Relational Cost Analysis for Functional-Imperative Programs (Appendix)

Weihao Qu

Boston University

Marco Gaboardi

Boston Univsersity

Deepak Garg

MPI-SWS

May 2020

Contents

List of Figures

List of Theorems and Lemmas

1 Syntax

Unary types
$$A ::= \mathsf{int} \mid \mathsf{int}[I] \mid \mathsf{unit} \mid \overset{\mathsf{exec}(L,U)}{\{P\}\ \exists\vec{\gamma}.A\ \{Q\}} \mid \overset{\mathsf{exec}(L,U)}{\forall i :: S.\ A} \mid \overset{\mathsf{exec}(L,U)}{A \longrightarrow A'} \mid A_1 \times A_2$$
$$\mathsf{Array}_\gamma[I]\ A \mid \mathsf{list}[I]\ A \mid A_1 + A_2 \mid C \& A \mid C \supset A \mid \exists i :: S.\ A$$

Relational types
$$\tau ::= \mathsf{int}_r \mid \mathsf{unit}_r \mid \mathsf{int}[I] \mid \overset{\mathsf{diff}(D)}{\{P\}\ \exists\vec{\gamma}.\tau\ \{Q\}} \mid \overset{\mathsf{diff}(D)}{\forall i :: S.\ \tau} \mid \overset{\mathsf{diff}(D)}{\tau \longrightarrow \tau'} \mid \mathsf{Array}_\gamma[I]\ \tau \mid$$
$$U\ (A_1, A_2) \mid \tau_1 \times \tau_2 \mid \mathsf{list}^\alpha[I]\ \tau \mid \Box\tau \mid \tau_1 + \tau_2 \mid C \& \tau \mid C \supset \tau \mid \exists i :: S.\ \tau$$

Index terms
$$I, L, U, D, \alpha, \beta ::= i \mid b \mid n \mid r \mid I_1 + I_2 \mid I_1 * I_2 \mid I_1 - I_2 \mid max(I_1, I_2) \mid min(I_1, I_2) \mid \frac{I_1}{I_2} \mid \sum_{i=I_1}^{I_n} I$$
$$\mid log_2(I) \mid \lfloor I \rfloor \mid \lceil I \rceil \mid \{I_i\}_{i \in K} \mid \beta \cup \beta \mid \beta \setminus \beta \mid \beta \cap \beta$$

Sort
$$S ::= \mathbb{R} \mid \mathbb{N} \mid \mathbb{B} \mid \mathbb{P} \mid \mathbb{L}$$

Terms
$$t ::= x \mid n \mid r \mid () \mid \lambda x.t \mid \mathsf{fix}\, f(x).t \mid t_1\ t_2 \mid \mathsf{let}\ x = t_1\ \mathsf{in}\ t_2 \mid \langle t_1, t_2 \rangle \mid \mathsf{inl}\ t \mid \mathsf{inr}\ t$$
$$\mid \mathsf{case}\ (t, x.t_1, y.t_2) \mid \Lambda.t \mid t\,[] \mid \mathsf{pack}\ t \mid \mathsf{unpack}\ t_1\ \mathsf{as}\ x\ \mathsf{in}\ t_2 \mid \mathsf{celim}\ t$$
$$\mathsf{nil} \mid \mathsf{cons}(t_1, t_2) \mid \big(\ \mathsf{case}\ t\ \mathsf{of}\ \mathsf{nil} \to t_1 \mid h :: tl \to t_2\big) \mid \pi_1(t) \mid \pi_2(t) \mid$$
$$\mid \mathsf{return}\ t \mid \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \mid \mathsf{alloc}\ t_1\ t_2 \mid \mathsf{read}\ t_1\ t_2 \mid \mathsf{updt}\ t_1\ t_2\ t_3$$

Values
$$v ::= n \mid l \mid r \mid () \mid \lambda x.t \mid \mathsf{fix}\, f(x).t \mid \mathsf{inl}\ v \mid \mathsf{inr}\ v \mid \Lambda.t \mid \langle v_1, v_2 \rangle \mid \mathsf{pack}\ v \mid \mathsf{nil}$$
$$\mid \mathsf{return}\ t \mid \mathsf{alloc}\ t_1\ t_2 \mid \mathsf{updt}\ t_1\ t_2\ t_3 \mid \mathsf{read}\ t_1\ t_2 \mid \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2$$

Unary Type Env.
$$\Omega ::= \emptyset \mid \Omega, x : A$$

Relational Type Env.
$$\Gamma ::= \emptyset \mid \Gamma, x : \tau$$

Sort Env.
$$\Delta ::= \emptyset \mid \Delta, i :: S$$

Loc Env.
$$\Sigma ::= \emptyset \mid \Sigma, \gamma :: \mathbb{L}$$

Constraint
$$C ::= I_1 = I_2 \mid I_1 < I_2 \mid \neg C \mid I_1 \in I_2$$

Constraint Env.
$$\Phi ::= \top \mid C \wedge \Phi$$

Assertions
$$P, Q ::= \mathsf{empty} \mid \gamma \to \beta \mid P \star Q \mid P \cup Q$$

Unary Heap
$$H ::= [] \mid [l \to z] \mid H_1 \uplus H_2$$

array
$$z ::= [v_1, \ldots, v_m]$$

$$n \in \mathbb{N},\ r \in \mathbb{R},\ x \in Var,\ i \in iVar,\ \gamma \in iLoc,\ l \in dLoc$$

Figure 1: Syntax of Values & Expressions

$\boxed{t; H \Downarrow_p^{c,k} v; H_1}$

$$\frac{t \Downarrow^{c_1,k_1} v' \qquad v'; H \Downarrow_f^{c_2,k_2} v; H'}{t; H \Downarrow_p^{c_1+c_2,k_1+k_2} v; H'}\ \text{E-P}$$

$\boxed{t \Downarrow^{c,k} v}$

$$\frac{}{v \Downarrow^{0,0} v}\ \textbf{e-val} \qquad \frac{t \Downarrow^{c,k} v}{\text{inl}\ t \Downarrow^{c,k} \text{inl}\ v}\ \textbf{e-inl} \qquad \frac{t \Downarrow^{c,k} v}{\text{inr}\ \ t \Downarrow^{c,k} \text{inr}\ \ v}\ \textbf{e-inr}$$

$$\frac{t \Downarrow^{c_1,k_1} \text{inl}\ v \qquad t_1[v/x] \Downarrow^{c_2,k_2} v_r}{\text{case}\ (t, x.t_1, y.t_2) \Downarrow^{c_1+c_2+c_{\text{case}},k_1+k_2+1} v_r}\ \textbf{e-casel} \qquad \frac{t \Downarrow^{c_1,k_1} \text{inr}\ \ v \qquad t_2[v/y] \Downarrow^{c_2,k_2} v_r}{\text{case}\ (t, x.t_1, y.t_2) \Downarrow^{c_1+c_2+c_{\text{case}},k_1+k_2+1} v_r}\ \textbf{e-caser}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} \lambda x.t' \qquad t_2 \Downarrow^{c_2,k_2} v \qquad t'[v/x] \Downarrow^{c_3,k_3} v_1}{t_1\ t_2 \Downarrow^{c_1+c_2+c_3+c_{\text{app}},k_1+k_2+k_3+1} v_1}\ \textbf{e-app} \qquad \frac{t_1 \Downarrow^{c_1,k_1} v \qquad t_2/[v/x] \Downarrow^{c_2,k_2} v_1}{\text{let}\ x = t_1\ \text{in}\ t_2 \Downarrow^{c_1+c_2+c_{\text{lt}},k_1+k_2+1} v_1}\ \textbf{e-lt}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} \text{fix}\, f\, x.t' \qquad t_2 \Downarrow^{c_2,k_2} v \qquad t'[\text{fix}\, f\, x.t'/f][v/x] \Downarrow^{c_3,k_3} v_1}{t_1\ t_2 \Downarrow^{c_1+c_2+c_3+c_{\text{app}},k_1+k_2+k_3+1} v_1}\ \textbf{e-fix}$$

$$\frac{t \Downarrow^{c,k} \Lambda t_b \qquad t_b \Downarrow^{(c',k')} v_r}{t\ [\,] \Downarrow^{c+c',k+k'} v_r}\ \textbf{e-iapp} \qquad \frac{t \Downarrow^{c,k} v}{\text{pack}\ t \Downarrow^{c,k} \text{pack}\ v}\ \textbf{e-pack}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} \text{pack}\ v \qquad t_2[v/x] \Downarrow^{(c_2,k_2)} v_r}{\text{unpack}\ t_1\ \text{as}\ x\ \text{in}\ t_2 \Downarrow^{c_1+c_2,k_1+k_2} v_r}\ \textbf{e-unpack} \qquad \frac{t_1 \Downarrow^{c_1,k_1} \Lambda t_b \qquad t_2 \Downarrow^{(c_2,k_2)} v_2}{\text{cons}(t_1, t_2) \Downarrow^{c_1+c_2,k_1+k_2} \text{cons}(v_1, v_2)}\ \textbf{e-cons}$$

$$\frac{t \Downarrow^{c,k} \text{nil} \qquad t_1 \Downarrow^{(c_r,k_r)} v_r}{\text{case}\ t\ \text{of nil} \to t_1 \mid h :: tl \to t_2 \Downarrow^{c+c_r+c_{caseL},k+k_r+1} v_r}\ \text{E-CASEL-NIL}\ \textbf{e-caseL-nil}$$

$$\frac{t \Downarrow^{c,k} \text{cons}(v_1, v_2) \qquad t_2[v_1/h, v_2/tl] \Downarrow^{(c_r,k_r)} v_r}{\text{case}\ t\ \text{of nil} \to t_1 \mid h :: tl \to t_2 \Downarrow^{c+c_r+c_{caseL},k+k_r+1} v_r}\ \textbf{e-caseL-cons}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} v_1 \qquad t_2 \Downarrow^{(c_2,k_2)} v_2}{\langle t_1, t_2\rangle \Downarrow^{c_1+c_2,k_1+k_2} \langle v_1, v_2\rangle}\ \textbf{e-prod} \qquad \frac{t \Downarrow^{c,k} \langle v_1, v_2\rangle}{\pi_1(t) \Downarrow^{c+c_{proj},k+1} v_1}\ \textbf{e-proj1}$$

$$\frac{t \Downarrow^{c,k} \langle v_1, v_2\rangle}{\pi_2(t) \Downarrow^{c+c_{proj},k+1} v_2}\ \textbf{e-proj2} \qquad \frac{t_1 \Downarrow^{c_1,k_1} v_1 \qquad t_2[v_1/x] \Downarrow^{c_r,k_r} v_r}{\text{clet}\ t_1\ \text{as}\ x\ \text{in} t_2 \Downarrow^{c_1+c_r,k+k_r} v_r}\ \textbf{e-clet}$$

Figure 2: Operational Semantics: (1) standard, (2) forcing evaluation.

$\boxed{t; H \Downarrow_f^{c,k} v; H_1}$

$$\frac{t \Downarrow^{c,k} v}{\mathsf{return}\ t; H \Downarrow_f^{c+c_{\mathrm{ret}},k+1} v; H}\ \textbf{f-ret}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} v \qquad v; H \Downarrow_f^{c_2,k_2} v_1; H_1 \qquad t_2[v_1/x] \Downarrow^{c_3,k_3} v_2 \qquad v_2; H_1 \Downarrow_f^{c_4,k_4} v_3; H_2}{\mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2; H \Downarrow_f^{c_1+c_2+c_3+c_4+c_{\mathrm{let}},k_1+k_2+k_3+k_4+1} v_3; H_2}\ \textbf{f-bind}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} l \qquad t_2 \Downarrow^{c_2,k_2} n \qquad t_3 \Downarrow^{c_3,k_3} v}{\mathsf{updt}\ t_1\ t_2\ t_3; H \Downarrow_f^{c_1+c_2+c_3+c_{\mathrm{update}},k_1+k_2+k_3+1} (); H(l)[n] \leftarrow v}\ \textbf{f-updt}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} l \qquad t_2 \Downarrow^{c_2,k_2} n_2 \qquad H(l)[n] = v}{\mathsf{read}\ t_1\ t_2; H \Downarrow_f^{c_1+c_2+c_{\mathrm{read}},k_1+k_2+1} v; H}\ \textbf{f-read}$$

$$\frac{t_1 \Downarrow^{c_1,k_1} n \qquad t_2 \Downarrow^{c_2,k_2} v \qquad z = [\overbrace{v,\ldots,v}^{n}] \qquad l\ \text{fresh}}{\mathsf{alloc}\ t_1\ t_2; H \Downarrow_f^{c_1+c_2+c_{\mathrm{alloc}},k_1+k_2+1} l; H, l \to z}\ \textbf{f-alloc}$$

Figure 3: Operational Semantics: forcing evaluation.

$$\begin{array}{rcl}
|.|_{i\in\{1,2\}} & : & \mathsf{Binarytype} \to \mathsf{Unarytype} \\
|\mathrm{int}_r|_i & = & \mathrm{int} \\
|\mathrm{unit}_r|_i & = & \mathrm{unit} \\
|\ \{P\}\ \exists^{\mathrm{diff}(c)}\vec{\gamma} : \tau\ \{Q\}\ |_i & = & \{|P|_i\}\ \exists^{\mathrm{exec}(0,\infty)}\vec{\gamma} : |\tau|_i\ \{|Q|_i\} \\
|\ \tau \overset{\mathrm{diff}((c))}{\longrightarrow} \tau'\ |_i & = & |\tau|_i \overset{\mathrm{exec}(0,\infty)}{\longrightarrow} |\tau'|_i \\
|\ \mathsf{Array}_\gamma[I]\ \tau\ |_i & = & \mathsf{Array}_\gamma[I]\ |\tau|_i \\
|U\ A|_i & = & A_i \\
|\Box\tau|_i & = & |\tau|_i \\
|\ \mathsf{list}^\alpha[I]\ \tau\ |_i & = & \mathsf{list}[I]\ |\tau|_i \\
|\tau_1 + \tau_2|_i & = & |\tau_1|_i + |\tau_2|_i \\
|C\&\tau|_i & = & C\&|\tau|_i \\
|C \supset \tau|_i & = & C \supset |\tau|_i \\
|\emptyset|_i & = & \emptyset \\
|\Gamma, x:\tau|_i & = & |\Gamma|_i, x:|\tau|_i \\
|\Gamma, x:\tau, x:U(A_1,A_2)|_i & = & |\Gamma|_i, x:A_i \\
|\mathsf{empty}|_i & = & \mathsf{empty} \\
|\gamma_n \to \beta_n|_i & = & \gamma_n \to \mathbb{N}
\end{array}$$

Figure 4: Binary to unary type erasure

$\boxed{\vdash \Delta}$

$$\frac{}{\vdash \emptyset}\ \textsc{senv-empty} \qquad \frac{\vdash \Delta \qquad i \in \mathsf{iVar} \qquad S \in \mathsf{Sort}}{\vdash \Delta, i :: S}\ \textsc{senv-iv}$$

Figure 5: sort environment well formedness

$\boxed{\vdash \Sigma}$

$$\frac{}{\vdash \emptyset}\ \textsc{lenv-empty} \qquad \frac{\vdash \Sigma \qquad \gamma \in \mathsf{iLoc}}{\vdash \Sigma, \gamma :: \mathbb{L}}\ \textsc{lenv-lv}$$

Figure 6: location environment well-formedness

$\boxed{\Sigma; \Delta \vdash P \quad wf}$

$$\frac{}{\Sigma; \Delta \vdash \text{empty} \quad wf}\ \textsc{wf-as-empty} \qquad \frac{\Sigma; \Delta \vdash \gamma :: \mathbb{L} \qquad \Sigma; \Delta \vdash \beta :: \mathbb{P}}{\Sigma; \Delta \vdash \gamma \to \beta \quad wf}\ \textsc{wf-as-referTo}$$

$$\frac{\Sigma; \Delta \vdash P \quad wf \qquad \Sigma; \Delta \vdash Q \quad wf}{\Sigma; \Delta \vdash P * Q \quad wf}\ \textsc{wf-as-star}$$

Figure 7: Assertion well-formedness

$\boxed{\Delta \vdash C \quad wf}$

$$\frac{\Delta \vdash I_1 :: S \qquad \Delta \vdash I_2 :: S \qquad S \in \{\mathbb{N}, \mathbb{R}\}}{\Delta \vdash I_1 = I_2 \quad wf}\ \textsc{wf-cs-eq} \qquad \frac{\Delta \vdash I_1 :: S \qquad \Delta \vdash I_2 :: S \qquad S \in \{\mathbb{N}, \mathbb{R}\}}{\Delta \vdash I_1 < I_2 \quad wf}\ \textsc{wf-cs-lt}$$

$$\frac{\Delta \vdash C \quad wf}{\Delta \vdash \neg C \quad wf}\ \textsc{wf-cs-neg} \qquad \frac{\Delta \vdash I_1 :: S \qquad \Delta \vdash I_2 :: S \qquad S \in \{\mathbb{N}, \mathbb{R}\}}{\Delta \vdash I_1 \in I_2 \quad wf}\ \textsc{wf-cs-in}$$

Figure 8: Constraint well-formedness

$\boxed{\Sigma;\Delta \vdash I :: S}$

$$\frac{\Delta(i) = S}{\Sigma;\Delta \vdash i :: S}\ \text{I-VAR} \qquad \frac{\Sigma(\gamma) = \mathbb{L}}{\Sigma;\Delta \vdash \gamma :: \mathbb{L}}\ \text{I-LOC} \qquad \frac{}{\Sigma;\Delta \vdash b :: \mathbb{B}}\ \text{I-BOOL}$$

$$\frac{}{\Sigma;\Delta \vdash n :: \mathbb{N}}\ \text{I-NAT} \qquad\qquad \frac{}{\Sigma;\Delta \vdash r :: \mathbb{R}}\ \text{I-REAL}$$

$$\frac{\Sigma;\Delta \vdash I_1 :: \mathbb{N} \qquad \Sigma;\Delta \vdash I_2 :: \mathbb{N} \qquad \diamond \in \{min, max, +, -, *, \}}{\Sigma;\Delta \vdash (I_1 \diamond I_2) :: \mathbb{N}}\ \text{I-BIN-N}$$

$$\frac{\Sigma;\Delta \vdash I_1 :: \mathbb{R} \qquad \Sigma;\Delta \vdash I_2 :: \mathbb{R} \qquad \diamond \in \{min, max, +, -, *, \}}{\Sigma;\Delta \vdash (I_1 \diamond I_2) :: \mathbb{R}}\ \text{I-BIN-R}$$

$$\frac{\Sigma;\Delta \vdash I :: \mathbb{R} \qquad \circ \in \{\lfloor\ \rfloor, \lceil\ \rceil\}}{\Sigma;\Delta \vdash (\circ I) :: \mathbb{N}}\ \text{I-UNARY-R}$$

$$\frac{\Sigma;\Delta \vdash I :: \mathbb{R}}{\Sigma;\Delta \vdash log_2(I) :: \mathbb{R}}\ \text{I-LOG}$$

$$\frac{\Sigma;\Delta \vdash I_i :: \mathbf{N} \quad \forall i \in K \subseteq \mathbb{N}}{\Sigma;\Delta \vdash \{I_i\}_{i \in K} :: \mathbf{P}}\ \text{I-NATSET} \qquad \frac{\Sigma;\Delta \vdash \beta_1 :: \mathbf{P} \qquad \Sigma;\Delta \vdash \beta_2 :: \mathbf{P} \quad \diamond \in \{\cup, \cap, /\}}{\Sigma;\Delta \vdash \beta_1 \diamond \beta_2 :: \mathbf{P}}\ \text{I-SET-OP}$$

$$\frac{\Sigma;\Delta \vdash I_1 :: \mathbb{N} \qquad \Sigma;\Delta \vdash I_n :: \mathbb{N} \qquad \Sigma;\Delta, i :: \mathbf{N} \vdash I :: \mathbb{S} \qquad S \in \{\mathbf{N}, \mathbb{R}\}}{\Sigma;\Delta \vdash \sum_{i=I_1}^{I_n} I :: \mathbb{S}}\ \text{I-SUM}$$

Figure 9: Sorting rules

$\boxed{\Sigma;\Delta;\Phi \vdash A \quad wf}$

$$\frac{}{\Sigma;\Delta;\Phi \vdash \emptyset \quad wf}\ \text{WF-A-OMEGA-EMPTYSET} \qquad \frac{\Sigma;\Delta;\Phi \vdash \Omega \quad wf \qquad \Sigma;\Delta;\Phi \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash \Omega, x : A \quad wf}\ \text{WF-A-OMEGA}$$

$$\frac{}{\Sigma;\Delta;\Phi \vdash unit \quad wf}\ \text{WF-A-UNIT} \qquad \frac{}{\Sigma;\Delta;\Phi \vdash Int \quad wf}\ \text{WF-A-INT} \qquad \frac{\Sigma;\Delta \vdash I :: \mathbb{N}}{\Sigma;\Delta;\Phi \vdash \text{int}[I] \quad wf}\ \text{WF-A-INT-I}$$

$$\frac{\Sigma;\Delta;\Phi \vdash A_1 \quad wf \qquad \Sigma;\Delta;\Phi \vdash A_2 \quad wf}{\Sigma;\Delta;\Phi \vdash A_1 + A_2 \quad wf}\ \text{WF-A-SUM}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi \vdash A \quad wf \qquad \Sigma;\Delta \vdash \gamma :: \mathbb{L} \\ \Sigma;\Delta \vdash L :: \mathbb{R} \qquad \Sigma;\Delta \vdash U :: \mathbb{R} \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Sigma, \vec{\gamma} : \vec{\mathbb{L}};\Delta \vdash Q \quad wf\end{array}}{\begin{array}{c}\text{exec}(L,U) \\ \Sigma;\Delta;\Phi \vdash \{P\}\ \exists \vec{\gamma} : A\ \{Q\} \quad wf\end{array}}\ \text{WF-A-MONAD}$$

$$\frac{\Sigma;\Delta;\Phi \vdash A \quad wf \qquad \Sigma;\Delta;\Phi \vdash A' \quad wf \qquad \Sigma;\Delta \vdash L :: \mathbb{R} \qquad \Sigma;\Delta \vdash U :: \mathbb{R}}{\begin{array}{c}\text{exec}(L,U) \\ \Sigma;\Delta;\Phi \vdash A \longrightarrow A' \quad wf\end{array}}\ \text{WF-A-ARROW}$$

$$\frac{\Sigma;\Delta \vdash \gamma :: \mathbb{L} \qquad \Sigma;\Delta \vdash I :: \mathbb{N} \qquad \Sigma;\Delta;\Phi \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash \mathsf{Array}_\gamma[I]\ A \quad wf}\ \text{WF-A-ARRAY}$$

$$\frac{\Sigma; i :: S,\Delta;\Phi \vdash A \quad wf \qquad \Sigma; i :: S,\Delta \vdash L :: \mathbb{R} \qquad \Sigma; i :: S,\Delta \vdash U :: \mathbb{R}}{\begin{array}{c}\text{exec}(L,U) \\ \Sigma;\Delta;\Phi \vdash \forall i :: S.\ A \quad wf\end{array}}\ \text{WF-A-FORALL}$$

$$\frac{\Sigma; i :: S,\Delta;\Phi \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash \exists i :: S.\ A \quad wf}\ \text{WF-A-EXIST}$$

$$\frac{\Delta \vdash I :: \mathbb{N} \qquad \Sigma;\Delta;\Phi \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash \mathsf{list}[I]\ A \quad wf}\ \text{WF-A-LIST}$$

$$\frac{\Delta \vdash C \quad wf \qquad \Sigma;\Delta;\Phi \wedge C \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash C \& A \quad wf}\ \text{WF-A-C-AND}$$

$$\frac{\Delta \vdash C \quad wf \qquad \Sigma;\Delta;\Phi \wedge C \vdash A \quad wf}{\Sigma;\Delta;\Phi \vdash C \supset A \quad wf}\ \text{WF-A-C-IMPLY}$$

Figure 10: Well-formedness of unary types

$\boxed{\Sigma;\Delta;\Phi \vdash \tau \quad wf}$

$$\frac{}{\Sigma;\Delta;\Phi \vdash \emptyset \quad wf}\ \text{WF-R-GAMMA-EMPTYSET} \qquad \frac{\Sigma;\Delta;\Phi \vdash \Gamma \quad wf \qquad \Sigma;\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash \Gamma, x:\tau \quad wf}\ \text{WF-R-GAMMA}$$

$$\frac{}{\Sigma;\Delta;\Phi \vdash \text{unit}_r \quad wf}\ \text{WF-R-UNIT} \qquad \frac{}{\Sigma;\Delta;\Phi \vdash \text{int}_r \quad wf}\ \text{WF-R-INT}$$

$$\frac{\Sigma;\Delta;\Phi \vdash \tau_1 \quad wf \qquad \Sigma;\Delta;\Phi \vdash \tau_2 \quad wf}{\Sigma;\Delta;\Phi \vdash \tau_1 + \tau_2 \quad wf}\ \text{WF-R-SUM}$$

$$\frac{\Sigma;\Delta;\Phi \vdash \tau_1 \quad wf \qquad \Sigma;\Delta;\Phi \vdash \tau_2 \quad wf}{\Sigma;\Delta;\Phi \vdash \tau_1 \times \tau_2 \quad wf}\ \text{WF-R-PROD}$$

$$\frac{\Sigma;\Delta \vdash P \quad wf \qquad \Sigma;\Delta;\Phi \vdash \tau \quad wf \qquad \Sigma;\Delta,\vec{\gamma}:\vec{\mathbb{L}} \vdash Q \quad wf \qquad \Sigma;\Delta \vdash D :: \mathbb{R}}{\overset{\text{diff}(D)}{\Sigma;\Delta;\Phi \vdash \{P\}\ \exists\vec{\gamma}:\tau\ \{Q\} \quad wf}}\ \text{WF-R-MONAD}$$

$$\frac{\Sigma;\Delta;\Phi \vdash \tau \quad wf \qquad \Sigma;\Delta;\Phi \vdash \tau' \quad wf \qquad \Sigma;\Delta \vdash D :: \mathbb{R}}{\overset{\text{diff}(D)}{\Sigma;\Delta;\Phi \vdash \tau \longrightarrow \tau' \quad wf}}\ \text{WF-R-ARROW}$$

$$\frac{\Sigma;i :: S,\Delta;\Phi \vdash \tau \quad wf \qquad \Sigma;i :: S,\Delta \vdash D :: \mathbb{R}}{\overset{\text{diff}(D)}{\Sigma;\Delta;\Phi \vdash \forall i{::}S.\ \tau \quad wf}}\ \text{WF-R-FORALL} \qquad \frac{\Sigma;i :: S,\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash \exists i :: S.\ \tau \quad wf}\ \text{WF-R-EXIST}$$

$$\frac{\Sigma;\Delta \vdash \gamma :: \mathbb{L} \qquad \Sigma;\Delta \vdash I :: \mathbb{N} \qquad \Sigma;\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash \mathsf{Array}_\gamma[I]\ \tau \quad wf}\ \text{WF-R-ARRAY}$$

$$\frac{\Sigma;\Delta;\Phi \vdash A_1 \quad wf \qquad \Sigma;\Delta;\Phi \vdash A_2 \quad wf}{\Sigma;\Delta;\Phi \vdash U\ (A_1, A_2) \quad wf}\ \text{WF-R-UA}$$

$$\frac{\Sigma;\Delta \vdash \alpha :: \mathbb{N} \qquad \Sigma;\Delta \vdash I :: \mathbb{N} \qquad \Sigma;\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash \mathsf{list}^\alpha[I]\ \tau \quad wf}\ \text{WF-R-LIST}$$

$$\frac{\Delta \vdash C \quad wf \qquad \Sigma;\Delta;\Phi \wedge C \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash C\&\tau \quad wf}\ \text{WF-R-C-AND}$$

$$\frac{\Delta \vdash C \quad wf \qquad \Sigma;\Delta;\Phi \wedge C \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash C \supset \tau \quad wf}\ \text{WF-R-C-IMPLY} \qquad \frac{\Sigma;\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi \vdash \Box\tau \quad wf}\ \text{WF-R-SQUARE}$$

Figure 11: Well-formedness of relational types

$\boxed{\Omega \vdash H \quad wf}$

$$\frac{}{\Omega \vdash empty \quad wf}\ \text{WF-H-EMPTY} \qquad \frac{\Omega \vdash H \quad wf \qquad x \notin dom(H)}{\Omega \vdash (H, x \to k) \quad wf}\ \text{WF-H-UPDATE}$$

Figure 12: Heap well-formedness

$\boxed{\Sigma;\Delta;\Phi;\Omega \vdash^{U}_{L} t : A}$

$$\frac{}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} n : \text{int}}\ \textbf{u-i} \qquad \frac{}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} n : \text{int}[n]}\ \textbf{u-int} \qquad \frac{\Sigma;\Delta;\Phi;\Omega, x : A \vdash^{U}_{L} t : B}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \lambda x.t : A \overset{\text{exec}(L,U)}{\longrightarrow} B}\ \textbf{u-abs}$$

$$\frac{\Omega(x) = A}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} x : A}\ \textbf{u-var} \qquad \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : A \overset{\text{exec}(L,U)}{\longrightarrow} B \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : A}{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1+U_2+U+U_{app}}_{L_1+L_2+L+L_{app}} t_1\ t_2 : B}\ \textbf{u-app}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : A \qquad \Sigma;\Delta;\Phi;\Omega, x : A \vdash^{U_2}_{L_2} t_2 : B}{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1+U_2+U_{lt}}_{L_1+L_2+L_{lt}} \text{let } x = t_1 \text{ in } t_2 : B}\ \textbf{u-lt} \qquad \frac{}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} () : \text{unit}}\ \textbf{u-unit}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : A_1 \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : A_2}{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1+U_2}_{L_1+L_2} \langle t_1, t_2 \rangle : A_1 \times A_2}\ \textbf{u-prod} \qquad \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U}_{L} t : A_1 \times A_2}{\Sigma;\Delta;\Phi;\Omega \vdash^{U+U_{proj}}_{L+L_{proj}} \pi_1(t) : A_1}\ \textbf{u-proj1}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U}_{L} t : A_1 \times A_2}{\Sigma;\Delta;\Phi;\Omega \vdash^{U+U_{proj}}_{L+L_{proj}} \pi_2(t) : A_2}\ \textbf{u-proj2} \qquad \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U}_{L} t : A \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{return } t : \{P\}\ \exists \gamma.A\ \{P\}}\ \textbf{u-ret}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \{P\}\ \exists \vec{\gamma}_1 : A\ \{P'\} \overset{\text{exec}(L,U)}{} \qquad \Sigma;\Delta,\vec{\gamma}_1;\Phi;\Omega, x : A \vdash^{U_2}_{L_2} t_2 : \{P'\}\ \exists \vec{\gamma}_2 : B\ \{Q\} \overset{\text{exec}(L',U')}{}}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{let } \{x\} = t_1 \text{ in } t_2 : \overset{\text{exec}(L+L'+L_1+L_2+L_l, U+U'+U_1+U_2+U_l)}{\{P\}\ \exists \vec{\gamma}_1, \vec{\gamma}_2 : B\ \{Q\}}}\ \textbf{u-bind}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \text{int}[I] \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : A \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{alloc } t_1\ t_2 : \{P\}\ \exists \gamma : \text{Array}_{\gamma}[I]\ A\ \{P \star \gamma \rightarrow \mathbb{N}\} \overset{\text{exec}(L_1+L_2+L_a, U_1+U_2+U_a)}{}}\ \textbf{u-alloc}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \text{Array}_{\gamma}[I]\ A \\ \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \text{int}[I'] \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_3}_{L_3} t_3 : A \qquad \Delta;\Phi \vDash I' \leq I \qquad \Delta;\Phi \vDash I' \in \beta \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{updt } t_1\ t_2\ t_3 : \{P \star \gamma \rightarrow \beta\}\ \exists \_ : \text{unit}\ \{P \star \gamma \rightarrow \beta\} \overset{\text{exec}(L_1+L_2+L_3+L_u, U_1+U_2+U_3+U_u)}{}}\ \textbf{u-updt}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \text{Array}_{\gamma}[I]\ A \\ \gamma \in dom(P) \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \text{int}[I'] \qquad \Delta;\Phi \vDash I' \leq I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{read } t_1\ t_2 : \overset{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)}{\{P\}\ \exists \_ : A\ \{P\}}}\ \textbf{u-read}$$

Figure 13: Unary typing judgment, part (1)

$$\frac{\Sigma;\Delta;\Phi;x:A,f:A\overset{\mathsf{exec}(L,U)}{\longrightarrow}B,\Omega\vdash^{U}_{L}t:B}{\Sigma;\Delta;\Phi;\Omega\vdash^{0}_{0}\mathsf{fix}\,f(x).t:A\overset{\mathsf{exec}(L,U)}{\longrightarrow}B}\ \textbf{u-fix}\qquad\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A_1\qquad\Delta;\Sigma;\Phi\vdash A_2\quad wf}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}\mathsf{inl}\;t:A_1+A_2}\ \textbf{u-inl}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A_2\qquad\Delta;\Sigma;\Phi\vdash A_1\quad wf}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}\mathsf{inr}\;t:A_1+A_2}\ \textbf{u-inr}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1}_{L_1}t:A_1+A_2\qquad\Sigma;\Delta;\Phi;x:\ A_1,\Omega\vdash^{U_2}_{L_2}t_1:A\qquad\Sigma;\Delta;\Phi;y:\ A_2,\Omega\vdash^{U_2}_{L_2}t_2:A}{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1+U_2+U_c}_{L_1+L_2+L_c}\mathsf{case}\;(t,x.t_1,y.t_2):A}\ \textbf{u-case}$$

$$\frac{}{\Sigma;\Delta;\Phi;\Omega\vdash^{0}_{0}\mathsf{nil}:\mathsf{list}[0]\;A}\ \textbf{u-nil}\qquad\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1}_{L_1}t_1:A\qquad\Sigma;\Delta;\Phi;\Omega\vdash^{U_2}_{L_2}t_2:\mathsf{list}[I]\;A}{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1+U_2}_{L_1+L_2}\mathsf{cons}(t_1,t_2):\mathsf{list}[I+1]\;A}\ \textbf{u-cons}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:\mathsf{list}[n]\;A\\ \Sigma;\Delta;\Phi\wedge n=0;\Omega\vdash^{U'}_{L'}t_1:A'\qquad\Sigma;i,\Delta;\Phi\wedge n=i+1;h:A,tl:\mathsf{list}[i]\;A,\Omega\vdash^{U'}_{L'}t_2:A'\end{array}}{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1+U_2}_{L_1+L_2}\mathsf{case}\;t\;\mathsf{of}\;\mathsf{nil}\to t_1\mid h::tl\to t_2:A'}\ \textbf{u-caseL}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A\qquad\Sigma;\Delta\vDash U\le U'\qquad\Sigma;\Delta\vDash L'\le L}{\Sigma;\Delta;\Phi;\Omega\vdash^{U'}_{L'}t:A}\ \textbf{u-cost-trans}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A\qquad\Sigma;\Delta;\Phi\vDash A\sqsubseteq A'\qquad\Sigma;\Delta\vDash U\le U'\qquad\Sigma;\Delta\vDash L'\le L}{\Sigma;\Delta;\Phi;\Omega\vdash^{U'}_{L'}t:A'}\ \textbf{u-sub}$$

$$\frac{\Sigma;i::S,\Delta;\Phi;\Omega\vdash^{U}_{L}t:A\qquad i\notin FIV(\Phi;\Omega)}{\Sigma;\Delta;\Phi;\Omega\vdash^{0}_{0}\Lambda.t:\forall i::S.\;A\overset{\mathsf{exec}(L,U)}{}}\ \textbf{u-iLam}\qquad\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1}_{L_1}t:\overset{\mathsf{exec}(L_2,U_2)}{\forall i::S.\;A}\qquad\Sigma;\Delta\vdash I::S}{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1+U_2}_{L_1+L_2}t\,[]:A\{I/i\}}\ \textbf{u-iapp}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A\{I/i\}\qquad\Sigma;\Delta\vdash I::S}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}\mathsf{pack}\,t:\exists i::S.\;A}\ \textbf{u-pack}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1}_{L_1}t_1:\exists i::S.\;A\qquad\Sigma;i::S,\Delta;\Phi;\Omega\vdash^{U_2}_{L_2}t_2:A_2\qquad i\notin FV(\Phi;\Omega;A_2,L_2,U_2)}{\Sigma;\Delta;\Phi;\Omega\vdash^{U_1+U_2}_{L_1+L_2}\mathsf{unpack}\,t_1\,\mathsf{as}\,x\,\mathsf{in}\;t_2:A_2}\ \textbf{u-unpack}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:C\supset A\qquad\Sigma;\Delta;\Phi\vDash C}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}\mathsf{celim}\;t:A}\ \textbf{u-celim}\qquad\frac{\Sigma;\Delta;\Phi\wedge C;\Omega\vdash^{U}_{L}t:A}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:C\supset A}\ \textbf{u-cimpl}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:C\&A_1\qquad\Sigma;\Delta;\Phi\wedge C;x:A_1,\Omega\vdash^{U'}_{L'}t:A_2}{\Sigma;\Delta;\Phi;\Gamma\vdash^{U+U'}_{L+L'}\mathsf{clet}\;t\;\mathsf{as}\;x\;\mathsf{in}\;t':A_2}\ \textbf{u-clet}$$

$$\frac{\Sigma;\Delta;\Phi\wedge C;\Omega\vdash^{U}_{L}t:A\qquad\Sigma;\Delta;\Phi\wedge\neg C;\Omega\vdash^{U}_{L}t:A}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:A}\ \textbf{u-split}$$

$$\frac{\Sigma;\Delta;\Phi\vDash\bot\qquad\vdash\Sigma;\Delta;\Phi\vDash\Omega\quad wf}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:C\&A}\ \textbf{u-contra}\qquad\frac{\Sigma;\Delta;\Phi\wedge C;\Omega\vdash^{U}_{L}t:A\qquad\Sigma;\Delta;\Phi\vDash C}{\Sigma;\Delta;\Phi;\Omega\vdash^{U}_{L}t:C\&A}\ \textbf{u-andI}$$

Figure 14: Unary typing judgment, part (2)

$\boxed{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim c : \tau}$

$$\frac{}{\Sigma;\Delta;\Phi;\Gamma \vdash n \ominus n \lesssim 0 : \mathrm{int}_r}\ \textbf{r-i} \qquad \frac{}{\Sigma;\Delta;\Phi;\Gamma \vdash n \ominus n \lesssim 0 : \mathrm{int}_r[n]}\ \textbf{r-int} \qquad \frac{}{\Sigma;\Delta;\Phi;\Gamma \vdash () \ominus () \lesssim 0 : \mathrm{unit}_r}\ \textbf{r-unit}$$

$$\frac{\Gamma(x) = \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash x \ominus x \lesssim 0 : \tau}\ \textbf{r-var} \qquad \frac{\Sigma;\Delta;\Phi;\Gamma, x:\tau \vdash t_1 \ominus t_2 \lesssim D : \sigma}{\Sigma;\Delta;\Phi;\Gamma \vdash \lambda x.t_1 \ominus \lambda x.t_2 \lesssim 0 : \tau \xrightarrow{\mathrm{diff}(D)} \sigma}\ \textbf{r-abs}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \tau \xrightarrow{\mathrm{diff}(D)} \sigma \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1\ t_2 \ominus t_1'\ t_2' \lesssim D + D_1 + D_2 : \sigma}\ \textbf{r-app}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \tau_1 \qquad \Sigma;\Delta;\Phi;\Gamma, x:\tau_1 \vdash t_2 \ominus t_2' \lesssim D_2 : \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ x = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ x = t_1'\ \mathsf{in}\ t_2' \lesssim D_1 + D_2 : \tau_2}\ \textbf{r-lt}$$

$$\frac{\Sigma;\Delta;\Phi;x:\tau, f\ :\tau \xrightarrow{\mathrm{diff}(D)} \sigma, \Gamma \vdash t_1 \ominus t_2 \lesssim D : \sigma}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{fix}\, f(x).t_1 \ominus \mathsf{fix}\, f(x).t_2 \lesssim 0 : \tau \xrightarrow{\mathrm{diff}(D)} \sigma}\ \textbf{r-fix}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t' \lesssim D : \tau_1 \qquad \Sigma;\Delta;\Phi \vdash \tau_2 \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{inl}\ t \ominus \mathsf{inl}\ t' \lesssim D : \tau_1 + \tau_2}\ \textbf{r-inl}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t' \lesssim D : \tau_2 \qquad \Sigma;\Delta;\Phi \vdash \tau_1 \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{inr}\ t \ominus \mathsf{inr}\ t' \lesssim D : \tau_1 + \tau_2}\ \textbf{r-inr}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t' \lesssim D_1 : \tau_1 + \tau_2 \\ \Sigma;\Delta;\Phi;\Gamma, x:\tau_1 \vdash t_1 \ominus t_1' \lesssim D_2 : \tau \qquad \Sigma;\Delta;\Phi;\Gamma, y:\tau_2 \vdash t_2 \ominus t_2' \lesssim D_2 : \tau\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash\ \mathsf{case}\ (t, x.t_1, y.t_2) \ominus\ \mathsf{case}\ (t', x.t_1', y.t_2') \lesssim D_1 + D_2 : \tau}\ \textbf{r-case}$$

