 \wedge k_1 \doteq k'_1 \wedge t_1 \doteq t'_1$

$$\Delta; k'_1, t'_1; \Phi_a; |\Gamma|_1 \vdash e'_1 \downarrow A_1, k'_1, t'_1 \Rightarrow \Phi'_1 \tag{5.33}$$

By Theorem 48.1 on the second premise , $\exists e_2'$ such that

d) $\Delta; \cdot; \Phi_a; |\Gamma|_2 \vdash e_2' \downarrow A_2, k_2, t_2 \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$

f) $|e_2'| = e_2$

By d), we can show that for $k_2', t_2' \in \text{fresh}(\mathbb{R})$ where $\Phi_2' = \Phi_2 \wedge k_2 \doteq k_2' \wedge t_2 \doteq t_2'$

$$\Delta; k_2', t_2'; \Phi_a; |\Gamma|_2 \vdash e_2' \downarrow A_2, k_2', t_2' \Rightarrow \Phi_2' \tag{5.34}$$

Then, we can conclude as follows

1. By using eqs. (5.33) and (5.34):
$$\frac{k_1', t_1', k_2', t_2' \in \text{fresh}(\mathbb{R}) \quad \Delta; k_1', t_1', \psi_a; \Phi; |\Gamma| \vdash e_1' \downarrow A, k_1', t_1' \Rightarrow \Phi_1' \qquad \Delta; k_2', t_2', \psi_a; \Phi; |\Gamma| \vdash e_2' \downarrow A, k_2', t_2' \Rightarrow \Phi_2'}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \texttt{switch}\ e_1' \ominus \texttt{switch}\ e_2' \downarrow t, U\, A \Rightarrow \exists k_1, t_1 :: \mathbb{R}.(\Phi_1' \wedge \exists k_2', t_2' :: \mathbb{R}.\Phi_2' \wedge t_1' - k_2' \doteq t)}\ \textbf{alg-r-switch}\downarrow.$$
2. By using b) and e) and the substitutions $k_i' = k_i$ and $t_i' = t_i$ for the fresh costs where $t = t_1 - k_2$.
3. By c) and f), we get $|\texttt{switch}\ e_i'| = \texttt{switch}\ |e_i|$

Case $$\frac{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t :^c \tau \qquad \Delta; \Phi_a \models \tau \equiv \tau' \qquad \Delta; \Phi_a \models t \leq t'}{\Delta; \Phi_a; \Gamma \vdash e \ominus e' \lesssim t' :^c \tau'}\ \textbf{c-r-}\equiv$$

By Theorem 48.2 on the first premise, $\exists e_1', e_2'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau, t \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|e_1'| = e$ and $|e_2'| = e'$

By Theorem 43 on the second premise,

d) $\Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$.

Then, we can conclude as follows

1. By using a) and d)
$$\frac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash e_1' \ominus e_2' \downarrow \tau, t \Rightarrow \Phi_1}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e_1' : \tau, t) \ominus (e_2' : \tau, t) \uparrow \tau \Rightarrow [\cdot], t, \Phi_1}\ \textbf{alg-r-anno-}\uparrow \qquad \Delta; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_2}{\Delta; \cdot; \Phi_a; \Gamma \vdash (e_1' : \tau, t) \ominus (e_2' : \tau, t) \downarrow \tau', t' \Rightarrow \Phi_1 \wedge \Phi_2 \wedge t \leq t'}\ \textbf{alg-r-}\uparrow\downarrow$$
2. By using b), e) and the third premise, we can show that $\delta; \Phi_a \models \Phi_1 \wedge \Phi_2 \wedge t \leq t'$
3. By c), $|(e_1' : \tau, t)| = e$ and $(e_2' : \tau, t) = e'$

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \tau_1 \xrightarrow{\mathbf{diff}(t)} \tau_2 \qquad \Delta; \Phi_a; \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_1}{\Delta; \Phi_a; \Gamma \vdash e_1\ e_2 \ominus e_1'\ e_2' \lesssim t_1 + t_2 + t :^c \tau_2}$ **c-r-app**

By Theorem 48.2 on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1 \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By Theorem 48.2 on the second premise, $\exists \overline{e}_2, \overline{e}_2'$ such that

d) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1, t_2 \Rightarrow \Phi_2$

e) $\Delta; \Phi_a \models \Phi_2$

f) $|\overline{e}_2| = e_2$ and $|\overline{e}_2'| = e_2'$

By d), we can show that for $t_2' \in \text{fresh}(\mathbb{R})$ where $\Phi_2' = \Phi_2 \wedge t_2 \doteq t_2'$

$$\Delta; t_2'; \Phi_a; \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1, t_2' \Rightarrow \Phi_2' \tag{5.35}$$

Then, we can conclude as follows

1.