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash_{L_1}^{U_1} t_1 : A_1 \qquad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash_{L_2}^{U_2} t_2 : A_2}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim U_1 - L_2 : U(A_1, A_2)}\ \textbf{r-switch}$$

$$\frac{\Sigma;\Delta;\Phi;\Omega \vdash t \ominus t \lesssim D : \tau \qquad \Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma;\Delta;\Phi \vDash D \leq D'}{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t \lesssim D' : \tau'}\ \textbf{r-sub}$$

$$\frac{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \qquad \Sigma;\Delta;\Phi \wedge \neg C;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau}\ \textbf{r-split}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \qquad \forall x \in dom(\Gamma).\Sigma;\Delta;\Phi \models \Gamma(x) \sqsubseteq \Box\Gamma(x)}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim 0 : \Box\tau}\ \textbf{r-nc}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau\{I/i\} \qquad \Sigma;\Delta \vdash I :: S}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{pack}\ t_1 \ominus \mathsf{pack}\ t_2 \lesssim D : \exists i :: S.\tau}\ \textbf{r-pack}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \exists i :: S.\tau_1 \qquad \Sigma; i :: S, \Delta;\Phi; x:\tau_1, \Gamma \vdash t_2 \ominus t_2' \lesssim D_1 + D_2 : \tau_2 \qquad i \notin FV(\Phi;\Gamma;\tau_2)}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{unpack}\ t_1\ \mathsf{as}\ x\ \mathsf{in}\ t_2 \ominus \mathsf{unpack}\ t_1'\ \mathsf{as}\ x\ \mathsf{in}\ t_2' \lesssim D_1 + D_2 : \tau_2}\ \textbf{r-pack}$$

Figure 15: Relational typing judgment, part (1)

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{return}\ t_1 \ominus \mathsf{return}\ t_2 \overset{\mathrm{diff}(D)}{\lesssim} 0 : \{P\}\ \exists \_.\tau\ \{P\}}\ \textbf{r-ret}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \overset{\mathrm{diff}(D)}{\lesssim} D_1 : \{P\}\ \exists \vec{\gamma}_1.\tau\ \{P'\} \qquad \Sigma;\Delta,\vec{\gamma_1} : \vec{\mathbb{L}};\Phi;\Gamma, x : \tau \vdash t_2 \ominus t_2' \overset{\mathrm{diff}(D')}{\lesssim} D_2 : \{P'\}\ \exists \vec{\gamma}_2.\sigma\ \{Q\}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \overset{\mathrm{diff}(D+D'+D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists \vec{\gamma}_1 \vec{\gamma}_2 : \sigma\ \{Q\}}\ \textbf{r-bind}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau \qquad \gamma\ \mathsf{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists \gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \mathbb{N}\}}\ \textbf{r-alloc}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \square\tau \qquad \gamma\ \mathsf{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists \gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\}}\ \textbf{r-allocb}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \gamma \in dom(P) \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta;\Phi \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists \_.\tau\ \{P\}}\ \textbf{r-read}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta;\Phi \models I' \le I \qquad \Sigma;\Delta;\Phi \models I' \notin \beta \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P \star \gamma \to \beta\}\ \exists \_.\square\tau\ \{P \star \gamma \to \beta\}}\ \textbf{r-readb}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 : \tau \qquad \Delta;\Phi \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \overset{\mathrm{diff}(D_1+D_2+D_3)}{\lesssim} 0 : \{P \star \gamma \to \beta\}\ \exists \_ : \mathrm{unit}_r\ \{P \star \gamma \to \beta \cup \{I'\}\}}\ \textbf{r-updt}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 : \square\tau \qquad \Delta;\Phi \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \overset{\mathrm{diff}(D_1+D_2+D_3)}{\lesssim} 0 : \{P \star \gamma \to \beta\}\ \exists \_ : \mathrm{unit}_r\ \{P \star \gamma \to \beta \setminus \{I'\}\}}\ \textbf{r-updtB}$$

Figure 16: Relational typing judgment, part (2)

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \qquad i \notin FV(\Phi;\Gamma)}{\Sigma;\Delta;\Phi;\Gamma \vdash \Lambda\ t_1 \ominus \Lambda\ t_2 \lesssim 0 : \overset{\mathrm{diff}(D)}{\forall i{::}S.\ \tau}}\ \textbf{r-ilam} \qquad \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \overset{\mathrm{diff}(D')}{\forall i{::}S.\ \tau} \qquad \Sigma;\Delta \vdash I :: S}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1\ [] \ominus t_2\ [] \lesssim D + D'[I/i] : \tau\{I/i\}}\ \textbf{r-iapp}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : C \supset \tau \qquad \Sigma;\Delta;\Phi \vDash C}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{celim}\ t_1 \ominus \mathsf{celim}\ t_2 \lesssim 0 : \tau}\ \textbf{r-celim} \qquad \frac{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim 0 : C \supset \tau}\ \textbf{r-cimpl}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : C\&\tau_1 \qquad \Sigma;\Delta;\Phi \wedge C;x:\tau_1,\Gamma \vdash t_1' \ominus t_2' \lesssim D' : \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{clet}\ t_1\ \mathsf{as}\ x\ \mathsf{in}\ t_1' \ominus \mathsf{clet}\ t_2\ \mathsf{as}\ x\ \mathsf{in}\ t_2' \lesssim D + D' : \tau_2}\ \textbf{r-clet}$$

$$\frac{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \qquad \Sigma;\Delta;\Phi \vDash C}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : C\&\tau}\ \textbf{r-andI}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;x:\tau_1,f\ :\overset{\mathrm{diff}(D)}{\tau_1 \longrightarrow \tau_2},\Gamma,f\ :U(A_1,A_2) \vdash t_1 \ominus t_2 \lesssim D : \tau_2 \\ \Sigma;\Delta;\Phi;|\Gamma|_1 \vdash_0^0 \mathsf{Fix}\, f(x).t_1 : A_1 \qquad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash_0^0 \mathsf{Fix}\, f(x).t_2 : A_2\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{Fix}\, f(x).t_1 \ominus \mathsf{Fix}\, f(x).t_2 \lesssim 0 : \overset{\mathrm{diff}(D)}{\tau_1 \longrightarrow \tau_2}}\ \textbf{r-fix-ext}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \tau_1 \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \langle t_1,t_2\rangle \ominus \langle t_1',t_2'\rangle \lesssim D_1 + D_2 : \tau_1 \times \tau_2}\ \textbf{r-prod}$$

$$\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t' \lesssim D : \tau_1 \times \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \pi_1(t) \ominus \pi_1(t_1') \lesssim D : \tau_1}\ \textbf{r-proj1} \qquad \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t' \lesssim D : \tau_1 \times \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \pi_2(t) \ominus \pi_2(t_1') \lesssim D : \tau_2}\ \textbf{r-proj2}$$

$$\frac{\Sigma;\Delta;\Phi \vdash \tau \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{nil} \ominus \mathsf{nil} \lesssim D : \mathsf{list}^{\alpha}[0]\ \tau}\ \textbf{r-nil}$$

$$\frac{\Sigma;\Delta;\Phi\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \tau \qquad \Sigma;\Delta;\Phi\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathsf{list}^{\alpha}[n]\ \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{cons}(t_1,t_2) \ominus \mathsf{cons}(t_1',t_2') \lesssim D_1 + D_2 : \mathsf{list}^{\alpha+1}[n+1]\ \tau}\ \textbf{r-cons1}$$

$$\frac{\Sigma;\Delta;\Phi\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \Box\tau \qquad \Sigma;\Delta;\Phi\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathsf{list}^{\alpha}[n]\ \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{cons}(t_1,t_2) \ominus \mathsf{cons}(t_1',t_2') \lesssim D_1 + D_2 : \mathsf{list}^{\alpha}[n+1]\ \tau}\ \textbf{r-cons2}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi\Gamma \vdash t \ominus t' \lesssim D : \mathsf{list}^{\alpha}[n]\ \tau \qquad \Sigma;\Delta;\Phi \wedge n = 0\Gamma \vdash t_1 \ominus t_1' \lesssim D' : \tau' \\ \Sigma;i,\Delta;\Phi \wedge n = i+1\Gamma,h:\Box\tau,tl:\mathsf{list}^{\alpha}[i]\ \tau \vdash t_2 \ominus t_2' \lesssim D' : \tau' \\ \Sigma;i,B\Delta;\Phi \wedge n = i+1 \wedge \alpha = B+1;\Gamma,h:\tau,tl:\mathsf{list}^{\beta}[i]\ \tau \vdash t_2 \ominus t_2' \lesssim D' : \tau'\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{case}\ t'\ \mathsf{of\ nil} \to t_1' \mid h :: tl \to t_2' \ominus \mathsf{case}\ t'\ \mathsf{of\ nil} \to t_1' \mid h :: tl \to t_2' \lesssim D + D' : \tau'}\ \textbf{r-caseL}$$

Figure 17: Relational typing judgment, part (3)

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : A_1 \qquad \Sigma;\Delta;\Phi;\Gamma, x : U(A_1, A_1) \vdash t_2 \ominus t_2' \lesssim D_2 : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ x = t_1\ \mathsf{in}\ t_2 \ominus t_2' \lesssim U_1 + D_2 + c_{lt} : \tau}\ \textbf{r-lt-e}$$

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t_1' : A_1' \qquad \Sigma;\Delta;\Phi;\Gamma, x : U(A_1', A_1') \vdash t_2 \ominus t_2' \lesssim D_2 : \tau'}{\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus \mathsf{let}\ x = t_1'\ \mathsf{in}\ t_2' \lesssim D_2 - L_1 - c_{lt} : \tau'}\ \textbf{r-e-lt}$$

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : A_1 \overset{\mathrm{exec}(L,U)}{\longrightarrow} A_2 \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : U(A_1, A_2')}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1\ t_2 \ominus t_2' \lesssim U_1 + U + D_2 + c_{app} : U(A_2, A_2')}\ \textbf{r-app-e}$$

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t_1' : A_1' \overset{\mathrm{exec}(L,U)}{\longrightarrow} A_2' \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : U(A_2, A_1')}{\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_1'\ t_2' \lesssim D_2 - L_1 - L - c_{app} : U(A_2, A_2')}\ \textbf{r-e-app}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t : A_1 + A_2 \\ \Sigma;\Delta;\Phi;\Gamma, x : U(A_1, A_1) \vdash t_1 \ominus t' \lesssim D_2 : \tau \qquad \Sigma;\Delta;\Phi;\Gamma, y : U(A_2, A_2) \vdash t_2 \ominus t' \lesssim D_2 : \tau\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash\ \mathsf{case}\ (t, x.t_1, y.t_2) \ominus t' \lesssim U_1 + D_2 + c_{case} : \tau}\ \textbf{r-case-e}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1'}_{L_1'} t' : A_1 + A_2 \\ \Sigma;\Delta;\Phi;\Gamma, x : U(A_1, A_1) \vdash t \ominus t_1' \lesssim D_2 : \tau \qquad \Sigma;\Delta;\Phi;\Gamma, y : U(A_2, A_2) \vdash t \ominus t_2' \lesssim D_2 : \tau\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus\ \mathsf{case}\ (t', x.t_1', y.t_2') \lesssim D_2 - L_1' - c_{case} : \tau}\ \textbf{r-e-case}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : \{P_1\}\ \overset{\mathrm{exec}(L,U)}{\exists \vec{\gamma}_1 : A_1}\ \{Q_1\} \qquad dom(P) = dom(P_1) \qquad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_2}_{L_2} t_2' : \{P_2\}\ \overset{\mathrm{exec}(L',U')}{\exists \vec{\gamma}_1 : A_1'}\ \{Q_2\} \\ \Sigma;\Delta;\Phi;\Gamma, x : U(A_1, A_1) \vdash t_2 \ominus t_2' \lesssim D_2 : \{P \sqcup P_1\}\ \overset{\mathrm{diff}(D')}{\exists \vec{\gamma}_1.\tau}\ \{Q\}\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus t_2' \lesssim -L_2 : \quad \overset{\mathrm{diff}(U_1+U+(D_2+U_2)+D'+c_{let})}{\{P\}\ \exists \vec{\gamma}_1.\tau}\ \{Q\}}\ \textbf{r-let-e}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t_1' : \{P_1\}\ \overset{\mathrm{exec}(L,U)}{\exists \vec{\gamma}_1 : A_1'}\ \{Q_1\} \qquad dom(P) = dom(P_1) \qquad \Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_2}_{L_2} t_2 : \{P_2\}\ \overset{\mathrm{exec}(L',U')}{\exists \vec{\gamma}_1 : A_1}\ \{Q_2\} \\ \Sigma;\Delta;\Phi;\Gamma, x : U(A_1', A_1') \vdash t_2 \ominus t_2' \lesssim D_2 : \{P \sqcup P_1\}\ \overset{\mathrm{diff}(D')}{\exists \vec{\gamma}_1.\tau'}\ \{Q\}\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \lesssim U_2 : \quad \overset{\mathrm{diff}(D'+(D_2-L_2)-L_1-L-c_{\mathrm{let}})}{\{P\}\ \exists \vec{\gamma}_1.\tau'}\ \{Q\}}\ \textbf{r-e-let}$$

Figure 18: Relational typing judgment, part (4), Asychonization rules

$\boxed{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_2}$

$$\frac{\Sigma;\Delta;\Phi \models A_1' \sqsubseteq A_1 \qquad \Sigma;\Delta;\Phi \models A_2 \sqsubseteq A_2' \qquad \Sigma;\Delta;\Phi \models L' \leq L \qquad \Sigma;\Delta;\Phi \models U \leq U'}{\Sigma;\Delta;\Phi \models A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2 \sqsubseteq A_1' \overset{\text{exec}(L',U')}{\longrightarrow} A_2'}\ \textbf{s-u-arrow}$$

$$\frac{\Sigma; i :: S,\Delta;\Phi \models A \sqsubseteq A' \qquad \Sigma; i :: S,\Delta;\Phi \models L' \leq L \qquad \Sigma;\Delta;\Phi \models U \leq U' \qquad i \notin FV(\Phi)}{\Sigma;\Delta;\Phi \models \forall i :: S.\ A \sqsubseteq \forall i :: S.\ A'}\ \textbf{s-a-forall}$$

$$\frac{\Sigma;\Delta;\Phi \models L' \leq L \qquad \Sigma;\Delta;\Phi \models U \leq U' \qquad \Sigma;\Delta;\Phi \models P \sqsubseteq P' \qquad \Sigma,\vec{\gamma}_1;\Delta;\Phi \models Q' \sqsubseteq Q \qquad \Sigma;\Delta;\Phi \vDash \vec{\gamma}_1 \subseteq \vec{\gamma}_2 \qquad \Sigma;\Delta;\Phi \models A \sqsubseteq A'}{\Sigma;\Delta;\Phi \models \{P\}\ \overset{\text{exec}(L,U)}{\exists \vec{\gamma}_1 : A}\ \{Q\} \sqsubseteq \{P'\}\ \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_2 : A'}\ \{Q'\}}\ \textbf{s-um}$$

$$\frac{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_1' \qquad \Sigma;\Delta;\Phi \models A_2 \sqsubseteq A_2'}{\Sigma;\Delta;\Phi \models A_1 \times A_2 \sqsubseteq A_1' \times A_2'}\ \text{S-A-PROD}\ \textbf{s-a-prod}$$

$$\frac{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_1' \qquad \Sigma;\Delta;\Phi \models A_2 \sqsubseteq A_2'}{\Sigma;\Delta;\Phi \models A_1 + A_2 \sqsubseteq A_1' + A_2'}\ \textbf{s-a-sum} \qquad \frac{\Sigma;\Delta;\Phi \models A \sqsubseteq A' \qquad \Sigma;\Delta;\Phi \models I = I'}{\Sigma;\Delta;\Phi \models \mathsf{Array}_\gamma[I]\ A \sqsubseteq \mathsf{Array}_\gamma[I']\ A'}\ \textbf{s-a-array}$$

$$\frac{\Sigma;\Delta;\Phi \models A \sqsubseteq A' \qquad \Sigma;\Delta;\Phi \models I = I'}{\Sigma;\Delta;\Phi \models \mathsf{list}[I]\ A \sqsubseteq \mathsf{list}[I']\ A'}\ \textbf{s-a-list} \qquad \frac{\Sigma; i :: S,\Delta;\Phi \models A \sqsubseteq A' \qquad i \notin FV(\Phi)}{\Sigma;\Delta;\Phi \models \exists i :: S.\ A \sqsubseteq \exists i :: S.\ A'}\ \textbf{s-a-exist}$$

Figure 19: Unary subtyping rules, part (1)

$$\frac{\Sigma; f\,\Delta;\Phi \models A \sqsubseteq A' \qquad \Sigma;\Delta;\Phi \wedge C \models C'}{\Sigma;\Delta;\Phi \models C \& A \sqsubseteq C' \& A'}\ \textbf{s-a-cand} \qquad \frac{\Sigma;\Delta;\Phi \models A \sqsubseteq A' \qquad \Sigma;\Delta;\Phi \wedge C' \models C}{\Sigma;\Delta;\Phi \models C \supset A \sqsubseteq C' \supset A'}\ \textbf{s-a-cimpl}$$

$$\frac{}{\Sigma;\Delta;\Phi \models A \sqsubseteq A}\ \textbf{s-a-refl} \qquad \frac{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_2 \qquad \Sigma;\Delta;\Phi \models A_2 \sqsubseteq A_3}{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_3}\ \textbf{s-a-trans}$$

Figure 20: Unary subtyping rules, part (2)

$\boxed{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_2}$

$$\frac{}{\Sigma;\Delta;\Phi \models \mathrm{int}_r \sqsubseteq \Box U(\mathrm{int},\mathrm{int})}\ \textbf{s-r-int-box} \qquad \frac{}{\Sigma;\Delta;\Phi \models \mathrm{int}_r[I] \sqsubseteq \Box U(\mathrm{int}[I],\mathrm{int}[I])}\ \textbf{s-r-int-i-box}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \Box U(\mathrm{int},\mathrm{int}) \sqsubseteq \Box \mathrm{int}_r}\ \textbf{s-r-uint-box} \qquad \frac{}{\Sigma;\Delta;\Phi \models \mathrm{unit} \sqsubseteq \Box\mathrm{unit}}\ \text{S-R-UNIT-BOX}\ \textbf{s-r-unit-box}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau_1' \sqsubseteq \tau_1 \qquad \Sigma;\Delta;\Phi \models \tau_2 \sqsubseteq \tau_2' \qquad \Sigma;\Delta;\Phi \models c \leq c'}{\Sigma;\Delta;\Phi \models \tau_1 \overset{\mathrm{diff}(c)}{\longrightarrow} \tau_2 \sqsubseteq \tau_1' \overset{\mathrm{diff}(c')}{\longrightarrow} \tau_2'}\ \textbf{s-r-diff}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \Box(\tau_1 \overset{\mathrm{diff}(c)}{\longrightarrow} \tau_2) \sqsubseteq \Box\tau_1 \overset{\mathrm{diff}(0)}{\longrightarrow} \Box\tau_2}\ \textbf{s-r-BD}$$

$$\frac{}{\Sigma;\Delta;\Phi \models U(A_1 \overset{\mathrm{exec}(L,U)}{\longrightarrow} A_2, A_1' \overset{\mathrm{exec}(L',U')}{\longrightarrow} A_2') \sqsubseteq U(A_1, A_1') \overset{\mathrm{diff}(U-L')}{\longrightarrow} U(A_2, A_2')}\ \text{S-R-EXECDIFF}\ \textbf{s-r-execdiff}$$

$$\frac{\Sigma; i :: S, \Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma; i :: S, \Delta;\Phi \models c \leq c' \qquad i \notin FV(\Phi)}{\Sigma;\Delta;\Phi \models \forall i{::}S.\ \tau \sqsubseteq \forall i{::}S.\ \tau_1'}\ \textbf{s-r-forall-diff}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \Box(\forall i{::}S.\ \tau) \sqsubseteq \forall i{::}S.\ \Box\tau}\ \textbf{s-r-forall-box} \qquad \frac{}{\Sigma;\Delta;\Phi \models U(\mathrm{unit},\mathrm{unit}) \sqsubseteq \mathrm{unit}}\ \textbf{s-r-unit}$$

$$\frac{}{\Sigma;\Delta;\Phi \models U(\forall i :: S.\ A, \forall i :: S.\ A') \sqsubseteq \forall i{::}S.\ U(A, A')}\ \textbf{s-r-forall-U}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_1' \qquad \Sigma;\Delta;\Phi \models \tau_2 \sqsubseteq \tau_2'}{\Sigma;\Delta;\Phi \models \tau_1 \times \tau_2 \sqsubseteq \tau_1' \times \tau_2'}\ \textbf{s-r-prod} \qquad \frac{}{\Sigma;\Delta;\Phi \models \Box\tau_1 \times \Box\tau_2 \sqsubseteq \Box(\tau_1 \times \tau_2)}\ \textbf{s-r-prod-box}$$

$$\frac{}{\Sigma;\Delta;\Phi \models U(A_1 \times A_2, A_1' \times A_2') \sqsubseteq U(A_1, A_1') \times U(A_2, A_2')}\ \textbf{s-r-prod-U}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_1' \qquad \Sigma;\Delta;\Phi \models \tau_2 \sqsubseteq \tau_2'}{\Sigma;\Delta;\Phi \models \tau_1 + \tau_2 \sqsubseteq \tau_1' + \tau_2'}\ \textbf{s-r-sum} \qquad \frac{}{\Sigma;\Delta;\Phi \models \Box\tau_1 + \Box\tau_2 \sqsubseteq \Box(\tau_1 + \tau_2)}\ \textbf{s-r-sum-box}$$

$$\frac{\Sigma; i :: S, \Delta;\Phi \models \tau \sqsubseteq \tau' \qquad i \notin FV(\Phi)}{\Sigma;\Delta;\Phi \models \exists i :: S.\ \tau \sqsubseteq \exists i :: S.\ \tau'}\ \textbf{s-r-exist} \qquad \frac{}{\Sigma;\Delta;\Phi \models \exists i :: S.\ \Box\tau \sqsubseteq \Box(\exists i :: S.\ \tau)}\ \textbf{s-r-exist-box}$$

Figure 21: Relational subtyping rules, part (1)

$\boxed{\Delta;\Phi \models \tau_1 \sqsubseteq \tau_2}$

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma;\Delta;\Phi \models I{=}I' \qquad \Sigma;\Delta;\Phi \models \gamma{=}\gamma'}{\Sigma;\Delta;\Phi \models \mathsf{Array}_{\gamma}[I]\ \tau \sqsubseteq \mathsf{Array}_{\gamma'}[I']\ \tau'}\ \textbf{s-r-array}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma;\Delta;\Phi \models I{=}I' \ \Sigma;\Delta;\Phi \models \alpha{\leq}\alpha'}{\Sigma;\Delta;\Phi \models \mathsf{list}^{\alpha}[I]\ \tau \sqsubseteq \mathsf{list}^{\alpha'}[I']\ \tau'}\ \textbf{s-r-list} \qquad \frac{\Sigma;\Delta;\Phi \models I{=}0}{\Sigma;\Delta;\Phi \models \mathsf{list}^{\alpha}[I]\ \tau \sqsubseteq \mathsf{list}^{\alpha}[I]\ \Box\tau}\ \textbf{s-r-list2}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \mathsf{list}^{\alpha}[I]\ \Box\tau \sqsubseteq \Box(\mathsf{list}^{\alpha}[I]\ \tau)}\ \textbf{s-r-listB} \qquad \frac{\Sigma;\Delta;\Phi \models A_1 \sqsubseteq A_1' \qquad \Sigma;\Delta;\Phi \models A_2 \sqsubseteq A_2'}{\Sigma;\Delta;\Phi \models U\ (A_1, A_2) \sqsubseteq U\ (A_1', A_2')}\ \textbf{s-r-ua}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau_1' \sqsubseteq \tau_1 \qquad \Sigma;\Delta;\Phi \models \tau_2 \sqsubseteq \tau_2' \qquad \Delta;\Phi \models D{\leq}D'}{\Sigma;\Delta;\Phi \models \Box(\tau_1 \overset{\mathrm{diff}((D))}{\longrightarrow} \tau_2) \sqsubseteq \Box\tau_1' \overset{\mathrm{diff}((D'))}{\longrightarrow} \Box\tau_2'}\ \textbf{s-r-a}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma;\Delta;\Phi \wedge C \models C'}{\Sigma;\Delta;\Phi \models C\&\tau \sqsubseteq C'\&\tau'}\ \textbf{s-r-c-and} \qquad \frac{}{\Sigma;\Delta;\Phi \models C\&\Box\tau \sqsubseteq \Box(C\&\tau)}\ \textbf{s-r-c-and-box}$$

$$\frac{}{\Sigma;\Delta;\Phi \models U(C \supset A, C \supset A') \sqsubseteq C \supset U(A, A')}\ \textbf{s-r-cimpl-U}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Sigma;\Delta;\Phi \wedge C' \models C}{\Sigma;\Delta;\Phi \models C \supset \tau \sqsubseteq C' \supset \tau'}\ \textbf{s-r-c-impl}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \Box(C \supset \tau) \sqsubseteq C \supset \Box\tau}\ \textbf{s-r-c-impl-box} \qquad \frac{}{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau}\ \textbf{s-r-refl} \qquad \frac{}{\Sigma;\Delta;\Phi \models \tau \sqsubseteq U(|\tau|_1, |\tau|_2)}\ \textbf{s-r-w}$$

$$\frac{}{\Sigma;\Delta;\Phi \models \Box\tau \sqsubseteq \tau}\ \textbf{s-r-T} \qquad \frac{}{\Sigma;\Delta;\Phi \models \Box\tau \sqsubseteq \Box\Box\tau}\ \textbf{s-r-D} \qquad \frac{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_2}{\Sigma;\Delta;\Phi \models \Box\tau_1 \sqsubseteq \Box\tau_2}\ \textbf{s-r-bb}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_2 \qquad \Sigma;\Delta;\Phi \models \tau_2 \sqsubseteq \tau_3}{\Sigma;\Delta;\Phi \models \tau_1 \sqsubseteq \tau_3}\ \textbf{s-r-trans}$$

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Delta;\Phi \models c{\leq}c' \qquad \Delta;\Phi \models P'{\subseteq}P \qquad \Delta;\Phi \models \vec{\gamma}_1 \subseteq \vec{\gamma}_2 \qquad \Sigma, \vec{\gamma}_1 : \vec{\mathbb{L}};\Delta;\Phi \models Q{\subseteq}Q'}{\Sigma;\Delta;\Phi \models \{P\}\ \overset{\mathrm{diff}(c)}{\exists \vec{\gamma}_1 : \tau}\ \{Q\} \sqsubseteq \{P'\}\ \overset{\mathrm{diff}(c')}{\exists \vec{\gamma}_2 : \tau'}\ \{Q'\}}\ \textbf{s-rm}$$

$$\frac{}{\begin{array}{l}\Sigma;\Delta;\Phi \models U\ (\{\gamma_i \to T_i\}\ \overset{\mathrm{exec}(L,U)}{\exists \vec{\gamma_1'} : A_1}\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \overset{\mathrm{exec}(L',U')}{\exists \vec{\gamma_2'} : A_1}\ \{Q_2\}) \sqsubseteq \\ \qquad \{\gamma_i \to \beta_i\}\ \overset{\mathrm{diff}(U-L')}{\exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2)}\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}\end{array}}\ \textbf{s-rum}$$

$$\begin{array}{lll} P \subseteq P' & \Leftrightarrow & P = \{\gamma_1 \to \beta_1, \gamma_2 \to \beta_2, \ldots, \gamma_n \to \beta_n\} \\ & & P' = \{\gamma_1 \to \beta_1', \gamma_2 \to \beta_2', \ldots, \gamma_n \to \beta_n'\} \\ & & \forall\, i \in \{1, \ldots, n\}.\beta_i \subseteq \beta_i' \end{array}$$

Figure 22: Relational subtyping rules, part (2)

$$
\begin{array}{ll}
[\![\mathrm{int}]\!]_{g,k} & = \{\, n \mid n \in \mathbb{N} \,\} \\
[\![\mathrm{unit}]\!]_{g,k} & = \{\, () \,\} \\
[\![\mathrm{int}[I]]\!]_{g,k} & = \{\, n \mid I = n \,\} \\
[\![\mathsf{Array}_\gamma[I]\ A]\!]_{g,k} & = \{\, l \mid I = n \ \wedge\ g(\gamma) = (l, A, n) \,\} \\
[\![A \xrightarrow{\mathrm{exec}(L,U)} A']\!]_{g,k} & = \left\{ \mathsf{fix}\, f(x).t \;\middle|\; \begin{array}{l} \forall k' < k,\ g' \supseteq g.\ \forall v.\ v \in [\![A]\!]_{g',k'} \implies \\ \quad t[v/x][\mathsf{fix}\, f(x).t/f] \in [\![A']\!]^{e}_{g',k',(L,U)} \end{array} \right\} \\
[\![\{P\}\ \exists \vec{\gamma}.A\ \{Q\}]\!]^{\mathrm{exec}(L,U)}_{g,k} & = \left\{ v \;\middle|\; \begin{array}{l} \forall g_1 \supseteq g.\ \forall k_1 \le k,\ k_2 < k_1.\ \forall c.\ \forall H. \\ H \vDash_{g_1,k_1} P \wedge\ v; H \Downarrow^{c,k_2}_{f} \implies \\ \quad \exists g_2 \supseteq g_1.\ \exists H_1,\ v_1,\ \vec{\gamma}.\ v; H \Downarrow^{c,k_2}_{f} v_1; H_1 \ \wedge \\ L \le c \le U \ \wedge\ H_1 \vDash_{g_2,k_1-k_2} Q \ \wedge\ v_1 \in [\![A]\!]_{g_2,k_1-k_2} \ \wedge \\ \Big(\exists n.\ P = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \ \wedge \\ \quad \forall i \in [1,n].\ g(\gamma_i) = (l_i, A, m) \implies \\ \quad \forall j. H[l_i][j] \neq H_1[l_i][j] \implies j \in T_i \Big) \end{array} \right\} \\
[\![\forall i :: S.\ A]\!]^{\mathrm{exec}(L,U)}_{g,k} & = \{\, \Lambda.t \mid \forall I.\ \vdash I :: S\,.\ \wedge\ t \in [\![A[I/i]]\!]^{E,(L[I/i],U[I/i])}_{g,k-1} \,\} \\
[\![\exists i :: S.\ A\,]\!]_{g,k} & = \{\, \mathsf{pack}\, v \mid \exists I.\ \vdash I :: S \ \wedge\ v \in [\![A[I/i]]\!]_{g,k-1} \,\} \\
[\![A_1 + A_2]\!]_{g,k} & = \{\, \mathrm{inl}\ v \mid v \in [\![A_1]\!]_{g,k} \,\} \cup \{\, \mathrm{inr}\ v \mid v \in [\![A_2]\!]_{g,k} \,\} \\
[\![C \supset A]\!]_{g,k} & = \{\, v \mid \nvDash C \ \vee\ v \in [\![A]\!]_{g,k-1} \,\} \\
[\![C \& A]\!]_{g,k} & = \{\, v \mid \vDash C \ \wedge\ v \in [\![A]\!]_{g,k-1} \,\} \\
[\![A]\!]^{e}_{g,k,(L,U)} & = \{\, t \mid \forall v,\ k'.\ k' \le k \wedge\ t \Downarrow^{c,k'} v \Rightarrow\ v \in [\![A]\!]_{g,k-k'} \ \wedge\ L \le c \le U \,\} \\
[\![\cdot]\!]_{g,k} & = \{\, \emptyset \,\} \\
[\![\Omega, x : A]\!]_{g,k} & = \{\, (\sigma[v/x]) \mid \sigma \in [\![\Omega]\!]_{g,k} \ \wedge\ v \in [\![A]\!]_{g,k} \,\}
\end{array}
$$

where

$$
\begin{array}{lll}
H \vDash_{g,k} empty & \mathsf{iff} & \mathrm{true} \\
H \vDash_{g,k} (\gamma \to \beta) & \mathsf{iff} & \exists l, A, n : g(\gamma) = (l, A, n) \\
 & & \wedge \big(\forall i \le n. \big(H[l][i]\big) \in (\!|A|\!)_{g,k-1}\big) \\
H \vDash_{g,k} (P * Q) & \mathsf{iff} & \exists H', H'', g', g'' : \\
 & & \big(H = H' \uplus H''\big) \wedge \big(g = g' \uplus g''\big) \\
 & & \wedge \big(H' \vDash_{g',k} P\big) \wedge \big(H'' \vDash_{g'',k} Q\big) \\
v; H \Downarrow^{c,k}_{f} & \triangleq & \exists v', H'. v; H \Downarrow^{c,k}_{f} v'; H'
\end{array}
$$

Figure 23: Unary interpretation of types

$$
\begin{aligned}
&(\!|\mathrm{int}_r|\!)_{G,k} && = \{\, (n_1, n_2) \mid n_1 = n_2 \,\} \\
&(\!|\mathrm{unit}_r|\!)_{G,k} && = \{\, ((), ()) \,\} \\
&(\!|\mathrm{int}[I]|\!)_{G,k} && = \{\, (n, n) \mid I = n \,\} \\
&(\!|\mathsf{Array}_\gamma[I]\ \tau|\!)_{G,k} && = \{\, (l_1, l_2) \mid I = n \wedge G(\gamma) = (l_1, l_2, \tau, n) \,\} \\
&(\!|\tau_1 \xrightarrow{\mathrm{diff}((D))} \tau_2|\!)_{G,k} && = \left\{ (\,\mathsf{fix}\, f(x).t_1,\ \mathsf{fix}\, f(x).t_2\,) \;\middle|\; \begin{array}{l} \forall G' \supseteq G.\ \forall k' \le k-1.\ \forall v_1,\ v_2. \\ (v_1, v_2) \in (\!|\tau_1|\!)_{G',k'} \implies \\ \quad (t_1[v_1/x][\mathsf{fix}\, f(x).t_1/f], \\ t_2[v_2/x][\mathsf{fix}\, f(x).t_2/f]) \in (\!|\tau_2|\!)^e_{G',k',D} \wedge \\ \forall j.\ \Big(\mathsf{fix}\, f(x).t_1 \in [\![|\tau_1|_1 \xrightarrow{\mathrm{exec}(0,\infty)} |\tau_2|_1]\!]_{G|_1,j} \wedge \\ \mathsf{fix}\, f(x).t_2 \in [\![|\tau_1|_2 \xrightarrow{\mathrm{exec}(0,\infty)} |\tau_2|_2]\!]_{G|_2,j}\Big) \end{array} \right\} \\
&(\!|\{P\}\ \exists\vec{\gamma}.\tau\ \{Q\}|\!)^{\mathrm{diff}(D)}_{G,k} && = \left\{ (v_1, v_2) \;\middle|\; \begin{array}{l} \forall G_1 \supseteq G.\ \forall k_1 \le k,\ k_2 < k_1,\ k_3,\ c_1,\ c_2,\ H_1,\ H_2. \\ (H_1, H_2) \vDash_{G_1,k_1} P \wedge v_1; H_1 \Downarrow_f^{c_1,k_2} \wedge v_2; H_2 \Downarrow_f^{c_2,k_3} \\ \implies \quad \exists G_2 \supseteq G_1,\ H_1',\ H_2',\ v_1',\ v_2'. \\ \Big(v_1; H_1 \Downarrow_f^{c_1,k_2} v_1'; H_1' \wedge v_2; H_2 \Downarrow_f^{c_2,k_3} v_2'; H_2' \wedge \\ \quad (H_1', H_2') \vDash_{G_2,k_1-k_2} Q \wedge (v_1', v_2') \in (\!|\tau|\!)_{G_2,k_1-k_2} \wedge \\ \quad c_1 - c_2 \le D\Big) \end{array} \right\} \\
&(\!|U(A_1, A_2)|\!)_{G,k} && = \{\, (v_1, v_2) \mid \forall k'.\ v_1 \in [\![A_1]\!]_{G|_1,k'} \wedge v_2 \in [\![A_2]\!]_{G|_2,k'} \,\} \\
&(\!|\forall i{::}S.\ \tau|\!)^{\mathrm{diff}((D))}_{G,k} && = \left\{ (\Lambda.t_1, \Lambda.t_2) \;\middle|\; \begin{array}{l} \forall I.\ \vdash I :: S \wedge (t_1, t_2) \in (\!|\tau[I/i]|\!)^{E,D[I/i]}_{G,k-1} \wedge \\ \forall j.\ t_1 \in [\![|\tau|_1]\!]^{E,(0,\infty)}_{G|_1,j} \wedge t_2 \in [\![|\tau|_2]\!]^{E,(0,\infty)}_{G|_2,j} \end{array} \right\} \\
&(\!|\exists i :: S.\ \tau\ |\!)_{G,k} && = \{\, (\mathsf{pack}\, v_1, \mathsf{pack}\, v_2) \mid \exists I.\ \vdash I :: S \wedge (v_1, v_2) \in (\!|\tau[I/i]|\!)_{G,k-1} \,\} \\
&(\!|\tau_1 + \tau_2|\!)_{G,k} && = \{\, (\mathrm{inl}\ v_1, \mathrm{inl}\ v_2) \mid (v_1, v_2) \in (\!|\tau_1|\!)_{G,k-1} \vee (v_1, v_2) \in (\!|\tau_2|\!)_{G,k-1} \} \\
&(\!|\Box\tau|\!)_{G,k} && = \{\, (v, v) \mid (v, v) \in (\!|\tau|\!)_{G,k} \,\} \\
&(\!|C \supset \tau|\!)_{G,k} && = \{\, (v_1, v_2) \mid \not\models C \vee (v_1, v_2) \in (\!|\tau|\!)_{G,k-1} \,\} \\
&(\!|C \& \tau|\!)_{G,k} && = \{\, (v_1, v_2) \mid \models C \wedge (v_1, v_2) \in (\!|\tau|\!)_{G,k-1} \,\} \\
&(\!|\tau|\!)^e_{G,k,D} && = \left\{ (t_1, t_2) \;\middle|\; \begin{array}{l} \forall k_1 \le k.\ \forall k_2,\ v_1,\ v_2.\ (t_1 \Downarrow^{c_1,k_1} v_1 \wedge t_2 \Downarrow^{c_2,k_2} v_2) \\ \implies \quad (v_1, v_2) \in (\!|\tau|\!)_{G,k-k_1} \wedge c_1 - c_2 \le D \end{array} \right\} \\
&(\!|\cdot|\!)_{G,k} && = \{\, \emptyset \,\} \\
&(\!|\Gamma|\!)_{G,k} && = \{\, (\sigma_1, \sigma_2) \mid \forall x \in dom(\Gamma).\ \forall \tau.\ x : \tau \in \Gamma.\ (\sigma_1(x), \sigma_2(x)) \in (\!|\tau|\!)_{G,k} \,\}
\end{aligned}
$$

where

$$
\begin{aligned}
&(H_1, H_2) \vDash_{G,k} empty && \text{iff} \quad \text{true} \\
&(H_1, H_2) \vDash_{G,k} (\gamma \mapsto \beta) && \text{iff} \quad \exists l_1, l_2, \tau, n : G(\gamma) = (l_1, l_2, \tau, n) \\
&&& \quad \wedge \big(\forall i \le n.\, \big(H_1[l_1][i], H_2[l_2][i]\big) \in (\!|\tau|\!)_{G,k-1}\big) \\
&&& \quad \wedge \big(\forall i \le n.\, \big(H_1[l_1][i] \neq H_2[l_2][i] \Rightarrow i \in \beta\big)\big) \\
&(H_1, H_2) \vDash_{G,k} (P * Q) && \text{iff} \quad \exists H_1', H_1'', H_2', H_2'', G', G'' : \\
&&& \quad \big(H_1 = H_1' \uplus H_1''\big) \wedge \big(H_2 = H_2' \uplus H_2''\big) \wedge \big(G = G' \uplus G''\big) \\
&&& \quad \wedge \big((H_1', H_2') \vDash_{G',k} P\big) \wedge \big((H_1'', H_2'') \vDash_{G'',k} Q\big) \\
&G(\gamma) = (l_1, l_2, \tau, n), && G|_1(\gamma) = (l_1, |\tau|_1, n), G|_2(\gamma) = (l_2, |\tau|_2, n)
\end{aligned}
$$

Figure 24: Relational interpretation of types

2 Logical relation

We use $\lambda x.t$ as syntatic sugar for $\text{fix}\, f(x).t$ when f does not show in t.

Lemma 1 (Monotonicity)**.**

1. *If $k' \le k$ and $G \subseteq G'$, then $(\!|\tau|\!)_{G,k} \subseteq (\!|\tau|\!)_{G',k'}$.*
2. *If $k' \le k$ and $g \subseteq g'$, then $[\![A]\!]_{g,k} \subseteq [\![A]\!]_{g',k'}$.*
3. *If $k' \le k$ and $G \subseteq G'$, then $(\!|\tau|\!)^{E,D}_{G,k} \subseteq (\!|\tau|\!)^{E,D}_{G',k'}$.*
4. *If $k' \le k$ and $g \subseteq g'$, then$[\![A]\!]^{E,(L,U)}_{g,k} \subseteq [\![A]\!]^{E,(L,U)}_{g',k'}$.*
5. *If $k' \le k$ and $G \subseteq G'$, then $(\!|\Gamma|\!)_{G,k} \subseteq (\!|\Gamma|\!)_{G',k'}$*
6. *If $k' \le k$ and $g \subseteq g'$, then$[\![\Omega]\!]_{g,k} \subseteq [\![\Omega]\!]_{g',k'}$*

This monotonicity lemma shows that the value interpretation, expression interpretation, and our interpretation of either the unary context or relational context are monotone. Take the value interpretation of relational type as an instance, if we know that a pair of values (v_2, v_2) is in the value interpretation of certain relational type τ for some step k in certain world G, then this pair of values should also be in the value interpretation of τ for some smaller step k' in a bigger world G'. We can read other cases in a similar way.

Proof. The proof of the 6 statements will be organized as follows. The statements of (1) and (3), related to the interpretations of relational types, are first proved simultaneously by induction on τ. Similarly, the statements about the interpretations of the unary types, (2), and (4) are proved simultaneously by induction on the unary type A. The statements about the contexts, (5) and (6), follows directly from statement (1) and (2). □

Lemma 2 (Heap extension)**.** *If $(H_1, H_2) \vDash_{G,k} P$, then for any H_1', H_2', $(H_1 \uplus H_1', H_2 \uplus H_2') \vDash_{G,k} P$.*

Proof. It is simply proved by case analysis on P. □

Lemma 3 (Heap evaluation extension)**.** *If $H; t \Downarrow_f^{c,k} H_1; v$, then for any H', $H \uplus H'; t \Downarrow_f^{c,k} H_1 \uplus H'; v$.*

Proof. The proof is by induction on evaluation derivation on operation semantics. □

Lemma 4 (Evaluation cost soundness)**.**

1. *If $\vdash_L^U t : A$ and $t \Downarrow^c v$, then $L \le c \le U$.*
2. *If $\vdash_L^U t : \{P\}\, \overset{\text{exec}(L',U')}{\exists \gamma : A}\, \{Q\}$ and $t \Downarrow_p^c v$, then $L + L' \le c \le U + U'$.*
3. *If $\vdash_L^U t : \{P_1\}^{U_1}_{L_1} \exists \gamma_1 : \{P_2\}^{U_2}_{L_2} \exists \gamma_2 : \ldots \{Q_2\}\{Q_1\}$ and $t; H \Downarrow_p^{c_1} v_1; H1$ and $H_1; v_1 \Downarrow_p^{c_2} v_2; H_2$ etc, then $L + L_1 + L_2 + \ldots L_n \le c_1 + c_2 + \cdots + c_n \le U + U_1 + U_2 + \cdots + U_n$.*
4. *If $\vdash t_1 \ominus t_2 \lesssim D : \tau$ and $t_1 \Downarrow_p^{c_1} v_1$ and $t_2 \Downarrow_p^{c_2} v_2$, then $c_1 - c_2 \le D$.*
5. *If $\vdash t_1 \ominus t_2 \lesssim D : \{P\}\, \overset{\text{diff}(D')}{\exists \gamma : \tau}\, \{Q\}$ and $t_1 \Downarrow_p^{c_1} v_1$ and $t_2 \Downarrow_p^{c_2} v_2$, then $c_1 - c_2 \le D + D'$.*
6. *If $\vdash t_1 \ominus t_2 \lesssim D : \{P_1\}^{\text{diff}(D_1)} \exists \gamma_1 : \{P_2\}^{\text{diff}(D_2)} \exists \gamma_2 : \ldots \{Q_2\}\{Q_1\}$ and $t; H \Downarrow_p^{c_1} v_1; H1$ and $v_1; H_1 \Downarrow_p^{c_2} v_2; H_2$ then $c_1 - c_2 \le D + D_1 + D_2 \ldots D_n$.*

Proof. (1) is directly proved from the definition of unary expression interpretation. If $\vdash_L^U t:\tau$, then $t \in [\![\tau]\!]_{G,k}^{E,(L,U)}$, unfold the definition, we get : $L \le c \le U$ and $t \Downarrow^c v$.
(2) $t \in [\![\{P\}\ \exists\gamma : A\ \{Q\}]\!]_{G,k}^{E,(L,U)}$, $t \Downarrow^{c_1} v'$ and $v' \Downarrow_f^{c_2} v$ unfold the definition, we know $c = c_1 + c_2$ and $L \le c_1 \le U$ and $L' \le c_2 \le U'$. It is proved.
(3) $t \in [\![\{P_1\}_{L_1}^{U_1} \exists\gamma_1 : \{P_2\}_{L_2}^{U_2} \exists\gamma_2 : \ldots \{Q_2\}\{Q_1\}]\!]_{G,k}^{E,(L,U)}$. Unfold the definition, we can get : $L + L_1 + L_2 + \ldots L_n \le c_1 + c_2 + \cdots + c_n \le U + U_1 + U_2 + \cdots + U_n$. This case is proved.
(4) From the fundamental theory, If $\vdash t_1 \ominus t_2 \lesssim D : \tau$, we know closed terms (t_1, t_2) in the interpretation of relational type τ, $(\!|\tau|\!)_{G,k}^{E,D}$, unfold the definition, $c_1 - c_2 \le D$ is proved.
(5) From the fundamental theory, If $\vdash t_1 \ominus t_2 \lesssim D : \{P\}\ \exists\gamma : \tau\ \{Q\}$, we know closed terms (t_1, t_2) in the interpretation $(\!|\{P\}\ \exists\gamma : \tau\ \{Q\}|\!)_{G,k}^{E,D}$, unfold the definition, $c_1 - c_2 \le D + D'$ is proved.
(6) From the fundamental theory, If $\vdash t_1 \ominus t_2 \lesssim D : \{P_1\}^{\mathsf{diff}(D_1)} \exists\gamma_1 : \{P_2\}^{\mathsf{diff}(D_2)} \exists\gamma_2 : \ldots \{Q_2\}\{Q_1\}$, we know closed terms (t_1, t_2) in the interpretation of nested relational type τ, $(\!|\tau|\!)_{G,k}^{E,D}$, unfold the definition level by level, $c_1 - c_2 \le D + D_1 + D_2 \cdots + D_n$ is proved.

□

Lemma 5 (Well-formedness)**.**

1. *If* $\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$, *then* $FV(t_1) \subseteq dom(\Gamma)$ *and* $FV(t_2) \subseteq dom(\Gamma)$.
2. *If* $\Delta;\Phi;\Omega \vdash_L^U t : A$, *then* $FV(t) \subseteq dom(\Omega)$.

Proof. The proof is by induction on the typing derivations. □

Lemma 6 (Value Projection)**.**

1. *If* $(v_1, v_2) \in (\!|\tau|\!)_{G,k}$, *then* $\forall k. v_1 \in [\![|\tau|_1]\!]_{G|_1,k}$ *and* $v_2 \in [\![|\tau|_2]\!]_{G|_2,k}$.
2. *If* $(\sigma_1, \sigma_2) \in (\!|\Gamma|\!)_{G,k}$, *then* $\forall k. \sigma_1 \in (\!||\Gamma|_1|\!)_{G|_1,k}$ *and* $\sigma_2 \in (\!||\Gamma|_2|\!)_{G|_2,k}$.

Proof. Proof of Statement (1) is by induction on $(\!|\tau|\!)_{G,k}$. Proof of Statement (2) follows by proof of (1). □

Lemma 7 (Heap monotonicity)**.**

1. *If* $k' \le k$ *and* $G' \supseteq G$, *then* $(H_1, H_2) \vDash_{G,k} P \Rightarrow (H_1, H_2) \vDash_{G',k'} P$
2. *If* $k' \le k$ *and* $g' \supseteq g$, *then* $H \vDash_{g,k} P \Rightarrow H \vDash_{g',k'} P$

Proof. **proof of statement (1)**

Assume $k' \le k$ and $G' \supseteq G$, and we have $H_1, H_2 \vDash_{G,k} P$., TS: $H_1, H_2 \vDash_{G',k'} P$
Unfold the definition of $H \vDash_{G',k'} P$, By case analysis on P, there are three cases:

case 1: P=empty It is vacuously true that $H \vDash_{g',k'} \mathsf{empty}$.

case 2: P= $\gamma \to \beta$, From $H_1, H_2 \vDash_{G,k} \gamma \to \beta$, we know:

$$G(\gamma) = (l_1, l_2, \tau, n) \tag{1}$$

$$\forall i \le n.(H_1(l_1)[i], H_2(l_2)[i]) \in (\!|\tau|\!)_{G,k-1} \tag{2}$$

$$\forall i \le n.(H_1(l_1)[i] \ne H_2(l_2)[i]) \Rightarrow i \in \beta \tag{3}$$

We conclude that $G'(\gamma) = (l_1, l_2, \tau, n)$ from assumption $G' \supseteq G$ and $\forall i \le n.H(l)[i] \in (\!|\tau|\!)_{G',k'}$ by Lemma 1 on (2).

case 3: P= $P_1 \star P_2$. Suppose $\exists n.P = \{\gamma_i \to \beta_i\}$ for $i \in [1,\ldots,n]$.
Unfold the definition of $H_1, H_2 \vDash_{G,k} \{\gamma_i \to \beta_i\}$ for $i \in [1,\ldots,n]$.
We know there exists $H_1 = (h_1^1) \uplus (h_1^2) \uplus \cdots \uplus (h_1^n)$ and $H_2 = (h_2^1) \uplus (h_2^2) \uplus \cdots \uplus (h_2^n)$ and $G = G_1 \uplus G_2 \uplus \cdots \uplus G_n$
so that $h_1^i, h_2^i \vDash_{G_i,k} \gamma_i \to \beta_i$ for any i.
Since $G' \supseteq G$, we know there exists $G_i' \supseteq G_i$ s.t. $G' = G_1' \uplus G_2' \uplus \cdots \uplus G_n' \wedge \forall i.G_i' \supseteq G_i$
Use the conclusion of case 2, we know:
$\forall i \in [1,n].h_1^i, h_2^i \vDash_{G_i',k'} \gamma_i \to \beta_i$.

proof of statement (2)
Assume $k' \le k$ and $g' \supseteq g$, and we have $H \vDash_{g,k} P$. TS: $H \vDash_{g',k'} P$
Unfold the definition of $H \vDash_{g',k'} P$, by case analysis on P, there are three cases:

case 1: P=empty, It is vacuously true that $H \vDash_{g',k'}$ empty.

case 2: P= $\gamma \to \beta$. From $H \vDash_{g,k} \gamma$, we know:

$$g(\gamma) = (l, A, n) \tag{1}$$
$$\forall i \le n.H(l)[i] \in [\![A]\!]_{g,k-1} \tag{2}$$

We conclude that $g'(\gamma) = (l, A, n)$ from assumption $g' \supseteq g$ and $\forall i \le n.H(l)[i] \in [\![A]\!]_{g',k'}$ by Lemma 1 on (2).

case 3: P = $P * Q$
Assume $p = \{\gamma_i\}$ for $i \in [1,\ldots,n]$.
From the definition of $H \vDash_{g,k} \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, we know
$\exists H = (h_1^1) \uplus (h_1^2) \uplus \cdots \uplus (h_1^n)$ and $g = g_1 \uplus g_2 \uplus \cdots \uplus g_n$ s.t. $h_1^i \vDash_{g_i,k} \gamma_i \to \beta_i$ for any i.
Since $g' \supseteq g$, we know: $\exists g_1', g_2', \ldots, g_n'.g' = g_1' \uplus g_2' \uplus \cdots \uplus g_n' \wedge \forall i.g_i' \supseteq g_i$.
By the conclusion of case 2. we know: $h_1^i \vDash_{g_i',k'} \gamma_i$ for any i.

This completes the proof of statement (2). □

Lemma 8 (Value interpretation containment)**.**

1. *If* $(v_1, v_2) \in (\!|\tau|\!)_{G,k}$,*then* $(v_1, v_2) \in (\!|\tau|\!)_{G,k}^{E,0}$
2. *If* $v \in [\![A]\!]_{g,k}$, *then* $\forall t \ge 0.v \in [\![A]\!]_{g,k}^{E,(0,t)}$

Proof. (1) Assume $(v_1, v_2) \in (\!|\tau|\!)_{G,k}$(*)
TS: $(v_1, v_2) \in (\!|\tau|\!)_{G,k}^{E,0}$
Following the definition of $(\!|\tau|\!)_{G,k}^{E,0}$, we get: $v_1 \Downarrow^0 v_1$ and $v_1 \Downarrow^0 v_1$ and $0-0 \le 0$ and $(v_1, v_2) \in (\!|\tau|\!)_{G,k-0}$. This completes the proof. □

Proof. Assume $v \in [\![A]\!]_{g,k}$(*)
TS: $\forall t \ge 0.v \in [\![A]\!]_{g,k}^{E,(0,t)}$ Following the definition of $[\![A]\!]_{g,k}^{E,(0,t)}$. We know $v \Downarrow^0 v$ and $v \in [\![A]\!]_{g,k}$. This completes the proof. □

Lemma 9 (Value evaluation)**.** $v \Downarrow^0 v$

Proof. By induction on the value term v. □

Lemma 10 (heap projection)**.** *If* $(H_1, H_2) \vDash_{G,k} \gamma_i \to \beta_i, i \in (1,n)$,*then* $\forall T, T'.H_1 \vDash_{G|_1,k} \gamma_i \to T_i$ *and* $H_2 \vDash_{G|_2,k} \gamma_i \to T_i'$

Proof. Suppose $H_1 = h_1 \uplus h_2 \ldots h_n$ and $H_2 = h'_1 \uplus h'_2 \ldots h'_n$

Unfold $(h_i, h'_i) \vDash_{G,k} \gamma_i \to \beta_i$, we know:

$$G(\gamma) = (l_1, l_2, \tau, n) \tag{1}$$
$$\forall i \le n.(h_i(l_1)[i], h'_i(l_2)[i]) \in (\!|\tau|\!)_{G,k-1} \tag{2}$$

From (1), we know:

$$G|_1(\gamma) = (l_1, \gamma, n) \tag{3}$$
$$G|_2(\gamma) = (l_2, \gamma, n) \tag{4}$$

From(2) and Lemma (Value projection) and ,we know:

$$\forall i \le n.(H_1(l_1)[i]) \in [\![|\tau|_1]\!]_{G|_1,k-1} \tag{5}$$
$$\forall i \le n.(H_2(l_1)[i]) \in [\![|\tau|_2]\!]_{G|_2,k-1} \tag{6}$$

Using (3)(5), (4)(6), this lemma is proved.

□

Lemma 11 (heap subtyping)**.** *If* $\vdash \delta : \Delta$ *and* $\vDash \delta\Phi$ *and* $\Delta;\Phi \vDash P \subseteq P'$ *and* $H_1, H_2 \vDash_{G,k} \delta P$*, then* $H_1, H_2 \vDash_{G,k} \delta P'$.

Proof. Assume we have: $H_1, H_2 \vDash_{G,k} \delta P$.
We want to show : $H_1, H_2 \vDash_{G,k} \delta P'$.
Suppose $\exists n.\delta P = \{\gamma_i \to \beta_i\}$ for $i \in [1, \ldots, n]$.
Unfold the definition of $H_1, H_2 \vDash_{G,k} \{\gamma_i \to \beta_i\}$ for $i \in [1, \ldots, n]$.
We know there exists $H_1 = (h_1^1) \uplus (h_1^2) \uplus \cdots \uplus (h_1^n)$ and $H_2 = (h_2^1) \uplus (h_2^2) \uplus \cdots \uplus (h_2^n)$ and $G = G_1 \uplus G_2 \uplus \cdots \uplus G_n$ so that $h_1^i, h_2^i \vDash_{G_i,k} \gamma_i \to \beta_i$ for any i.
Unfold the definition of $\Delta;\Phi \vDash \delta P \subseteq \delta P'$, we know: $\delta P' = \{\gamma_i \to \beta'_i\}, i \in [1, \ldots, n]$.
STS: $\forall i \in [1, \ldots, n].\exists x_1^i, x_2^i \vDash_{G_i,k]} \gamma_i \to \beta'_i$ where $H_1 = (x_1^1) \uplus (x_1^2) \uplus \cdots \uplus (x_1^n)$ and $H_2 = (x_2^1) \uplus (x_2^2) \uplus \cdots \uplus (x_2^n)$
Pick i.Choose $x_1^i = h_1^i$ and $x_2^i = h_2^i$, by unfolding the definition of $h_1^i, h_2^i \vDash_{G_i,k} \gamma_i \to \beta_i$.
we have: $G_i(\gamma_i) = (l_1^i, l_2^i, \tau^i, n)$ and $\forall m \le n.(h_1^i(l_1^i[m]) \neq h_2^i(l_2^i)[m]) \Rightarrow m \in \beta_i$
Since we know $\beta_i \subseteq \beta'_i$, we get: $\forall i \in [1, \ldots, n].G_i(\gamma_i) = (l_1^i, l_2^i, \tau_i, n) \wedge \forall m \le n.(x_1^i(m) \neq x_2^i(m)) \Rightarrow m \in \beta'_i$.
For any $i \in [1, \ldots, n]$, choose $x_1^i = h_1^i$ and $x_2^i = h_2^i$, we know: $x_1^i, x_2^i \vDash_{G_i,k} \gamma_i \to \beta'_i$
We also know $H_1 = (x_1^1) \uplus (x_1^2) \uplus \cdots \uplus (x_1^n)$ and $H_2 = (x_2^1) \uplus (x_2^2) \uplus \cdots \uplus (x_2^n)$ from assumption.
This completes the proof of the lemma. □

Lemma 12 (Subtyping soundness)**.**

1. *If* $\Delta;\Phi \vDash \tau \sqsubseteq \tau'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $(v, v') \in (\!|\delta\tau|\!)_{G,k}$*, then* $(v, v') \in (\!|\delta\tau'|\!)_{G,k}$.
2. *If* $\Delta;\Phi \vDash A \sqsubseteq A'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $(v) \in [\![\delta A]\!]_{g,k}$*, then* $(v) \in [\![\delta A']\!]_{g,k}$.
3. *If* $\Delta;\Phi \vDash \tau \sqsubseteq \tau'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $(t, t') \in (\!|\delta\tau|\!)_{G,k}^{E,D}$ *and* $D \le D'$*, then* $(t, t') \in (\!|\delta\tau'|\!)_{G,k}^{E,D'}$.
4. *If* $\Delta;\Phi \vDash A \sqsubseteq A'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $t \in [\![\delta A]\!]_{g,k}^{E,(L,U)}$ *and* $L' \le L$ *and* $U \le U'$*, then* $t \in [\![\delta A']\!]_{g,k}^{E,(L',U')}$.
5. *If* $\Delta;\Phi \vDash \tau \sqsubseteq \tau'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $\forall i \in \{1,2\}.\ (v) \in [\![|\delta\tau|_i]\!]_{G|_i,k}$*, then* $v \in [\![|\delta\tau'|_i]\!]_{G|_i,k}$.
6. *If* $\Delta;\Phi \vDash \tau \sqsubseteq \tau'$ *and* $\vdash \delta : \Delta$ *and* $\vDash \Phi\delta$ *and* $\forall i \in \{1,2\}.\ (e) \in [\![|\delta\tau|_i]\!]_{G|_i,k}^{E,(L,U)}$ *and* $L' \le L$ *and* $U \le U'$*, then* $e \in [\![|\delta\tau'|_i]\!]_{G|_i,k}^{E,(L,U)}$.

Proof Statements (1) and (2) (5) are by proven simultaneously by induction on the subtyping derivation. (3),(4),(6) will be proved first.

Proof. **Proof of statement (3)** Assume that $\Delta;\Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi\delta$ and $(t,t') \in (\!|\delta\tau|\!)^{E,D}_{G,k}$ and $D \le D'$.
TS: $(t,t') \in (\!|\delta\tau'|\!)^{E,D'}_{G,k}$.
Assume that 1) $t \Downarrow^{s,c} v$, 2) $t' \Downarrow^{s',c'} v'$, 3) $s < k$.
Unfold the assumption $(t,t') \in (\!|\delta\tau|\!)^{E,D}_{G,k}$ using assumption (1)(2)(3), we know that $c - c' \le D(a)$, $(v,v') \in (\!|\delta\tau|\!)_{G,k-s}(b)$.
we conclude: $c - c' \le D'$ from assumption $D \le D'$. and by IH1 on (b), we know $(v,v') \in (\!|\delta\tau'|\!)_{G,k-s}$.

Proof of statement (4) Assume $\Delta;\Phi \models A \sqsubseteq A'$ and $\vdash \delta : \Delta$ and $\models \Phi\delta$ and $e \in [\![\delta A]\!]^{E,(L,U)}_{g,k}$ and $L' \le L$ and $U \le U'$. We want to show: $t \in [\![\delta A]\!]^{E,(L,U)}_{g,k}$
Assume: $t \Downarrow^{s,c}$, $s < k$.
Unfold the assumption $t \in [\![\delta A]\!]^{E,(L,U)}_{g,k}$, we know: $L \le c \le U(a)$, $v \in [\![\delta A]\!]_{g,k-s}(b)$.
Then, we conclude: $L' \le c \le U'$ from assumption. By IH2 with the first premise on (b), we get: $v \in [\![\delta A']\!]_{g,k-s}$
.
Proof of statement (6) Assume $\Delta;\Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi\delta$ and $\forall i \in \{1,2\}$. $(e) \in [\![|\delta\tau|_i]\!]^{E,(L,U)}_{G|_i,k}$ and $L' \le L$ and $U \le U'$
TS: $t \in [\![|\delta\tau'|_i]\!]^{E,(L,U)}_{G|_i,k}$
Assume: $t \Downarrow^{s,c}$, $s < k$. Unfold the assumption $t \in [\![|\delta\tau'|_i]\!]^{E,(L,U)}_{G|_i,k}$, we know: $L \le c \le U(a)$, $v \in [\![|\delta\tau'|_i]\!]_{G|_i,k-s}(b)$.
We conclude: $L' \le c \le U'$ from assumption, by IH5 with the first premise on (b), we get: $v \in [\![|\delta\tau'|_i]\!]_{G|_i,k-s}$
.
Proof of statement (1)
The proof is by induction on the subtyping derivation, we will detail some of the most interesting cases.
Case

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \qquad \Delta;\Phi \models D \le D' \qquad \Delta;\Phi \models P' \subseteq P \qquad \Sigma, \vec{\gamma}_1 : \vec{\mathbb{L}};\Delta;\Phi \models Q \subseteq Q' \qquad \Sigma \models \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2}{\Sigma;\Delta;\Phi \models \{P\}\ \exists^{\mathrm{diff}(D)} \vec{\gamma}_1 : \tau\ \{Q\} \sqsubseteq \{P'\}\ \exists^{\mathrm{diff}(D')} \vec{\gamma}_2 : \tau'\ \{Q'\}}\ \text{S-RM}$$

Assume δ such that $\vdash \delta : \Delta$ and $\models \delta\Phi$. Assume also that we have G, k, v_1, v_2 such that

$$(v_1, v_2) \in (\!|\{\delta P\}\ \exists^{\mathrm{diff}(\delta D)} \vec{\gamma}_1 : \delta\tau\ \{\delta Q\}|\!)_{\delta G,k} \tag{1}$$

We want to show

$$(v_1, v_2) \in (\!|\{\delta P'\}\ \exists^{\mathrm{diff}(\delta D')} \vec{\gamma}_2 : \delta\tau'\ \{\delta Q'\}|\!)_{\delta G,k} \tag{2}$$

By definition, consider $G_1 \supseteq \delta G, k_1 \le k, k_2 < k, H_1, H_2$ and assume

$$(H_1, H_2) \models_{G_1,k_1} \delta P' \wedge H_1; v_1 \downarrow^{k_2}_f \wedge H_2; v_2 \not\downarrow^f$$

Since $\Delta;\Phi \models P' \subseteq P$ we also have $(H_1, H_2) \models_{G_1,k_1} \delta P$.
Thus, by Equation (1) we have that there exist $G_2 \supseteq G_1, H_1', H_2', v_1', v_2', c_1, c_2$ such that

$$v_1; H_1 \Downarrow^{c_1,k_2}_f v_1'; H_1' \wedge v_2; H_2 \Downarrow^{c_2,k_2}_f v_2'; H_2' \wedge (H_1', H_2') \models_{G_2,k_1-k_2} \delta Q \wedge (v_1', v_2') \in (\!|\delta\tau|\!)_{G_2,k_1-k_2} \wedge c_1 - c_2 \le \delta D$$

Since $\Sigma, \vec{\gamma}_1 : \vec{\mathbb{L}};\Delta;\Phi \models Q \subseteq Q'$ we also have $(H_1', H_2') \models_{G_2,k_1-k_2} \delta Q'$ by using Lemma 11 on $(H_1', H_2') \models_{G_2,k_1-k_2} \delta Q$.
Additionally, from $\Delta;\Phi \models \tau \sqsubseteq \tau'$ and $\Delta;\Phi \models D \le D'$
By IH on the first premise, we also get
$(v_1', v_2') \in (\!|\delta\tau'|\!)_{G_2,k_1-k_2}$ and $c_1 - c_2 \le \delta D'$ respectively.
So we can derive Equation (2), and this completes the proof of case-sub-r-monad.

$$\Sigma;\Delta;\Phi \vDash U\ (\{\gamma_i \to T_i\}\ \exists \vec{\gamma_1'} : A_1\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \exists \vec{\gamma_2'} : A_1\ \{Q_2\}) \sqsubseteq \{\gamma_i \to \beta_i\}\ \exists \vec{\gamma_1}, \vec{\gamma_2} : U(A_1, A_2)\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}$$

Pick G,k,v,v', we assume $\vdash \delta : \Delta$ and $\vDash \Phi\delta$, we have:

$$(v, v') \in (\!|\delta U\ (\{\gamma_i \to T_i\}\ \exists \vec{\gamma_1} : A_1\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \exists \vec{\gamma_1} : A_2\ \{Q_2\})|\!)_{G,k} \tag{a}$$

TS: $(v, v') \in (\!|\delta\ \{\gamma_i \to \beta_i\}\ \exists \vec{\gamma_1}, \vec{\gamma_2} : U(A_1, A_2)\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}|\!)_{G,k}$

Unfold the definition of $(\!|\delta\ \{\gamma_i \to \beta_i\}\ \exists \vec{\gamma_1}, \vec{\gamma_2} : U(A_1, A_2)\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}|\!)_{G,k}$.

Pick $G' \supseteq G, k' \le k, k'' < k, H_1, H_2$, we assume $(H_1, H_2) \vDash_{G',k'} \gamma_i \to \beta_i(1)$, $H_1; v \downarrow_f^{k''} \wedge H_2; v' \not\downarrow^f$ (2).

Still To Show:$\exists G'' \supseteq G, H_1', H_2', v_1, v_1',\ H_1; v \Downarrow_f^{c_1,k''} H_1'; v_1(3),\ H_2; v' \Downarrow_f^{c_2} H_2'; v_1'(4)\ H_1', H_2' \vDash_{G'',k'-k''} \gamma_i \to \beta_i \cup T_i \cup T_i'(5)$, $(v_1, v_1') \in (\!|U(A_1, A_2)|\!)_{G'',k'-k''}(6)$.

From (a), unfold its definition of $(\!|\delta U\ (\{\gamma_i \to T_i\}\ \exists \vec{\gamma_1} : A_1\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \exists \vec{\gamma_1} : A_2\ \{Q_2\})|\!)_{G,k}$

$$\forall k', v \in [\![\{\gamma_i \to T_i\}\ \exists \vec{\gamma_1} : A_1\ \{Q_1\}]\!]_{G|_1,k'} \tag{b}$$

$$\forall k', v' \in [\![\{\gamma_i \to T_i'\}\ \exists \vec{\gamma_2} : A_2\ \{Q_2\}]\!]_{G|_2,k'} \tag{c}$$

Pick k' as k, unfold its definition in (b), we already got: $G'|_1 \supseteq G|_1$ and $k' \le k$ and $k'' < k'$ and $H_1; v \downarrow_f^{k''}$ and $H_1 \vDash_{G'|_1,k'} \gamma_i \to T_i$ holds by using Lemma (heap projection) on (1).

Derive from the definition, we know: $\exists g_1'' \supseteq G'|_1, H_1', v_1., H_1; v \Downarrow_f^{c_1,k''} H_1'; v_1(d), H_1' \vDash_{g_1'',k'-k''} Q_1(e), v_1 \in [\![A_1]\!]_{g_1'',k'-k''}(f)\ \forall i.g_1'(\gamma_i) = (l_i, A, m) \Rightarrow H_1(l)[n] \neq H_1'(l)[n] \Rightarrow n \in T_i(*)$.

Unfold its definition in (c), we already got: $G'|_2 \supseteq G|_2$ and $k' \le k$ and $k'' < k'$ and $H_2; v \downarrow_f$ and $H_2 \vDash_{G'|_2,k'} \gamma_i \to T_i'$ holds by using Lemma 10 (heap projection) on (1).

There are two cases for $H_2; v \downarrow_f \Rightarrow H_2; v \downarrow_f^{k''}$:

1. $H_2; v \downarrow_f^{k_1}$ and $k_1 > k'$. Using Lemma 1, we get $H_2; v \downarrow_f^{k''}$
2. $H_2; v \downarrow_f^{k_1}$ and $k_1 \le k'$. We choose $k'' = k_1$. we get $H_2; v \downarrow_f^{k''}$

Derive from the definition, we know: $\exists g_2'' \supseteq G'|_2, H_2', v_1'., H_2; v \Downarrow_f^{c_2,k''} H_2'; v_1'(g), H_2' \vDash_{g_2'',k'-k''} Q_2(h), v_1' \in [\![A_2]\!]_{g_2'',k'-k''}(i).\ \forall i.g_2'(\gamma_i) = (l_i, A, m) \Rightarrow H_2(l)[n] \neq H_2'(l)[n] \Rightarrow n \in T_i'(**)$.

(3),(4) already proved. STS1: $H_1', H_2' \vDash_{G'',k'-k''} \gamma_i \to \beta_i \cup T_i \cup T_i'$

From (e), we assume: $H_1' = h_1 \uplus h_2 \uplus \ldots h_n(7)$, $H_2' = h_1' \uplus h_2' \uplus \ldots h_n'(8)$.

From (*),(**), we know: $\forall i, n.(H_1(l_i)[n] = (H_2(l_i)[n]) \Rightarrow n \notin \beta_i$, otherwise, $n \in \beta_i$.

for all the n not in β_i, if $H_1(l_i)[n] \neq H_1'(l_i)[n]$, then $n \in T_i$. Similarly, if $H_2(l_i)[n] \neq H_2'(l_i)[n]$, then $n \in T_i'$.

Unfold the definition of $H_1', H_2' \vDash_{G,k} \gamma_i \to \beta_i \cup T_i \cup T_i'$. $\forall n.H_1'(l_i)[n] \neq H_2'(l_i)[n] \Rightarrow n \in \beta_i \cup T_i \cup T_i'$ because (1) at position n, values differs in H_1, H_2 means n in β_i (2) at position n, value same in H_1, H_2. and differ in H_1', H_2' means n must in T_i or T_i' or both.

This completes the proof of case sub-r-u-monad

Proof of statement (2). Proof is by induction on the subtyping derivation.

$$\frac{\Delta;\Phi \vDash U \le U' \qquad \begin{array}{ccc}\Delta;\Phi \vDash A \sqsubseteq A' & \Delta;\Phi \vDash L' \le L & \\ \Delta;\Phi \vDash P \sqsubseteq P' & \Sigma, \vec{\gamma_1}; \Delta;\Phi \vDash Q' \sqsubseteq Q & \vec{\gamma_1} \subseteq \vec{\gamma_2}\end{array}}{\Sigma;\Delta;\Phi \vDash \{P\}\ \exists \vec{\gamma_1} : A\ \{Q\} \sqsubseteq \{P'\}\ \exists \vec{\gamma_2} : A'\ \{Q'\}}\ \textsc{S-A-monad}$$

Pick g,k,v. We assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$, and then we know : $v \in \{P\}\ \overset{\text{exec}(L,U)}{\exists \vec{\gamma}_1 : A}\ \{Q\}$ (a).
To Show: $v \in \{P'\}\ \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_2 : A'}\ \{Q'\}$ (b).
By unfolding the definition of $\{P\}\ \overset{\text{exec}(L,U)}{\exists \vec{\gamma}_1 : A}\ \{Q\}$, we pick $g' \supseteq g, k' \leq k, k'' < k', H$.
Assume $H \vDash_{g',k'} P(1)$, $H; v \downarrow_f^{k''}$ (2). We know: $\exists g'' \supseteq g, H', v_1$, $H; v \Downarrow_f^{c_1,k''} H'; v_1(3)$, $H' \vDash_{g'',k'-k''} Q(4)$, $v_1 \in [\![A]\!]_{g'',k'-k''}(5)$, $P = \gamma_i \to T_i \Rightarrow \forall i.g''(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T_i (*)$.
Pick g'', H', v_1. By IH 2 on the first premise $\Delta; \Phi \vDash A \sqsubseteq A'$ with (5), we know: $v_1 \in [\![A']\!]_{g'',k'-k''}(6)$.
TS: $v \in \{P'\}\ \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_2 : A'}\ \{Q'\}$, unfold its definition, we know: $H \vDash_{g',k'} P'$ from (1) and $P \supseteq P'$ because P and P' have the same γ_i. together with (2)
Still To Show : $H; v \Downarrow_f^{c_1,k''} H'; v_1(c)$, $H' \vDash_{g'',k'-k''} Q'(d)$, $v_1 \in [\![A']\!]_{g'',k'-k''}(e)$, $P' = \gamma_i \to T_i' \Rightarrow \forall i.g_1'(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T_i' (**)$.
(c) is proved by assumption (3), (e) is proved by (6), (d) is proved by (4) and $Q' \supseteq Q$, (**) is proved by (*). Because $P \supseteq P' \Rightarrow \forall i.T_i \supseteq T_i'$. For all the n in T_i, it is also in T_i'. This completes the proof of sub-A-monad.'

Proof of statement (5). Proof is by induction on the subtyping derivation. We will prove i=1, the proof of i=2 is similar

s-RUM

$$\frac{}{\Sigma; \Delta; \Phi \vDash U\ (\{\gamma_i \to T_i\}\ \overset{\text{exec}(L,U)}{\exists \vec{\gamma}_1' : A_1}\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_2' : A_1}\ \{Q_2\}) \sqsubseteq \{\gamma_i \to \beta_i\}\ \overset{\text{diff}(U-L')}{\exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2)}\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$. we have

$$v \in [\![|\delta U\ (\{\gamma_i \to T_i\}\ \overset{\text{exec}(L,U)}{\exists \vec{\gamma}_1 : A_1}\ \{Q_1\}, \{\gamma_i \to T_i'\}\ \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_1 : A_2}\ \{Q_2\})|_i]\!]_{|G|_i,k} \tag{a}$$

TS:$v \in [\![|\delta\ \{\gamma_i \to \beta_i\}\ \overset{\text{diff}(U-L')}{\exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2)}\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}\ |_i]\!]_{|G|_i,k}(b)$ Choose i=1. From (a), we know: $v \in [\![\delta\ \{\gamma_i \to \overset{\text{exec}(L,U)}{T_i}\}\ \exists \vec{\gamma}_1 : A_1\ \{Q_1\}]\!]_{|G|_1,k}(c)$.
From (b), Still To Show:$v \in [\![|\delta\ \{\gamma_i \to \beta_i\}\ \overset{\text{exec}(0,\infty)}{\exists \vec{\gamma}_1, \vec{\gamma}_2 : A_1}\ \{\gamma_i \to \beta_i \cup T_i \cup T_i'\}]\!]_{|G|_1,k}(d)$.
Assume $g' \supseteq G|_1, k' \leq k, k'' < k'$. $H \vDash_{g',k'} \gamma_i \to \mathbb{N}$ (1) , $H; v \downarrow_f^{k''}$ (2).
STS: $\exists g'' \supseteq g', H', v'$, $H; v \Downarrow_f^{k'',c_1} H'; v'(3)$, $H' \vDash_{g'',k'-k''} \gamma_i \to \mathbb{N}$ (4), $v' \in [\![A_1]\!]_{g'',k'-k''}$ (5), $P' = \gamma_i \to \mathbb{N} \Rightarrow \forall i.g_1'(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in N (**)$.
From (c), unfold its definition, we know (2) and $H \vDash_{g',k'} \gamma_i \to T_i$ holds from (1).
Then we get: $\exists g'' \supseteq g', H', v'$, $H; v \Downarrow_f^{k'',c_1} H'; v'$ (e), $H' \vDash_{g'',k'-k''} Q_1(f)$, $v' \in [\![A_1]\!]_{g'',k'-k''}$ (g), $P = \gamma_i \to T_i \Rightarrow \forall i.g_1'(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T_i$ $(**)$.
(3) is proved by (e), (4) is proved by (f) because Q_1 contains at least as many γ_i as its precondition $\gamma_i \to T_i$, (*) is proved by (**) because $T_i \supseteq \mathbb{N}$.
This completes the proof of this case.

$$\frac{\Sigma; \Delta; \Phi \vDash \tau \sqsubseteq \tau' \qquad \Delta; \Phi \vDash D \leq D' \qquad \Delta; \Phi \vDash P' \sqsubseteq P \qquad \Sigma, \vec{\gamma}_1 : \vec{\mathbb{L}}; \Delta; \Phi \vDash Q \sqsubseteq Q' \qquad \Sigma \vDash \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2}{\Sigma; \Delta; \Phi \vDash \{P\}\ \overset{\text{diff}(D)}{\exists \vec{\gamma}_1 : \tau}\ \{Q\} \sqsubseteq \{P'\}\ \overset{\text{diff}(D')}{\exists \vec{\gamma}_2 : \tau'}\ \{Q'\}}\ \textbf{s-RM}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$, we have $v \in [\![|\ \{P\}\ \overset{\text{diff}((c))}{\exists \vec{\gamma}_1 : \tau}\ \{Q\}\ |_i]\!]_{|G|_i,k}$ (a).

TS: $v \in [\![|\delta\ \{P'\}\ \exists \vec{\gamma_2} : \tau'\ \{Q'\}\ |_i]\!]_{|G|_i,k}(b)$. $^{\text{diff}((c'))}$

Choose i=1. From (a), we know: $v \in [\![\delta\ \{|P|_1\}\ \exists \vec{\gamma_1} : |\tau|_1\ \{|Q|_1\}]\!]_{|G|_1,k}(c)$. $^{\text{exec}(0,\infty)}$

From (c),Still To Show: $v \in [\![\delta\ \{|P'|_1\}\ \exists \vec{\gamma_2} : |\tau'|_1\ \{|Q'|_1\}]\!]_{|G|_1,k}\ (d)$ $^{\text{exec}(0,\infty)}$

From $P' \supseteq P$ and $Q \supseteq Q'$, we know $|P|_1 = |P'|_1 \wedge |Q|_1 = |Q'|_1$, it is trivially proved.

This completes the proof of case sub-r-monad.

□

Theorem 2.1 (Fundamental Theorem)**.**

1. *If* $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ *and* $\vdash \delta : \Delta$ *and* $\vDash \delta\Phi$ *and* $(\sigma_1,\sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$*, then* $(\sigma_1 t_1, \sigma_2 t_2) \in (\!|\delta\tau|\!)^e_{G,k,(\delta D)}$.
2. *If* $\Sigma;\Delta;\Phi;\Omega \vdash^U_L t : A$ *and* $\vdash \delta : \Delta$ *and* $\vDash \delta\Phi$ *and there exists* Ω'*. s.t.* $FV(t) \subseteq dom(\Omega')$ *and* $\Omega' \subseteq \Omega$ *and* $\sigma \in (\!|\delta\Omega'|\!)_{g,k}$*, then* $(\sigma t) \in [\![\delta A]\!]^e_{g,k,(\delta L,\delta U)}$.
3. *If* $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ *and* $\vdash \delta : \Delta$ *and* $\vDash \delta\Phi$*, then for* $i \in \{1,2\}$*. if there exists* Γ'_i *s.t.* $FV(t_i) \subseteq dom(\Gamma'_i)$ *and* $\Gamma'_i \subseteq \Gamma$ *and* $\sigma_i \in [\![|\delta\Gamma'_i|_i]\!]_{G|_i,k}$*, then* $\sigma_i t_i \in [\![|\delta\tau_i|]\!]^e_{G|_i,k,(0,\infty)}$.

Corollary 2.1.1. *If* $\forall k.\bullet \vdash t_1 \ominus t_2 \lesssim n : \tau$*, then* $(t_1,t_2) \in (\!|\tau|\!)^e_{G,k,\delta(n)}$.

The notation $(\!|\tau|\!)^e_{G,k,D)}$ and $(\!|\tau|\!)^{E,D}_{G,k}$ can be regarded as interchangable.

Proof of statement (1). if $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ and $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1,\sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$ then $(\sigma_1 t_1, \sigma_2 t_2) \in (\!|\delta\tau|\!)^{E,\delta(D)}_{G,k}$.