$$\dfrac{\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1 \Rightarrow \Phi_1}{\Delta; \cdot; \Phi_a; \Gamma \vdash (\overline{e}_1 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1) \ominus (\overline{e}_1' : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1) \uparrow \tau_1 \xrightarrow{\text{diff}(t)} \tau_2 \Rightarrow [\cdot], t_1, \Phi_1}\ \textbf{alg-r-anno-}\uparrow \qquad t_2' \in \text{fresh}(\mathbb{R}) \qquad \Delta; t_2'; \Phi_a; \Gamma \vdash \overline{e}_2 \ominus \overline{e}_2' \downarrow \tau_1, t_2' \Rightarrow \Phi_2'}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E_2 \uparrow \tau_2 \Rightarrow [t_2'], t_1 + t + t_2', \Phi_1 \wedge \Phi_2'}\ \textbf{c-r-app}}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E_2 \downarrow \tau_2, t_1 + t + t_2 \Rightarrow \exists t_2' :: \mathbb{R}. \Phi_1 \wedge \Phi_2' \wedge t_1 + t + t_2' \leq t_1 + t + t_2}\ \textbf{alg-r-}\uparrow$$

where $E_1 = (\overline{e}_1 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1)\ \overline{e}_2$ and $E_2 = (\overline{e}_1' : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1)\ \overline{e}_2'$.

2. By using b) and e) and the substitution $t_2' = t_2$ for the fresh cost.

3. Using c) and f), $|(\overline{e}_1 : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1)\ \overline{e}_2| = e_1\ e_2$ and $|(\overline{e}_1' : \tau_1 \xrightarrow{\text{diff}(t)} \tau_2, t_1)\ \overline{e}_2'| = e_1'\ e_2'$.

Case $\dfrac{\Delta; \Phi_a; \Gamma \vdash e_1 \ominus e_1' \lesssim t_1 :^c \exists i{::}S.\, \tau_1 \qquad i :: S, \Delta; \Phi_a; x : \tau_1, \Gamma \vdash e_2 \ominus e_2' \lesssim t_2 :^c \tau_2 \qquad i \notin FV(\Phi_a; \Gamma, \tau_2, t_2)}{\Delta; \Phi_a; \Gamma \vdash \mathbf{unpack}\ e_1\ \mathbf{as}\ (x, i)\ \mathbf{in}\ e_2 \ominus \mathbf{unpack}\ e_1'\ \mathbf{as}\ (x, i)\ \mathbf{in}\ e_2' \lesssim t_1 + t_2 :^c \tau_2}$ **c-r-unpack1**

By Theorem 48.2 on the first premise, $\exists \overline{e}_1, \overline{e}_1'$ such that

a) $\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}_1' \downarrow \exists i{::}S.\, \tau_1, t_1 \Rightarrow \Phi_1$

b) $\Delta; \Phi_a \models \Phi_1$

c) $|\overline{e}_1| = e_1$ and $|\overline{e}_1'| = e_1'$

By Theorem 48.2 on the second premise, $\exists \overline{e}_2, \overline{e}'_2$ such that

d) $i :: S, \Delta; \cdot; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}'_2 \downarrow \tau_2, t_2 \Rightarrow \Phi_2$

e) $i :: S, \Delta; \Phi_a \models \Phi_2$

f) $|\overline{e}_2| = e_2$ and $|\overline{e}'_2| = e'_2$

By d), we can show that for $t'_2 \in \text{fresh}(\mathbb{R})$ where $\Phi'_2 = \Phi_2 \wedge t_2 \doteq t'_2$

$$i :: S, \Delta; t'_2; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}'_2 \downarrow \tau_2, t'_2 \Rightarrow \Phi'_2 \tag{5.36}$$

Then, we can conclude as follows

1.

$$\dfrac{\dfrac{\Delta; \cdot; \Phi_a; \Gamma \vdash \overline{e}_1 \ominus \overline{e}'_1 \downarrow \exists i{::}S.\, \tau_1, t_1 \Rightarrow \Phi_1 \quad (a)}{\Delta; \cdot; \Phi_a; \Gamma \vdash E_1 \ominus E'_1 \uparrow \exists i{::}S.\, \tau_1 \Rightarrow [\cdot], t_1, \Phi_1}\ \textbf{alg-r-anno-}\uparrow \quad \begin{array}{c} i :: S, \Delta; t'_2; \Phi_a; x : \tau_1, \Gamma \vdash \overline{e}_2 \ominus \overline{e}'_2 \downarrow \tau_2, t'_2 \Rightarrow \Phi'_2 \quad (eq.\ (5.36)) \\ \Phi' = \exists t'_2 :: \mathbb{R}.\Phi_1 \wedge \Phi'_2 \wedge t_1 + t'_2 \doteq t_1 + t_2 \end{array}}{\Delta; \cdot; \Phi_a; \Gamma \vdash \text{unpack } E_1 \text{ as } (x, i) \text{ in } \overline{e}_2 \ominus \text{unpack } E'_1 \text{ as } (x, i) \text{ in } \overline{e}'_2 \downarrow \tau_2, t_1 + t_2 \Rightarrow \Phi'}\ \textbf{alg-r-unpack-}\downarrow$$

where $E_1 = (\overline{e}_1 : \exists i{::}S.\, \tau_1, t_1,)$ and $E_2 = (\overline{e}'_1 : \exists i{::}S.\, \tau_1, t_1,)$