Proof by induction on typing derivation:

$$\frac{\Sigma;\Delta;\Phi;x:\tau_1, f\ :\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2, \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{Fix}\, f(x).t_1 \ominus \mathsf{Fix}\, f(x).t_2 \lesssim 0 : \tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2}\ \text{R-FIX}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$ and $(\sigma_1,\sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$

TS: $(\sigma_1 \mathsf{fix}\, f(x).t_1, \sigma_2 \mathsf{fix}\, f(x).t_2) \in (\!|\delta(\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2)|\!)^{E,0}_{G,k}$

By unfolding the definition and the fact that $\mathsf{fix}\, f(x).t_1$ and $\mathsf{fix}\, f(x).t_2$ are values

STS: $(\sigma_1 \mathsf{fix}\, f(x).t_1, \sigma_2 \mathsf{fix}\, f(x).t_2) \in (\!|\delta\tau_1 \overset{\text{diff}((\delta D))}{\longrightarrow} \delta\tau_2|\!)_{G,k}$

Set $F = \sigma_1 \mathsf{fix}\, f(x).t_1, F' = \sigma_2 \mathsf{fix}\, f(x).t_2$, We want to prove a general statement: $\forall m \le k.(F,F') \in (\!|\delta\tau_1 \overset{\text{diff}(\delta D)}{\longrightarrow} \delta\tau_2|\!)_{G,m}$.

By sub-induction on m, there are two cases to show.

subcase 1: m=0, we unfold the definition of function type interpretation, there are two parts to show:

subsubcase 1: $\forall k' < m.(v_1,v_2) \in (\!|\tau_1|\!)_{G,k'} \Rightarrow \ldots$. It is vacuous by definition to show $k' \le -1$.

subsubcase 2: STS: $\forall j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j} \wedge F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$.Pick j.

1. STS 1: $F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j}$, we prove the more general statement.

$$\forall m' \le j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'} \tag{2.1}$$

By subinduction on m', there are two csases:

1.1. m'=0, we unfold the definition of $[\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'}$, there is no non-negative $k' < m'(0)$, it is vacuously true.

1.2. m'=m"+1. By sub-IH on m"

$$F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m''} \tag{2.2}$$

STS: $F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'}$
Unfold the definition of $[\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m''+1}$.
Pick $j'' < m''+1, g' \supseteq G|_1$ and assume that $v \in [\![|\delta\tau_1|_1]\!]_{g',j''}$.
STS: $\sigma_1 t_1[v/x][\text{fix}\, f(x).t_1/f] \in [\![|\tau_2|_1]\!]^{E,(0,\infty)}_{g',j''}$
This follows by IH 3 on the premise instantiated with
$\sigma_1[v/x][\text{fix}\, f(x).t_1/f] \in [\![\delta(x:|\tau_1|_1, f:|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1, |\Gamma_1|)]\!]_{g',j''}$ which holds because

i. $FV(t_1) \subseteq dom(x:\tau_1, f:\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2, \Gamma)$ using Lemma 5.
ii. $\sigma_1 \in [\![|\delta\Gamma|_1]\!]_{g',j''}$ by Lemma 6 and then Lemma 1.
iii. $v \in [\![|\delta\tau_1|_1]\!]_{g',j''}$ from the assumption.
iv. $F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{g',j''}$ by Lemma 1 on (3.2).

2. STS 2: $F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$
Its proof is similar with STS 1.

subcase 2: $m = m'+1 \le k$

By the definition of function types, there are two parts to show:

subsubcase 1: $\forall k' < m.(v_1, v_2) \in (\!|\tau_1|\!)_{G,k} \Rightarrow \ldots$
Assume $G' \supseteq G, k' \le (m-1)$. Pick v_1, v_2 s.t. $(v_1, v_2) \in (\!|\delta\tau_1|\!)_{G',k'}$.
STS: $(t_1[v_1/x][\text{fix}\, f(x).t_1/f], t_2[v_2/x][\text{fix}\, f(x).t_2/f]) \in (\!|\delta\tau_2|\!)^{E,(\delta D)}_{G',k'}$
By IH on premise instantiated with $\sigma_1' = \sigma_1[v_1/x][\text{fix}\, f(x).t_1/f], \sigma_2' = \sigma_2[v_2/x][\text{fix}\, f(x).t_2/f]$
and $(\sigma_1', \sigma_2') \in (\!|(\Gamma, x:\tau, f:\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2)\delta|\!)_{G',k'}$ which holds because

1. $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by assumption and Lemma 1.
2. $(v_1, v_2) \in (\!|\tau|\!)_{G',k'}$ from assumption
3. $(F_1, F_2) \in (\!|\delta(\tau_1 \to \tau_2)|\!)_{G',k'}$ from sub-IH $(F_1, F_2) \in (\!|\delta(\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2)|\!)_{G,m'}$ and Lemma 1 ($k' < m', G' \supseteq G$)

subsubcase 2: $\forall j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j} \wedge F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$
which is already proved above

This completes the proof of this case.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;x:\tau_1, f:\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2, \Gamma, f:U(A_1,A_2) \vdash t_1 \ominus t_2 \lesssim D : \tau_2 \quad \Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^0_0 \text{Fix}\, f(x).t_1 : A_1 \quad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^0_0 \text{Fix}\, f(x).t_2 : A_2}{\Sigma;\Delta;\Phi;\Gamma \vdash \text{Fix}\, f(x).t_1 \ominus \text{Fix}\, f(x).t_2 \lesssim 0 : \tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2}\ \text{R-FIX-EXT}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$.
To Show $(\sigma_1 \text{fix}\, f(x).t_1, \sigma_2 \text{fix}\, f(x).t_2) \in (\!|\delta(\tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2)|\!)^{E,0}_{G,k}$.
By unfolding the definition and the fact that $\text{fix}\, f(x).t_1$ and $\text{fix}\, f(x).t_2$ are values
STS: $(\sigma_1 \text{fix}\, f(x).t_1, \sigma_2 \text{fix}\, f(x).t_2) \in (\!|\delta\tau_1 \overset{\text{diff}(\delta D)}{\longrightarrow} \delta\tau_2|\!)_{G,k}$, we set $F = \sigma_1 \text{fix}\, f(x).t_1, F' = \sigma_2 \text{fix}\, f(x).t_2$, We want to prove a general statement.
$\forall m \le k.(F, F') \in (\!|\delta\tau_1 \overset{\text{diff}((\delta D))}{\longrightarrow} \delta\tau_2|\!)_{G,m}$
By sub-induction on m, there are two cases to show.

subcase 1: m=0. Unfold the definition of function type interpretation, there are two parts to show:

subsubcase 1: $\forall k' < m.(v_1, v_2) \in (\!|\tau_1|\!)_{G,k'} \Rightarrow \ldots$. It is vacuous by definition to show $k' \leq -1$.

subsubcase 2: STS: $\forall j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j} \wedge F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$.Pick j.

1. STS 1: $F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j}$. We prove the more general statement

$$\forall m' \leq j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'} \tag{2.3}$$

   By subinduction on m', there are two csases:

   1.1. m'=0. Unfold the definition of $[\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'}$, there is no non-negative $k' < 0(m')$, it is vacuously true.

   1.2. m'=m"+1. By sub-IH on m'.'

$$F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m''} \tag{2.4}$$

   STS: $F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m'}$
   Unfold the definition of $[\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,m''+1}$.
   Pick $j'' < m''+1, g' \supseteq G|_1$ and assume that $v \in [\![|\delta\tau_1|_1]\!]_{g',j''}$.
   STS: $\sigma_1 t_1[v/x][\text{fix}\, f(x).t_1/f] \in [\![|\tau_2|_1]\!]^{E,(0,\infty)}_{g',j''}$
   This follows by IH 3 on the premise instantiated with
   $\sigma_1[v/x][\text{fix}\, f(x).t_1/f] \in [\![\delta|(x:\tau_1, f:\overset{\text{diff}(D)}{\longrightarrow}\delta\tau_1 \longrightarrow \delta\tau_2, f:U(A_1,A_2),\Gamma_1)|_1]\!]_{g',j''} \Rightarrow$
   $\sigma_1[v/x][\text{fix}\, f(x).t_1/f] \in [\![\delta(x:|\tau_1|_1, f:A_1, |\Gamma_1|_1)]\!]_{g',j''}$ which holds because

   i. $FV(t_1) \subseteq dom(x:\tau_1, f:\overset{\text{diff}(D)}{\longrightarrow}\tau_2, f:U(A_1,A_2),\Gamma)$ using Lemma 5.

   ii. $\sigma_1 \in [\![|\delta\Gamma|_1]\!]_{g',j''}$ by Lemma 6 and then Lemma 1.

   iii. $v \in [\![|\delta\tau_1|_1]\!]_{g',j''}$ from assumption.

   iv. $F \in [\![\delta A_1]\!]_{g',j''}$ by IH2 on the second premise of FIX-EXT instantiated with $\Omega' = |\Gamma|_1$ to show $F \in [\![\delta A_1]\!]_{G|_1,k}$. Then use Lemma 1 .

2. STS 2: $F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$
   Its proof is similar with STS 1.

subcase 2: $m = m'+1 \leq k$. By the definition of function types, there are two parts to show:

subsubcase 1: $\forall k' < m.(v_1, v_2) \in (\!|\tau_1|\!)_{G,k} \Rightarrow \ldots$
Assume $G' \supseteq G, k' \leq (m-1)$. Pick v_1, v_2 s.t. $(v_1, v_2) \in (\!|\delta\tau_1|\!)_{G',k'}$.
STS: $(t_1[v_1/x][\text{fix}\, f(x).t_1/f], t_2[v_2/x][\text{fix}\, f(x).t_2/f]) \in (\!|\delta\tau_2|\!)^{E,(\delta D)}_{G',k'}$
By IH on premise instantiated with $\sigma_1' = \sigma_1[v_1/x][\text{fix}\, f(x).t_1/f], \sigma_2' = \sigma_2[v_2/x][\text{fix}\, f(x).t_2/f]$ and $(\sigma_1', \sigma_2') \in (\!|(\Gamma, x:\tau, f:\overset{\text{diff}(D)}{\longrightarrow}\tau_1 \tau_2, f:U(A_1,A_2))\delta|\!)_{G',k'}$ which holds because

1. $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by assumption and Lemma 1.
2. $(v_1, v_2) \in (\!|\tau|\!)_{G',k'}$ from assumption
3. $(F_1, F_2) \in (\!|\delta(\tau_1 \to \tau_2)|\!)_{G',k'}$ from sub-IH $(F_1, F_2) \in (\!|\delta(\tau_1 \overset{\text{diff}(D)}{\longrightarrow}\tau_2)|\!)_{G,m'}$ and Lemma 1 ($k' < m', G' \supseteq G$)
4. $(F_1, F_2) \in (\!|U(A_1,A_2)|\!)_{G',k}$. STS: $\forall n, i \in \{1,2\}.F_i \in (\!|A_i|\!)_{G|_i,n}$. By Lemma 6, we know $\forall k.\sigma_i \in (\!|\delta|\Gamma|_i|\!)_{g,k}$, it is proved by IH2 on the second and third premises respectively.

subsubcase 2: $\forall j.F \in [\![|\delta\tau_1|_1 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_1]\!]_{G|_1,j} \wedge F' \in [\![|\delta\tau_1|_2 \overset{\text{exec}(0,\infty)}{\longrightarrow} |\delta\tau_2|_2]\!]_{G|_2,j}$
which is already proved above

This proof is the same as case Fix.

$$\textbf{Case}\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \lesssim 0 : \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \mathbb{N}\}}^{\mathrm{diff}(D_1+D_2)}\ \textbf{r-alloc}$$

By assumption we have $\vdash \delta : \Delta$, $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in [\![\delta\Gamma]\!]_{G,k}$, we need to show:

$$(\mathsf{alloc}\,(\sigma_1\ t_1)\,(\sigma_1\ t_2), \mathsf{alloc}\,(\delta\,\sigma_2\ t_1')\,(\delta\,\sigma_2\ t_2')) \in (\!|\{P\}\ \exists\vec{\gamma}.\ \mathsf{Array}_\gamma[I]\ \delta\tau\ \{P \star \gamma \to \mathbb{N}\}|\!)^{e}_{G,k,0}$$

Since $\mathsf{alloc}\,(\sigma_1\ t_1)\,(\sigma_1\ t_2)$ and $\mathsf{alloc}\,(\delta\,\sigma_2\ t_1')\,(\delta\,\sigma_2\ t_2')$ are values, it is sufficient to show:

$$(\mathsf{alloc}\,(\sigma_1\ t_1)\,(\sigma_1\ t_2), \mathsf{alloc}\,(\delta\,\sigma_2\ t_1')\,(\delta\,\sigma_2\ t_2')) \in (\!|\{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ (\delta\,\tau)\ \{P \star \gamma \to \mathbb{N}\}|\!)_{G,k}$$

Unfold its definition, when we are given arbitrary $G' \supseteq G$, $k' \le k$, and heaps H_1, H_2 and assume $(H_1, H_2) \vDash_{G',k'} P$. Suppose the evaluation of $\mathsf{alloc}\,(\sigma_1\ t_1)\,(\sigma_1\ t_2)$ and $\mathsf{alloc}\,(\delta\,\sigma_2\ t_1')\,(\delta\,\sigma_2\ t_2')$ of the following form:

$$\frac{\sigma_1\ t_1 \Downarrow^{c_1,k_1} n \quad \sigma_1\ t_2 \Downarrow^{c_2,k_2} v \quad z = [\overbrace{v,\ldots,v}^{n}] \quad l \text{ fresh}}{\mathsf{alloc}\,(\sigma_1\ t_1)\,(\sigma_1\ t_2); H_1 \Downarrow_f^{c_1+c_2+c_{\mathsf{alloc}},k_1+k_2+1} l; H_1 \uplus [l \to z]}\ \textbf{f-alloc}$$

$$\frac{\sigma_2\ t_1' \Downarrow^{c_1',k_1'} n' \quad \sigma_2\ t_2' \Downarrow^{c_2,k_2} v' \quad z' = [\overbrace{v',\ldots,v'}^{n'}] \quad l' \text{ fresh}}{\mathsf{alloc}\,(\sigma_2\ t_1')\,(\sigma_2\ t_2'); H_2 \Downarrow_f^{c_1'+c_2'+c_{\mathsf{alloc}},k_1'+k_2'+1} l'; H_2 \uplus [l \to z']}\ \textbf{f-alloc}$$

, with $k_1 + k_2 + 1 < k' \le k$. Now taking $G_2 = G'[r \to (l, l', \delta\,\tau, \delta\,I)$, $H_1' = H_1 \uplus [l \to z]$ and $H_2' = H_2 \uplus [l' \to z']$,we must show that

1. $(H_1', H_2') \vDash_{G_2,k'-(k_1+k_2+1)} P \star \gamma \to \mathbb{N}$
2. $(l, l') \in (\!|\mathsf{Array}_\gamma[\delta\,I]\ (\delta\,\tau)|\!)_{G_2,k'-(k_1+k_2+1)}$.
3. $(c_1 + c_2 + c_{\mathsf{alloc}}) - (c_1' + c_2' + c_{\mathsf{alloc}}) \le (\delta\,D_1 + \delta\,D_2)$.

By induction hypothesis on the first premise $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I]$, we conclude:

$$(\sigma_1\ t_1, \sigma_2\ t_1') \in (\!|\mathrm{int}_r[\delta\,I]|\!)^{e}_{G',k',\delta\,D_1}$$

, which we unwind to conclude that: $(n, n') \in (\!|\mathrm{int}_r[\delta\,I]|\!)_{G',k'-k_1}$, which in turn tells us that

$$n = n' = \delta\,I \ \wedge\ c_1 - c_1' \le \delta\,D_1$$

By induction hypothesis on the second premise $\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau$, we conclude that:

$$(\sigma_1\ t_2, \sigma_2\ t_2') \in (\!|\delta\,\tau|\!)^{e}_{G',k',\delta D_2}$$

, which we unfold to conclude that

$$(v, v') \in (\!|\delta\,\tau|\!)_{G',k'-k_2}\ \wedge\ c_2 - c_2' \le \delta\,D_2$$

We can now conclude that:

1. $(H_1', H_2') \vDash_{G_2,k'-(k_1+k_2+1)} P \star \gamma \to \mathbb{N}$. When $\gamma_i \neq \gamma$, the corresponding arrays is not changed, it is true from our assumption $(H_1, H_2) \vDash_{G',k'} P$. When $\gamma_i = \gamma$, we can show that $\forall\, i < \delta\,I.H_1'(l)[i] \neq H_2'(l')[i] \Rightarrow i \in \mathbb{N}$.

2. $(l, l') \in (\!|\mathsf{Array}_\gamma[\delta\, I]\ (\delta\, \tau)|\!)_{G_2, k'-(k_1+k_2+1)}$. It is proved by the its definition using $G_2 = G'[r \to (l, l', \delta\,\tau, \delta\, I)$.

3. $(c_1 + c_2 + c_{\mathsf{alloc}}) - (c_1' + c_2' + c_{\mathsf{alloc}}) \leq (\delta\, D_1 + \delta\, D_2)$. It is proved by the previous conclusions.

This proof is complete.

$$\text{CASE}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I] \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \Box\tau \qquad \gamma\ \mathsf{fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\, t_1\, t_2 \ominus \mathsf{alloc}\, t_1'\, t_2' \lesssim 0 : \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \Box\tau\ \{P \star \gamma \to \emptyset\}}^{\mathrm{diff}(D_1+D_2)}\ \text{R-ALLOCB}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$

TS: $\left(\sigma_1(\mathsf{alloc}\, t_1\ t_2), (\sigma_2(\mathsf{alloc}\, t_1'\ t_2')\right) \in (\!|\delta(\{P\}\ \exists\vec{\gamma}.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\})|\!)^{E,0}_{G,k}$

By unfolding the definition of $(\!|\tau|\!)^E_{G,k}$

STS: $\left(\sigma_1(\mathsf{alloc}\, t_1\ t_2), (\sigma_2(\mathsf{alloc}\, t_1'\ t_2')\right) \in (\!|\delta(\{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\})|\!)_{G,k}$

Unfold the definition of $(\!|\delta\ \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\}|\!)_{G,k}$.

Pick $G' \supseteq G$, H_1, H_2,$k' \leq k, k'' < k', k'''$. Assume $\sigma_1(\mathsf{alloc}\, t_1\ t_2); H_1 \Downarrow_f^{k''}$ (1), $\sigma_2(\mathsf{alloc}\, t_1'\ t_2'); H_2 \Downarrow_f^{k'''}$ (2).

Because $\sigma_1\, t_1$,$\sigma_2\, t_2$ are sub terms of $\sigma_1\ \mathsf{alloc}\, t_1\ t_2$, $\sigma_2\, t_1'$,$\sigma_2\, t_2'$ are sub terms of $\sigma_2\ \mathsf{alloc}\, t_1'\ t_2'$. From (1)(2), we get:

$\sigma_1\, t_1 \Downarrow_f^{k_1} \wedge \sigma_1\, t_2 \Downarrow_f^{k_2}$ (3), $\sigma_2\, t_1' \Downarrow_f \wedge \sigma_2\, t_2' \Downarrow_f$ (4).

From (3),(4), we get: $\exists v_1, v_2, k_1, k_2.\, \sigma_1\, t_1 \Downarrow^{k_1, c_1} v_1 \wedge \sigma_1\, t_2 \Downarrow^{k_2, c_2} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_2 + c_{alloc}$ (a)

$\exists v_1', v_2', k_1', k_2'.\, \sigma_2\, t_1' \Downarrow^{k_1', c_1'} v_1' \wedge \sigma_2\, t_2' \Downarrow^{k_2', c_2'} v_2' \wedge k''' = k_1' + k_2' + 1 \wedge c' = c_1' + c_2' + c_{alloc}$ (b)

From (a),(b) and evaluation rules we get:

$$\exists l.(\sigma_1\ \mathsf{alloc}\, t_1\ t_2); H_1 \Downarrow_f^{k'', c} l; H_1, l \to [v_2, \ldots v_2] \tag{c}$$

$$\exists l'.(\sigma_2\ \mathsf{alloc}\, t_1'\ t_2'); H_2 \Downarrow_f^{c'} l'; H_2, l' \to [v_2', \ldots v_2'] \tag{d}$$

By IH on the first premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G', k''}$ using lemma 1, we get:

$$(\sigma_1\, t_1, \sigma_2\, t_1') \in (\!|\mathrm{int}_r[I]|\!)^{E, \delta D_1}_{G', k''} \tag{*}$$

Unfold the definition of $(\!|\mathrm{int}_r[I]|\!)^E_{G', k''}$ and using (a) and $k_1 \leq k''$, we know:

$$(v_1, v_1') \in (\!|\mathrm{int}_r[I]|\!)_{G', k''-k_1} \Rightarrow v_1 = v_1' = I \wedge c_1 - c_1' \leq \delta D_1 \qquad \text{ih1}$$

By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G', k'-k_1}$ using lemma 1, we get:

$$(\sigma_1\, t_2, \sigma_2\, t_2') \in (\!|\tau|\!)^{E, \delta D_2}_{G', k'-k_1} \tag{**}$$

Unfold the definition of (**) and using (a), (b) and $k_2 \leq k' - k_1$, we know

$$(v_2, v_2') \in (\!|\Box\tau|\!)_{G', k'-k_1-k_2} = (\!|\Box\tau|\!)_{G', k'-k''+1} \wedge c_2 - c_2' \leq \delta D_2 \Rightarrow v_2 = v_2' \qquad \text{ih2}$$

Let us assume: $G'' = G'[r \to (l, l', \tau, I)]$ (e), $H_1' = H_1, l \to [v_2, v_2, \ldots, v_2]$$(f)$, $H_2' = H_2, l' \to [v_2', v_2', \ldots, v_2']$$(g)$. we want to show 3 cases.

TS1 $(l, l') \in (\!|\mathsf{Array}_\gamma[I]\ \tau|\!)_{G'', k'-k''}$
it is proved by unfolding the definition and using the assumption (e).

TS2 $(H_1', H_2') \vDash_{G'', k'-k''} P \star \gamma \to \emptyset$
$G''(\gamma) = (l, l', \tau, I)$, $H_1'' = l \to [v_2, v_2, \ldots, v_2]$, $H_2'' = l' \to [v_2', v_2', \ldots, v_2']$

STS1 $\forall i \leq I, (H_1''(l)[i], H_2''(l')[i]) \in (\!|\tau|\!)_{G'', k'-k''-1}$
It is proved by using ih2 and Lemma 1.

STS2 $\forall i \le I, H_1'(l)[i] \neq H_2''(l')[i] \rightarrow i \in \emptyset$

We know that $\forall i \le I. H_1(l)[i] = v_2, H_1'(l')[i] = v_2'$ and $v_2 = v_2'$ from ih2. We can not find any i that make $H_1'(l)[i] \neq H_2''(l')[i]$ hold. It is proved.

TS3 $c - c' \le \delta(D_1 + D_2)$ from ih1 and ih2.

This proof is complete.

$$Case\frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \tau_1 \xrightarrow{\text{diff}((D))} \tau_2 \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau_1}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1\ t_2 \ominus t_1'\ t_2' \lesssim D + D_1 + D_2 : \tau_2}\ \textbf{r-app}$$

By assumption we have $\vdash \delta : \Delta$, $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in [\![\delta\Gamma]\!]_{G,k}$, we need to show:

$$\left((\sigma_1\ t_1)\ (\sigma_1\ t_2), (\sigma_2\ t_1')\ (\sigma_2\ t_2')\right) \in (\!|\delta\ \tau_2|\!)^e_{G,k,(\delta D + \delta D_1 + \delta D_2)}$$

Unfold the definition, suppose the evaluation of $(\sigma_1\ t_1)\ (\sigma_1\ t_2)$ and $(\sigma_2\ t_1')\ (\sigma_2\ t_2')$ of the following form:

$$\frac{\sigma_1\ t_1 \Downarrow^{c_1,k_1} \mathsf{fix}\, f(x).t \quad \sigma_1\ t_2 \Downarrow^{c_2,k_2} v \quad t[\mathsf{fix}\, f(x).t/f][v/x] \Downarrow^{c_3,k_3} v_1}{(\sigma_1\ t_1)\ (\sigma_1\ t_2) \Downarrow^{c_1+c_2+c_3+c_{\mathsf{fapp}},k_1+k_2+k_3+1} v_1}\ \textbf{e-fix}$$

$$\frac{\sigma_1\ t_1' \Downarrow^{c_1',k_1'} \mathsf{fix}\, f(x).t' \quad \sigma_1\ t_2' \Downarrow^{c_2',k_2'} v' \quad t'[\mathsf{fix}\, f(x).t'/f][v'/x] \Downarrow^{c_3',k_3'} v_1'}{(\sigma_2\ t_1')\ (\sigma_2\ t_2') \Downarrow^{c_1'+c_2'+c_3'+c_{\mathsf{fapp}},k_1'+k_2'+k_3'+1} v_1'}\ \textbf{e-fix}$$

we must show that:

1. $(v_1, v_1') \in (\!|\delta\ \tau_2|\!)_{G,k-(k_1+k_2+k_3+1)}$.
2. $(c_1 + c_2 + c_3 + c_{\mathsf{fapp}}) - (c_1' + c_2' + c_3' + c_{\mathsf{fapp}}) \le \delta D + \delta D_1 + \delta D_2$

By induction hypothesis on the first premise $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D_1 : \tau_1 \xrightarrow{\text{diff}((D))} \tau_2$, instantiated with $(\sigma_1, \sigma_2) \in [\![\delta\Gamma]\!]_{G,k}$ and the index variable substitution δ. We conclude that:

$$(\sigma_1\ t_1, \sigma_2\ t_1') \in (\!|\delta\tau_1 \xrightarrow{\text{diff}((\delta D))} \delta\tau_2|\!)^e_{G,k,\delta D_1}$$

, which we unfold to conclude that

$$(\mathsf{fix}\, f(x).t, \mathsf{fix}\, f(x).t') \in (\!|\delta\tau_1 \xrightarrow{\text{diff}((\delta D))} \delta\tau_2|\!)_{G,k-k_1} \ \wedge\ c_1 - c_1' \le \delta D_1$$

By induction hypothesis on the second premise $\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau_1$, we conclude that:

$$(\sigma_1\ t_2, \sigma_2\ t_2') \in (\!|\delta\ \tau_1|\!)^e_{G,k,\delta D_2}$$

, which we unwind to conclude that

$$(v, v') \in (\!|\delta\ \tau_1|\!)_{G,k-k_2} \ \wedge\ c_2 - c_2' \le \delta D_2$$

We unfold $(\mathsf{fix}\, f(x).t, \mathsf{fix}\, f(x).t') \in (\!|\delta\tau_1 \xrightarrow{\text{diff}((\delta D))} \delta\tau_2|\!)_{G,k-k_1}$, when we choose $G' = G$ and $k' = k-(k_1+k_2+1) \le k - k_1 - 1$, we know that $(v, v') \in (\!|\delta\ \tau_1|\!)_{G,k-(k_2+k_1+1)}$, we conclude that

$$(t[\mathsf{fix}\, f(x).t/f][v/x], t'[\mathsf{fix}\, f(x).t'/f][v'/x]) \in (\!|\delta\ \tau_2|\!)^e_{G,k-(k_1+k_2+1),\delta D}$$

, which we unwind to conclude that

$$(v_1, v_1') \in (\!|\delta\ \tau_2|\!)_{G,k-(k_1+k_2+k_3+1),\delta D} \wedge c_3 - c_3' \le \delta D$$

Now we can conclude as follows:

1. $(v_1, v_1') \in (\!|\delta\, \tau_2|\!)_{G,k-(k_1+k_2+k_3+1)}$. It is already shown.
2. $(c_1 + c_2 + c_3 + c_{\mathsf{fapp}}) - (c_1' + c_2' + c_3' + c_{\mathsf{fapp}}) \leq \delta\, D + \delta\, D_1 + \delta\, D_2$. It is proved by our aforementioned conclusions.

This completes the proof of case r-app.

$$\text{CASE}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta \models I' \leq I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \lesssim 0 : \{P \star \gamma \to \beta\}\ \exists\_.\tau\ \{P \star \gamma \to \beta\}}^{\mathrm{diff}(D_1+D_2)}\ \text{R-R}$$

Assume that $\vdash \delta : \Delta$ and $\models \Phi\delta$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\delta\sigma_1(\mathsf{read}\ t_1\ t_2), \sigma_2(\mathsf{read}\ t_1'\ t_2')) \in (\!|\delta\ \{P \star \gamma \to \beta\}\ \exists\_.\tau\ \{P \star \gamma \to \beta\}|\!)^{E,0}_{G,k}$
Because $\mathsf{read}\ t_1\ t_2$ is a value.
STS: $(\delta\sigma_1(\mathsf{read}\ t_1\ t_2), \sigma_2(\mathsf{read}\ t_1'\ t_2')) \in (\!|\delta\ \{P \star \gamma \to \beta\}\ \exists\_.\tau\ \{P \star \gamma \to \beta\}|\!)_{G,k}$
Unfold the definition of $(\!|\delta\ \{P \star \gamma \to \beta\}\ \exists\_.\tau\ \{P \star \gamma \to \beta\}|\!)_{G,k}$.
Pick $G' \supseteq G$, H_1, H_2, $k' \leq k$, $k'' < k'$, k'''. Assume $(H_1, H_2) \models_{G',k'} P \star \gamma \to \beta$ (1), $\sigma_1(\mathsf{read}\ t_1\ t_2); H_1 \Downarrow_f^{k''}$ (2), $\sigma_2(\mathsf{read}\ t_1'\ t_2'); H_2 \Downarrow_f^{k''}$ (3).
Because t_1, t_2 are sub terms of $\mathsf{read}\ t_1\ t_2$, t_1', t_2' are sub terms of $\mathsf{read}\ t_1'\ t_2'$. From (2)(3), we get: $\sigma_1 t_1 \Downarrow_f^{k_1} \wedge \sigma_1 t_2 \Downarrow_f^{k_2}$ (4), $\sigma_2 t_1' \Downarrow_f^{k_1'} \wedge \sigma_2 t_2' \Downarrow_f^{k_2'}$ (5).
From (4),(5), we get: $\exists l, n, k_1, k_2.\, \sigma_1 t_1 \Downarrow^{k_1, c_1} l \wedge \sigma_1 t_2 \Downarrow^{k_2, c_1'} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_1' + c_{read}$ (a).
$\exists l', n', k_1', k_2'.\, \sigma_2 t_1' \Downarrow^{k_1', c_2} l' \wedge \sigma_2 t_2' \Downarrow^{k_2', c_2'} n' \wedge k''' = k_1' + k_2' + 1 \wedge c' = c_2 + c_2' + c_{read}$ (b)
From (a),(b) and the evaluation rule we get: $\exists v.\, \sigma_1(\mathsf{read}\ t_1\ t_2); H_1 \Downarrow_f^{k'', c} v; H_1 \wedge H_1(l)[n] = v$ (c)
$\exists v'.\, \sigma_2(\mathsf{read}\ t_1'\ t_2'); H_2 \Downarrow_f^{k''', c'} v'; H_2 \wedge H_2(l')[n'] = v'$ (d)
By IH on the first premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. we get:

$$(\sigma_1 t_1, \sigma_2 t_1') \in (\!|\mathsf{Array}_\gamma[I]\ \delta\tau|\!)^{E,D_1}_{G',k'} \tag{6}$$

Unfold (6), since $k_1 \leq k'$ and $\sigma_1 t_1 \Downarrow^{k_1} l$ and $\sigma_2 t_1' \Downarrow l'$, we know

$$(l, l') \in (\!|\mathsf{Array}_\gamma[I]\ \delta\tau|\!)_{G',k'-k_1} \wedge c_1 - c_2 \leq \delta D_1 \tag{e}$$

From (e), we know : $G'(\gamma) = (l, l', \tau, I)$ (7).
By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. we get:

$$(\sigma_1 t_2, \sigma_2 t_2') \in (\!|\delta \mathrm{int}_r[I']|\!)^{E}_{G',k'} \tag{8}$$

Unfold (8), since $k_2 \leq k'$,we know $(n, n') \in (\!|\mathrm{int}_r[I']|\!)_{G',k'-k_2} \Rightarrow n = n' = I' \wedge c_1' - c_2' \leq \delta D_2(f)$. Let us assume: $G' = G'$ (g).

STS1: $(H_1, H_2) \models_{G',k'-k''} P \star \gamma \to \beta$. By Lemma 7 and (1),this is proved.

STS2: $(v, v') \in (\!|\tau|\!)_{G',k'-k''}$. From (c),(d). we know $H_1(l)[n] = v$ and $H_2(l')[n'] = v'$. based on (e) and (f), unfold (1), we know: $\forall i \leq n, (H_1(l)(i), H_2(l')(i)) \in (\!|\tau|\!)_{G',k'-1}$ s.t we know $(H_1(l)[n], H_2(l')[n']) \in (\!|\tau|\!)_{G',k'-1}$. Because $k' - k'' \leq k' - 1$, By Lemma 1, We get: $(v, v') \in (\!|\tau|\!)_{G',k'-k''}$

STS3: $c - c' \leq \delta(D_1 + D_2)$, which is proved by (e),(f).

This completes the proof of case read.

$$\text{CASE}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma\vdash t_1\ominus t_1'\lesssim D_1:\mathsf{Array}_\gamma[I]\ \tau\\ \Sigma;\Delta;\Phi;\Gamma\vdash t_2\ominus t_2'\lesssim D_2:\mathrm{int}[I'] \qquad \Sigma;\Delta\models I'\le I \qquad \Sigma;\Delta;\Phi\models I'\notin\beta \qquad \Sigma;\Delta\vdash P\quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma\vdash \mathsf{read}\ t_1\ t_2\ominus \mathsf{read}\ t_1'\ t_2'\lesssim 0:\{P\star\gamma\to\beta\}\ \exists\_.\Box\tau\ \{P\star\gamma\to\beta\}}^{\mathrm{diff}(D_1+D_2)}\ \text{R-RB}$$

Assume that $\vdash\delta:\Delta$ and $\models\Phi\delta$ and $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G,k}$

TS: $(\delta\sigma_1(\mathsf{read}\ t_1\ t_2),\sigma_2(\mathsf{read}\ t_1'\ t_2'))\in(\!|\delta\ \{P\star\gamma\to\beta\}\ \exists\_.\tau\ \{P\star\gamma\to\beta\}|\!)^{E,0}_{G,k,}$
Because read $t_1\ t_2$ is a value.

STS: $(\delta\sigma_1(\mathsf{read}\ t_1\ t_2),\sigma_2(\mathsf{read}\ t_1'\ t_2'))\in(\!|\delta\ \{P\star\gamma\to\beta\}\ \exists\_.\tau\ \{P\star\gamma\to\beta\}|\!)_{G,k}$

Unfold the definition of $(\!|\delta\ \{P\star\gamma\to\beta\}\ \exists\_.\tau\ \{P\star\gamma\to\beta\}|\!)_{G,k}$.
Pick $G^{'}\supseteq G$, H_1, H_2,$k'\le k$,$k''<k'$,k'''. Assume $(H_1,H_2)\models_{G',k'}P\star\gamma\to\beta$ (1), $\sigma_1(\mathsf{read}\ t_1\ t_2);H_1\Downarrow_f^{k''}$ (2), $\sigma_2(\mathsf{read}\ t_1'\ t_2');H_2\Downarrow_f^{k'''}$ (3).

Because t_1,t_2 are sub terms of read $t_1\ t_2$, t_1',t_2' are sub terms of read $t_1'\ t_2'$. From (2)(3), we get $\sigma_1 t_1\Downarrow_f^{k_1}\wedge\sigma_1 t_2\Downarrow_f^{k_2}$ (4), $\sigma_2 t_1'\Downarrow_f^{k_1'}\wedge\sigma_2 t_2'\Downarrow_f^{k_2'}$ (5).

From (4),(5), we get: $\exists l,n,k_1,k_2.\sigma_1 t_1\Downarrow^{k_1,c_1}l\wedge\sigma_1 t_2\Downarrow^{k_2,c_1'}n\wedge k''=k_1+k_2+1\wedge c=c_1+c_1'+c_{read}$ (a),
$\exists l',n',k_1',k_2'.\sigma_2 t_1'\Downarrow^{k_1',c_2}l'\wedge\sigma_2 t_2'\Downarrow^{k_2',c_2'}n'\wedge k'''=k_1'+k_2'+1\wedge c'=c_2+c_2'+c_{read}$ (b).
From (a),(b) and evaluation rules we get:

$$\exists v.H_1;\sigma_1(\mathsf{read}\ t_1\ t_2)\Downarrow_f^{k'',c}v;H_1\wedge H_1(l)[n]=v \tag{c}$$

$$\exists v'.H_2;\sigma_2(\mathsf{read}\ t_1'\ t_2')\Downarrow_f^{k''',c'}v';H_2\wedge H_2(l')[n']=v' \tag{d}$$

By IH on the first premise instantiated with $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. we get: $(\sigma_1 t_1,\sigma_2 t_1')\in(\!|\mathsf{Array}_\gamma[I]\ \delta\tau|\!)^{E,D_1}_{G',k',}$ (6). Unfold (6), since $k_1\le k'$ and $\sigma_1 t_1\Downarrow^{k_1}l$ and $\sigma_2 t_1'\Downarrow l'$, we know:

$$(l,l')\in(\!|\mathsf{Array}_\gamma[I]\ \delta\tau|\!)_{G',k'-k_1}\wedge c_1-c_2\le\delta D_1 \tag{e}$$

From (e), we know : $G'(\gamma)=(l,l',\tau,I)$ (7). By IH on the second premise instantiated with $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. we get: $(\sigma_1 t_2,\sigma_2 t_2')\in(\!|\delta\mathrm{int}_r[I']|\!)^{E}_{G',k'}$ (8). Unfold (8), since $k_2\le k'$,we know that $(n,n')\in(\!|\mathrm{int}_r[I']|\!)_{G',k'-k_2}\Rightarrow n=n'=I'\wedge c_1'-c_2'\le\delta D_2$ (f).
Let us assume: $G'=G'$ (g).

STS1: $(H_1,H_2)\models_{G',k'-k''}P\star\gamma\to\beta$. By Lemma 7 and (1),this is proved.

STS2: $(v,v')\in(\!|\Box\tau|\!)_{G',k'-k''}$. from (c),(d). we know $H_1(l)[n]=v$ and $H_2(l')[n']=v'$, based on (e) and (f),unfold (1), we know: $\forall i\le n,(H_1(l)(i),H_2(l')(i))\in(\!|\tau|\!)_{G',k'-1}$. s.t we know that $(H_1(l)[n],H_2(l')[n'])\in(\!|\tau|\!)_{G',k'-1}$. Because $k'-k''\le k'-1$, By Lemma 1, We get: $(v,v')\in(\!|\tau|\!)_{G',k'-k''}$,consider the premise $i\notin\beta$, we know $v=v'$, so that we prove $(v,v')\in(\!|\Box\tau|\!)_{G',k'-k''}$.

STS3: $c-c'\le\delta(D_1+D_2)$, which is proved by (e),(f).

This completes the proof of case read-box.

$$\text{CASE}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma\vdash t_1\ominus t_1'\lesssim D_1:\mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi;\Gamma\vdash t_2\ominus t_2'\lesssim D_2:\mathrm{int}[I']\\ \Sigma;\Delta;\Phi;\Gamma\vdash t_3\ominus t_3'\lesssim D_3:\tau \qquad \Delta;\Phi\models I'\le I \qquad \Sigma;\Delta\vdash P\quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma\vdash \mathsf{updt}\ t_1\ t_2\ t_3\ominus\mathsf{updt}\ t_1'\ t_2'\ t_3'\lesssim 0:\{P\star\gamma\to\beta\}\ \exists\_:\mathrm{unit}_r\ \{P\star\gamma\to\beta\cup\{I'\}\}}^{\mathrm{diff}(D_1+D_2+D_3)}\ \text{R-U}$$

Assume that $\vdash\delta:\Delta$ and $\models\Phi\delta$ and $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G,k}$

TS: $(\delta\sigma_1(\mathsf{updt}\ t_1\ t_2\ t_3),\delta\sigma_2(\mathsf{updt}\ t_1'\ t_2'\ t_3'))\in(\!|\delta(\{\gamma\to\beta\}\ \exists\_.\mathrm{unit}\ \{\gamma\to\beta\cup\{I'\}\})|\!)^{E,0}_{G,k}$

Because updt $t_1\ t_2\ t_3$ is value.

STS: $(\delta\sigma_1(\text{updt } t_1\ t_2\ t_3), \delta\sigma_2(\text{updt } t_1'\ t_2'\ t_3')) \in (\!|\overset{\text{diff}(D_1+D_2)}{\delta(\{\gamma \to \beta\}\ \exists\_.\text{unit}\ \{\gamma \to \beta \cup \{I'\}\})}|\!)_{G,k}$

Unfold the definition of $(\!|\overset{\text{diff}(D_1+D_2)}{\delta(\{\gamma \to \beta\}\ \exists\_.\text{unit}\ \{\gamma \to \beta \cup \{I'\}\})}|\!)_{G,k}$.

Pick $G^{'} \supseteq G$, H_1, H_2,$k' \le k, k'' < k', k'''$. Assume $(H_1, H_2) \vDash_{G',k'} P \star \gamma \to \beta \wedge H_1 = H_{p1} \uplus H_{l1} \wedge H_{l1} = l \to [v,\ldots,v] \wedge H_2 = H_{p2} \uplus H_{l2} \wedge H_{l2} = l' \to [v',\ldots,v'](1)$,$\sigma_1(\text{updt } t_1\ t_2\ t_3); H_1 \Downarrow_f^{k''} (2)$, $\sigma_2(\text{updt } t_1'\ t_2'\ t_3'); H_2 \Downarrow_f^{k'''}$ (3).

Because t_1, t_2, t_3 are sub terms of updt $t_1\ t_2\ t_3$, t_1', t_2', t_3' are sub terms of updt $t_1'\ t_2'\ t_3'$. From (2)(3), we get $\sigma_1 t_1 \Downarrow_f^{k_1} \wedge \sigma_1 t_2 \Downarrow_f^{k_2} \wedge \sigma_1 t_3 \Downarrow_f^{k_3}$ (4), $\sigma_2 t_1' \Downarrow_f^{k_1'} \wedge \sigma_2 t_2' \Downarrow_f^{k_2'} \wedge \sigma_2 t_3' \Downarrow_f^{k_3'}$ (5).

From (4),(5), we get:$\exists l, n, v, k_1, k_2, k_3. \sigma_1 t_1 \Downarrow^{k_1,c_1} l \wedge \sigma_1 t_2 \Downarrow^{k_2,c_1'} n \wedge \sigma_1 t_3 \Downarrow^{k_3,c_1''} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c_1' + c_1'' + c_{update}(a)$. and $\exists l', n', v', k_1', k_2', k_3'. \sigma_2 t_1' \Downarrow^{k_1',c_2} l' \wedge \sigma_2 t_2' \Downarrow^{k_2',c_2'} n' \wedge \sigma_2 t_3' \Downarrow^{k_3',c_2''} v' \wedge c' = c_2 + c_2' + c_2'' + c_{update}(b)$.

From evaluation rule: E-Update-F and (a)(b), we know:

$$\text{updt } t_1\ t_2\ t_3; H_1 \Downarrow^{k'',c} (); H_1(l)[n] \leftarrow v \tag{c}$$

$$\text{updt } t_1'\ t_2'\ t_3'; H_2 \Downarrow^{k''',c'} (); H_2(l')[n'] \leftarrow v' \tag{d}$$

By IH on first premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. We get:

$$(\sigma_1 t_1, \sigma_2 t_1') \in (\!|\text{Array}_\gamma[I]\ \delta\tau|\!)_{G',k'}^{E,\delta D_1} \tag{6}$$

Similarly, By IH on the second premise and third premise, we get: $(\sigma_1 t_2, \sigma_2 t_2') \in (\!|\text{int}_r[I]|\!)_{G',k'}^{E,\delta D_2}$(7), $(\sigma_1 t_3, \sigma_2 t_3') \in (\!|\tau|\!)_{G',k'}^{E,\delta D_3}$(8). From (6),(7),(8), we know:

$$(l, l') \in (\!|\text{Array}_\gamma[I]\ \tau|\!)_{G',k'-k_1} \Rightarrow G'(\gamma) = (l, l', \tau, I) \wedge c_1 - c_2 \le D_1 \tag{e}$$

$$(n, n') \in (\!|\text{int}_r[I]|\!)_{G,k'-k_2} \Rightarrow n = n' = I \wedge c_1' - c_2' \le D_2 \tag{f}$$

$$(v, v') \in (\!|\tau|\!)_{G',k'-k_3} \wedge c_1'' - c_2'' \le D_3 \tag{g}$$

Let us assume: $G'' = G'$(9), $H_1' = H_1(l)[n] \leftarrow v \wedge H_2' = H_2(l')[n'] \leftarrow v'$(10)

STS1: $(H_1', H_2') \vDash_{G',k'-k''} P \star \gamma \to \beta \cup \{I'\}$

Unfold (1) and consider (e), we know: $\forall i \le I.(H_{l1}(l)[i], H_{l2}(l')[i]) \in (\!|\tau|\!)_{G',k'-1}$(11), $\forall i \le I.(H_{l1}(l)[i] \neq H_{l2}(l')[i]) \Rightarrow i \in \beta$(12). Since (e), we need to prove two sub cases.

subgoal 1: $\forall i \le I.(H_{l1}'(l)[i], H_{l2}'(l')[i]) \in (\!|\tau|\!)_{G',k'-k''}$

when i=n, we prove $(H_{l1}'(l)[i], H_{l2}'(l')[i]) \in (\!|\tau|\!)_{G',k'-k_3}$ from (g) and (10), and then use lemma 1.

when $i \neq n$, proved by using lemma 1 on (11)

subgoal 2: $\forall i \le I.(H_{l1}'(l)[i] \neq H_{l2}'(l')[i]) \Rightarrow i \in \beta \cup \{I'\}$

when i=n, $n \in \{I'\}$ from (f) $\Rightarrow n \in \beta \cup \{I'\}$

when $i \neq n$, proved by using lemma 1 on (12)

STS2: $((), ()) \in (\!|\text{unit}_r|\!)_{G',k'-k''}$. It is proved by unfolding the definition.

STS3 $c - c' \le \delta(D_1 + D_2 + D_3)$. It is proved by (e),(f),(g).

This completes the proof of case update.

$$\text{CASE}\ \frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \text{Array}_\gamma[I]\ \tau \qquad \Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \text{int}[I'] \\ \Delta;\Phi;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 : \square\tau \qquad \Delta;\Phi \vDash I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Delta;\Phi;\Gamma \vdash \text{updt } t_1\ t_2\ t_3 \ominus \text{updt } t_1'\ t_2'\ t_3' \lesssim 0 : \overset{\text{diff}(D_1+D_2+D_3)}{\{P \star \gamma \to \beta\}\ \exists\_ : \text{unit}_r\ \{P \star \gamma \to \beta \setminus \{I'\}\}}}\ \text{R-UB}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \Phi\delta$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$

TS: $(\delta\sigma_1(\mathsf{updt}\ t_1\ t_2\ t_3), \delta\sigma_2(\mathsf{updt}\ t_1'\ t_2'\ t_3')) \in (\!|\delta(\{P \star \gamma \to \beta\}\ \exists\_.\mathsf{unit}\ \{P \star \gamma \to \beta \setminus \{I'\}\})|\!)^{E,0}_{G,k}$
Because $\mathsf{updt}\ t_1\ t_2\ t_3$ is value.

STS: $(\delta\sigma_1(\mathsf{updt}\ t_1\ t_2\ t_3), \sigma_2(\mathsf{updt}\ t_1'\ t_2'\ t_3')) \in (\!|\delta(\{P \star \gamma \to \beta\}\ \exists\_.\mathsf{unit}\ \{P \star \gamma \to \beta / \{I'\}\})|\!)_{G,k}$

Unfold the definition of $(\!|\delta(\{P \star \gamma \to \beta\}\ \exists\_.\mathsf{unit}\ \{P \star \gamma \to \beta \setminus \{I'\}\})|\!)_{G,k}$.
Pick $G^{'} \supseteq G$, H_1, H_2, $k' \le k$, $k'' < k'$, k'''. Assume $(H_1, H_2) \vDash_{G',k'} P \star \gamma \to \beta \wedge H_1 = H_{p1} \uplus H_{l1} \wedge H_{l1} = l \to [v, \ldots, v] \wedge H_2 = H_{p2} \uplus H_{l2} \wedge H_{l2} = l' \to [v', \ldots, v']$(1) $\sigma_1(\mathsf{updt}\ t_1\ t_2\ t_3); H_1 \Downarrow^{k''}_f$ (2) $\sigma_2(\mathsf{updt}\ t_1'\ t_2'\ t_3'); H_2 \Downarrow^{k'''}_f$ (3).
Because t_1, t_2, t_3 are sub terms of $\mathsf{updt}\ t_1\ t_2\ t_3$, t_1', t_2', t_3' are sub terms of $\mathsf{updt}\ t_1'\ t_2'\ t_3'$ From (2)(3), we get: $\sigma_1 t_1 \Downarrow^{k_1}_f \wedge \sigma_1 t_2 \Downarrow^{k_2}_f \wedge \sigma_1 t_3 \Downarrow^{k_3}_f$ (4), $\sigma_2 t_1' \Downarrow^{k_1'}_f \wedge \sigma_2 t_2' \Downarrow^{k_2'}_f \wedge \sigma_2 t_3' \Downarrow^{k_3'}_f$ (5).
From (4),(5), we get: $\exists l, n, v, k_1, k_2, k_3. \sigma_1 t_1 \Downarrow^{k_1, c_1} l \wedge \sigma_1 t_2 \Downarrow^{k_2, c_1'} n \wedge \sigma_1 t_3 \Downarrow^{k_3, c_1''} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c_1' + c_1'' + c_{update}(a)$
$\exists l', n', v', k_1', k_2', k_3'. \sigma_2 t_1' \Downarrow^{k_1', c_2} l' \wedge \sigma_2 t_2' \Downarrow^{k_2', c_2'} n' \wedge \sigma_2 t_3' \Downarrow^{k_3', c_2''} v' \wedge c' = c_2 + c_2' + c_2'' + c_{update}(b)$.
From evaluation rule: **E-Update-F** and (a)(b), we know:

$$\mathsf{updt}\ t_1\ t_2\ t_3; H_1 \Downarrow^{k'', c} (); H_1(l)[n] \leftarrow v \tag{c}$$

$$\mathsf{updt}\ t_1'\ t_2'\ t_3'; H_2 \Downarrow^{k''', c'} (); H_2(l')[n'] \leftarrow v' \tag{d}$$

By IH on first premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. We get:

$$(\sigma_1 t_1, \sigma_2 t_1') \in (\!|\mathsf{Array}_\gamma[I]\ \delta\tau|\!)^{E,\delta D_1}_{G',k'} \tag{6}$$

Similarly, By IH on the second premise and third premise, we get

$$(\sigma_1 t_2, \sigma_2 t_2') \in (\!|\delta\mathsf{int}_r[I]|\!)^{E,\delta D_2}_{G',k'} \tag{7}$$

$$(\sigma_1 t_3, \sigma_2 t_3') \in (\!|\Box\delta\tau|\!)^{E,\delta D_3}_{G',k'} \tag{8}$$

From (6),(7),(8), we know:

$$(l, l') \in (\!|\mathsf{Array}_\gamma[I]\ \tau|\!)_{G',k'-k_1} \Rightarrow G'(\gamma) = (l, l', \tau, I) \wedge c_1 - c_2 \le \delta D_1 \tag{e}$$

$$(n, n') \in (\!|\mathsf{int}_r[I]|\!)_{G,k'-k_2} \Rightarrow n = n' = I \wedge c_1' - c_2' \le \delta D_2 \tag{f}$$

$$v = v' \wedge (v, v') \in (\!|\tau|\!)_{G',k'-k_3} c_1'' - c_2'' \le \delta D_3 \tag{g}$$

Let us assume: $G'' = G'(9)$, $H_1' = H_1(l)[n] \leftarrow v \wedge H_2' = H_2(l')[n'] \leftarrow v'(10)$.

STS1: $(H_1', H_2') \vDash_{G',k'-k''} P \star \gamma \to \beta / \{I'\}$
Unfold (1) and consider (e), we know $\forall i \le I.(H_{l1}(l)[i], H_{l2}(l')[i]) \in (\!|\delta\tau|\!)_{G',k'-1}(11)$ $\forall i \le I.(H_{l1}(l)[i] \neq H_{l2}(l')[i]) \Rightarrow i \in \beta(12)$. Since (e), we need to prove two sub cases.

subgoal 1: $\forall i \le I.(H_{l1}'(l)[i], H_{l2}'(l')[i]) \in (\!|\tau|\!)_{G',k'-k''}$
when i=n, we prove $(H_{l1}'(l)[i], H_{l2}'(l')[i]) \in (\!|\tau|\!)_{G',k'-k_3}$ from (g) and (10), and then use lemma 1.
when $i \neq n$, proved by using lemma 1 on (11)

subgoal 2: $\forall i \le I.(H_{l1}'(l)[i] \neq H_{l2}'(l')[i]) \Rightarrow i \in \beta / \{I'\}$
when i=n, $H_1'(l)[i] = H_2'(l)[i]$, it is trivially proved. when $i \neq n$, proved by using lemma 1 on (12)

STS2: $((), ()) \in (\!|\mathsf{unit}_r|\!)_{G',k'-k''}$ It is proved by unfolding the definition.

STS3 $c - c' \le \delta(D_1 + D_2 + D_3)$. It is proved by (e),(f),(g).

This completes the proof of case update-box.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Gamma\vdash t_1\ominus t_2\lesssim D:\tau}{\Sigma;\Delta;\Phi;\Gamma\vdash \mathsf{return}\ t_1\ominus \mathsf{return}\ t_2\lesssim 0:\{P\}\overset{\mathrm{diff}(D)}{\exists\_.\tau}\{P\}}\ \text{R-T}$$

Assume that $\vdash\delta:\Delta$ and $\vDash\Phi\delta$ and $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G,k}$ TS: $(\delta\sigma_1(\mathsf{return}\ t_1),\sigma_2(\mathsf{return}\ t_2))\in(\!|\delta\ \{P\}\overset{\mathrm{diff}(D)}{\exists\_.\tau}\{P\}|\!)^{E,0}_{G,k}$

Because return t is value, STS: $(\delta\sigma_1(\mathsf{return}\ t_1),\sigma_2(\mathsf{return}\ t_2))\in(\!|\delta\ \{P\}\overset{\mathrm{diff}(D)}{\exists\_.\tau}\{P\}|\!)_{G,k}$

Unfold the definition of $(\!|\delta\ \{P\}\overset{\mathrm{diff}(D)}{\exists\_.\tau}\{P\}|\!)_{G,k}$. Pick $G^{'}\supseteq G$, H_1, H_2,$k'\le k$,$k''<k'$,k'''. Assume $(H_1,H_2)\vDash_{G',k'}$ P (1), $\sigma_1(\mathsf{return}\ t_1);H_1\Downarrow_f^{k''}$ (2), $\sigma_2(\mathsf{return}\ t_2);H_2\Downarrow_f^{k'''}$ (3).

Because t_1 is the subterm of return t_1 . From (2)(3), We know: $\sigma_1 t_1\Downarrow^{k_1}$ (4), $\sigma_2 t_2\Downarrow_f^{k'''}$ (5).
From (4),(5), we know:

$$\exists v_1,k_1,\sigma_1 t_1\Downarrow^{k_1,c_1} v_1\wedge k''=k_1+1\wedge c=c_1+c_{ret} \tag{a}$$

$$\exists v_2,k_1',\sigma_2 t_2\Downarrow^{k_1',c_2} v_2\wedge k'''=k_1'+1\wedge c'=c_2+c_{ret} \tag{b}$$

From (a),(b) and the evaluation rule **R-ret**: $H_1;\mathsf{return}\ t_1\Downarrow_f^{k'',c} v_1;H_1(c)$, $H_2;\mathsf{return}\ t_2\Downarrow_f^{k''',c'} v_2;H_2(d)$.
By IH on the premise instantiated with $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G',k'}$ by lemma 1. We get: $(\sigma_1 t_1,\sigma_2 t_2)\in(\!|\delta\tau|\!)^{E,\delta D}_{G',k'}$(6).
Unfold (6), we get: $(v_1,v_2)\in(\!|\delta\tau|\!)_{G',k'-k_1}$(7).
Let us assume: $G''=G'$(8), $H_1'=H_1'\wedge H_2'=H_2'\leftarrow v'$(9).

STS1: $(H_1',H_2')\vDash_{G',k'-k''}P$. It is trivially true.

STS2: $\big(v_1,v_2\big)\in(\!|\tau|\!)_{G',k'-k''}$. It is proved by (7) using Lemma 1.

STS3: $c-c'\le\delta D$ which is proved from (a),(b).

This completes the proof.

$$\text{CASE}\ \frac{\begin{array}{c}P=P_1\star P_2\qquad \Delta;\Phi;\Gamma\vdash t_1\ominus t_2\lesssim D_1:\{P_1\}\overset{\mathrm{diff}(D)}{\exists\vec{\gamma}_1.\tau_1}\{Q_1\star Q_2\}\\ \Delta,\vec{\gamma}_1:\vec{\mathbb{L}};\Phi;\Gamma,x:\tau_1\vdash t_1'\ominus t_2'\lesssim D_2:\{Q_1\star P_2\}\overset{\mathrm{diff}(D')}{\exists\vec{\gamma}_2.\tau_2}\{Q\}\end{array}}{\Delta;\Phi;\Gamma\vdash \mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1'\ominus\mathsf{let}\ \{x\}=t_2\ \mathsf{in}\ t_2'\lesssim 0:\{P\}\overset{\mathrm{diff}(D_1+D_2+D+D')}{\exists\vec{\gamma}_1\vec{\gamma}_2:\tau_2}\{Q\star Q_2\}}\ \text{R-BIND}$$

Assume that $\vdash\delta:\Delta$ and $\vDash\delta\Phi$ and $(\sigma_1,\sigma_2)\in(\!|\delta\Gamma|\!)_{G,k}$
TS: $(\delta\sigma_1(\mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1'),\sigma_2(\mathsf{let}\ \{x\}=t_2\ \mathsf{in}\ t_2'))\in(\!|\delta\ \{P\}\overset{\mathrm{diff}(D+D'+D_1+D_2)}{\exists\vec{\gamma}_1\vec{\gamma}_2:\tau_2}\{Q\star Q_2\}|\!)^{E,0}_{G,k}$
Because $\mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1'$ is value.
STS: $(\delta\sigma_1(\mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1'),\sigma_2(\mathsf{let}\ \{x\}=t_2\ \mathsf{in}\ t_2'))\in(\!|\delta\ \{P\}\overset{\mathrm{diff}(D+D'+D_1+D_2)}{\exists\vec{\gamma}_1\vec{\gamma}_2:\tau_2}\{Q\star Q_2\}|\!)_{G,k}$
Unfold the definition of $(\!|\delta\ \{P\}\overset{\mathrm{diff}(D+D'+D_1+D_2)}{\exists\vec{\gamma}_1\vec{\gamma}_2:\tau_2}\{Q\star Q_2\}|\!)_{G,k}$.
Pick $G^{'}\supseteq G$, H_1, H_2,$k'\le k$,$k''<k'$,k'''. Assume $(H_1,H_2)\vDash_{G',k'}$ P(1), $\sigma_1(\mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1');H_1\Downarrow_f^{k''}$ (2), $\sigma_2(\mathsf{let}\ \{x\}=t_2\ \mathsf{in}\ t_2');H_2\Downarrow_f^{k'''}$ (3).
Because t_1 is the subterm of $\mathsf{let}\ \{x\}=t_1\ \mathsf{in}\ t_1'$. From (2) ,we know: $\exists v_1,v_1',v_1'',v_1''',k_1,k_2,k_3,k_4,H_1',H_1''.\sigma_1 t_1\Downarrow^{k_1,c_1}$ $v_1\wedge v_1;H_1\Downarrow_f^{k_2,c_1'}v_1';H_1'\sigma_1 t_1'[v_1'/x]\Downarrow^{k_3,c_1''}v_1''\wedge v_1'';H_1'\Downarrow_f^{k_4,c_1'''}v_1''';H_1''\wedge k''=k_1+k_2+k_3+k_4+1\wedge c=c_1+c_1'+$ $c_1''+c_1'''+c_{let}(a)$.
Similarly, from (3) , we get: $\exists v_2,v_2',v_2'',v_2''',k_1',k_2',k_3',k_4',H_2',H_2''.\quad\sigma_1 t_2\Downarrow^{k_1',c_2}v_2$ and $v_2;H_2\Downarrow_f^{k_2',c_2'}v_2';H_2'$ and

$\sigma_1 t_2'[v_2'/x] \Downarrow^{k_3', c_2''} v_2''$ and $v_2''; H_2' \Downarrow_f^{k_4', c_2'''} v_2'''; H_2'' \wedge k'' = k_1' + k_2' + k_3' + k_4' + 1 \wedge c' = c_2 + c_2' + c_2'' + c_2''' + c_{let}(b)$.
From evaluation rule: E-Let-F and (a)(b), we know:

$$\exists H_a, H_c, H_a', H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v1; H_a \Downarrow_f v_1'; H_a' \wedge H_a' = H1_{Q1} \uplus H1_{Q2} \\ \wedge v_1''; H1_{Q1} \uplus H_c \Downarrow_f v_1'''; H1_Q \wedge H_1'' = H1_Q \uplus H1_{Q2} \quad (c)$$

$$\exists H_a, H_c, H_a', H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v1; H_a \Downarrow_f v_1'; H_a' \wedge H_a' = H1_{Q1} \uplus H1_{Q2} \\ \wedge v_1''; H1_{Q1} \uplus H_c \Downarrow_f v_1'''; H1_Q \wedge H_1'' = H1_Q \uplus H1_{Q2} \quad (d)$$

From (1) : $(H_1, H_2) \vDash_{G',k'} P_1 \star P_2$, we assume that

$$\exists G_a, G_b : (H_1 = H_a \uplus H_c) \quad \text{and} \quad (H_2 = H_b \uplus H_d) \\ (G' = G_a \uplus G_c) \quad \text{and} \quad (H_a, H_b) \vDash_{G_a,k'} P_1 \quad \text{and} \quad (H_c, H_d) \vDash_{G_c,k'} P_2 \quad (*)$$

STS1: $(H_1'', H_2'') \vDash_{G'',k'-k''} Q \star Q_2$. STS2: $(v_1''', v_2''') \in (\!|\tau_2|\!)_{G'',k'-k''}$. STS3: $c - c' \leq \delta(D_1 + D_2 + D + D')$.
By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G',k'}$, we know:$(\sigma_1 t_1, \sigma_2 t_2) \in (\!|\{P_1\}\ \exists\vec{\gamma}_1.\tau\ \{Q_1 \star Q_2\}|\!)^{E,D_1}_{G',k'}$(4).
Unfold the definition of $(\!|\{P_1\}\ \exists\vec{\gamma}_1.\tau\ \{Q_1 \star Q_2\}|\!)^{E,D_1}_{G',k'}$,use (a), we get: $(v_1, v_2) \in (\!|\{P_1\}\ \exists\vec{\gamma}_1.\tau_1\ \{Q_1 \star Q_2\}|\!)_{G',k'-k_1} \wedge$
$c_1 - c_2 \leq D_1(e)$.
To unfold (e), we choose: $k_f \leq k' - k_1$ and $k_f' < k_f$ and $G_f \supseteq G'$ and $H_a \uplus H_c$,$H_b \uplus H_d$ and (1),(a),(b). We have $(H_a, H_b) \vDash_{G_f,k_f} P_1 \wedge v_1; H_a \Downarrow_f \wedge v_2; H_b \Downarrow_f$.
Unfold the definition, we get: $v_1; H_a \Downarrow_f^{c_1'} v_1'; H_a' \wedge v_2; H_b \Downarrow_f^{c_2'} v_2'; H_b' \wedge H_a', H_b' \vDash Q_1 \star Q_2$.

$$(v_1', v_2') \in (\!|\tau_1|\!)_{G',k'-k_1} \wedge c_1' - c_2' \leq \delta D \quad (k)$$

By IH on the third premise, we know: $(\sigma_1 v_1'', \sigma_2 v_2'') \in (\!|\{Q_1 \star P_2\}\ \exists\vec{\gamma}_1.\tau\ \{Q\}|\!)^{E,D_2}_{G',k'}$(5).
Unfold the definition of $(\!|\{Q_1 \star P_2\}\ \exists\vec{\gamma}_1.\tau_2\ \{Q\}|\!)^{E,D_2}_{G',k'}$,use (a), we get:
$(v_1, v_2) \in (\!|\{Q_1 \star P_2\}\ \exists\vec{\gamma}_1.\tau_2\ \{Q\}|\!)_{G',k'-k_1} \wedge c_1'' - c_2'' \leq D_2(f)$ We have $(H_a', H_b') \vDash_{G_f,k_f} Q_1 \star Q_2 \Rightarrow H1_{Q1}, H2_{Q1} \vDash$
$Q_1 \wedge H1_{Q2}, H2_{Q2} \vDash Q_2 \wedge H_c, H_d \vDash P_2,\ H1_{Q1} \uplus H_c; v_1''' \Downarrow_f \wedge H2_{Q1} \uplus H_d; v_2''' \Downarrow_f$.// Unfold the definition of (f), we get: $H1_{Q1} \uplus H_c; v_1''' \Downarrow_f^{c_1'''} H1_Q; v_1'''' \wedge H2_{Q1} \uplus H_d; v_2'''' \Downarrow_f^{c_2'''} H2_Q; v_2'''' \wedge H1_Q, H2_Q \vDash Q$.

$$(v_1''', v_2''') \in (\!|\tau_2|\!)_{G',k'-k_1} \wedge c_1''' - c_2''' \leq \delta D' \quad (g)$$

STS1: Use Lemma 2, we know that $H1_Q \uplus H1_{Q2}, H2_Q \uplus H2_{Q2} \vDash Q \star Q_2$.
STS2: it is proved from (g).
STS3: it is proved from (e),(k),(f),(g).
This completes the proof of let case.

$$\text{CASE}\ \frac{\Delta; \Phi; |\Gamma|_1 \vdash_{L_1}^{U_1} t_1 : A_1 \qquad \Delta; \Phi; |\Gamma|_2 \vdash_{L_2}^{U_2} t_2 : A_2}{\Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim U_1 - L_2 : U(A_1, A_2)}\ \text{R-SWITCH}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$. TS: $(\sigma_1(t_1), \sigma_2(t_2)) \in (\!|\delta U(A_1, A_2)|\!)^{E,(\delta(U_1 - L_2)}_{G,k,\delta(U_1-L_2)}$
Unfold the definition. Assume: $\sigma_1 t_1 \Downarrow^{k_1,c} v_1(a)$, $\sigma_2 t_2 \Downarrow^{k_2,c'} v_2(b)$, $k_1 < k(c)$.
TS 1: $(v_1, v_2) \in (\!|U(\delta A_1, \delta A_2)|\!)_{G,k-k_1}$. TS 2: $c - c' \leq \delta(U_1 - L_2)$
We prove TS 1 first. Unfold its definition, STS: $\forall k'. v_1 \in (\!|\delta A_1|\!)_{G|_1,k'}$ and $v_2 \in (\!|\delta A_2|\!)_{G|_2,k'}$
Pick k'. By IH 2 on the first premise using $FV(t_1) \subseteq dom(\delta|\Gamma|_1)$ using Lemma 5. $\vDash \delta\Phi, \forall k. \sigma_1 \in (\!|\delta|\Gamma|_1|\!)_{G|_1,k}$.
We get: $\forall k. \delta\sigma_1 t_1 \in [\![\delta A_1]\!]^{E,(\delta L_1, \delta U_1)}_{G|_1,k}$(1). Unfold the definition of (1), choose $k = k_1 + k'$. we know $v_1 \in$
$[\![\delta A_1]\!]_{G|_1,k'} \wedge \delta L_1 \leq c \leq \delta U_1(d)$.
By IH 2 on the second premise using $\forall k. \sigma_2 \in (\!|\delta|\Gamma|_2|\!)_{G|_2,k}$. We get: $\forall k. \delta\sigma_2 t_2 \in [\![\delta A_2]\!]^{E,(\delta L_2, \delta U_2)}_{G|_2,k}$ (2). Unfold

the definition of (1), choose $k = k_2 + k'$. we know $v_2 \in [\![\delta A_2]\!]_{G|_2,k'} \wedge \delta L_2 \le c' \le \delta U_2(e)$.
(d),(e) finished the proof of STS1. (d),(e) finished the proof of STS2.
This completes the proof of switch case.

$$\text{CASE}\ \frac{\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2 \qquad \Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : U(A_1, A_2')}{\Delta;\Phi;\Gamma \vdash t_1\ t_2 \ominus t_2' \lesssim U_1 + U + D_2 + c_{app} : U(A_2, A_2')}\ \text{R-APP-E}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\sigma_1(t_1\ t_2), \sigma_2(t_2')) \in (\!|\delta U\,(A_2, A_2')|\!)^{E,\delta(U_1+U+D_2+c_{app})}_{G,k}$.
Following the definition of $(\!|\delta U\,(A_2, A_2')|\!)^{E,\delta(U_1+U+D_2+c_{app})}_{G,k}$, assume that:

$$\frac{\sigma_1 t_1 \Downarrow^{c_1,k_1} \text{fix} f\,x.t'\ (\star) \qquad \sigma_1 t_2 \Downarrow^{c_2,k_2} v_2\ (\diamond) \qquad t'[\text{fix} f\,x.t'/f][v_2/x] \Downarrow^{c_3,k_3} v_r\ (\spadesuit)}{(\sigma_1 t_1)\,(\sigma_1 t_2) \Downarrow^{c_1+c_2+c_3+c_{\text{app}},k_1+k_2+k_3+1} v_r}\ \text{E-F}$$

and $\sigma_2(t_2') \Downarrow^{c',k'} v'\ (\diamond\diamond)$ and $k_1 + k_2 + k_3 + 1 < k$.
TS1: $c_1 + c_2 + c_3 + c_{\text{app}} - c' \le \delta(U_1 + U + D_2 + c_{app})$. TS2: $(v_r, v') \in (\!|\delta U\,(A_2, A_2')|\!)_{G,k-(k_1+k_2+k_3+1)}$.
Frist show the second statement. By unrolling the definition of $(\!|\delta U\,(A_2, A_2')|\!)_{G,k-(k_1+k_2+k_3+1)}$, STS: $\forall j.v_r \in [\![\delta A_2]\!]_{|G|_1,j} \wedge v' \in [\![\delta A_2']\!]_{|G|_2,j}$.
Pick j. By IH 2 on the first premise using $FV(t_1) \subseteq dom(\delta|\Gamma|_1)$ using Lemma 5. $\vDash \delta\Phi$, $\forall k.\sigma_1 \in (\!|\delta|\Gamma|_1|\!)_{G|_1,k}$.
we get $\forall k.\sigma_1 t_1 \in [\![\delta(A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2)]\!]^{E,(\delta L_1,\delta U_1)}_{G|_1,k}$ (1).

Instantiating (1) with $X = j + k_1 + k_3 + 1$, we get $\sigma_1 t_1 \in [\![\delta(A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2)]\!]^{E,(\delta L_1,\delta U_1)}_{G|_1,X}$ (2).
Following the definition, we get: $\sigma_1 t_1 \Downarrow^{c_1,k_1} \text{fix} f\,x.t'$ and $\text{fix} f\,x.t' \in [\![\delta(A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2)]\!]_{G|_1,X-k_1}$ (a) and $\delta L_1 < c_1 < \delta U_1$ (e).
By IH on the second premise, we get: $(\sigma_1 t_2, \sigma_2 t_2') \in (\!|\delta U\,(A_1, A_2')|\!)^{E,\delta(D_2)}_{G,k}$ Unfold the defintion, by $(\diamond)$ and $(\diamond\diamond)$, we get : $c_2 - c' \le \delta D_2$ (f) and $(v_2, v') \in (\!|\delta U\,(A_1, A_2')|\!)_{G,k-k_2}$ (b).
From (b), we know : $\forall k.v_2 \in [\![\delta A_1]\!]_{G|_1,k} \wedge v' \in [\![\delta A_2]\!]_{G|_2,k}$. Instantiate k with $X - k_1 - 1$, we get: $v_2 \in [\![\delta A_1]\!]_{G|_1,X-k_1-1}$ (3), $v' \in [\![\delta A_2]\!]_{G|_2,X-k_1-1}$ (4).
Unfold (a), we get: $t'[\text{fix} f\,x.t'/f][v_2/x] \in [\![\delta A_2]\!]^{E,\delta(L,U)}_{G|_1,X-k_1-1}$ (c).
Unfold (c) using $(\spadesuit)$, we get : $v_r \in [\![\delta A_2]\!]_{G|_1,X-k_1-k_3-1}$ (h), $\delta L \le c_3 \le \delta U$ (g).
STS1 is proved by $(e),(f),(g)$. STS2 is proved by (h) and (3) using Lemma 1.

$$\text{CASE}\ \frac{\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t_1' : A_1' \overset{\text{exec}(L,U)}{\longrightarrow} A_2' \qquad \Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : U(A_2, A_1')}{\Delta;\Phi;\Gamma \vdash t_2 \ominus t_1'\ t_2' \lesssim D_2 - L_1 - L - c_{app} : U(A_2, A_2')}\ \text{R-E-APP}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\delta\sigma_1(t_2), \sigma_2(t_1'\ t_2')) \in (\!|\delta U\,(A_2, A_2')|\!)^{E,\delta(D_2-L_1-L-c_{app})}_{G,k}$.
Following the definition of $(\!|\delta U\,(A_2, A_2')|\!)^{E,\delta(D_2-L_1-L-c_{app})}_{G,k}$, assume that:

$$\frac{\sigma_2 t_1' \Downarrow^{c_1',k_1'} \text{fix} f\,x.t'\ (\star) \qquad \sigma_2 t_2' \Downarrow^{c_2',k_2'} v_2'\ (\diamond) \qquad t'[\text{fix} f\,x.t'/f][v_2'/x] \Downarrow^{c_3',k_3'} v_r'\ (\spadesuit)}{(\sigma_2 t_1')\,(\sigma_2 t_2') \Downarrow^{c_1'+c_2'+c_3'+c_{\text{app}},k_1'+k_2'+k_3'+1} v_r'}\ \text{E-F}$$

and $\sigma_1 t_2 \Downarrow^{c_1,k_1} v_r\ (\diamond\diamond)$ and $k_1 < k$.
TS1: $c_1 - (c_1' + c_2' + c_3' + c_{\text{app}}) \le \delta(D_2 - U_1 - U - c_{app})$.
TS2: $(v_r, v_r') \in (\!|\delta U\,(A_2, A_2')|\!)_{G,k-k_1}$.
Frist show the second statement. By unrolling the definition of $(\!|\delta U\,(A_2, A_2')|\!)_{G,k-k_1}$, STS: $\forall j.v_r \in [\![\delta A_2]\!]_{|G|_1,j} \wedge v_r' \in [\![\delta A_2']\!]_{|G|_2,j}$.
Pick j. By IH 2 on the first premise using $FV(t_1') \subseteq dom(\delta|\Gamma|_2)$ using Lemma 5. $\vDash \delta\Phi$, $\forall k.\sigma_2 \in (\!|\delta|\Gamma|_2|\!)_{G|_2,k}$.
we get $\forall k.\sigma_2 t_1' \in [\![\delta(A_1' \overset{\text{exec}(L,U)}{\longrightarrow} A_2')]\!]^{E,(\delta L_1,\delta U_1)}_{G|_2,k}$ (1). Instantiating (1) with $X = j + k_1' + k_3' + 1$, we get $\sigma_2 t_1' \in$

$[\![\delta(A_1' \xrightarrow{\text{exec}(L,U)} A_2')]\!]^{E,(\delta L_1,\delta U_1)}_{G|_2,X}$ (2).
Following the definition, we get:
$\sigma_2 t_1' \Downarrow^{c_1',k_1'} \text{fix} f\, x.t'$ and $\text{fix} f\, x.t' \in [\![\delta(A_1' \xrightarrow{\text{exec}(L,U)} A_2')]\!]_{G|_2,X-k_1'}$ (a) and $\delta L_1 < c_1' < \delta U_1$ (e).
By IH on the second premise, we get: $(\sigma_1 t_2, \sigma_2 t_2') \in (\!|\delta U(A_2, A_1')|\!)^{E,\delta(D_2)}_{G,k}$ Unfold the defintion, by $(\diamond)$ and $(\diamond\diamond)$, we get :
$c_1 - c_2' \le \delta D_2$ (f) and $(v_r, v_2') \in (\!|\delta U(A_2, A_1')|\!)_{G,k-k_1}$ (b). From (b), we know : $\forall k. v_r \in [\![\delta A_2]\!]_{G|_1,k} \wedge v_2' \in [\![\delta A_1']\!]_{G|_2,k}$.
Instantiate k with $X - k_1' - 1$, we get: $v_r \in [\![\delta A_2]\!]_{G|_1,X-k_1'-1}$ (3), $v_2' \in [\![\delta A_1']\!]_{G|_2,X-k_1'-1}$ (4).
Unfold (a), we get: $t'[\text{fix} f\, x.t'/f][v_2'/x] \in [\![\delta A_2']\!]^{E,\delta(L,U)}_{G|_2,X-k_1'-1}$ (c).
Unfold (c) using $(\spadesuit)$, we get : $v_r' \in [\![\delta A_2']\!]_{G|_2,X-k_1'-k_3'-1}$ (h), $\delta L \le c_3' \le \delta U$ (g).
STS1 is proved by $(e), (f), (g)$. STS2 is proved by (h) and (4) using Lemma 1.

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : A_1 \qquad \Sigma;\Delta;\Phi;\Gamma, x : U(A_1,A_1) \vdash t_2 \ominus t_2' \lesssim D_2 : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash \text{let } x = t_1 \text{ in } t_2 \ominus t_2' \lesssim U_1 + D_2 + c_{lt} : \tau}\ \text{R-LT-E}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\delta\sigma_1(\text{let } x = t_1 \text{ in } t_2), \sigma_2(t_2')) \in (\!|\delta\tau|\!)^{E,\delta(D_2+U_1+c_{lt})}_{G,k}$.
Following the definition of $(\!|\delta\tau|\!)^{E,\delta(D_2+U_1+c_{lt})}_{G,k}$, assume that:

$$\frac{\sigma_1 t_1 \Downarrow^{c_1,k_1} v_1\ (\star) \qquad \sigma_1 t_2[v_1/x] \Downarrow^{c_r,k_r} v_r\ (\spadesuit)}{\text{let } x = \sigma_1 t_1 \text{ in } \sigma_1 t_2 \Downarrow^{c_1+c_r+c_{\text{lt}},k_1+k_r+1} v_r}\ \text{E-LET}$$

and $\sigma_2 t_2' \Downarrow^{c',k'} v'$ $(\diamond\diamond)$. STS1: $(v_r, v') \in (\!|\delta\tau|\!)_{G,k-(k_1+k_r+1)}$. STS2: $c_1 + c_r + c_{lt} - c' \le \delta(D_2 + U_1 + c_{lt}$.
To prove : $\forall k. v_1 \in [\![\delta A_1]\!]_{|G|_1,k})$.

Proof. Pick k. IH2 on the first premise.

1. $FV(t_1) \subseteq |\delta\Gamma|_1$ using Lemma 5.
2. $\sigma_1 \in (\!||\delta\Gamma|_1|\!)_{|G|_1,k+\delta U_1+1}$ by Lemma 6 using assumptions.

We get: $\sigma_1 t_1 \in [\![\delta A_1]\!]^{E,(\delta L_1,\delta U_1)}_{|G|_1,k+\delta U_1+1}$. Unfold the defintion, we know: $\delta L_1 \le c_1 \le \delta U_1$ (a) and $v_1 \in [\![\delta A_1]\!]_{|G|_1,k+\delta u_1+1-c_1}$.
So, we know: $v_1 \in [\![\delta A_1]\!]_{|G|_1,k}$ by Lemma 1. ■

By IH on the second premise using

1. $(\sigma_1[v_1/x], \sigma_2[v_1/x]) \in (\!|\delta\Gamma, x : U(\delta A_1, \delta A_1)|\!)_{G,k}$ using
   1.1. $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
   1.2. $(v_1, v_1) \in (\!|U(\delta A_1, \delta A_1)|\!)_{G,k}$

we get: $(\sigma_1[v_1/x]t_2, \sigma_2[v_1/x]t_2') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$.
Since x does not occur free in t_2', we have : $(\sigma_1[v_1/x]t_2, \sigma_2 t_2') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$. Unfold the defintion, we have:
$c_r - c_1' \le \delta D_2$ (b), $(v_r, v') \in (\!|\delta\tau|\!)_{G,k-k_r}$ (c).
STS1 is proved by using Lemma 1 on (c) because $k - (k_1 + k_r + 1) < k - k_r$. STS2 is proved by (a) and (b).
This completes the proof of case R-lt-e.

$$\text{CASE}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t : A_1 + A_2 \\ \Sigma;\Delta;\Phi;\Gamma, x : U(A_1,A_1) \vdash t_1 \ominus t' \lesssim D_2 : \tau \qquad \Sigma;\Delta;\Phi;\Gamma, y : U(A_2,A_2) \vdash t_2 \ominus t' \lesssim D_2 : \tau\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \text{ case } (t, x.t_1, y.t_2) \ominus t' \lesssim U_1 + D_2 + c_{case} : \tau}\ \text{R-CASE-E}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\delta\sigma_1(\text{ case } (t, x.t_1, y.t_2)), \sigma_2(t')) \in (\!|\delta\tau|\!)^{E,\delta(D_2+U_1+c_{case})}_{G,k}$.
Following the definition of $(\!|\delta\tau|\!)^{E,\delta(D_2+U_1+c_{case})}_{G,k}$, assume that: σ_1 case $(t, x.t_1, y.t_2) \Downarrow^{C,K} v_r$ and $\sigma_2 t' \Downarrow^{c',k'} v'$ $(\diamond\diamond)$ and $C < k$.
STS1: $(v_r, v') \in (\!|\delta\tau|\!)_{G,k-K}$. STS2: $C - c' \leq \delta(D_2 + U_1 + c_{case})$.

To prove : $\forall k. v_1 \in [\![\delta A_1]\!]_{|G|_1,k}$. Depending on what $\sigma_1 t$ evaluates to, there are two cases.

Subcase1:

$$\frac{\sigma_1 t \Downarrow^{c_1,k_1} \mathsf{inl} v_1 \ (\star) \qquad \sigma_1 t_1[v_1/x] \Downarrow^{c_r,k_r} v_r \ (\spadesuit)}{\sigma_1 \text{ case } (t, x.t_1, y.t_2) \Downarrow^{c_1+c_r+c_{\text{case}},k_1+k_r+1} v_r} \text{CASE-INL}$$

To prove: $\forall k.(\mathsf{inl} v_1) \in (\!|\delta A_1 + \delta A_2|\!)_{G,k}$.