2. By using b) and e) and the substitution $t'_2 = t_2$ for the fresh cost.

3. Using c) and f), $|\text{unpack } (\overline{e}_1 : \exists i{::}S.\, \tau_1, t_1) \text{ as } (x, i) \text{ in } \overline{e}_2| = \text{unpack } e_1 \text{ as } (x, i) \text{ in } e_2$ and $|\text{unpack } (\overline{e}'_1 : \exists i{::}S.\, \tau_1, t_1) \text{ as } (x, i) \text{ in } \overline{e}'_2| = \text{unpack } e'_1 \text{ as } (x, i) \text{ in } e'_2$.

□

6 Experimental Evaluation

In this section, we present example programs that we have typechecked with BiRelCost.

Example (map) Consider the standard list `map` function that applies the mapping function `map` to all the elements of the list.

$\Lambda.\text{fix } \texttt{map}(f).\Lambda.\Lambda.\lambda l.\ \text{case } l \text{ of nil} \to \text{nil}$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \mid h :: tl \to \text{cons}(f\ h,\ \texttt{map}\ f[\,]\,[\,]\ tl)$

Two runs of `map` can be typed with 0 cost and the following type:

$$\vdash \texttt{map} \ominus \texttt{map} \lesssim 0 : \forall t.(\square\,(\tau_1 \xrightarrow{\text{diff}(t)} \tau_2)) \to \forall n, \alpha.\text{list}[n]^\alpha\, \tau_1 \xrightarrow{\text{diff}(t\cdot\alpha)} \text{list}[n]^\alpha\, \tau_2\ . \tag{6.1}$$

Example (filter) Consider the standard list `filter` function that goes over the list and only returns the list elements that satisfy the filtering function f.

$\Lambda.\text{fix } \texttt{filter}(f).\Lambda.\Lambda.\lambda l.\ \text{case } l \text{ of nil} \to \text{pack nil}$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \mid h :: tl \to \text{let } r' = \texttt{filter}\ f[\,]\,[\,]\ tl \text{ in}$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \text{let } b = f\ h \text{ in unpack } r' \text{ as } r \text{ in if } b \text{ then pack cons}(h, r)$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \text{else pack } r$

Two runs of `filter` can be typed with 0 cost and the following type:

$$\vdash \texttt{filter} \ominus \texttt{filter} \lesssim 0 : \forall t.(\square\,(\tau \xrightarrow{\text{diff}(t)} \text{bool})) \to \forall n, \alpha.\text{list}[n]^{\alpha}\,\tau \xrightarrow{\text{diff}(t\cdot\alpha)} \exists j{::}S.\,U\,(\text{list}[j]\,|\tau|)\,. \tag{6.2}$$

Example (square-and-multiply) Consider the square-and-multiply algorithm, a fast way of computing the positive powers of a number based on the observation that $x^m = x \cdot (x^2)^{\frac{m-1}{2}}$ when m is odd, and $x^m = (x^2)^{\frac{m}{2}}$ when m is even. The following function, `sam`, implements this idea, assuming that m is stored in binary form in a list l of 0s and 1s, with the least significant bit at the head.

fix $\texttt{sam}(x).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to$ `contra`
| $b :: bs$ $\to$ case bs of
    nil $\to$ if $x = 0$ then 1 else x
  | $\_ :: \_$ $\to$ let r= $\texttt{sam}\,x[\,]\,[\,]\ bs$ in
      if $b = 0$ then r^2 else $x \cdot r^2$

Assuming that multiplication $(\cdot)$ and equality $(=)$ are primitives with types $(\text{int} \times \text{int}) \xrightarrow{\text{exec}(1,1)} \text{int}$ and $(\text{int} \times \text{int}) \xrightarrow{\text{exec}(1,1)} \text{bool}$, respectively, two runs of `sam` can be typed with 0 cost and the following type:

$$\vdash \texttt{sam} \ominus \texttt{sam} \lesssim 0 : \text{int}_r \xrightarrow{\text{diff}(0)} \forall n > 0, \alpha{::}\mathbb{N}.\ \text{list}[n]^{\alpha}\,U\,\text{int} \xrightarrow{\text{diff}(\alpha))} U\,\text{int} \tag{6.3}$$

Example (constant-time comparison) Consider the following comparison function `comp` that checks the equality of two passwords represented as equal-length lists of bits. In BiRelCost, we can show that `comp` is constant-time, i.e. its relative cost wrt. itself is 0.

fix $\texttt{comp}(_).\Lambda.\Lambda.\Lambda.\lambda(l_1, l_2).$case l_1 of
  nil $\to$ true
| $h_1 :: tl_1$ $\to$ case l_2 of
    nil $\to$ false
  | $h_2 :: tl_2$ $\to$ $\texttt{boolAnd}\,(\texttt{comp}\,()[\,]\,[\,]\,[\,]\ (tl_1, tl_2), \texttt{eq}\,(h_1, h_2))$

Assuming that `boolAnd` and `eq` are constant-time primitives with the same type $(U\,\text{bool} \times U\,\text{bool}) \xrightarrow{\text{diff}(0)} U\,\text{bool}$, two runs of `comp` can be typed with 0 cost and the following type:

$$\vdash \texttt{comp} \ominus \texttt{comp} \lesssim 0 : \text{unit}_r \xrightarrow{\text{diff}(0)} \forall n, \alpha, \beta{::}\mathbb{N}.\,(\text{list}[n]^{\alpha}\,U\,\text{int} \times \text{list}[n]^{\beta}\,U\,\text{int}) \xrightarrow{\text{diff}(0)} U\,\text{bool} \tag{6.4}$$

Example (two-dimensional count) Consider the following function `2Dcount` that counts the number of rows of a matrix M (represented as a list of lists in row-major form) that both contain a key x and satisfy a predicate p. The function takes as argument another function `find` that returns 1 when a given row l contains x, else returns 0.

fix $\texttt{2Dcount}(\texttt{find}).\Lambda.\Lambda.\lambda x.\lambda M.$case M of
  nil $\to 0$
| $l :: M'$ $\to$ let $r = \texttt{2Dcount}[\,]\,[\,]\ \texttt{find}\,x\,M'$ in
    let $r' = \texttt{find}\,x[\,]\ l$ in if $p\,l$ then $r + r'$ else r

Consider the following two different implementations of `find`.

fix $\texttt{find1}(x).\Lambda.\lambda l.$case l of
 nil $\rightarrow 0$
| $h :: tl \;\rightarrow\;$ if $h = x$ then 1 else $\texttt{find1}\ x[\,]\ tl$

fix $\texttt{find2}(x).\Lambda.\lambda l.$case l of
 nil $\rightarrow 0$
| $h :: tl \;\rightarrow\;$ if $(\texttt{find2}\ x[\,]\ tl) = 1$ then 1
  else if $(h = x)$ then 1 else 0

where find1 and find2 can be given the following weakest relational type:

$$\vdash \texttt{find1} \ominus \texttt{find2} \lesssim 0 : \; U\,(\mathrm{int} \to \forall n{::}\mathbb{N}.\,(\mathrm{list}[n]\,\mathrm{int}) \xrightarrow{\mathrm{exec}(\_,4\cdot n)} \mathrm{int}, \mathrm{int} \to \forall n{::}\mathbb{N}.\,(\mathrm{list}[n]\,\mathrm{int}) \xrightarrow{\mathrm{exec}(4\cdot n,\_)} \mathrm{int}). \tag{6.5}$$

Assuming that the predicate p is a constant-time primitive with type $U\,(\forall n, t \overset{\mathrm{exec}(\mathrm{list}[,]\,n)}{::} S.\,\mathrm{int} \xrightarrow{\mathrm{exec}(t,t)}$ bool), then we can type 2Dcount find1 and 2Dcount find2 with 0 cost as follows:

$$\vdash \texttt{2Dcount find1} \ominus \texttt{2Dcount find2} \lesssim 0 : \mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} \forall i, j{::}\mathbb{N}.\,\mathrm{list}[n]^0\,(\mathrm{list}[j]^0\,U\,\mathrm{int}) \xrightarrow{\mathrm{diff}(0)} U\,\mathrm{int} \tag{6.6}$$

Example (Mergesort) Consider the standard mergesort function.

fix $\texttt{msort}(_).\Lambda.\Lambda.\lambda l.$case l of
 nil $\rightarrow$ nil
| $h_1 :: tl_1 \;\rightarrow\;$ case tl_1 of
  nil $\rightarrow \mathrm{cons}(h_1, \mathrm{nil})$
 | $\_ :: \_ \;\rightarrow\; r = \texttt{bsplit}\ ()[\,][\,]\ l$ in
   unpack r as r' in unpack r' as r''
   clet r'' as z in $\texttt{merge}\ ()[\,][\,]\ (\texttt{msort}\ ()[\,][\,]\ (\pi_1 z), \texttt{msort}\ ()[\,][\,]\ (\pi_2 z))$

There are two helper functions: bsplit and merge. The helper function bsplit splits an input list into two nearly equal lists by alternating the input's elements to the two outputs.