Proof. Pick k. IH2 on the first premise.

1. $FV(t) \subseteq |\delta\Gamma|_1$ using Lemma 5.
2. $\sigma_1 \in (\!||\delta\Gamma|_1|\!)_{|G|_1,k+\delta U_1+1}$ by Lemma 6 using assumptions.

We get: $\sigma_1 t \in [\![\delta A_1 + \delta A_2]\!]^{E,(\delta L_1,\delta U_1)}_{|G|_1,k+\delta U_1+1}$. Unfold the defintion, we know: $\sigma_1 t \Downarrow^{c_1,k_1} \mathsf{inl} v_1$ and $\delta L_1 \leq c_1 \leq \delta U_1$ (a) and $\mathsf{inl} v_1 \in [\![\delta A_1 + \delta A_2]\!]_{|G|_1,k+\delta u_1+1-c_1}$.
So, we know: $\mathsf{inl} v_1 \in [\![\delta A_1 + \delta A_2]\!]_{|G|_1,k}$ by Lemma 1. ■

By IH on the second premise using

1. $(\sigma_1[v_1/x], \sigma_2[v_1/x]) \in (\!|\delta\Gamma, x : U(\delta A_1, \delta A_1)|\!)_{G,k}$ using
   1.1. $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
   1.2. $(v_1, v_1) \in (\!|U(\delta A_1, \delta A_1)|\!)_{G,k}$

we get: $(\sigma_1[v_1/x]t_1, \sigma_2[v_1/x]t') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$.
Since x does not occur free in t', we have : $(\sigma_1[v_1/x]t_1, \sigma_2 t') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$.
Unfold its defintion, we get: $c_r - c_1' \leq \delta D_2$ (b). $(v_r, v') \in (\!|\delta\tau|\!)_{G,k-k_r}$ (c).
STS1 is proved by using Lemma 1 on (c) because $k - (k_1 + k_r + 1) < k - k_r$. STS2 is proved by (a) and (b).

Subcase 2:

$$\frac{\sigma_1 t \Downarrow^{c_1,k_1} \mathsf{inr} v_1 \ (\star) \qquad \sigma_1 t_2[v_1/x] \Downarrow^{c_r,k_r} v_r \ (\spadesuit)}{\sigma_1 \text{ case } (t, x.t_1, y.t_2) \Downarrow^{c_1+c_r+c_{\text{case}},k_1+k_r+1} v_r} \text{CASE-INR}$$

Prove similarily : $\forall k.(\mathsf{inr} v_1) \in (\!|\delta A_1 + \delta A_2|\!)_{G,k}$.
IH on the third premise using

1. $(\sigma_1[v_1/x], \sigma_2[v_1/x]) \in (\!|\delta\Gamma, x : U(\delta A_1, \delta A_1)|\!)_{G,k}$ using
   1.1. $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
   1.2. $(v_1, v_1) \in (\!|U(\delta A_1, \delta A_1)|\!)_{G,k}$

we get: $(\sigma_1[v_1/x]t_2, \sigma_2[v_1/x]t') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$. Since x does not occur free in t', we have : $(\sigma_1[v_1/x]t_2, \sigma_2 t') \in (\!|\delta\tau|\!)^{E,\delta D_2}_{G,k}$.
Unfold its defintion, we get: $c_r - c_1' \leq \delta D_2$ (b) ,$(v_r, v') \in (\!|\delta\tau|\!)_{G,k-k_r}$ (c).
STS1 is proved by using Lemma 1 on (c) because $k - (k_1 + k_r + 1) < k - k_r$. STS2 is proved by (a) and (b).

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : \{P_1\}\ \overset{\text{exec}(L,U)}{\exists\vec{\gamma}_1} : A_1\ \{Q_1\} \qquad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_2}_{L_2} t_2' : \{P_2\}\ \overset{\text{exec}(L',U')}{\exists\vec{\gamma}_1} : A_1'\ \{Q_2\}}{\Sigma;\Delta;\Phi;\Gamma, x : U(A_1,A_1) \vdash t_2 \ominus t_2' \lesssim D_2 : \{P \sqcup P_1\}\ \overset{\text{diff}(D')}{\exists\vec{\gamma}_1}.\tau\ \{Q\} \qquad dom(P) = dom(P_1)}\ \text{R-BIND-E}$$
$$\Sigma;\Delta;\Phi;\Gamma \vdash \text{let}\ \{x\} = t_1\ \text{in}\ t_2 \ominus t_2' \lesssim -L_2 : \overset{\text{diff}(U_1+U+(D_2+U_2)+D'+c_{let})}{\{P\}\ \exists\vec{\gamma}_1.\tau\ \{Q\}}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and $(\sigma_1,\sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$
TS: $(\sigma_1(\text{let}\ \{x\} = t_1\ \text{in}\ t_2), \sigma_2(t_2')) \in (\!|\delta(\{P\}\ \overset{\text{diff}(-L_2)}{\exists\vec{\gamma}_1}.\tau\ \{Q\})|\!)^{E,\delta(U_1+U+(D_2+U_2)+D'+c_{let})}_{G,k}$.
Assume $\text{let}\ \{x\} = t_1\ \text{in}\ t_2 \Downarrow^{0,0} \text{let}\ \{x\} = t_1\ \text{in}\ t_2$ and $\sigma_2 t_2' \Downarrow^{c_1',k_1'} v_1'$.
Unfold the definition, STS1: $0 - c_1' \le -(\delta L_2)$. By IH2 on the second premise, we get: $\delta L_2 \le c_1' \le \delta U_2$, which proves STS1.
STS2: $(\delta\sigma_1(\text{let}\ \{x\} = t_1\ \text{in}\ t_2), v_1') \in (\!|\delta(\overset{\text{diff}(U_1+U+(D_2+U_2)+D'+c_{let})}{\{P\}\ \exists\vec{\gamma}_1.\tau\ \{Q\}})|\!)_{G,k}$.
Unfold its defintion, assume : $G \supseteq G, k' < k, k_2 < k'.H_1, H_1' \vDash_{G,k'} P$ and $\text{let}\ \{x\} = \sigma_1 t_1\ \text{in}\ \sigma_1 t_2; H_1 \Downarrow^{c_2,k_2}_f v_{let}; H_2$ (a) and $v_1'; H_1' \Downarrow^{c_2',k_2'}_f v_2'; H_2'$ (b).
STS1: $(H_2, H_2') \vDash_{G,k'-k_2} Q$. STS2: $(v_{let}, v_2') \in (\!|\delta\tau|\!)_{G,k'-k_2}$. STS3: $c_2 - c_2' \le \delta((D_2+U_2)+D'+U+U_1+c_{let})$.
By IH2 on the first premise, we get: $\sigma_1 t_1 \in (\!|\delta(\{P_1\}\ \overset{\text{exec}(L,U)}{\exists\vec{\gamma}_1} : A_1\ \{Q_1\})|\!)^{E,\delta(L_1,U_1)}_{|G|_1,k}$ (c).
Unfold (c), we assume $\sigma_1 t_1 \Downarrow^{c_a,k_a} v$, we get : $v \in (\!|\delta(\{P_1\}\ \overset{\text{exec}(L,U)}{\exists\vec{\gamma}_1} : A_1\ \{Q_1\})|\!)_{|G|_1,k-k_a}$ (d) and $\delta L_1 \le c_a \le \delta U_1$ (e). Unfold (d), pick $k_1 \le k - k_a$. assume $H_1 \vDash_{|G|_1,k_1} P_1$,
we know: $v; H_1 \Downarrow^{c_3,k_3}_f v_2; H_3$ and $H_3 \vDash_{|G|_1,k_1-k_3} Q_1$ and $v_a \in (\!|A|\!)_{|G|_1,k_1-k_3} \wedge \delta L \le c_3 \le \delta U$ (f)
and $\exists n.P_1 = \{\gamma_1 \to T_1, \ldots \gamma_n \to T_n\} \wedge \forall i \in [1,n].|G|_1(\gamma_i) = (l_i, A, m) \implies \forall j.H_1[l_i][j] \neq H_3[l_i][j] \implies j \in T_i$ (z).
From (f), we know : $(\sigma_1[v_a/x], \sigma_2[v_a/x]) \in (\!|\delta(\Gamma, x : U(A,A))|\!)_{G,k}$ (h).

IH on the third premise instantiated with (h), we get:

$$(\sigma_1[v_a/x](t_2), \sigma_2[v_a/x](t_2')) \in (\!|\delta(\{P \sqcup P_1\}\ \overset{\text{diff}(D')}{\exists\vec{\gamma}_1}.\tau\ \{Q\})|\!)^{E,\delta(D_2)}_{G,k}\ (i)$$

Unfold (i), assume : $\sigma_1[v_a/x] t_2 \Downarrow^{c_4,k_4} v_4$ and $\sigma_2[v_a/x] t_2' \Downarrow^{c_1',k_1'} v_1'$.
we get: $(v_4, v_1') \in (\!|\delta(\{P \sqcup P_1\}\ \overset{\text{diff}(D')}{\exists\vec{\gamma}_1}.\tau\ \{Q\})|\!)_{G,k}$ (j) and $c_4 - c_1' \le \delta D_2$ (k).
Unfold (j), we assume :Pick $k_6 = k' - ka - k_3 - k_4 - 1 \le k - k_4$.
We first show the conclusion: $(H_3, H_1') \vDash_{G,k_6} P \sqcup P_1$ $(\clubsuit)$.

Proof. We have the assuption (z) and the assumption $H_1, H_1' \vDash_{G,k} P$.
Unfold $(\clubsuit)$, we need to prove :
$\forall\gamma \in dom(P) \cup dom(P_1).\exists l_1, l_2, \tau, n.G(\gamma) = (l_1, l_2, \tau, n) \wedge \forall i < n.H_3[\gamma][i] \neq H_1'[\gamma][i] \implies i \in (P \sqcup P_1)[\gamma]$. $(\star)$
Unfold the definition of $H_1, H_1' \vDash_{G,k} P$, we know:
$\forall\gamma \in dom(P).\exists l_1, l_2, \tau, n.G(\gamma) = (l_1, l_2, \tau, n) \wedge \forall i < n.H_1[\gamma][i] \neq H_1'[\gamma][i] \implies i \in (P)[\gamma]$. $(\heartsuit)$
From the premise $dom(P) = dom(P_1)$, there is only one case:

1. $\gamma \in dom(P), \gamma \in dom(P_1)$
   From $(\heartsuit)$, we know
   $\forall\gamma \in dom(P) \cap dom(P_1).\exists l_1, l_2, \tau, n.G(\gamma) = (l_1, l_2, \tau, n) \wedge \forall i < n.H_1[\gamma][i] \neq H_1'[\gamma][i] \implies i \in (P)[\gamma]$,
   which means H_1 and H_1' differ at most at indices in $P[\gamma]$, also means $\forall i < n.i \notin P[\gamma] \implies H_1[\gamma][i] = H_1'[\gamma][i]$.
   From (z), we also know that $\forall\gamma \in dom(P_1), \forall i < n.H_1[\gamma][i] \neq H_3[\gamma][i] \implies i \in P_1[\gamma]$, which means H_1 and H_3 differ at most at all the indices in $P_1[\gamma]$.
   So we know H_3 and H_1' differ at most at indices appear in P or P_1, which is $\forall\gamma \in (dom(P) \cap$

$dom(P_1)).\exists l_1, l_2, \tau, n. G(\gamma) = (l_1, l_2, \tau, n) \wedge \forall i < n. H_1[\gamma][i] \neq H_1'[\gamma][i] \implies i \in P[\gamma] \vee i \in P_1[\gamma]$.
This subcase is proved because $i \in P[\gamma] \vee i \in P_1[\gamma] \implies i \in P[\gamma] \cup P_1[\gamma] \implies (P \sqcup P_1)[\gamma]$.

□

Also assume $v_4; H_3 \Downarrow_f^{c_5,k_5} v_5; H_2$ and $v_1'; H_1' \Downarrow_f^{c_2',k_2'} v_2'; H_2'$.
We get: $H_2, H_2' \vDash_{G,k_6-k_5} Q\ (l)$ and $(v_5, v_2') \in (\!|\delta\tau|\!)_{G,k_6-k_5}\ (m)$ and $c_5 - c_2' \leq \delta D'\ (n)$

From the forcing evaluation rule

$$\frac{\delta\, t_1 \Downarrow^{c_a,k_a} v \qquad v; H_1 \Downarrow_f^{c_3,k_3} v_a; H_3 \qquad t_2[v_a/x] \Downarrow^{c_4,k_4} v_4 \qquad v_4; H_3 \Downarrow_f^{c_5,k_5} v_5; H_2}{\mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2; H_1 \Downarrow_f^{c_a+c_3+c_4+c_5+c_{\mathsf{let}},k_a+k_3+k_4+k_5+1} v_5; H_2}\ \text{F-E}$$

STS1 is proved by (l). STS2 is proved by (m) using Lemma 1.
STS3 : $c_2 - c_2' = c_a + c_3 + c_4 + c_5 + c_{\mathsf{let}} - c_2' \leq \delta(U + U_1 + D' + (D_2 + U_2) + c_{\mathsf{let}})$ by (e),(k),(n), (f).

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash_{L_1}^{U_1} t_1' : \{P_1\}\ \exists\vec{\gamma}_1 : A_1'\ \{Q_1\}^{\mathsf{exec}(L,U)} \qquad \Sigma;\Delta;\Phi;|\Gamma|_1 \vdash_{L_2}^{U_2} t_2 : \{P_2\}\ \exists\vec{\gamma}_1 : A_1\ \{Q_2\}^{\mathsf{exec}(L',U')}}{dom(P) = dom(P_1) \qquad \Sigma;\Delta;\Phi;\Gamma, x : U(A_1', A_1') \vdash t_2 \ominus t_2' \lesssim D_2 : \{P \sqcup P_1\}\ \exists\vec{\gamma}_1.\tau'\ \{Q\}^{\mathsf{diff}(D')}}\ \textbf{r-e-bind}$$
$$\Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \lesssim U_2 : \quad \{P\}\ \exists\vec{\gamma}_1.\tau'\ \{Q\}^{\mathsf{diff}(D'+(D_2-L_2)-L_1-L-c_{\mathsf{let}})}$$

By assumption we have $\vdash \delta : \Delta$, $\vDash \delta\Phi$ and $(\sigma_1, \sigma_2) \in [\![\delta\Gamma]\!]_{G,k}$, we need to show:

$$(\sigma_1\ t_2, \mathsf{let}\ \{x\} = (\sigma_2\ t_1')\ \mathsf{in}\ (\sigma_2\ t_2')) \in (\!| \quad \{P\}\ \exists\vec{\gamma}_1.\delta\,\tau'\ \{Q\} \quad |\!)^{e}_{G,k,\delta\,U_2}{}^{\mathsf{diff}(\delta(D'+(D_2-L_2)-L_1-L-c_{\mathsf{let}}))}$$

Unfold this definition, suppose the pure evaluation of $\sigma_1\ t_2$ of the form $\sigma_1\ t_2 \Downarrow^{c_1,k_1} v_1$, and the evaluation of the monadic bind of the form

$$(\mathsf{let}\ \{x\} = \sigma_2\ t_1'\ \mathsf{in}\ \sigma_2\ t_2') \Downarrow^{0,0} \mathsf{let}\ \{x\} = \sigma_2\ t_1'\ \mathsf{in}\ \sigma_2\ t_2'$$

, we need to show:

1. $c_1 - 0 \leq \delta\, U_2$.
2. $(v_1, (\mathsf{let}\ \{x\} = (\sigma_2\ t_1')\ \mathsf{in}\ (\sigma_2\ t_2'))) \in (\!|\delta(\quad \{P\}\ \exists\vec{\gamma}_1.\delta\,\tau\ \{Q\} \quad)|\!)_{G,k-k_1}{}^{\mathsf{diff}(\delta(D'+(D_2-L_2)-L_1-L-c_{\mathsf{let}}))}$.

By Theorem **??** on the second premise $\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash_{L_2}^{U_2} t_2 : \{P_2\}\ \exists\vec{\gamma}_1 : A_1\ \{Q_2\}^{\mathsf{exec}(L',U')}$, instantiated with $\sigma_1 \in [\![\delta|\Gamma|_1]\!]_{|G|_1,k}$ inferred from $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$, and the index variable substitution δ. We conclude that:

$$\sigma_1\ t_2 \in [\![\{P_2\}\ \exists\vec{\gamma}_1 : \delta\, A_1\ \{Q_2\}]\!]^{e}_{|G|_1,k,(\delta\,L_2,\delta\,U_2)}{}^{\mathsf{exec}(\delta\,L',\delta\,U')}$$

, which we unfold we conclude that

$$v_1 \in [\![\{P_2\}\ \exists\vec{\gamma}_1 : \delta\, A_1\ \{Q_2\}]\!]_{|G|_1,k}{}^{\mathsf{exec}(\delta\,L',\delta\,U')} \wedge \delta\, L_2 \leq c_1 \leq \delta\, U_2$$

, which show our first goal $c_1 - 0 \leq \delta\, U_2$.

Then there is one thing left we need to show:

$$(v_1, \mathsf{let}\ \{x\} = \sigma_2\ t_1'\ \mathsf{in}\ \sigma_2\ t_2') \in (\!| \quad \{P\}\ \exists\vec{\gamma}_1.\delta\,\tau\ \{Q\} \quad |\!)_{G,k-k_1}{}^{\mathsf{diff}(\delta(D'+(D_2-L_2)-L_1-L-c_{\mathsf{let}}))}$$

We unfold it definition, when are given arbitrary $G' \supseteq G$, $k' \leq k - k_1$, heaps H, H' so that $H, H' \vDash_{G',k'} P$, suppose the forcing evaluation of v_1 and let $\{x\} = \sigma_2\ t_1'$ in $\sigma_2\ t_2'$ of the form:

$$v_1; H \Downarrow_f^{c_2,k_2} v_2; H_2$$

$$\frac{\sigma_2\ t_1' \Downarrow^{c_1',k_1'} v' \quad v'; H' \Downarrow_f^{c_2',k_2'} v_1'; H_1' \quad \sigma_2\ t_2'[v_1'/x] \Downarrow^{c_3',k_3'} v_2' \quad v_2'; H_1' \Downarrow_f^{c_4',k_4'} v_3'; H_2'}{\mathsf{let}\ \{x\} = \sigma_2\ t_1'\ \mathsf{in}\ \sigma_2\ t_2'; H' \Downarrow_f^{c_1'+c_2'+c_3'+c_4'+c_{\mathsf{let}},k_1'+k_2'+k_3'+k_4'+1} v_3'; H_2'}\ \textbf{f-bind}$$

, with $k_2 < k' \leq k - k_1$. Now taking $G_2 = G'$, we must show:

1. $(H_2, H_2') \vDash_{G',k'-k_2} Q$.
2. $(v_2, v_3') \in (\!|\delta\,\tau|\!)_{G',k'-k_2}$.
3. $c_2 - (c_1' + c_2' + c_3' + c_4' + c_{\mathsf{let}}) \leq \delta(D' + (D_2 - L_2) - L_1 - L - c_{\mathsf{let}})$.

By induction hypothesis using Theorem **??** on the first premise, instantiated with $\sigma_2 \in [\![\delta|\Gamma|_2]\!]_{G|_2,k}$ inferred from $(\sigma_1, \sigma_2) \in (\!|\delta\Gamma|\!)_{G,k}$, and the index variable substitution δ. We can conclude that:

$$\sigma_2\ t_1' \in [\![(\{P_1\}\ \exists^{\mathrm{exec}(\delta\,L,\delta\,U)} \vec{\gamma}_1 : \delta\,A_1'\ \{Q_1\})]\!]^e_{|G|_2,k',(\delta\,L_1,\delta\,U_1)}$$

, which we unwind to conclude:

$$v' \in [\![\{P_1\}\ \exists^{\mathrm{exec}(\delta\,L,\delta\,U)} \vec{\gamma}_1 : \delta\,A_1'\ \{Q_1\}]\!]_{|G|_2,k'-k_1'} \ \wedge\ \delta\,L_1 \leq c_1' \leq \delta\,U_1$$

, which we unfold its definition, taking $g_2 = |G|_2$, assume $H' \vDash_{g_2,k'}$ to conclude that

$$H_1' \vDash_{g_2,k'-k_2'} Q_1 \ \wedge\ v_1' \in [\![\delta\,A_1']\!]_{g_2,k'-k_2'} \ \wedge\ \delta\,L \leq c_2' \leq \delta\,U$$

From $v_1' \in [\![\delta\,A_1']\!]_{g_2,k'-k_2'}$, we conclude that:

$$(\sigma_1[v_1'/x], \sigma_2[v_1'/x]) \in (\!|\delta(\Gamma, x : U(A_1', A_1'))|\!)_{G_2,k}$$

By induction hypothesis on the third premise instantiated with $(\sigma_1[v_1'/x], \sigma_2[v_1'/x]) \in (\!|\delta(\Gamma, x : U(A_1', A_1'))|\!)_{G_2,k}$, we know that:

$$(\sigma_1\ t_2[v_1'/x], \sigma_2\ t_2'[v_1'/x]) \in (\!|\{P \cup P_1\}\ \exists^{\mathrm{diff}(\delta\,D')} \vec{\gamma}_1.\delta\,\tau'\ \{Q\}|\!)^e_{G_2,k,\delta\,D_2}$$

, which we unfold to conclude that

$$(v_1, v_2') \in (\!|\{P \cup P_1\}\ \exists^{\mathrm{diff}(\delta\,D')} \vec{\gamma}_1.\delta\,\tau'\ \{Q\}|\!)_{G_2,k-k_1} \ \wedge\ c_1 - c_3' \leq \delta\,D_2$$

. We unfold $(v_1, v_2') \in (\!|\{P \cup P_1\}\ \exists^{\mathrm{diff}(\delta\,D')} \vec{\gamma}_1.\delta\,\tau'\ \{Q\}|\!)_{G_2,k-k_1}$, we can show that $(H, H_1') \vDash_{G',k'} P \sqcup P_1$ by the definition of the heap relation using the premise $dom(P) = dom(P_1)$. Then we can conclude that when taking $G_2 = G'$:

$$(v_2, v_3') \in (\!|\delta\tau'|\!)_{G',k'-k_2} \ \wedge\ c_2 - c_4' \leq \delta\,D' \ \wedge\ (H_2, H_2') \vDash_{G',k'-k_2}$$

Now we can conclude that

1. $(H_2, H_2') \vDash_{G',k'-k_2} Q$. It is already proved by the previous conclusion.
2. $(v_2, v_3') \in (\!|\delta\,\tau|\!)_{G',k'-k_2}$. It is already proved by the previous conclusion.
3. $c_2 - (c_1' + c_2' + c_3' + c_4' + c_{\mathsf{let}}) \leq \delta(D' + (D_2 - L_2) - L_1 - L - c_{\mathsf{let}})$. It is proved by previous conclusions.

Proof of statement (2) if $\Delta;\Phi;\Omega \vdash^{U}_{L} t : A$ and $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and there exists Ω'. s.t. $FV(t) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in (\!|\delta\Omega'|\!)_{g,k}$. Then, $(\sigma\, t) \in (\!|A|\!)^{E,(\delta L,\delta U)}_{g,k}$
Proof by induction on typing derivation:

$$\text{CASE}\ \frac{\xi;\Delta;\Phi;x : A, f : \overset{\text{exec}(L,U)}{A \longrightarrow} B,\Omega \vdash^{U}_{L} t : B}{\Delta;\Phi;\Omega \vdash^{0}_{0} \text{fix}\, f(x).t : \overset{\text{exec}(L,U)}{A \longrightarrow} B}\ \text{U-F}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and there exists Ω'. s.t. $FV(t) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in (\!|\delta\Omega'|\!)_{g,k}$
TS: $(\sigma \text{fix}\, f(x).t) \in [\![\delta\, \overset{\text{exec}(L,U)}{A \longrightarrow} B]\!]^{E,(0,0)}_{g,k}$. Because $\text{fix}\, f(x).e$ is value.
STS: $(\sigma \text{fix}\, f(x).t) \in [\![\delta\, \overset{\text{exec}(L,U)}{A \longrightarrow} B]\!]_{g,k}$
We prove the more general statement $\forall k' \le k.\sigma(\text{fix}\, f(x).t) \in [\![\delta \overset{\text{exec}(L,U)}{A \longrightarrow} \delta B]\!]_{g,k}(1)$.
By sub-induction on k'.There are two cases:

subcase 1: k'=0. there is no non-negative k such that k ≤ 0, the goal is vacuously true.

subcase 2: $k' = k'' + 1 \le k$

By sub-IH, we know: $\sigma(\text{fix}\, f(x).t) \in [\![\delta A \overset{\text{exec}(L,U)}{\longrightarrow} \delta B]\!]_{g,k''}(2)$.
STS: $(\text{fix}\, f(x).\sigma e) \in [\![\delta A \overset{\text{exec}(L,U)}{\longrightarrow} \delta B]\!]_{g,k''+1}$. Pick j < k"+1 and assume that $v \in [\![\delta A]\!]_{g,j}$. STS: $\sigma[v/x][\text{fix}\, f(x).e/f]) \in [\![\delta B]\!]^{E,(\delta L,\delta U)}_{g,j}$
This follows by IH on the premise instantiated with

1. FV(e) $\subseteq dom(x : A, f : \overset{\text{exec}(L,U)}{A \longrightarrow} B,\Omega')$ from Lemma (well-formness) and $(x : A, f : \overset{\text{exec}(L,U)}{A \longrightarrow} B,\Omega') \subseteq (x : A, f : \overset{\text{exec}(L,U)}{A \longrightarrow} B,\Omega)$
2. $(\sigma[v/x][\text{fix}\, f(x).t/f]) \in [\![\delta(x : A, f : \overset{\text{exec}(L,U)}{A \longrightarrow} B,\Omega')]\!]_{g,j}$ which holds because
   2.1. $\sigma \in [\![\delta\Omega']\!]_{g,j}$ by Lemma downward closure and $j < k'' + 1 \le k$
   2.2. $v \in [\![A]\!]_{g,j}$ from assumption
   2.3. $(\sigma \text{fix}\, f(x).e) \in [\![\delta A \overset{\text{exec}(L,U)}{\longrightarrow} \delta B]\!]_{g,j}$ from Lemma 1 on (2).

This completes the proof of unary fix case.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \text{int}[I] \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : A \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \text{alloc}\, t_1\, t_2 : \{P\}\ \exists\gamma : \overset{\text{exec}(L_1+L_2+L_a,U_1+U_2+U_a)}{\text{Array}_\gamma}[I]\ A\ \{P \star \gamma \to \mathbb{N}\}}\ \text{U-ALLOC}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and there exists Ω'. s.t. $FV(t) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in (\!|\delta\Omega'|\!)_{g,k}$
TS: $(\sigma\ \text{alloc}\, t_1\, t_2) \in [\![\{P\}\ \exists\gamma : \overset{\text{exec}(L_1+L_2+L_a,U_1+U_2+U_a)}{\text{Array}_\gamma}[I]\ A\ \{P\gamma \to \mathbb{N}\}]\!]^{E,(0,0)}_{g,k}$
Because $\text{alloc}\, t_1\, t_2$ is value. STS: $(\sigma\ \text{alloc}\, t_1\, t_2) \in [\![\{P\}\ \exists\gamma : \overset{\text{exec}(L_1+L_2+L_a,U_1+U_2+U_a)}{\text{Array}_\gamma}[I]\ A\ \{P\gamma \to \mathbb{N}\}]\!]_{g,k}$
Unfold the definition of $[\![\{P\}\ \exists\gamma : \overset{\text{exec}(L_1+L_2+L_a,U_1+U_2+U_a)}{\text{Array}_\gamma}[I]\ A\ \{P\gamma \to \mathbb{N}\}]\!]_{g,k}$. Pick $g' \supseteq g$, $k' \le k$, $k'' < k'$ and H. Assume $H \vDash_{g',k'} P$ and $H;\text{alloc}\, t_1\, t_2 \Downarrow^{k''}$.
From $H;\text{alloc}\, t_1\, t_2 \downarrow^{k''}$, because t_1, t_2 are sub terms of σ $\text{alloc}\, t_1\, t_2$,we get $\sigma\, t_1 \Downarrow^{k_1}_{f} \wedge \sigma\, t_2 \Downarrow^{k_2}_{f}$ (1). From (1) , we get:
$\exists v_1, v_2, k_1, k_2.\sigma\, t_1 \Downarrow^{k_1,c_1} v_1 \wedge \sigma\, t_2 \Downarrow^{k_2,c'_1} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c'_1 + c_{alloc}(a)$.
From (a) and evaluation rules we get: $\exists l.H, (\text{alloc}\, t_1\, t_2)\sigma \Downarrow^{k'',c}_{f} l; H[l \to [v_2, \ldots v_2](b)$.
By IH on the first premise instantiated with $\sigma \in [\![\delta\Omega']\!]_{g,k''}$using lemma 1 and $FV(t_1) \subseteq dom(\Omega')$ and

$\Omega' \subseteq \Omega$, we get $(\sigma\, t_1) \in [\![\text{int}_r\, [I]]\!]^{E,(\delta L_1, \delta U_1)}_{g',k''}(*)$.
Unfold the definition of $[\![\text{int}[I]]\!]^{E,(\delta L_1,\delta U_1)}_{g',k''}$ and using (a) and $k_1 \le k''$, we know:

$$(v_1) \in [\![\text{int}[I]]\!]_{G',k''-k_1} \Rightarrow v_1 = I \wedge \delta L_1 \le c_1 \le \delta U_1 \qquad \text{IH1}$$

By IH on the second premise instantiated with $(\sigma) \in [\![\delta\Gamma]\!]_{g',k'-k_1}$ using lemma 1 and $FV(t_2) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$, we get: $(\sigma\, t_2) \in [\![\delta A]\!]^{E,(\delta L_2,\delta U_2)}_{g',k'-k_1}(**)$.
Unfold the definition of (**) and using (a), (b) and $k_2 \le k' - k_1$, we know

$$(v_2) \in [\![\delta A]\!]_{g',k'-k_1-k_2} = [\![\delta A]\!]_{g',k'-k''+1} \wedge \delta L_2 \le c_1' \le \delta U_2 \qquad \text{IH2}$$

Let us assume: $g'' = g'[r \to (l, A, I)](c)$, $H' = H, \left[l \to [v_2, v_2, \ldots, v_2]\right](d)$. We want to show 4 cases.

TS1 $(l) \in [\![\mathsf{Array}_\gamma [I]\ A]\!]_{g'',k'-k''}$
it is proved by unfolding the definition and using the assumption (c).

TS2 $(H') \vDash_{g'',k'-k''} p \star \gamma \to \mathbb{N}$
we know that $g''(\gamma) = (l, A, I)$, $H' = H \uplus H_l \wedge H_l = l \to [v_2, v_2, \ldots, v_2]$

STS1 $\forall i \le I, (H_l(l)[i] \in [\![A]\!]_{g'',k'-k''-1}$. It is proved by using IH2 and Lemma 1.

TS3 $\delta(L_1 + L_2 + L_a) \le c \le \delta(U_1 + U_2 + U_a)$
It is proved by IH1 and IH2 and $\delta L_a \le c_{alloc} \le \delta U_a$.

TS4 $\exists n.P = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \ne H'[l_i][j] \Rightarrow j \in T_i \Big\}$
Because new allocated γ is not in P, so for all the γ_i, it is not changed.

This proof is complete for case alloc.

$$Case\frac{\gamma \in dom(P) \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \mathsf{Array}_\gamma [I]\ A \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \text{int}[I'] \qquad \Delta;\Phi \vDash I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \mathsf{read}\ t_1\ t_2 : \overset{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)}{\{P\}\ \exists\_ : A\ \{P\}}}\ \textbf{u-read}$$

By assumption we have $\vdash \delta : \Delta$, $\vDash \delta\Phi$ and $\sigma \in [\![\delta\Omega]\!]_{g,k}$, we need to show:

$$\mathsf{read}\ (\sigma\ t_1)\ (\sigma\ t_2) \in [\![\delta \overset{\text{exec}(L_1+L_2+L_{\mathsf{read}}, U_1+U_2+U_{\mathsf{read}})}{\{P\}\ \exists\_ : A\ \{P\}}]\!]^e_{g,k,(0,0)}.$$

Since $\mathsf{read}\ (\sigma\ t_1)\ (\sigma\ t_2)$ is a value, and its evaluation incurs no cost, it is sufficient to show:

$$\mathsf{read}\ (\sigma\ t_1)\ (\sigma\ t_2) \in [\![\delta \overset{\text{exec}(L_1+L_2+L_{\mathsf{read}}, U_1+U_2+U_{\mathsf{read}})}{\{P\}\ \exists\_ : A\ \{P\}}]\!]_{g,k}$$

Unfold the definition, when we are given an arbitrary $k' \le k$, an arbitrary $g' \supseteq g$, an arbitrary heap H so that $H \vDash_{g',k'} P$, and suppose that the evaluation of $(\mathsf{read}\ (\sigma\ t_1)\ (\sigma\ t_2))$ has the following form

$$\frac{\sigma\ t_1 \Downarrow^{c_1,k_1} l \quad \sigma\ t_2 \Downarrow^{c_2,k_2} n_2 \quad H(l)[n] = v}{\mathsf{read}\ (\sigma\ t_1)\ (\sigma\ t_2); H \Downarrow^{c_1+c_2+c_{\mathsf{read}}, k_1+k_2+1}_{f} v; H}\ \textbf{f-read}$$

with $k_1 + k_2 + 1 < k' \le k$. Now taking $g_2 = g'$ and $H_1 = H$, we must show that:

1. $\delta L_1 + \delta L_2 + \delta L_{\mathsf{read}} \le c_1 + c_2 + c_{\mathsf{read}} \le \delta U_1 + \delta U_2 + \delta U_{\mathsf{read}}$.
2. $H \vDash_{g',k'-(k_1+k_2+1)} P$.
3. $v \in [\![\delta A]\!]_{g',k'-(k_1+k_2+1)}$.

4. $\exists n.P = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \ \wedge \ \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H[l_i][j] \Rightarrow j \in T_i$.

By induction hypothesis on the first premise $\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} \sigma\ t_1 : \mathsf{Array}_\gamma[I]\ A$, instantiated with $\sigma \in [\![\delta\Omega]\!]_{g',k'}$ inferred from our original assumption $\sigma \in [\![\delta\Omega]\!]_{g,k}$ by lemma 1 , and the index variable substitution δ. We conclude that:

$$\sigma\ t_1 \in [\![\mathsf{Array}_\gamma[\delta\ I]\ (\delta\ A)]\!]^e_{g',k',(\delta\ L_1, \delta\ U_1)}$$

, which we unfold its definition to conclude:

$$l \in [\![\mathsf{Array}_\gamma[\delta\ I]\ (\delta\ A)]\!]_{g',k'-k_1} \ \wedge \ \delta\ L_1 \leq c_1 \leq \delta\ U_1$$

, which in turn tell us: $g'(\gamma) = (l, \delta\ A, \delta\ I)$.

By induction hypothesis on the second premise $\Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \mathrm{int}[I']$, instantiated with $\sigma \in [\![\delta\Omega]\!]_{g',k'}$ and δ, we can conclude :

$$\sigma\ t_2 \in [\![\mathrm{int}[\delta\ I']]\!]^e_{g',k',(\delta\ L_2, \delta\ U_2)}$$

, which we unfold to conclude that:$n_2 \in [\![\mathrm{int}[\delta\ I']]\!]_{g',k'-k_2}$, in turn tells us that

$$n_2 = \delta\ I' \ \wedge \ \delta\ L_2 \leq c_2 \leq \delta\ U_2$$

Now we can conclude as follows:

1. $\delta\ L_1 + \delta\ L_2 + \delta\ L_{\mathsf{read}} \leq c_1 + c_2 + c_{\mathsf{read}} \leq \delta\ U_1 + \delta\ U_2 + \delta\ U_{\mathsf{read}}$, using the fact that $L_{\mathsf{read}} \leq c_{\mathsf{read}} < U_{\mathsf{read}}$ in our language and L_{read} and U_{read} are constants , which means $\delta\ L_{\mathsf{read}} = L_{\mathsf{read}}$ and $\delta\ U_{\mathsf{read}} = U_{\mathsf{read}}$.
2. $H \vDash_{g',k'-(k_1+k_2+1)} P$, which is proved by unfolding the definition of our assumption $H \vDash_{g',k'} P$.
3. $v \in [\![\delta\ A]\!]_{g',k'-(k_1+k_2+1)}$. From the heap relation $H \vDash_{g',k'} P$ and the premise $\gamma \in dom(P)$, we know $\forall i \leq n, (H_1(l)(i)) \in [\![A]\!]_{g',k'-1}$, which in turn tells us $v \in [\![A]\!]_{g',k'-1}$. We can then show our goal by using Lemma 1.
4. $\exists n.P = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \ \wedge \ \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H[l_i][j] \Rightarrow j \in T_i$. Because the heap H is not changed, so the claim is trivial.

$$Case\ \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \mathsf{Array}_\gamma[I]\ A \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_3}_{L_3} t_3 : A \qquad \Delta;\Phi \vDash I' \leq I \qquad \Sigma;\Delta;\vdash P \quad wf \qquad \Delta;\Phi \vDash I' \in \beta}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{\mathrm{exec}(L_1+L_2+L_3+L_{\mathsf{updt}}, U_1+U_2+U_3+U_{\mathsf{updt}})} \mathsf{updt}\ t_1\ t_2\ t_3 : \ \{P \star \gamma \to \beta\}\ \exists\_ : \mathrm{unit}\ \{P \star \gamma \to \beta\}}\ \textbf{u-updt}$$

By assumption we have $\vdash \delta : \Delta$, $\vDash \delta\Phi$ and $\sigma \in [\![\delta\Omega]\!]_{g,k}$, we need to show:

$$\mathsf{updt}\ (\sigma\ t_1)\ (\sigma\ t_2)\ (\sigma\ t_3) \in [\![\delta \ \{P \star \gamma \to \beta\}\ \exists\_ : \mathrm{unit}\ \{P \star \gamma \to \beta\}\ ]\!]^{e}_{g,k,(0,0)}{}^{\mathrm{exec}(L_1+L_2+L_3+L_{\mathsf{updt}}, U_1+U_2+U_3+U_{\mathsf{updt}})}$$

Since $\mathsf{updt}\ (\sigma\ t_1)\ (\sigma\ t_2)\ (\sigma\ t_3)$ is a value, it is sufficient to show:

$$\mathsf{updt}\ (\sigma\ t_1)\ (\sigma\ t_2)\ (\sigma\ t_3) \in [\![\delta \ \{P \star \gamma \to \beta\}\ \exists\_ : \mathrm{unit}\ \{P \star \gamma \to \beta\}\ ]\!]_{g,k}{}^{\mathrm{exec}(L_1+L_2+L_3+L_{\mathsf{updt}}, U_1+U_2+U_3+U_{\mathsf{updt}})}$$

Unfold the definition, when we are given an arbitrary $k' \leq k$, an arbitrary $g' \supseteq g$, an arbitrary heap H so that $H \vDash_{g',k'} P \star \gamma \to \beta$, and suppose that the evaluation of $(\mathsf{updt}\ (\sigma\ t_1)\ (\sigma\ t_2)\ (\sigma\ t_3))$ has the following form

$$\frac{\sigma\ t_1 \Downarrow^{c_1,k_1} l \quad \sigma\ t_2 \Downarrow^{c_2,k_2} n \quad \sigma\ t_3 \Downarrow^{c_3,k_3} v}{\mathsf{updt}\ (\sigma\ t_1)\ (\sigma\ t_2)\ (\sigma\ t_3); H \Downarrow_f^{c_1+c_2+c_3+c_{\mathsf{update}}, k_1+k_2+k_3+1} (); H(l)[n] \leftarrow v}\ \textbf{f-updt}$$

With $k_1 + k_2 + k_3 + 1 < k'$. Now we take $g_2 = g'$ and $H_1 = H(l)[n] \leftarrow v$, we must show:

1. $\delta\, L_1 + \delta\, L_2 + \delta\, L_3 + \delta\, L_{\mathsf{updt}} \leq c_1 + c_2 + c_3 + c_{\mathsf{updt}} \leq \delta\, U_1 + \delta\, U_2 + \delta\, U_3 + \delta\, U_{\mathsf{updt}}$.
2. $(H(l)[n] \leftarrow v) \vDash_{g',k'-(k_1+k_2+k_3+1)} P \star \gamma \rightarrow \beta$.
3. $v \in [\![\mathsf{unit}]\!]_{g',k'-(k_1+k_2+k_3+1)}$.
4. $\exists n.P \star \gamma \rightarrow \beta = \{\gamma_1 \rightarrow T_1, \ldots, \gamma_n \rightarrow T_n\} \ \wedge \ \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq (H(l)[n] \leftarrow v)[l_i][j] \Rightarrow j \in T_i$.