fix $\texttt{bsplit}(\_)\Lambda.\Lambda.\lambda l.$ case l of
 nil $\rightarrow \langle \mathrm{nil}, \mathrm{nil}\rangle$
| $h_1 :: tl_1 \;\rightarrow\;$ case tl_1 of nil $\rightarrow \langle \mathrm{cons}(h_1, \mathrm{nil}), \mathrm{nil}\rangle$
  | $h_2 :: tl_2 \;\rightarrow\;$ let $r = \texttt{bsplit}\ tl_2$ in unpack r as r' in
   clet r' as z in pack $\langle \mathrm{cons}(h_1, \pi_1 z), \mathrm{cons}(h_2, \pi_2 z)\rangle$

It can be given the following relational type:

$$\texttt{bsplit} : (\mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\,\mathrm{list}[n]^\alpha\,\tau \xrightarrow{\mathrm{diff}(0)} \exists \beta{::}\mathbb{N}.\,(\mathrm{list}[\left\lceil \frac{n}{2} \right\rceil]^\beta\,\tau \times \mathrm{list}[\left\lfloor \frac{n}{2} \right\rfloor]^{\alpha-\beta}\,\tau)) \tag{6.7}$$

The helper function merge takes two sorted lists and merges them into a new sorted list.

fix $\texttt{merge}(\_).\Lambda.\Lambda.\lambda y$ case x of
 nil $\rightarrow y$
| $a :: as \;\rightarrow\;$ case y of
  nil $\rightarrow x$
 | $b :: bs \;\rightarrow\;$ if $a \leq b$ then $\mathrm{cons}(a, \texttt{merge}\ ()[\,][\,]\ as\ y)$
   else $\mathrm{cons}(b, \texttt{merge}\ ()[\,][\,]\ x\ bs)$

Assuming that the comparison operation ($\leq$) is unit cost primitive with type $(\text{int} \times \text{int}) \xrightarrow{\text{exec}(1,1,\text{bool})}$, then we can type merge as follows:

$$\vdash^0_0 \texttt{merge} : \ \text{int} \to \forall n, m{::}\mathbb{N}.\, (\text{list}[n]\,\text{int} \times \text{list}[m]\,\text{int}) \xrightarrow{\text{exec}(5\cdot(\texttt{min}(n,m)),5\cdot(n+m))} \text{list}[n+m]\,\text{int} \quad (6.8)$$

Using the unary type of merge, two runs of merge can be given a relational type by encapsulating into an unrelated type.
Then, we can give msort the following relational type

$$\texttt{msort} : \Box\,(\forall n, \alpha{::}\mathbb{N}.\, \text{list}[n]^{\alpha}\, U\,\text{int} \xrightarrow{\text{diff}(Q(n,\alpha))} U\,(\text{list}[n]\,\text{int})) \quad (6.9)$$

where $Q(n,\alpha) = \sum_{i=0}^{\lceil \texttt{log}_2(n) \rceil} 5 \cdot (\left\lceil \frac{2^i}{2} \right\rceil) \cdot \texttt{min}(\alpha, 2^{\lceil \texttt{log}_2(n)\rceil - i}) \in O(n \cdot (1 + \texttt{log}_2(\alpha)))$

Example (selection sort) Consider the standard selection sort algorithm that finds the smallest among a value and the elements in a list using the function select and then sorts the remaining list recursively. In BiRelCost, we can show that ssort is a constant time algorithm, i.e. its relative cost is 0.

fix $\texttt{select}(x).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to$ pack $\langle x, \text{nil}\rangle$
| $h :: tl$ $\to$ let $(small, big) = (\text{if } x < h \text{ then } \langle x, h\rangle \text{ else } \langle h, x\rangle : U\,(\text{int} \times \text{int}), 0)$
    let $r' = \texttt{select}\ small[\,]\,[\,]\,tl$ in
    unpack r' as $rpair$ in pack $\langle \pi_1\, rpair, \text{cons}(big, \pi_2\, rpair)\rangle$

The helper function select can be given the following relational type:

$$\vdash \texttt{select} \ominus \texttt{select} \lesssim 0 : U\,\text{int} \xrightarrow{\text{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\, \text{list}[n]^{\alpha}\, U\,\text{int} \xrightarrow{\text{diff}(0)} \exists \beta{::}\mathbb{N}.\, (U\,\text{int} \times \text{list}[n]^{\beta}\, U\,\text{int})\ . \quad (6.10)$$

Then, we can type ssort in BiRelCost as follows:

fix $\texttt{ssort}(l).$case l of
  nil $\to$ pack nil
| $h :: tl$ $\to$ let $r = \texttt{select}\ h[\,]\,[\,]\,tl$ in
    unpack r as $rpair$ in unpack $\texttt{ssort}\ ()[\,]\,[\,]\,(\pi_2\, rpair)$ as $rest$ in pack $\text{cons}(\pi_1\ rpair, rest)$

$$\vdash \texttt{ssort} \ominus \texttt{ssort} \lesssim 0 : \ \text{unit}_r \xrightarrow{\text{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\, \text{list}[n]^{\alpha}\, U\,\text{int} \xrightarrow{\text{diff}(0)} \exists \beta{::}\mathbb{N}.\, \text{list}[n]^{\beta}\, U\,\text{int}. \quad (6.11)$$

Example (approximate sum) Consider two implementations of the mean calculation of a list of numbers. The first function computes the sum of a list of numbers and divides the sum by the length of the list whereas the second function (its approximate version) only computes the sum of the half of the elements, divides this sum by the total length of the list and then doubles the result afterwards.

fix $\texttt{sum}(acc).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to acc$
| $h :: tl$ $\to$ case tl of
    nil $\to h + acc$
  | $h' :: tl'$ $\to$ $\texttt{sum}\ (h +$ h'+ acc $)[\,]\,[\,]\ tl'$

fix $\texttt{sumAppr}(acc).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to acc$
| $h :: tl \ \to$ case tl of
    nil $\to h + acc$
    | $h' :: tl' \ \to$ $\texttt{sum}$ (h'+ acc)[] [] tl'

$$\vdash \texttt{sum} \ominus \texttt{sumAppr} \lesssim 0 : (U\,\text{int}) \xrightarrow{\text{diff}(0)} \forall n{::}\mathbb{N}.\,\text{list}[n]^{\alpha}\,U\,\text{int} \xrightarrow{\text{diff}(n)} U\,\text{int}\ .$$

Example (balanced fold) The following balanced fold function is often used in incremental computing. Given an *associative and commutative* binary function f, a list can be folded by splitting it into two nearly equal sized lists, folding the sublists recursively and then applying f to the two results.

fix $\texttt{bfold}(f).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to 0$
| $h_1 :: tl_1 \ \to$ case tl_1 of
    nil $\to h$
    | $\_ :: \_ \ \to$ $r = \texttt{bsplit}\ ()[\,]\,[\,]\ l$ in
      unpack r as r' in clet r' as z in $\texttt{f}$ ($\texttt{bfold}\ f[\,]\,[\,]\ (\pi_1 z), \texttt{bfold}\ f[\,]\,[\,]\ (\pi_2 z)$)

The helper function $\texttt{bsplit}$ splits an input list into two nearly equal lists by alternating the input's elements to the two outputs. Its relational type is shown at eq. (6.7). Then, like $\texttt{msort}$, we can give $\texttt{bfold}$ the following relational type

$$\vdash \texttt{bfold} \ominus \texttt{bfold} \lesssim 0 : \Box\,(\Box\,(U\,((\text{int} \times \text{int}) \xrightarrow{\text{exec}(1,2)} \text{int})) \xrightarrow{\text{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\,\text{list}[n]^{\alpha}\,U\,\text{int} \xrightarrow{\text{diff}(P(n,\alpha))} U\,\text{int}) \tag{6.12}$$

$$\text{where } P(n, \alpha) = \sum_{i=0}^{\lceil \texttt{log}_2(n) \rceil} \texttt{min}(\alpha, 2^{\lceil \texttt{log}_2(n) \rceil - i}) \in O(\alpha + \alpha * \texttt{log}_2(n/\alpha))$$

Λ.fix $\texttt{bfold}(f).\Lambda.\Lambda.\lambda l.$case l of
  nil $\to 0$
| $h_1 :: tl_1 \ \to$ case tl_1 of
    nil $\to h$
    | $\_ :: \_ \ \to$ $r = \texttt{bsplit}\ ()[\,]\,[\,]\ l$ in
      unpack r as r' in clet r' as z in $\texttt{f}$ ($\texttt{bfold}\ f[\,]\,[\,]\ (\pi_1 z), \texttt{bfold}\ f[\,]\,[\,]\ (\pi_2 z)$)

$$\vdash \texttt{bfold} \ominus \texttt{bfold} \lesssim 0 : \forall k{::}\mathbb{N}.\,\Box\,(\Box\,(U\,(\text{int} \times \text{int}) \xrightarrow{\text{diff}(k)} U\,\text{int})) \xrightarrow{\text{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\,\text{list}[n]^{\alpha}\,U\,\text{int} \xrightarrow{\text{diff}(P(n,\alpha,k))} U\,\text{int}) \tag{6.13}$$

$$\text{where } P(n, \alpha, k) = \sum_{k=0}^{\lceil \texttt{log}_2(n) \rceil} k * \texttt{min}(\alpha, 2^{\lceil \texttt{log}_2(n) \rceil - k}) \in O(k * (\alpha + \alpha * \texttt{log}_2(n/\alpha)))$$