By induction hypothesis on the first premise $\Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t_1 : \mathsf{Array}_\gamma[I]\ A$, instantiated with $\sigma \in [\![\delta\Omega]\!]_{g',k'}$ inferred from our original assumption $\sigma \in [\![\delta\Omega]\!]_{g,k}$ by lemma 1 , and the index variable substitution δ. We conclude that:

$$\sigma\ t_1 \in [\![\mathsf{Array}_\gamma[\delta\ I]\ (\delta\ A)]\!]^e_{g',k',\delta\,(\delta\,L_1,\delta\,U_1)}$$

, which we unwind to conclude that

$$l \in [\![\mathsf{Array}_\gamma[\delta\ I]\ (\delta\ A)]\!]_{g',k'-k_1} \Rightarrow g'(\gamma) = (l, \delta\ A, \delta\ I) \ \wedge \ \delta\, L_1 \leq c_1 \leq \delta\, U_1$$

By induction hypothesis on the second premise $\Sigma;\Delta;\Phi;\Omega \vdash^{U_2}_{L_2} t_2 : \mathrm{int}[I']$, we conclude that

$$\sigma\ t_2 \in [\![\mathrm{int}[\delta\ I]]\!]^e_{g',k',(\delta\,L_2,\delta\,U_2)}$$

, which we unfold its definition to conclude: $n \in [\![\mathrm{int}[\delta\ I]]\!]_{g',k'-k_2}$, in turn tells us that

$$n = I \ \wedge \ \delta\, L_2 \leq c_2 \leq \delta\, U_2$$

By induction hypothesis on the third premise $\Sigma;\Delta;\Phi;\Omega \vdash^{U_3}_{L_3} t_3 : A$, we conclude that:

$$\sigma\ t_3 \in [\![\delta\ A]\!]^e_{g',k',(\delta\,L_3,\delta\,U_3)}$$

, which we unwind to conclude that

$$v \in [\![\delta\ A]\!]_{g',k'-k_3} \ \wedge \ \delta\, L_3 \leq c_1'' \leq \delta\, U_3$$

Now we can conclude as follows:

1. $\delta\, L_1 + \delta\, L_2 + \delta\, L_3 + \delta\, L_{\mathsf{updt}} \leq c_1 + c_2 + c_3 + c_{\mathsf{updt}} \leq \delta\, U_1 + \delta\, U_2 + \delta\, U_3 + \delta\, U_{\mathsf{updt}}$, using the fact that $L_{\mathsf{updt}} \leq c_{\mathsf{updt}} < U_{\mathsf{updt}}$ in our language and L_{updt} and U_{updt} are constants , which means $\delta\, L_{\mathsf{updt}} = L_{\mathsf{updt}}$ and $\delta\, U_{\mathsf{updt}} = U_{\mathsf{updt}}$.
2. $(H(l)[n] \leftarrow v) \vDash_{g',k'-(k_1+k_2+k_3+1)} P \star \gamma \rightarrow \beta$. We conlude that $\forall i \leq I.H(l)[i] \in (\!|\delta\ A|\!)_{g',k'-1}$ by unfolding the definition of our assumption $H \vDash_{g',k'} P \star \gamma \rightarrow \beta$. Then we can prove that $\forall i \leq I.(H(l)[n] \leftarrow v)(l)[i] \in (\!|\delta\ A|\!)_{g',k'-(k_3+1)}$ because of the our previous conclusion $v \in [\![\delta\ A]\!]_{g',k'-k_3}$. Then our goal can be proved by using the monotonicity lemma 1.
3. $v \in [\![\mathsf{unit}]\!]_{g',k'-(k_1+k_2+k_3+1)}$, which is proved by the definition of the interpretation of unit type.
4. $\exists n.P \star \gamma \rightarrow \beta = \{\gamma_1 \rightarrow T_1, \ldots, \gamma_n \rightarrow T_n\} \ \wedge \ \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq (H(l)[n] \leftarrow v)[l_i][j] \Rightarrow j \in T_i$. When $\gamma_i \neq \gamma$, the array is not changed so that $H[l_i] = (H(l)[n] \leftarrow v)[l_i]$. When $\gamma_i = \gamma$, then $T_i = \beta$, we can show that the only updated index n at this array $H[l]$ is in β from our premise $\Delta;\Phi \vDash I' \in \beta$ in the rule, which proves this case.

This completes the proof of case unary update.

$$\textsc{case}\ \frac{\Sigma;\Delta;\Phi;\Omega \vdash^{U}_{L} t : A \qquad \Delta;\Phi;\Omega \vdash p \quad wf}{\Sigma;\Delta;\Phi;\Omega \vdash^{0}_{0} \mathsf{return}\ t : \{P\}\ \exists\gamma.A\ \{P\}}\ \overset{\mathrm{exec}(L,U)}{\textsc{u-t}}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and there exists Ω'. s.t. $FV(t) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in (\!|\delta\Omega'|\!)_{g,k}$

TS: $(\sigma \text{ return } t) \in [\![\delta \ \{P\} \ \overset{\text{exec}(L,U)}{\exists\gamma.A} \ \{P\}]\!]^{E,(0,0)}_{g,k}$

Because return t is value. STS: $(\sigma \text{ return } t) \in [\![\delta \ \{P\} \ \overset{\text{exec}(L,U)}{\exists\gamma.A} \ \{P\}]\!]_{g,k}$

Unfold the definition of $[\![\delta \ \{P\} \ \overset{\text{exec}(L,U)}{\exists\gamma.A} \ \{P\}]\!]_{g,k}$. Pick $g^{'} \supseteq g$, $H_1, k' \leq k, k'' < k'$. Assume $H_1 \vDash_{g',k'} P(1)$, $H_1; \sigma(\text{return } t) \Downarrow_f^{k''}$ (2).

Because t is the subterm of return t , From (2), We know: $\sigma_1 t \Downarrow^{k''}$ (3). From (3), we know:

$$\exists v_1, k_1, \sigma_1 t \Downarrow^{k_1, c_1} v_1 \wedge k'' = k_1 + 1 \wedge c = c_1 + c_{ret} \tag{a}$$

From (a) and the evaluation rule **U-ret**: $\exists H_1'.H_1; \text{return } t \Downarrow_f^{k'',c} v_1; H_1'(b)$.

By IH on the premise instantiated with $(\sigma) \in [\![\delta\Omega']\!]_{g',k'}$ by lemma 1. We get: $(\sigma t_1) \in [\![\delta A]\!]^{E,\delta(L,U)}_{G',k'}(5)$.
Unfold (5), we get: $v_1 \in (\!|\delta A|\!)_{g',k'-k_1} \wedge \delta L_1 \leq c_1 \leq \delta U_1(6)$.
Let us assume: $g'' = g'(7)$, $H_1' = H_1'(8)$.

STS1: $H_1' \vDash_{g',k'-k''} P$. it is from assumption by using lemma 1.

STS2: $v_1 \in (\!|\tau|\!)_{g',k'-k''}$. It is proved by (6) using Lemma 1.

STS3: $\delta L \leq c \leq \delta U$ which is proved from (6)

STS4: $\exists n.P = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
H is not changed, it is trivial true.

This completes the proof of case unary return.

$$\frac{P = P_1 \star P_2 \qquad \Sigma;\Delta;\Phi;\Omega \vdash^{U_1}_{L_1} t : \{P_1\} \ \overset{\text{exec}(L,U)}{\exists\vec{\gamma}_1} : A \ \{Q_1 \star Q_2\} \qquad \Sigma;\Delta,\vec{\gamma}_1;\Phi;\Omega, x : A \vdash^{U_2}_{L_2} u : \{Q_1 \star P_2\} \ \overset{\text{exec}(L',U')}{\exists\vec{\gamma}_2} : B \ \{Q\}}{\Sigma;\Delta;\Phi;\Omega \vdash^{U_1+U_2+U_l}_{L_1+L_2+L_l} \text{let } \{x\} = t \text{ in } u : \{P\} \ \overset{\text{exec}(L+L',U+U')}{\exists\vec{\gamma}_1,\vec{\gamma}_2} : B \ \{Q \star Q_2\}} \text{ U-E}$$

Assume that $\vdash \delta : \Delta$ and $\vDash \delta\Phi$ and there exists Ω'. s.t. $FV(t) \subseteq dom(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in (\!|\delta\Omega'|\!)_{g,k}$

TS: $(\delta\sigma_1(\text{let } \{x\} = t_1 \text{ in } t_1')) \in (\!|\delta \ \{P\} \ \overset{\text{exec}(L+L',U+U')}{\exists\vec{\gamma}_1,\vec{\gamma}_2} : B \ \{Q \star Q_2\}|\!)^{(E,(U_1+U_2,L_1+L_2))}_{g,k}$

Unfold the definition . Pick $g^{'} \supseteq g$, $H_1, k' \leq k, k'' < k'$. Assume $H_1 \vDash_{G',k'} P(1)$, $H_1; \sigma_1(\text{let } \{x\} = t_1 \text{ in } t_1') \Downarrow_f^{k''}$ (2).

Because t_1 is the subterm of let $\{x\} = t_1$ in t_1' , From (2), we know: $\exists v_1, v_1', v_1'', v_1''', k_1, k_2, k_3, k_4, H_1', H_1''$. $\sigma_1 t_1 \Downarrow^{k_1,c_1} v_1$ and $H_1; v_1 \Downarrow_f^{k_2,c_2} H_1'; v_1'$ and $\sigma_1 t_1'[v_1'/x] \Downarrow^{k_3,c_3} v_1''$ and $H_1'; v_1'' \Downarrow_f^{k_4,c_4} H_1''; v_1'''$, $k'' = k_1 + k_2 + k_3 + k_4 + 1 \wedge c = c_1 + c_2 + c_3 + c_4 + c_{let}(a)$.
From evaluation rule: E-Let-F and (a)(b), we know:

$$\begin{aligned}\exists H_a, H_c, H_a', H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v1; H_a \Downarrow_f v_1'; H_a' \wedge H_a' = H1_{Q1} \uplus H1_{Q2} \\ \wedge v_1''; H1_{Q1} \uplus H_c \Downarrow_f v_1'''; H1_Q \wedge H_1'' = H1_Q \uplus H1_{Q2}\end{aligned} \tag{c}$$

From (1) : $H \vDash_{g',k'} P_1 \star P_2$, we assume that $\exists g_a, g_b : (H_1 = H_a \uplus H_c)$ $(g' = g_a \uplus g_c)$ and $(H_a) \vDash_{g_a,k'} P_1$ and $(H_c) \vDash_{g_c,k'} P_2(*)$.
STS1: $(H_1'') \vDash_{g'',k'-k''} Q \star Q_2$. STS2: $(v_1''') \in (\!|\delta B|\!)_{G'',k'-k''}$. STS3: $\delta(L+L') \leq c_f \leq \delta(U+U') \wedge \delta(L_1+L_2+L_{let}) \leq c_p \leq \delta(U_1+U_2+U_{let})$. STS4: $P = \gamma_i \to \beta_i \forall i.g''(\gamma_i) = (l_i, m, A) \to H_1(l_i)[n] \neq H_1''(l_i)[n] \Rightarrow n \in \beta_i$.
By IH on the second premise instantiated with $(\sigma_1) \in (\!|\delta\Omega|\!)_{g',k'}$, we know:$(\sigma_1 t_1) \in (\!|\{P_1\} \ \overset{\text{exec}\delta(L,U)}{\exists\vec{\gamma}_1} : A \ \{Q_1 \star Q_2\}|\!)^{E,(L_1,U_1)}_{g',k'}(4)$.
Unfold the definition ,use (a), we get:

$$\begin{aligned}(v_1) \in (\!|\delta \ \{P_1\} \ \overset{\text{exec}\delta(L,U)}{\exists\vec{\gamma}_1} : A \ \{Q_1 \star Q_2\}|\!)_{g',k'} \wedge L_1 \leq c_1 \leq U_1 \wedge P_1 = \gamma_i \to \beta_i \Rightarrow \\ \forall i.g''(\gamma_i) = (l_i, m, A) \to H_1(l_i)[n] \neq H_1'(l_i)[n] \Rightarrow n \in \beta_i\end{aligned} \tag{e}$$

To unfold (e), we choose: $k_f \leq k' - k_1$ and $k'_f < k_f$ and $g_f \supseteq g'$ and (1),(a).
We have $(H_a \vDash_{g_f,k_f} P_1 \wedge H_a; v_1 \Downarrow_f \wedge H_b; v_2 \Downarrow_f$.
Unfold the definition, we get: $H_a; v_1 \Downarrow_f^{c_2} H'_a; v'_1 \wedge H'_a \vDash Q_1 \star Q_2$.
$(v'_1) \in (\!|A|\!)_{g',k'-k_1} \wedge \delta L \leq c_2 \leq \delta U(k)$.
By IH on the third premise, we know: $(\sigma_1 v''_1) \in (\!|\{Q_1 \star P_2\} \overset{\text{exec}\delta D'}{\exists \vec{\gamma}_1.\delta B} \{Q\}|\!)^{E,\delta D_2}_{G',k'}$(5).
Unfold the definition, we get: $v_1 \in (\!|\{Q_1 \star P_2\} \overset{\text{exec}(L',U')}{\exists \vec{\gamma}_1.\tau_2} \{Q\}|\!)_{g',k'-k_1} \wedge l_2 \leq c_3 \leq U_2 \wedge Q_1 \uplus P_2 = \gamma_i \rightarrow \beta_i$
$\Rightarrow \forall i.g''(\gamma_i) = (l_i, m, A) \rightarrow H'_1(l_i)[n] \neq H''_1(l_i)[n] \Rightarrow n \in \beta_i(f)$.
We have $(H'_a) \vDash_{g_f,k_f} Q_1 \star Q_2 \Rightarrow H1_{Q1} \vDash Q_1 \wedge H1_{Q2} \vDash Q_2 \wedge H_c \vDash P_2$. $H1_{Q1} \uplus H_c; v'''_1 \Downarrow_f$.
Unfold the definition (f), we get: $H1_{Q1} \uplus H_c; v'''_1 \Downarrow_f^{c_4} H1_Q; v''''_1 \wedge H1_Q \vDash Q$, $(v'''_1) \in (\!|\delta B|\!)_{g',k'-k_1} \wedge \delta L' \leq c_4 \leq \delta U'(g)$.
STS1: Use Lemma 2, we know that $H1_Q \uplus H1_{Q2} \vDash Q \star Q_2$, which proves STS1.
STS2: it is proved from (g).
STS3: it is proved from (e),(k),(f),(g) and $c = c_f + c_p. \wedge c_f = c_2 + c_4$
STS4: from (e),(f), we know all the n in P will appear in P_1 or $Q_1 \uplus P_2$. It is proved.
This completes the proof of Unary let.

Proof of statement (3) if $\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ and $\vdash \delta : \Delta$ and $\vDash \delta\Phi$. Then for $i \in \{1,2\}$. if there exists Γ_i' s.t. $\mathrm{FV}(e_i) \subseteq dom(\Gamma_i'), \Gamma_i' \subseteq \Gamma$ and $(\sigma_i) \in [\![|\delta\Gamma_i'|_i]\!]_{G|_i,k}$, then $\sigma_i e_i \in [\![|\delta\tau|_i]\!]^{E,(0,\infty)}_{G|_i,k}$
Because it is similar for the two case, we only show the case when i=1.

$$\text{CASE}\ \frac{\Delta;\Phi;x:\tau_1, f\ :\tau_1 \xrightarrow{\text{diff}(D)} \tau_2, \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau_2}{\Delta;\Phi;\Gamma \vdash \mathsf{Fix}\, f(x).t_1 \ominus \mathsf{Fix}\, f(x).t_2 \lesssim 0 : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2}\ \text{R-FIX}$$

Assume that $\vDash \Phi\delta$ and there exists Γ' s.t. $\mathrm{FV}(\mathsf{fix}\, f(x).t_1) \subseteq dom(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in [\![|\delta\Gamma'|_1]\!]_{G|_1,k}$.
TS: $(\mathsf{fix}\, f(x).\sigma t_1) \in [\![(|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1)]\!]^{E,(0,\infty)}_{G|_1,k}$
By Lemma 31,STS: $(\mathsf{fix}\, f(x).\sigma t_1) \in [\![(|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1)]\!]_{G|_1,k}$. We prove the more general statement:

$$\forall m' \le k.(\mathsf{fix}\, f(x).\sigma t_1) \in [\![(|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1)]\!]_{G|_1,m} \tag{1}$$

By subinduction on m'. There are two cases.

subcase 1: $m' = 0$. there is no non-negative k' < 0, it is vacuously true.

subcase 2: $m' = m'' + 1 \le k$

By sub-IH: $(\mathsf{fix}\, f(x).\sigma t_1) \in [\![(|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1)]\!]_{G|_1,m''}$. STS: $(\mathsf{fix}\, f(x).\sigma t_1) \in [\![(|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1)]\!]_{G|_1,m''+1}$
Unfold the definition. Pick $j \le m''+1$, $g' \supseteq G|_1$. Assume $v \in [\![|\delta\tau_1|_1]\!]_{g',j}$, STS: $\delta\sigma t_1[v/x][\mathsf{fix}\, f(x).t_1/f] \in [\![|\delta\tau_2|_1]\!]^{E,(0,\infty)}_{g',j}$

By IH 3 on the premise instantiated with $\sigma[v/x][\mathsf{fix}\, f(x).t_1/f] \in [\![\delta(x:|\tau_1|_1, f:|\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1, |\Gamma'|)]\!]_{g',j}$ which holds because

1. $\sigma \in [\![|\delta\Gamma'|_1)]\!]_{g',j}$ by Lemma 1.
2. $v \in [\![|\tau_1|_1]\!]_{g',j}$ from assumption.
3. $\mathsf{fix}\, f(x).t_1 \in [\![|\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\tau_2|_1]\!]_{g',j}$ by Lemma 1 on the sub-IH conclusion.

and premises: $FV(t_1) \subseteq dom(x:\tau_1, f:\tau_1 \xrightarrow{\text{diff}(D)} \tau_2, \Gamma')$ and $x:\tau_1, f:\tau_1 \xrightarrow{\text{diff}(D)} \tau_2, \Gamma' \subseteq x:\tau_1, f:\tau_1 \xrightarrow{\text{diff}(D)} \tau_2, \Gamma$.

This proof is complete.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathrm{int}[I] \qquad \Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \tau \qquad \gamma\ \mathsf{fresh}}{\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\, t_1\ t_2 \ominus \mathsf{alloc}\, t_1'\ t_2' \lesssim 0 : \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \mathbb{N}\}}\ \text{ALLOC}$$

Assume that $\vDash \Phi\delta$ and there exists Γ' s.t. $\mathrm{FV}(\mathsf{alloc}\, t_1\ t_2) \subseteq dom(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in [\![|\delta\Gamma'|_1]\!]_{G|_1,k}$
TS: $\sigma\ \mathsf{alloc}\, t_1\ t_2 \in [\![|\delta(\{|P|_1\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ |\tau|_1\ \xrightarrow{\text{exec}(0,\infty)}\ \{|P|_1 \star \gamma \to \mathbb{N}\})|_1]\!]^{E,(0,\infty)}_{G|_1,k}$
STS: $\sigma\ \mathsf{alloc}\, t_1\ t_2 \in [\![|\delta(\{|P|_1\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ |\tau|_1\ \xrightarrow{\text{exec}(0,\infty)}\ \{|P|_1 \star \gamma \to \mathbb{N}\})|_1]\!]_{G|_1,k}$
Unfold the definition. Pick $g' \supseteq G|_1, k' \le k, k'' < k'$ and assume $H \vDash_{g',k'} |P|_1$ and $H;\mathsf{alloc}\, t_1\ t_2 \Downarrow_f^{k''}$
STS: $H;v \Downarrow_f^{k'',c} v', H'$ and $0 \le c \le \infty$ and $H' \vDash_{g'',k'-k''} \gamma$ and $v' \in [\![\mathsf{Array}_\gamma[I]\ |\tau|_1]\!]_{g'',k'-k''}$
From $H;\mathsf{alloc}\, t_1\ t_2 \Downarrow^{k''}$, because t_1, t_2 are sub terms of $\sigma\ \mathsf{alloc}\, t_1\ t_2$,we get: $\sigma t_1 \Downarrow_f^{k_1} \wedge \sigma t_2 \Downarrow_f^{k_2}$ (1). From (1), we get:

$$\exists v_1, v_2, k_1, k_2.\sigma t_1 \Downarrow^{k_1,c_1} v_1 \wedge \sigma t_2 \Downarrow^{k_2,c_1'} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_1' + c_{alloc} \tag{a}$$

From (a) and evaluation rules we get: $\exists l.H,(\mathsf{alloc}\ t_1\ t_2)\sigma \Downarrow_f^{k'',c} l; H[l \to [v_2,\ldots v_2](b)$.
By IH 3 on the first premise instantiated with $\sigma \in [\![|\delta\Gamma'|_1]\!]_{g,k''}$ using lemma 1 and $FV(t_1) \subseteq dom(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get: $(\sigma\, t_1) \in [\![|\delta \mathsf{int}_r[I]|_1]\!]^{E,(0,\infty)}_{g',k''}(*)$.
Unfold the definition of $[\![\mathsf{int}[I]]\!]^{E,(0,\infty)}_{g',k''}$ and using (a) and $k_1 \le k''$, we know:

$$(v_1) \in [\![\mathsf{int}[I]]\!]_{G',k''-k_1} \Rightarrow v_1 = I \wedge 0 \le c_1 \le \infty \qquad \text{IH1}$$

By IH 3 on the second premise instantiated with $(\sigma) \in [\![|\delta\Gamma'|_1]\!]_{g',k'-k_1}$ using lemma 1 and $FV(t_2) \subseteq dom(|\Gamma'|_1)$ and $\Gamma' \subseteq \Gamma$, we get: $(\sigma\, t_2) \in [\![|\delta\tau|_1]\!]^{E,(0,\infty)}_{g',k'-k_1}(**)$. Unfold the definition of (**) and using (a), (b) and $k_2 \le k' - k_1$, we know

$$(v_2) \in [\![|\delta\tau|_1]\!]_{g',k'-k_1-k_2} = [\![|\delta\tau|_1]\!]_{g',k'-k''+1} \wedge 0 \le c_1' \le \infty \qquad \text{IH2}$$

Let us assume: $g'' = g'[\gamma \to (l,|\tau|_1,I)](c)$, $H_1' = H_1, \big[l \to [v_2,v_2,\ldots,v_2]\big](d)$. We want to show 2 cases.

TS1 $(v' = l) \in [\![\mathsf{Array}_\gamma[I]\ |\tau|_1]\!]_{g'',k'-k''}$. it is proved by unfolding the definition and using the assumption (c).

TS2 $(H_1') \vDash_{g'',k'-k''} |P|_1 \star \gamma \to \mathbb{N}$
we know that $g''(\gamma) = (l,|\tau|_1,I)$, $H_1' = H_1 \uplus H_l \wedge H_l = l \to [v_2,\ldots,v_2]$, $H_1 \vDash |P|_1$ from assumption.

STS1 $\forall i \le I, (H_l(l)[i] \in [\![A]\!]_{g'',k'-k''-1}$. It is proved by using IH2 and Lemma 1.

TS3 $0 \le c \le \infty$. It is proved by IH1 and IH2 and $0 \le c_{alloc} \le \infty$.

TS4: $\exists n.|P|_1 = \{\gamma_1 \to T_1,\ldots,\gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i,A,m) \Rightarrow \forall j.(H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
Because γ not in $|P|_1$. For all γ_i, H,H' is the same.

This proof is complete.

$$\text{CASE}\ \frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathsf{int}_r[I'] \qquad \Delta;\Phi \vDash I' \le I\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \lesssim^{\mathrm{diff}(D_1+D_2)} 0 : \{P\}\ \exists\_.\tau\ \{P\}}\ \text{R-READ}$$

Assume that $\vDash \Phi\delta$ and there exists Γ' s.t. FV(read $t_1\ t_2$) $\subseteq dom(\Gamma')$, $\Gamma' \subseteq \Gamma$ and $\sigma \in [\![|\delta\Gamma'|_1]\!]_{G|_1,k}$
TS: σ read $t_1\ t_2 \in [\![|\delta(\{|P|_1\}\ \exists\_.|\tau|_1\ \{|P|_1\})|_1]\!]^{E,(0,\infty)}_{G|_1,k}$
STS: σ read $t_1\ t_2 \in [\![|\delta(\{|P|_1\}\ \exists\_.|\tau|_1\ \{|P|_1\})|_1]\!]_{G|_1,k}$
Unfold the definition. Pick $g' \supseteq G|_1$, $k' \le k, k'' < k'$ and assume $H \vDash_{g',k'} |P|_1$ and H; read $t_1\ t_2 \Downarrow_f^{k''}$
STS: H; read $t_1\ t_2 \Downarrow_f^{k'',c} v', H'$ and $0 \le c \le \infty$ and $H' \vDash_{g'',k'-k''} \gamma$ and $v' \in [\![|\tau|_1]\!]_{g'',k'-k''}$
From H; read $t_1\ t_2 \Downarrow^{k''}$, because t_1, t_2 are sub terms of σ read $t_1\ t_2$,we get:$\sigma\, t_1 \Downarrow_f^{k_1} \wedge \sigma\, t_2 \Downarrow_f^{k_2}$ (1). From (1), we get:

$$\exists l,n,v,k_1,k_2.\sigma\, t_1 \Downarrow^{k_1,c_1} l \wedge \sigma\, t_2 \Downarrow^{k_2,c_1'} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_1' + c_{read} \wedge H(l)[n] = v \qquad \text{(a)}$$

From (a) and evaluation rules we get: $H;\sigma(\mathsf{read}\ t_1\ t_2) \Downarrow_f^{k'',c} H; v\ (b)$.
By IH 3 on the first premise instantiated with $\sigma \in [\![|\delta\Gamma'|_1]\!]_{g,k''}$ using lemma 1 and $FV(t_1) \subseteq dom(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:$(\sigma\, t_1) \in [\![\mathsf{Array}_\gamma[I]\ |\tau|_1]\!]^{E,(0,\infty)}_{g',k'}$(6).
Unfold (6), since $k_1 \le k'$ and $\sigma_1\, t_1 \Downarrow^{k_1,c_1} l$, we know $l \in [\![\mathsf{Array}_\gamma[I]\ |\tau|_1]\!]_{g',k'-k_1} \wedge 0 \le c_1 \le \infty(e)$.
From (e), we know : $g'(\gamma) = (l,|\tau|_1,I)(7)$.
By IH 3 on the first premise instantiated with $\sigma \in [\![|\delta\Gamma'|_1]\!]_{g,k''}$ using lemma 1 and $FV(t_2) \subseteq dom(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get: $(\sigma_1\, t_2') \in [\![\mathsf{int}[I']]\!]^{E,(0,\infty)}_{g',k'}$(8). Unfold (8), since $k_2 \le k'$,we know

$$n \in [\![\mathsf{int}[I']]\!]_{g',k'-k_2} \Rightarrow n = I' \wedge 0 \le c_1' \le \infty \qquad \text{(f)}$$

Let us assume: $g' = g'\ (g)$.

STS1: $H \vDash_{g',k'-k''} |P|_1$. It is proved by Lemma 1 on assumption $H \vDash_{g',k'} |P|_1$

STS2: $(v' = v) \in [\![|\tau|_1]\!]_{g',k'-k''}$
from (a). we know $H_1(l)[n] = v$. From $H \vDash_{g',k'-k''} \gamma$, we know: $\forall i \le n, (H_1(l)(i)) \in [\![A]\!]_{g',k'-1}$.
s.t we know $(H_1(l)[n]) \in [\![A]\!]_{g',k'-1}$. Because $k' - k'' \le k' - 1$, By Lemma 1, We get: $v \in [\![A]\!]_{g',k'-k''}$

STS3: $0 \le c \le \infty$. Proved by the fact that $0 \le c_{read} \le \infty$, it is proved by (e)(f).

STS4: $\exists n.|P|_1 = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
H is not changed , it is trivially true.

This completes the proof of case read.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta;\Phi \models I' \le I \qquad \Delta;\Phi \models I' \in \beta}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \lesssim 0 : \{\gamma \to \beta\}\ \exists\_.\square\tau\ \{\gamma \to \beta\}}^{\mathrm{diff}(D_1+D_2)}\ \text{R-RB}$$

The proof is same as case read above.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 : \tau \qquad \Delta;\Phi \models I' \le I}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \lesssim 0 : \{P \star \gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{P \star \gamma \to \beta \cup \{I'\}\}}^{\mathrm{diff}(D_1+D_2+D_3)}\ \text{R-U}$$

Assume that $\vDash \Phi\delta$ and there exists Γ' s.t. $\mathrm{FV}(\mathsf{updt}\ t_1\ t_2\ t_3) \subseteq dom(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in [\![|\delta\Gamma'|_1]\!]_{G|_1,k}$
TS: $\sigma\ \mathsf{updt}\ t_1\ t_2\ t_3 \in [\![|\delta(\{|P|_1 \star \gamma \to \mathbb{N}\}\ \exists\_.|\mathrm{unit}_r|_1\ \{|P|_1 \star \gamma \to \mathbb{N}\})|_1]\!]^{E,(0,\infty)}_{G|_1,k}$
STS: $\sigma\ \mathsf{updt}\ t_1\ t_2\ t_3 \in [\![|\delta(\{|P|_1 \star \gamma \to \mathbb{N}\}\ \exists\_.|\mathrm{unit}_r|_1\ \{|P|_1 \star \gamma \to \mathbb{N}\})|_1]\!]_{G|_1,k}$
Unfold the definition.
Pick $g' \supseteq G|_1, k' \le k, k'' < k'$ and assume $H \vDash_{g',k'} |P|_i \star \gamma \to \mathbb{N}$ and $H; \mathsf{updt}\ t_1\ t_2\ t_3 \downarrow_f^{k''}$
STS: $H; \mathsf{updt}\ t_1\ t_2\ t_3 \Downarrow_f^{k'',c} v', H'$ and $0 \le c \le \infty$ and $H' \vDash_{g'',k'-k''} \gamma$ and $v' \in [\![|\mathrm{unit}_r|_1]\!]_{g'',k'-k''}$
From $H; \mathsf{updt}\ t_1\ t_2\ t_3 \Downarrow^{k''}$, because t_1, t_2 are sub terms of $\sigma\ \mathsf{updt}\ t_1\ t_2\ t_3$,we get: $\sigma t_1 \Downarrow_f^{k_1} \wedge \sigma t_2 \Downarrow_f^{k_2} \wedge \sigma t_3 \Downarrow_f^{k_3}$ (1).
From (1), we get:

$$\exists l, n, v, k_1, k_2, k_3.\sigma t_1 \Downarrow^{k_1,c_1} l \wedge \sigma t_2 \Downarrow^{k_2,c_1'} n \wedge \sigma t_3 \Downarrow^{k_3,c_1''} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c_1' + c_1'' + c_{update} \tag{a}$$

From (a) and evaluation rules we get:$H; \sigma(\mathsf{updt}\ t_1\ t_2\ t_3) \Downarrow_f^{k'',c} H[l][n] \leftarrow v; ()(b)$.
By IH 3 on the first premise instantiated with $\sigma \in [\![|\delta\Gamma'|_1]\!]_{g,k''}$ using lemma 1 and $FV(t_2) \subseteq dom(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:$(\sigma t_1) \in [\![\mathsf{Array}_\gamma[I]\ |\tau|_1]\!]^{E,(0,\infty)}_{G',k'}$(2).
Similarly, By IH 3 on the second premise and third premise, we get: $(\sigma t_2) \in [\![|\mathrm{int}[I]|_1]\!]^{E,(0,\infty)}_{g',k'}$(3), $(\sigma t_3) \in [\![|\tau|_1]\!]^{E,(0,\infty)}_{g',k'}$(4). From (2),(3),(4), we know:

$$l \in [\![\mathsf{Array}_\gamma[I]\ A]\!]_{g',k'-k_1} \Rightarrow g'(\gamma) = (l, A, I) \wedge 0 \le c_1 \le \infty \tag{c}$$

$$n \in [\![\mathrm{int}[I]]\!]_{g,k'-k_2} \Rightarrow n = I \wedge 0 \le c_1' \le \infty \tag{d}$$

$$v \in [\![A]\!]_{g',k'-k_3} \wedge 0 \le c_1'' \le \infty \tag{e}$$

Let us assume: $g'' = g'(5)$, $H_1' = H(l)[n] \leftarrow v(6)$.

STS1: $(H_1') \vDash_{g',k'-k''} |P|_1 \star \gamma \to \mathbb{N}$
$H_1 = H_p \uplus H_l$, $H_1' = H_p \uplus H_l(l) \leftarrow v$, From assumption $H_1 \vDash |P|_1 \star \gamma \to \mathbb{N}$
STS: $H_l' = H_l(l) \leftarrow v \vDash \gamma \to \mathbb{N}$, which is proved by the assumption that $H_l \vDash \gamma \to \mathbb{N}$.

STS2: $(()) \in [\![\text{unit}]\!]_{g',k'-k''}$. It is proved by unfolding the definition.

STS3: $0 \le c \le \infty$. by the fact that $\delta L_u \le c_{update} \le \delta U_u$, it is proved by (c)(d)(e).

STS4: $\exists n.|P|_i \star \gamma \to \mathbb{N} = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
When $l_i \neq l$, the heap is not changed. it is true;
When $l_i = l$, $H(l)[n] \neq H'(l)[n]$, To show $n \in \beta$, which the premise in the typing rule.

This completes the proof of case unary update.

$$\text{CASE}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{return}\ t_1 \ominus \mathsf{return}\ t_2 \lesssim 0 : \{P\}\ \overset{\text{diff}(D)}{\exists\_.\tau}\ \{P\}}\ \text{R-T}$$

Assume that $\vDash \Phi\delta$ and there exists Γ' s.t. $\text{FV}(\mathsf{return}\ t_1) \subseteq dom(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in [\![|\delta\Gamma'|_1]\!]_{G|_1,k}$
TS: $\sigma\ \mathsf{return}\ t_1 \in [\![|\delta(\{|P|_1\}\ \overset{\text{exec}(0,\infty)}{\exists\_.|\tau|_1}\ \{|P|_1\})|_1]\!]^{E,(0,\infty)}_{G|_1,k}$
STS: $\sigma\ \mathsf{return}\ t_1 \in [\![|\delta(\{|P|_1\}\ \overset{\text{exec}(0,\infty)}{\exists\_.|\tau|_1}\ \{|P|_1\})|_1]\!]_{G|_1,k}$
Unfold the definition. Pick $g' \supseteq G|_1, k' \le k, k'' < k'$ and assume $H \vDash_{g',k'} |P|_1$ and $H; \mathsf{return}\ t_1 \Downarrow_f^{k''}$
STS: $H; \mathsf{return}\ t_1 \Downarrow_f^{k'',c} v', H'$ and $0 \le c \le \infty$ and $H' \vDash_{g'',k'-k''} \gamma$ and $v' \in [\![|\tau|_1]\!]_{g'',k'-k''}$
Because t_1 is the subterm of return t_1 ,we know: $\sigma_1 t_1 \Downarrow^{k''}$ (3). From (3), we know:

$$\exists v_1, k_1, \sigma_1 t_1 \Downarrow^{k_1,c_1} v_1 \wedge k'' = k_1 + 1 \wedge c = c_1 + c_{ret} \tag{a}$$

From (a),(b) and the evaluation rule **R-ret**: $H_1; \mathsf{return}\ t_1 \Downarrow_f^{k'',c} v_1; H_1(b)$.
By IH 3 on the first premise instantiated with $\sigma \in [\![|\delta\Gamma'|_1]\!]_{g,k''}$ using lemma 1 and $FV(t_2) \subseteq dom(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get: $(\sigma t_1) \in [\![|\delta\tau|_1]\!]^{E,(0,\infty)}_{G',k'}$ (5).
Unfold (5), we get: $v_1 \in (\!|\delta A|\!)_{g',k'-k_1} \wedge 0 \le c_1 \le \infty$(6). Let us assume: $g'' = g'$(7) $H'_1 = H_1$(8).

STS1: $H_1 \vDash_{g',k'-k''} |P|_1$. It is proved by the assumption on Lemma (monotonicity).

STS2: $(v_1) \in (\!|\tau|_1|\!)_{g',k'-k''}$. It is proved by (6) using Lemma 1.

STS3: $0 \le c \le \infty$ which is proved from (6)

STS4: $\exists n.|P|_1 = \{\gamma_1 \to T_1, \ldots, \gamma_n \to T_n\} \wedge \forall i \in [1,n].g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j.(H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
H is not changed, it is true.

This completes the proof of case unary return.

3 Example revisited

3.1 InPlaceMap

Example 1: In Place Map

$$\begin{array}{l}\mathsf{InPlaceMap} = \mathsf{fix\ map}\,(f).\lambda a.\lambda k.\lambda n.\\ \quad \mathsf{if}\ k \le n\ \mathsf{then}\ \big(\mathsf{let}\ \{x\} = \big(\mathsf{read}\ a\ k\big)\ \mathsf{in}\\ \qquad \mathsf{let}\{\_\} = \big(\mathsf{updt}\ a\ k\ (f\ x)\big)\ \mathsf{in}\ \big((\mathsf{celim}\ (\mathsf{map}\ f))\ a\ (k+1)\ n\big)\\ \quad \mathsf{else\ return}()\end{array}$$

As a first example we want to prove

$$\vdash \quad \mathsf{InPlaceMap} \quad : \quad \begin{array}{l}\forall L.\forall U : \mathbb{N}.(A \overset{\mathrm{exec}(L,U)}{\longrightarrow} A') \longrightarrow \forall n : \mathbb{N}.\ \forall i : \mathbb{N}.\ \forall \gamma : \mathbb{L}.\\ (i \le n) \supset \big(\mathsf{Array}_\gamma[n]\ A \longrightarrow \mathrm{int}[i] \longrightarrow \overset{\mathrm{exec}((L+c_r+c_u)*(n-i),(U+c_r+c_u)*(n-i))}{\{\gamma \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to \mathbb{N}\}}\big)\end{array}$$

where c_r is the cost for one reading operation and c_u denotes the cost for one updating operation.//

$$\vdash \mathsf{InPlaceMap} \ominus \mathsf{InPlaceMap} \lesssim 0 : \begin{array}{l}\forall r.\square(\tau \overset{\mathrm{diff}(r)}{\longrightarrow} \tau) \longrightarrow \forall n, i : \mathbb{N}.\ \forall \gamma : \mathsf{ref}.\ \forall \beta : \mathbb{P}.\\ (i \le n) \supset \big(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i] \longrightarrow \{\gamma \to \beta\}\ \exists\_.\mathrm{unit}\ \{\gamma \to \beta\}\big)\end{array}$$

$$\vdash \mathsf{InPlaceMap} \ominus \mathsf{InPlaceMap} \lesssim 0 : \begin{array}{l}\forall r.(\tau \overset{\mathrm{diff}(r)}{\longrightarrow} \tau) \longrightarrow \forall n, i : \mathbb{N}.\ \forall \gamma : \mathsf{ref}.\ \forall \beta : \mathbb{P}.\ (i \le n) \supset\\ \big(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i] \longrightarrow \{\gamma \to \beta\}\ \exists\_.\mathrm{unit}\ \{\gamma \to \beta \cup [i,n]\}\big)\end{array}$$

Let us call M the following term: $\big(\mathsf{celim}(\mathsf{map}\,f)\,a\,(k+1)\,n\big)$. and call T the following term: $\Big(\mathsf{let}\ \{x\} = \big(\mathsf{read}\ a\ k\big)\ \mathsf{in}\ \mathsf{let}\{\_\} = \big(\mathsf{updt}\ a\ i\ (f\ x)\big)\ \mathsf{in}\ \big(\mathsf{celim}(\mathsf{map}\,f)\,a\,(k+1)\,n\big)\Big)$.
and call Z the following term: $\mathsf{let}\{\_\} = \big(\mathsf{updt}\ a\ i\ (f\ x)\big)\ \mathsf{in}\big(\mathsf{celim}(\mathsf{map}\,f)\,a\,(k+1)\,n\big)$.
and set V as: if $k < n$,then T else (return ()).
Before we discussing the two types, we can apply simple rules first to make the relational type easy to read. We apply $R-iLam$ and $R-abs$ several times. Before we start to derivate the relational type of part V, We apply relational rule $R-Cimpl$ to introduce the constraint $i \le n$ into the constraint environment Φ.

Type 1: Same function We have two possible cases, one is that position $I \in \beta$ which means the values in the two arrays of two runs may be different, the other case is $I \notin \beta$, which means in index I, the values are the same in the two arrays:

Before we go to the two cases, we will first apply R-split. We need to show two cases and their types, with different constraint environment $\Phi \wedge i \in \beta$ and $\Phi \wedge i \notin \beta$ respectively.