Example (list append) Consider the standard list $\texttt{append}$ function that appends a list to another list.

fix $\texttt{append}(_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda l_1.\lambda l_2.$case l_1 of
  nil $\to l_2$
| $h :: tl \ \to$ cons($h, \texttt{append}\ ()[\,]\,[\,]\,[\,]\,[\,]\ tl\ l_2$)

We can type two runs of $\texttt{append}$ as follows:

$$\vdash \texttt{append} \ominus \texttt{append} \lesssim 0 : \text{unit}_r \xrightarrow{\text{diff}(0)} \forall i, j, \alpha, \beta{::}\mathbb{N}.\,\text{list}[i]^{\alpha}\,U\,\text{int} \xrightarrow{\text{diff}(0)} \text{list}[j]^{\beta}\,U\,\text{int} \xrightarrow{\text{diff}(0)} \text{list}[i+j]^{\alpha+\beta}\,U\,\text{int}\ .$$

Example (list reverse) Consider the standard list `reverse` function that reverses the list.

fix $\texttt{rev}(_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda l.\lambda acc.$case l_1 of
  nil $\to l_2$
| $h :: tl \;\to\; \texttt{rev}\ ()[\,]\,[\,]\,[\,]\,[\,]\, tl\ \mathrm{cons}(h, acc)$

We can type two runs of `rev` as follows:

$$\vdash \texttt{rev} \ominus \texttt{rev} \lesssim 0 : \mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} \forall i, j, \alpha, \beta{::}\mathbb{N}.\, \mathrm{list}[i]^{\alpha}\, U\, \mathrm{int} \xrightarrow{\mathrm{diff}(0)} \mathrm{list}[j]^{\beta}\, U\, \mathrm{int} \xrightarrow{\mathrm{diff}(0)} \mathrm{list}[i+j]^{\alpha+\beta}\, U\, \mathrm{int}\ .$$

Example (list flatten) Consider the standard list `flatten` function that flattens a lists of lists.

fix $\texttt{flatten}(_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda M.$case M of
  nil $\to$ nil
| $l :: M' \;\to$ let $r = \texttt{flatten}\ ()[\,]\,[\,]\,[\,]\,[\,]\, M'$ in
      $\texttt{append}\ ()[\,]\,[\,]\,[\,]\,[\,]\ l\ r$

We can type two runs of `flatten` as follows:

$$\vdash \texttt{flatten} \ominus \texttt{flatten} \lesssim 0 : \mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} \forall i, j, \alpha, \beta{::}\mathbb{N}.\, \mathrm{list}[i]^{\alpha}\, (\mathrm{list}[j]^{\beta}\, U\, \mathrm{int}) \xrightarrow{\mathrm{diff}(0)} \mathrm{list}[i \cdot j]^{i \cdot j}\, U\, \mathrm{int}\ .$$

Example (list zip)

fix $\texttt{zip}(\_)\Lambda.\Lambda.\Lambda.\lambda x.$ case $(\pi_1 x)$ of
  nil $\to$ nil
| $h_1 :: tl_1 \;\to$ case $(\pi_2 x)$ of nil $\to \cdots$
      | $h_2 :: tl_2 \;\to$ let $rest = \texttt{zip}\ f[\,]\,[\,]\,[\,]\ (tl_1, tl_2)$ in
        let $fh = f\ \langle h, h' \rangle$ in $\mathrm{cons}(fh, rest)$

It can be given the following relational type:

$$\vdash \texttt{zip} \ominus \texttt{zip} \lesssim 0 : (\Box\, U\, ((\tau_1 \times \tau_2) \xrightarrow{\mathrm{exec}(1,1)} \tau_3)) \xrightarrow{\mathrm{diff}(0)} \forall n, \alpha, \beta{::}\mathbb{N}.\, (\mathrm{list}[n]^{\alpha}\, \tau_1 \times \mathrm{list}[n]^{\beta}\, \tau_2) \xrightarrow{\mathrm{diff}(0)} \mathrm{list}[n]^{min(n,\alpha+\beta)}\, \tau_3 \tag{6.14}$$

Example(multi_sort)

fix $\texttt{multi\_sort}(\_)\Lambda.\Lambda.\lambda l.$
  let $l_1 = \texttt{ssort}\ ()[\,]\,[\,]\, l$ in
  unpack l_1 as l_2 in
  let $l_3 = \texttt{rev}\ ()[\,]\,[\,]\,[\,]\,[\,]\, l_2$ nil in
  let $l_4 = \texttt{ssort}\ ()[\,]\,[\,]\, l_3$ in
  unpack l_4 as l_5 in
  let $l_6 = \texttt{rev}\ ()[\,]\,[\,]\,[\,]\,[\,]\, l_5$ nil in
  let $l_7 = \texttt{msort}\ ()[\,]\,[\,]\, l$ in
  let $l_8 = \texttt{msort}\ ()[\,]\,[\,]\, l$ in
  let $l_9 = \texttt{msort}\ ()[\,]\,[\,]\, l$ in
  l_7

We can give `multi_sort` the following relational type

$$\vdash \texttt{multi\_sort} \ominus \texttt{multi\_sort} \lesssim 0 :: \Box \left(\mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} (\forall n, \alpha{::}\mathbb{N}.\, \mathrm{list}[n]^{\alpha}\, U\, \mathrm{int} \xrightarrow{\mathrm{diff}(3 * Q(n,\alpha))} U\, (\mathrm{list}[n]\, \mathrm{int}))\right) \tag{6.15}$$

where $Q(n, \alpha) = \sum_{i=0}^{\lceil \texttt{log}_2(n) \rceil} 5 \cdot \left(\left\lceil \frac{2^i}{2} \right\rceil\right) \cdot \texttt{min}(\alpha, 2^{\lceil \texttt{log}_2(n) \rceil - i}) \in O(n \cdot (1 + \texttt{log}_2(\alpha)))$

Example (ssort_list)

$$
\begin{aligned}
&\text{fix } \texttt{ssort\_list}(\_)\Lambda.\Lambda.\lambda l. \\
&\qquad\qquad \text{let } l_1 = \texttt{ssort}\ ()[\,]\,[\,]\, l \text{ in} \\
&\qquad\qquad \text{unpack } l_1 \text{ as } l_2 \text{ in} \\
&\qquad\qquad \text{let } l_3 = \texttt{rev}\ ()[\,]\,[\,]\,[\,]\,[\,]\, l_2 \text{ nil in} \\
&\qquad\qquad \text{let } l_4 = \texttt{ssort}\ ()[\,]\,[\,]\, l_3 \text{ in} \\
&\qquad\qquad \text{unpack } l_4 \text{ as } l_5 \text{ in} \\
&\qquad\qquad \text{let } len = \texttt{length}\ ()[\,]\,[\,]\, l_5 \text{ in} \\
&\qquad\qquad \text{let } l_6 = \texttt{append}\ ()[\,]\,[\,]\,[\,]\,[\,]\, l_5\ l_2 \text{ in} \\
&\qquad\qquad \text{let } l_7 = \texttt{ssort}\ ()[\,]\,[\,]\, l_6 \text{ in} \\
&\qquad\qquad \text{unpack } l_7 \text{ as } l_8 \text{ in} \\
&\qquad\qquad \text{let } l_9 = \texttt{rev}\ ()[\,]\,[\,]\,[\,]\,[\,]\, l_8 \text{ nil in} \\
&\qquad\qquad l_7
\end{aligned}
$$

We can give `ssort_list` the following relational type

$$
\vdash \texttt{ssort\_list} \ominus \texttt{ssort\_list} \lesssim 0 :\ \mathrm{unit}_r \xrightarrow{\mathrm{diff}(0)} \forall n, \alpha{::}\mathbb{N}.\, \mathrm{list}[n]^{\alpha}\, U\, \mathrm{int} \xrightarrow{\mathrm{diff}(0)} \exists \beta{::}\mathbb{N}.\, \mathrm{list}[2{*}n]^{\beta}\, U\, \mathrm{int}. \tag{6.16}
$$

6.1 Experimental results

Table 1 demonstrates experimental evaluation results for all the benchmark programs described above. In all cases, the total typechecking (including existential elimination and SMT solving) takes less than 1s. A "-" indicates a negligible value. Our experiments were performed on a 3.10GHz 2-core Intel Core i5 processor with 8 GB of RAM.

Benchmark	Total time(s)	Type-checking	Existential elim.	Constraint solving	Loc	# of Annotations
`filter`	0.167	0.005	-	0.162	8	1
`append`	0.173	0.005	-	0.167	15	1
`rev`	0.170	0.005	-	0.164	12	1
`map`	0.172	0.005	-	0.167	7	1
`comp`	0.171	0.005	-	0.166	16	3
`sam`	0.174	0.005	-	0.169	13	1
`find`	0.172	0.005	-	0.167	21	3
`2Dcount`	0.168	0.005	-	0.163	17	4
`ssort`	0.181	0.005	-	0.175	26	3
`bsplit`	0.180	0.005	-	0.175	15	1
`flatten`	0.173	0.005	-	0.167	16	2
`appSum`	0.171	0.005	-	0.166	17	3
`merge`	0.207	0.005	-	0.201	26	3
`zip`	0.187	0.005	-	0.181	11	1
`msort`	0.384	0.008	0.003	0.373	29	3
`bfold`	0.216	0.006	0.001	0.208	22	2
`multi_sort`	0.958	0.012	0.015	0.930	81	9
`ssort_list`	0.207	0.006	0.004	0.197	72	9

Table 1: Statistics for BiRelCost example programs. All times are in seconds.