Case 1: different values from read, $\Phi \wedge i \in \beta$

$$\frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i]\\ \Delta;\Phi \models i \le n \qquad \Delta;\vdash \mathsf{empty} \quad wf\end{array}}{\Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \overset{\mathrm{diff}(0)}{\exists\_}.\tau\ \{\gamma \to \beta\}}\ \text{R-R}$$

and

$$\frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i]\\ \Delta;\Phi;\Gamma, x : \tau \vdash f\ x \ominus f\ x \lesssim r : \tau \qquad \Delta;\Phi \models i \le n \qquad \Delta;\vdash \mathsf{empty} \quad wf\end{array}}{\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \overset{\mathrm{diff}(r)}{\exists\_} : \mathrm{unit}_r\ \{\gamma \to \beta \cup \{i\}\}}\ \text{R-U}$$

where $\Delta = n:\mathbb{N}, i:\mathbb{N}, r:\mathbb{R}$
$\Sigma = \gamma:\mathbb{L}, \Phi = i \le n$,
$\Gamma = a:\mathsf{Array}_\gamma[n]\ \tau, k:\mathrm{int}[i], f:\square(\tau \xrightarrow{\mathrm{diff}((r))} \tau), map:\square(\tau \xrightarrow{\mathrm{diff}((r))} \tau) \xrightarrow{\mathrm{diff}(0)} \forall n:\mathbb{N}.\ \forall i:\mathbb{N}.\ \forall \gamma:\mathbb{L}.\ (i \le n) \supset \left(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i] \longrightarrow \mathrm{int}[n] \longrightarrow \{\gamma \to \beta\}\ \mathrm{unit}\ \{\gamma \to \beta\}\right)^{\mathrm{diff}(|\beta \cap [i,n]| * r)}$.
Because at position i, $a_1[i] \neq a_2[i]$,s.t. $\beta \cup \{i\} = \beta$.
We will show:

$$\dfrac{\dfrac{\text{S-RM}}{\Delta;\Phi \vDash \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\} \sqsubseteq \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(0)}{}^{\mathrm{diff}(|\beta \cap [i,n]| * r)}}{\Delta;\Phi;\Gamma \vdash \mathsf{return}\ () \ominus \mathsf{return}\ () \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(0)}\ \text{R-EXEC}}{\Delta;\Phi;\Gamma \vdash \mathsf{return}\ () \ominus \mathsf{return}\ () \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(|\beta \cap [i,n]| * r)}$$

$$\dfrac{\Delta;\Phi;\Gamma \vdash k < n \ominus k < n \lesssim 0 : \mathrm{unit} + \mathrm{unit} \qquad \Delta;\Phi;\Gamma \vdash\ T \ominus\ T \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}{\Delta;\Phi;\Gamma,\{x:\tau\} \vdash V \ominus V \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(|\beta \cap [i,n]| * r)}\ \text{R-CASE}$$

Using R-Let rules, we get

$$\dfrac{P_2 = \mathsf{empty} \qquad \Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{int}\ \{\gamma \to \beta\}^{\mathrm{diff}(0)} \qquad \Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}^{\mathrm{diff}(|\beta \cap [i,n]| * r)}}{\Delta;\Phi;\Gamma,\{x:\mathrm{int}\} \vdash T \ominus T \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(|\beta \cap [i,n]| * r)}\ \text{R-LET}$$

,

$$\dfrac{P_2 = \mathsf{empty} \qquad \Delta;\Phi;\Gamma \vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}^{\mathrm{diff}(r)} \qquad \Delta;\Phi;\Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}^{\mathrm{diff}(|\beta \cap [i+1,n]| * r)}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}}^{\mathrm{diff}(|\beta \cap [i+1,n]| * r + r)}\ \text{R-LET}$$

Because i in β, we know: $|\beta \cap [i+1,n]| * r + r = |\beta \cap [i,n]|$ and $\beta \cup \{i\} = \beta$.
We want to show the typing of M. By applying R-IApp (when instantiating $\forall i$, we use i+1) and R-App(when instantiating $\lambda.k$, we use k+1), we can easily get

$$\dfrac{\dfrac{\text{R-CELIM}}{\Delta;\Phi \vDash i+1 \le n \qquad \Delta;\Phi\Gamma \vdash \left((\mathsf{map}\ f)\right) \ominus \left((\mathsf{map}\ f)\right) \lesssim 0 :}{(i+1 \le n) \supset \left(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i+1] \longrightarrow \mathrm{int}[n] \longrightarrow \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}\ \{\gamma \to \beta\}\right)^{\mathrm{diff}(|\beta \cap [i+1,n]| * r)}}}{\Delta;\Phi;\Gamma \vdash \left(\mathsf{celim}(\mathsf{map}\ f)\right) \ominus \left(\mathsf{celim}(\mathsf{map}\ f)\right) \lesssim 0 :\ \mathsf{Array}_\gamma[n]\ \tau \to \mathrm{int}[i+1] \to \mathrm{int}[n] \to \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}^{\mathrm{diff}(|\beta \cap [i+1,n]| * r)}}$$

,

$$\Delta;\Phi;\Gamma \vdash \left(\mathsf{celim}(\mathsf{map}\ f)\ a\ (k+1)\ n\right) \ominus \left(\mathsf{celim}(\mathsf{map}\ f)\ a\ (k+1)\ n\right) \lesssim 0 :\ \{\gamma \to \beta\}\ \exists\_:\mathrm{unit}_r\ \{\gamma \to \beta\}^{\mathrm{diff}(|\beta \cap [i+1,n]| * r)}$$

Case 2: the same value from read, $\Phi \wedge i \notin \beta$

$$\dfrac{\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i] \qquad \Delta;\Phi \vDash i \le n \qquad \Delta;\Phi \vDash i \notin \beta}{\Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \exists\_.\square\tau\ \{\gamma \to \beta\}}^{\mathrm{diff}(0)}\ \text{R-RB}$$

and

$$\dfrac{\dfrac{\dfrac{\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i]}{\Delta;\Phi \models \Box(\tau \xrightarrow{\mathrm{diff}((r))} \tau) \sqsubseteq \Box\tau \xrightarrow{\mathrm{diff}((0))} \Box\tau}\ \text{S-R-BOX-DIFF}}{\Delta;\Phi \vdash f \ominus f \lesssim 0 : \Box\tau \xrightarrow{\mathrm{diff}((0))} \Box\tau} \qquad \Delta;\Phi \models i \le n}{\Delta;\Phi;\Gamma x : \Box\tau \vdash f\ x \ominus f\ x \lesssim 0 : \Box\tau}\ \text{R-UB}$$

$$\Delta;\Phi;\Gamma,\vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{\gamma \to \beta/\{i\}\}$$

where $\Delta = n : \mathbb{N}, i : \mathbb{N}, r : \mathbb{R}$
$\Sigma = \gamma : \mathbb{L}, \Phi = i \le n, \Gamma = a$:$\mathsf{Array}_\gamma[n]\ \tau, k : \mathrm{int}[i], f : \Box(\tau \xrightarrow{\mathrm{diff}((r))} \tau), map : \Box(\tau \xrightarrow{\mathrm{diff}((r))} \tau) \xrightarrow{\mathrm{diff}(0)} \forall n : \mathbb{N}.\ \forall i : \mathbb{N}.\ \forall \gamma, \beta : .$
$(i \le n) \supset \big(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i] \longrightarrow \mathrm{int}[n] \longrightarrow \exists \beta.\ \{\gamma \xrightarrow{\mathrm{diff}(|\beta \cap [i,n]| * r)} \beta\}\ \mathrm{unit}\ \{\gamma \to \beta\}\big)$.
Because at position i, $a_1[i] \neq a_2[i]$,s.t. $\beta \cup \{i\} = \beta$.
Using R-Let rules, we get

$$\dfrac{P_2 = \mathsf{empty} \qquad \Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\mathrm{diff}(0)}{\mathrm{int}}\ \{\gamma \to \beta\} \qquad \Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{\gamma \to \beta\}}{\Delta;\Phi;\Gamma, \{x : \mathrm{int}\} \vdash T \ominus T \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{\gamma \to \beta\}}\ \text{R-LET}$$

,

$$\dfrac{P_2 = \mathsf{empty} \qquad \Delta;\Phi;\Gamma \vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\mathrm{diff}(0)}{\mathrm{unit}_r}\ \{\gamma \to \beta\} \qquad \Delta;\Phi;\Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{\gamma \to \beta\}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \mathrm{unit}_r\ \{\gamma \to \beta\}}\ \text{R-LET}$$

Because i not in β, we know: $|\beta \cap [i+1, n]| * r + 0 = |\beta \cap [i, n]|$.
The type of M is the same as we showed in previous case.

Type 2: Different functions :

$$\vdash \mathsf{InPlaceMap} \ominus \mathsf{InPlaceMap} \lesssim 0 : \begin{array}{l} \forall r : \mathbb{N}.(\tau \xrightarrow{\mathrm{diff}((r))} \tau) \longrightarrow \forall n : \mathbb{N}.\ \forall i : \mathbb{N}.\ \forall \gamma : \mathbb{L}.\ \forall \beta : \mathbb{P}. \\ (i \le n) \supset \big(\mathsf{Array}_\gamma[n]\ \tau \longrightarrow \mathrm{int}[i] \longrightarrow \{\gamma \to \beta\}\ \overset{\mathrm{diff}(((n-i)*r))}{\mathrm{unit}}\ \{\gamma \to \beta \cup [i,n]\}\big) \end{array}$$

Let start with read and update from index k.

$$\dfrac{\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i] \qquad \Delta;\Phi \models i \le n \qquad \Delta;\vdash \mathsf{empty} \quad wf}{\Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \overset{\mathrm{diff}(0)}{\exists\_.\tau}\ \{\gamma \to \beta\}}\ \text{R-R}$$

and

$$\dfrac{\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \qquad \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i] \qquad \Delta;\Phi;\Gamma, x : \tau \vdash f\ x \ominus f\ x \lesssim r : \tau \qquad \Delta;\Phi \models i \le n \qquad \Delta;\vdash \mathsf{empty} \quad wf}{\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\mathrm{diff}(r)}{\mathrm{unit}_r}\ \{\gamma \to \beta \cup \{i\}\}}\ \text{R-U}$$

where $\Delta = n : \mathbb{N}, i : \mathbb{N}, r : \mathbb{R}$
$\Sigma = \gamma : \mathbb{L}, \Phi = i \le n, \Gamma = a$:$\mathsf{Array}_\gamma[n]\ \tau, k : \mathrm{int}[i], f : (\tau \xrightarrow{\mathrm{diff}((r))} \tau)$,

$map : (\tau \xrightarrow{\text{diff}((r))} \tau) \to \forall n, i, \gamma, \beta.(i \le n) \supset \big(\mathsf{Array}_\gamma[n]\ \tau \to \text{int}[i] \to \text{int}[n]\ \{\gamma \to \beta\}\ \overset{\text{diff}((n-i)*r)}{\text{unit}}\ \{\gamma \to \beta \cup [i,n]\}\big)$.
We will show:

$$\dfrac{\dfrac{\text{S-RM}}{\models \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(0)}{\text{unit}_r}\ \{\gamma \to \beta\} \sqsubseteq \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}} \quad \Delta;\Phi;\Gamma \vdash \mathsf{return}\ () \ominus \mathsf{return}\ () \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(0)}{\text{unit}_r}\ \{\gamma \to \beta\}}{\Delta;\Phi;\Gamma \vdash \mathsf{return}\ () \ominus \mathsf{return}\ () \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}\ \text{R-EXEC}$$

$$\dfrac{\Delta;\Phi;\Gamma \vdash k < n \ominus k < n \lesssim 0 : \text{unit} + \text{unit} \qquad \Delta;\Phi;\Gamma \vdash\ T \ominus\ T \lesssim 0 : \{\gamma \to S\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}{\Delta;\Phi;\Gamma, \{x : \tau\} \vdash V \ominus V \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}\ \text{R-CASE}$$

Using R-Let rules,instantiate P_1 with $\{\gamma \to \beta\}$ and $P_2 = \mathsf{empty}, Q = \{\gamma \to \beta \cup [i,n]\}$, we get

$$\dfrac{\Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ k \ominus \mathsf{read}\ a\ k \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(0)}{\text{int}}\ \{\gamma \to \beta\} \quad \Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}{\Delta;\Phi;\Gamma, \{x : \text{int}\} \vdash T \ominus T \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}\ \text{R-LET}$$

Because we know: $(\beta \cup \{i\}) \cap [i+1,n] = \beta \cap [i+1,n]$ and $\beta \cup \{i\} \cup [i+1,n] = \beta \cup [i,n]$.
Using R-Let rules,instantiate P_1 with $\{\gamma \to \beta\}$ and $P_2 = \mathsf{empty}, Q_1 = \{\gamma \to \beta \cup \{i\}\}, Q_2 = \mathsf{empty}, Q = \{\gamma \to \beta \cup \{i\} \cup [i+1,n]\}$, we get

$$\dfrac{\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ a\ k\ (f\ x) \ominus \mathsf{updt}\ a\ k\ (f\ x) \lesssim 0 : \{\gamma \to \beta\}\ \exists\_ : \overset{\text{diff}(r)}{\text{unit}_r}\ \{\gamma \to \beta \cup \{i\}\} \quad \Delta;\Phi;\Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \to \beta \cup \{i\}\}\ \exists\_ : \overset{\text{diff}(|(\beta \cup \{i\}) \cap [i+1,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup \{i\} \cup [i+1,n]\}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to S\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i+1,n]| * r + r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}}\ \text{R-LET}$$

We want to show the typing of M. By applying $R - IApp$, instantiating β with $\beta \cup \{i\}$ and i with $i+1$. Then apply $R - App$ (instantiating k with $k+1$) several times, we can easily get

$$\Delta;\Phi;\Gamma \vdash \big(\mathsf{celim}(\mathsf{map}\ f)\ a\ (k+1)\ n\big) \ominus \big(\mathsf{celim}(\mathsf{map}\ f)\ a\ (k+1)\ n\big) \lesssim 0 : \{\gamma \to \beta \cup \{i\}\}\ \exists\_ : \overset{\text{diff}(|\beta \cap [i+1,n]| * r)}{\text{unit}_r}\ \{\gamma \to \beta \cup [i,n]\}$$

When apply **R-Celim**, We also check the constraint $i + 1 \le n$.

3.2 MergeSort

$$\begin{array}{l}
\mathsf{copy} = \mathsf{fix}\ cp\ (_).\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda y.\lambda l.\lambda u. \\
\quad \mathsf{if}\ l \le u\ \mathsf{then} \\
\quad\quad \mathsf{let}\ \{a\} = \big(\ \mathsf{read}\ x\ l\ \big)\ \mathsf{in} \\
\quad\quad \mathsf{let}\{_\} = \big(\ \mathsf{updt}\ y\ l\ a\ \big)\ \mathsf{in} \\
\quad\quad \big(\mathsf{celim}(cp()\ []\ []\ []\ []\ []\ []\big)\ x\ y\ (l+1)\ u\big) \\
\quad \mathsf{else}\ \ \mathsf{return}()
\end{array}$$

Type 1: copy $S \cap [l,u] = \emptyset$

$$\begin{array}{ll}
\vdash \mathsf{copy} \ominus \mathsf{copy} \lesssim 0 : & \forall n, L, U, \gamma_1, \gamma_2, S, S'.\ L \le U \le n \wedge S' \cap [L,U] = \emptyset \supset \\
& \mathsf{Array}_{\gamma_1}[n]\ U(\text{int},\text{int}) \to \mathsf{Array}_{\gamma_2}[n]\ U(\text{int},\text{int}) \\
& \to \text{int}[L] \to \text{int}[U] \to \{\gamma_1 \to S, \gamma_2 \to S'\}\ \overset{\text{diff}(0)}{\exists_.\text{unit}}\ \{\gamma_1 \to S, \gamma_2 \to \mathbb{N}\}
\end{array}$$

Type 2: copy $S \cap [l,u] \neq \emptyset$

$$\vdash \mathsf{copy} \ominus \mathsf{copy} \lesssim 0 : \begin{array}{l} \forall n, L, U, \gamma_1, \gamma_2, S.\ L \leq U \leq n \supset \mathsf{Array}_{\gamma_1}[n]\ U(\mathrm{int},\mathrm{int}) \to \mathsf{Array}_{\gamma_2}[n]\ U(\mathrm{int},\mathrm{int}) \\ \to \mathrm{int}[L] \to \mathrm{int}[U] \to \{\gamma_1 \to S, \gamma_2 \to \mathbb{N}\}\ \exists\_.\mathrm{unit}^{\mathrm{diff}(0)}\ \{\gamma_1 \to S \cup [L,U], \gamma_2 \to \mathbb{N}\} \end{array}$$

$$\begin{array}{l} \mathsf{merge}_{\mathsf{lp}} = \mathsf{fix}\ \mathsf{mg}\ (\_).\lambda.(k : \mathrm{int}[K]).\lambda a : \mathsf{Array}_{\gamma_1}[n]\ \mathrm{int}. \\ \quad \lambda l_s : \mathrm{int}[i], \lambda l_e : \mathrm{int}[j], \lambda r_s : \mathrm{int}[m], \lambda r_e : \mathrm{int}[u].\lambda b : \mathsf{Array}_{\gamma_2}[n]\ \mathrm{int}. \\ \quad \mathsf{if}\ l_s < l_e\ \mathsf{then} \\ \quad\quad \mathsf{if}\ r_s < r_e\ \mathsf{then} \\ \quad\quad\quad \mathsf{let}\ \{x\} = (\mathsf{read}\ a\ l_s)\ \mathsf{in} \\ \quad\quad\quad\quad \mathsf{let}\ \{y\} = (\mathsf{read}\ a\ r_s)\ \mathsf{in} \\ \quad\quad\quad\quad\quad \mathsf{if}\ x < y\ \mathsf{then} \\ \quad\quad\quad\quad\quad\quad \Big(\mathsf{let}\ \{\_\} = (\mathsf{updt}\ b\ k\ x)\ \mathsf{in} \\ \quad\quad\quad\quad\quad\quad\quad \big((\mathsf{celim}\,\mathsf{mg}\ ()\ (k+1))\ a\ (l_s+1)\ l_e\ r_s\ r_e\ b\big)\Big) \\ \quad\quad\quad\quad\quad \mathsf{else} \\ \quad\quad\quad\quad\quad\quad \Big(\mathsf{let}\ \{\_\} = (\mathsf{updt}\ b\ k\ y)\ \mathsf{in} \\ \quad\quad\quad\quad\quad\quad\quad \big((\mathsf{celim}\,\mathsf{mg}\ ()\ (k+1))\ a\ l_s\ l_e\ (r_s+1)\ r_e,\ b\big)\Big) \\ \quad\quad\quad \mathsf{else} \\ \quad\quad\quad\quad \Big(\mathsf{let}\{x\} = (\mathsf{read}\ a\ l_s)\ \mathsf{in} \\ \quad\quad\quad\quad\quad \mathsf{let}\{\_\} = (\mathsf{updt}\ b\ k\ x)\ \mathsf{in} \\ \quad\quad\quad\quad\quad\quad \big((\mathsf{celim}\,\mathsf{mg}\ ()\ (k+1))\ a\ (l_s+1)\ l_e\ r_s\ r_e\ b\big)\Big) \\ \quad\quad \mathsf{else} \\ \quad\quad\quad \mathsf{if}\ r_s < r_e \\ \quad\quad\quad\quad \Big(\mathsf{let}\{x\} = (\mathsf{read}\ a\ r_s)\ \mathsf{in} \\ \quad\quad\quad\quad\quad \mathsf{let}\{\_\} = (\mathsf{updt}\ b\ k\ x)\ \mathsf{in} \\ \quad\quad\quad\quad\quad\quad \big((\mathsf{celim}\,\mathsf{mg}\ ()\ (k+1))\ a\ l_s\ l_e\ (r_s+1)\ r_e\ b\big)\Big) \\ \quad\quad\quad \mathsf{else}\,(\mathsf{return}()) \end{array}$$

$$\vdash \mathsf{merge}_{\mathsf{lp}} : \begin{array}{l} \forall n, l_s, l_e, r_s, r_e, K, \gamma, \gamma'.\ l_s \leq l_e \leq r_s \leq r_e \leq n \supset \mathrm{unit}_r \to \mathrm{int}[K] \to \mathsf{Array}_{\gamma}[n]\ \mathrm{int} \to \mathrm{int}[l_s] \to \\ \mathrm{int}[l_e] \to \mathrm{int}[r_s] \to \mathrm{int}[r_e] \to \mathsf{Array}_{\gamma'}[n]\ \mathrm{int} \to \{\gamma \to \emptyset, \gamma' \to^{\mathrm{exec}(U,L)} [K, r_e]\}\ \exists\_.\mathrm{unit}\ \{\gamma, \gamma'\} \end{array}$$

$L = (c_r + c_r + c_u) * min(l_e - l_s, r_e - r_s) + (c_r + c_u) * max(l_e - l_r, r_e - e_s)$
$U = (c_r + c_r + c_u) * ((l_e - l_s) + (r_e - r_s))$

In $\mathsf{merge}_{\mathsf{lp}}$, we have 4 positions in the input array, l_s, l_e indicating the left part of the array, r_s, r_e for right part. we need to merge the two part, which means total $(l_e - l_s)$ elements for left part of the input array and $(r_e - r_s)$ elements for right part. In the body of $merge_{lp}$, for every element to be merged and added to the buffer array, there are two possible different branches: in one branch (if $l_s < l_e \wedge r_s < r_e$, which means we still have elements waiting to be merged in both left and right parts), the cost will be $c_r + c_r + c_u$, c_r for the cost of read from the input array, c_u for the cost of updating the buffer array. in

the other branch, (either $l_s < l_e$ or $r_s < r_e$, we need to only merge one part into buffer array), the cost is $c_r + c_u$.

For the L, the minimal possible cost is that we run out the shorter part ($min(l_e - l_s, r_e - r_s)$) with the cost of $c_r + c_r + c_u$ first and then merge the rest part with the cost of $c_r + c_u$. For the maximal cost, merge all the elements with the cost of $c_r + c_r + c_u$. If we consider the cost for update as 1 unit cost and cost for read is 1 unit cost, $c_r + c_r + c_u = 3, c_r + c_u = 2$.

Type 1: mergeLp $S \cap [K, r_e] = \emptyset$

$$\begin{aligned}\vdash \mathsf{merge_{lp}} \ominus \mathsf{merge_{lp}} \lesssim 0 : \; & \forall n, l_s, l_e, r_s, r_e, K, \gamma, \gamma'. l_s \le l_e \le r_s \le r_e \le n \wedge S \cap [K, r_e] = \emptyset \\ & \supset \mathrm{unit}_r \to \mathrm{int}[K] \to \mathsf{Array}_\gamma[n]\ U(\mathrm{int}, \mathrm{int}) \to \mathrm{int}[l_s] \to \mathrm{int}[l_e] \to \mathrm{int}[r_s] \\ & \to \mathrm{int}[r_e] \to \mathsf{Array}_{\gamma'}[n]\ U(\mathrm{int}, \mathrm{int}) \\ & \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\overset{\mathrm{diff}(0)}{\mathrm{unit}}\ \{\gamma \to S, \gamma' \to \mathbb{N} \setminus [l, u]\}\end{aligned}$$

Because in the range $[K, r_e]$, all the values in the two arrays are exact the same, the two programs will go to the same branch for every step.
In the read rule, i equals to either l_s or r_s.

$$\frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash a \ominus a \lesssim 0 : \mathsf{Array}_\gamma[n]\ \tau \\ \Delta;\Phi;\Gamma \vdash l_s[r_s] \ominus l_s[r_s] \lesssim 0 : \mathrm{int}[i] \qquad \Delta;\Phi \models i \le n \qquad \Delta;\Phi \vdash \mathsf{empty} \quad wf\end{array}}{\Delta;\Phi;\Gamma \vdash \mathsf{read}\ a\ l_s[r_s] \ominus \mathsf{read}\ a\ l_s[r_s] \lesssim 0 : \{\gamma \to S\}\ \overset{\mathrm{diff}(0)}{\exists\_.\tau}\ \{\gamma \to S\}}\ \text{R-R}$$

and

$$\frac{\begin{array}{c}\Delta;\Phi;\Gamma \vdash b \ominus b \lesssim 0 : \mathsf{Array}_{\gamma'}[n]\ \tau \\ \Delta;\Phi;\Gamma \vdash k \ominus k \lesssim 0 : \mathrm{int}[i] \qquad \Delta;\Phi;\Gamma, x : \tau \vdash x \ominus x \lesssim 0 : \mathrm{int} \qquad \Delta;\Phi \models i \le n \qquad \Delta;\Phi \vdash \mathsf{empty} \quad wf\end{array}}{\Delta;\Phi;\Gamma \vdash \mathsf{updt}\ b\ k\ x[y] \ominus \mathsf{updt}\ b\ k\ x[y] \lesssim 0 : \{\gamma' \to N\}\ \overset{\mathrm{diff}(0)}{\exists\_ : \mathrm{unit}_r}\ \{\gamma' \to N \cup [i]\}}\ \text{R-U}$$

Using R-Case for two times, we have four similar cases, we will have a look at one of them.

$$\Delta;\Phi;\Gamma \vdash (\mathsf{celim}(\mathsf{mg}\ ())\ k+1\ a\ (l_s+1)\ l_e\ r_s\ r_s \ominus (\mathsf{celim}(\mathsf{mg}\ ())\ k+1\ a\ (l_s+1)\ l_e\ r_s\ r_s \lesssim 0 :$$

Its type is $\{\gamma \to S, \gamma' \to N\}\ \overset{\mathrm{diff}(0)}{\exists\_ : \mathrm{unit}_r}\ \{\gamma \to S, \gamma' \to N\}$. By applying $R-Celim$ to check the constraint and then applying $R-Iapp$ and $R-App$ for several times.

Type 2: mergeLp $S \cap [K, r_e] \neq \emptyset$

Because if $S \cap [K, r_e] \neq \emptyset$, $merge_{lp}$ in the two runs are likely to go into different branches at any step, We will use **R-S** to handle this cases because the two runs are likely to go to different branches in every step.

$$\frac{\begin{array}{c}\Delta;\Phi;|\Gamma|_1 \vdash^0_0 \mathsf{merge_{lp}}\ ()\ k\ a\ l_s\ l_e\ r_s\ r_e\ b : \{\gamma \to \emptyset, \gamma' \to \overset{\mathrm{exec}(L,U)}{[K, r_e]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\} \\ \Delta;\Phi;|\Gamma|_2 \vdash^0_0 \mathsf{merge_{lp}}\ ()\ k\ a\ l_s\ l_e\ r_s\ r_e\ b : \{\gamma \to \emptyset, \gamma' \to \overset{\mathrm{exec}(L,U)}{[K, r_e]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\}\end{array}}{\begin{array}{c}\Delta;\Phi;\Gamma \vdash \mathsf{merge_{lp}}\ ()\ k\ a\ l_s\ l_e\ r_s\ r_e\ b \ominus \mathsf{merge_{lp}}\ ()\ k\ a\ l_s\ l_e\ r_s\ r_e\ b \lesssim 0 : \\ U(\{\gamma \to \emptyset, \gamma' \to \overset{\mathrm{exec}(L,U)}{[K, r_e]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\}, \{\gamma \to \emptyset, \gamma' \to \overset{\mathrm{exec}(L,U)}{[K, r_e]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\})\end{array}}\ \text{R-S}$$

and we know from **S-RUM**.

$$\begin{aligned}\models U(\{\gamma, \gamma' \to \overset{\mathrm{exec}(L,U)}{[K, r_e]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\}, \{\gamma, \gamma' \to \overset{\mathrm{exec}(L,U)}{[L, U]}\}\ \exists\_ : \mathrm{unit}\ \{\gamma, \gamma'\}) \sqsubseteq \\ \{\gamma \to S, \gamma' \to S'\}\ \overset{\mathrm{diff}(U-L)}{\exists\_ : \mathrm{unit}_r}\ \{\gamma \to S, \gamma' \to S' \cup [K, r_e]\}\end{aligned}$$

$$\vdash \mathsf{merge}_{\mathsf{lp}} \ominus \mathsf{merge}_{\mathsf{lp}} \lesssim 0 : \begin{array}{l} \forall n, l_s, l_e, r_s, r_e, K, \gamma, \gamma'. l_s \le l_e \le r_s \le r_e \le n \supset \mathrm{unit}_r \to \mathrm{int}[K] \to \mathsf{Array}_\gamma[n]\ U(\mathrm{int},\mathrm{int}) \\ \to \mathrm{int}[l_s] \to \mathrm{int}[l_e] \to \mathrm{int}[r_s] \to \mathrm{int}[r_e] \to \mathsf{Array}_{\gamma'}[n]\ U(\mathrm{int},\mathrm{int}) \\ \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N} \cup [K, r_e]\} \end{array}^{\mathrm{diff}((U-L))}$$

Function merge

$$\begin{array}{l} \mathsf{merge} = \lambda a.\lambda b.\lambda l.\lambda u.\lambda mid. \\ \quad \mathsf{let}\ \{\_\} = \big(celim\ (merge_{lp}\ ()\ l)\ a\ l\ mid\ mid+1\ u\ b\big)\ \mathsf{in} \\ \quad \big(celim\ \mathsf{copy}\ l\ )\ a\ b\ u\big). \end{array}$$

Type 1: merge $S \cap [K, r_e] = \emptyset \to S \cap [l, u] = \emptyset$

$$\vdash \mathsf{merge} \ominus \mathsf{merge} \lesssim 0 : \begin{array}{l} \forall n, l, m, u, \gamma, \gamma', S. l \le m \le u \le n \wedge S \cap [l, u] = \emptyset \supset \mathsf{Array}_\gamma[n]\ U(\mathrm{int},\mathrm{int}) \to \\ \mathsf{Array}_{\gamma'}[n]\ U(\mathrm{int},\mathrm{int}) \to \mathrm{int}[l] \to \mathrm{int}[m] \to \mathrm{int}[u] \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N}\} \end{array}^{\mathrm{diff}(0)}$$

R-LET

$$\vdash celim\ merge_{lp}\ l\ a\ l\ m\ (m+1)\ u\ b \ominus celim\ merge_{lp}\ l\ a\ l\ m\ (m+1)\ u\ b \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N} \setminus [l, u]\}^{\mathrm{diff}(0)}$$

$$\frac{\Delta;\Phi;\Gamma \vdash \mathsf{celim\ copy}\ l\ a\ b\ u \ominus \mathsf{celim\ copy}\ l\ a\ b\ u \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N} \setminus [l, u]\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N} \setminus [l, u]\}^{\mathrm{diff}(0)}}{\Delta;\Phi;\Gamma \vdash \mathsf{merge}\ a\ b\ l\ m\ u \ominus \mathsf{merge}\ a\ b\ l\ m\ u \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N}\}^{\mathrm{diff}(0)}}$$

we know: $\models \mathbb{N}/[l, u] \sqsubseteq \mathbb{N}$, By **S-RM**, we can get the post condition of $merge$ to be

$$\{\gamma \to S, \gamma' \to \mathbb{N}\}$$

Type 2:merge $S \cap [K, r_e] \neq \emptyset \to S \cap [l, u] \neq \emptyset$

$$\vdash \mathsf{merge} \ominus \mathsf{merge} \lesssim 0 : \begin{array}{l} \forall n, l, m, u, \gamma, \gamma', S. l \le m \le u \le n \supset \mathsf{Array}_\gamma[n]\ U(\mathrm{int},\mathrm{int}) \to \mathsf{Array}_{\gamma'}[n]\ U(\mathrm{int},\mathrm{int}) \to \mathrm{int}[l] \to \\ \mathrm{int}[m] \to \mathrm{int}[u] \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S \cup [l, u], \gamma' \to \mathbb{N}\} \end{array}^{\mathrm{diff}(max(m-l, u-(m+1)))}$$

R-LET

$$\Delta;\Phi;\Gamma \vdash \mathsf{celim\ merge}_{\mathsf{lp}}\ l\ a\ l\ m\ (m+1)\ u\ b \ominus \mathsf{celim\ merge}_{\mathsf{lp}}\ l\ a\ l\ m\ (m+1)\ u\ b \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S, \gamma' \to \mathbb{N} \cup [l, u]\}^{\mathrm{diff}(max(m-l, u-(m+1)))}$$

$$\frac{\Delta;\Phi;\Gamma \vdash \mathsf{celim}\ copy\ l\ a\ b\ u \ominus \mathsf{celim}\ copy\ l\ a\ b\ u \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S \cup [l, u], \gamma' \to \mathbb{N}\}^{\mathrm{diff}(0)}}{\Delta;\Phi;\Gamma \vdash \mathsf{merge}\ a\ b\ l\ m\ u \ominus \mathsf{merge}\ a\ b\ l\ m\ u \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\mathrm{unit}\ \{\gamma \to S \cup [l, u], \gamma' \to \mathbb{N}\}^{\mathrm{diff}(max(m-l, u-(m+1)))}}$$

msort

$$\begin{array}{l} \mathsf{msort} = \mathsf{fix\ sort}\ (a : \mathsf{Array}_{\gamma_1}[n]\ \tau)\ b : \mathsf{Array}_{\gamma_2}[n]\ \tau\ .\lambda l.\lambda u. \\ \quad \mathsf{if}\ (l \ge u)\ \mathsf{then} \\ \quad \big(\mathsf{return}()\big) \\ \quad \mathsf{else}\ \Big(\\ \quad \mathsf{let}\ \_ = \big(celim\ (sort)\ a\ b\ l\ (l + \lfloor (u-l)/2 \rfloor)\big)\ \mathsf{in} \\ \quad \mathsf{let}\ \_ = \big(celim\ (sort)\ a\ b\ (l + \lfloor (u-l)/2 \rfloor + 1)\ u\big)\ \mathsf{in} \\ \quad \big(celim\ merge\ a\ b\ l\ (l + \lfloor (u-l)/2 \rfloor)\ u\big)\Big) \end{array}$$

$$\vdash \quad \text{msort} \quad : \quad \begin{array}{l} \forall n, l, u, \gamma, \gamma'. l \le u \le n \supset \text{Array}_{\gamma}[n]\ \text{int} \\ \to \text{int}[l] \to \text{int}[u] \to \text{Array}_{\gamma'}[n]\ \text{int} \to \{\gamma \to [l,u], \gamma' \to [l,u]\}\ \exists\_.\text{unit}^{\text{exec}(0,(u-l)*log(u-l))}\ \{\gamma, \gamma'\} \end{array}$$

Let us call M the following term: celim $merge\ a\ b\ l\ (l+\lfloor (u-l)/2 \rfloor)\ u$, and call $S1$ the following term: celim $(sort)\ a\ b\ l\ l+\lfloor (u-l)/2 \rfloor$, and call $S2$ the following term: celim $(sort)\ a\ b\ l+\lfloor (u-l)/2 \rfloor + 1\ u$, and call Z the following term: let_ = $S1$ in let _ = $S2$ in M, and set V as: if $l = u$, then (return ())else Z .

Type 1: msort $S \cap [l,u] = \emptyset$

$$\vdash \text{msort} \ominus \text{msort} \lesssim 0 : \begin{array}{l} \forall n, l, u, \gamma, \gamma' : \mathbb{N}, S. l \le u \le n \wedge S \cap [l,u] = \emptyset \supset \text{Array}_{\gamma}[n]\ U(\text{int},\text{int}) \to \text{int}[l] \to \text{int}[u] \\ \to \text{Array}_{\gamma'}[n]\ U(\text{int},\text{int}) \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\text{unit}^{\text{diff}(0)}\ \{\gamma \to S, \gamma' \to \mathbb{N}\} \end{array}$$

$$\dfrac{\begin{array}{c} \Delta;\Phi;\Gamma \vdash S2 \ominus S2 \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\} \\ \Delta;\Phi;\Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\} \qquad P_2 = \text{empty} \end{array}}{\Delta;\Phi;\Gamma \vdash let\_ = S2\ in\ M \ominus let\_ = S2\ in M \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\}}\ \text{R-LET}$$

$$\dfrac{\begin{array}{c} \Delta;\Phi;\Gamma \vdash S1 \ominus S1 \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\} \qquad P_2 = \text{empty} \\ \Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\} \\ \Delta;\Phi;\Gamma \vdash l = u \ominus l = u \lesssim 0 : \text{unit} + \text{unit} \\ \Delta;\Phi;\Gamma \vdash \text{return}\ () \ominus \text{return}\ () \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\} \end{array}}{\Delta;\Phi;\Gamma \vdash V \ominus V \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma \to \mathbb{N}\}}\ \text{R-CASE}$$

Type 2: msort $S \cap [l,u] \neq \emptyset$

$$\vdash \text{msort} \ominus \text{msort} \lesssim 0 : \begin{array}{l} \forall n, l, u, \gamma, \gamma' : \mathbb{N}, S. l \le u \le n \supset \text{Array}_{\gamma}[n]\ U(\text{int},\text{int}) \to \text{int}[l] \to \text{int}[u] \\ \to \text{Array}_{\gamma'}[n]\ U(\text{int},\text{int}) \to \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_.\text{unit}^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\} \end{array}$$

where $Q(n,\alpha) = \Sigma_{i=0}^{H} h(\lceil 2^{i-1} \rceil).min(\alpha, 2^{H-i})$, $H = \lceil log_2(n) \rceil$

$$\dfrac{\begin{array}{c} \text{S-RM} \\ \hline \models \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S, \gamma' \to \mathbb{N}\} \sqsubseteq \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\} \\ \Delta;\Phi;\Gamma \vdash \text{return}\ () \ominus \text{return}\ () \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(0)}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\} \end{array}}{\Delta;\Phi;\Gamma \vdash \text{return}\ () \ominus \text{return}\ () \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\}}\ \text{R-EXEC}$$

$$\dfrac{\begin{array}{c} \Delta;\Phi;\Gamma \vdash \text{return}\ () \ominus \text{return}\ () \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\} \\ \Delta;\Phi;\Gamma \vdash l = u \ominus l = u \lesssim 0 : \text{unit} + \text{unit} \\ \Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\} \end{array}}{\Delta;\Phi;\Gamma \vdash V \ominus V \lesssim 0 : \{\gamma \to S, \gamma' \to \mathbb{N}\}\ \exists\_ : \text{unit}_r^{\text{diff}(Q(u-l+1,|S\cap[l,u]|))}\ \{\gamma \to S \cup [l,u], \gamma' \to \mathbb{N}\}}\ \text{R-CASE}$$

$$\frac{\begin{array}{c} P_2 = \mathsf{empty} \\ \Delta;\Phi;\Gamma \vdash S2 \ominus S2 \lesssim 0 : \{\gamma \to S \cup [l,\lfloor (u-l)/2 \rfloor], \gamma \to \mathbb{N}\}^{\mathrm{diff}(Q(u-l-\lfloor (u-l)/2 \rfloor, |S \cap [l+\lfloor (u-l)/2 \rfloor+1, u]|))} \exists\_ : \mathrm{unit}_r \ \{\gamma \to S \cup [l,u], \gamma \to \mathbb{N}\} \\ \Delta;\Phi;\Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \to S \cup [l,u], \gamma \to \mathbb{N}\}^{\mathrm{diff}(max((l+\lfloor (u-l)/2 \rfloor)-l, u-((l+\lfloor (u-l)/2 \rfloor)+1))} \exists\_ : \mathrm{unit}_r \ \{\gamma \to S \cup [l,u], \gamma \to \mathbb{N}\} \end{array}}{\begin{array}{c} \Delta;\Phi;\Gamma \vdash let\_ = S2 \ in \ M \ominus let\_ = S2 \ in M \lesssim 0 : \\ \{\gamma \to S \cup [l,\lfloor (u-l)/2 \rfloor], \gamma \to \mathbb{N}\}^{\mathrm{diff}(Q(u-l-\lfloor (u-l)/2 \rfloor, |S \cap [l+\lfloor (u-l)/2 \rfloor+1, u]|+(u-l))} \exists\_ : \mathrm{unit}_r \ \{\gamma \to S \cup [l,u], \gamma \to \mathbb{N}\} \end{array}} \text{R-LET}$$

$$\frac{\begin{array}{cc} P_2 = \mathsf{empty} & \Delta;\Phi;\Gamma \vdash S1 \ominus S1 \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}^{\mathrm{diff}(Q(\lfloor (u-l)/2 \rfloor+1, |S \cap [l, l+\lfloor (u-l)/2 \rfloor]|))} \exists\_ : \mathrm{unit}_r \ \{\gamma \to S \cup [l,\lfloor (u-l)/2 \rfloor], \gamma \to \mathbb{N}\} \end{array}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \to S, \gamma \to \mathbb{N}\}^{\mathrm{diff}(Q(u-l+1, |S \cap [l,u]|))} \exists\_ : \mathrm{unit}_r \ \{\gamma \to S \cup [l,u], \gamma \to \mathbb{N}\}} \text{R-LET}$$

If we simplify the relative cost for M, it is $max((\lfloor (u-l)/2 \rfloor), u-l-(\lfloor (u-l)/2 \rfloor)+1) = \lceil \frac{u-l}{2} \rceil$. Let us look at the simplified version of msort:

$$\begin{array}{rcl} msort(in,l,u,buf) & = & msort(in,l,l+\lfloor \frac{u-l}{2} \rfloor, buf); \\ & & msort(in,l+\lfloor \frac{u-l}{2} \rfloor+1,u,buf); \\ & & merge(in,l,l+\lfloor \frac{u-l}{2} \rfloor+1,u,buf). \end{array}$$

To show : $Q(\lfloor \frac{u-l}{2} \rfloor+1, |S \cap [l,l+\lfloor \frac{u-l}{2} \rfloor]|) +$
$Q(u-l-\lfloor \frac{u-l}{2} \rfloor, |S \cap [l+\lfloor \frac{u-l}{2} \rfloor+1,u]|) + \lceil \frac{u-l}{2} \rceil$
$\leq Q(u-l+1, |S \cap [l,u]|)$
Let us simplify the inequality with n=l-u, $\alpha = |S \cap [l,l+\lfloor \frac{u-l}{2} \rfloor]|$ and $\beta = |S \cap [l+\lfloor \frac{u-l}{2} \rfloor+1,u]|$:
$\lceil \frac{n}{2} \rceil$ + Q($\lfloor \frac{n}{2} \rfloor$,$\alpha$) + Q($\lceil \frac{n}{2} \rceil$,$\beta$) $\leq$ Q(n, $\alpha+\beta$), it is arithmetically tautology
$\alpha+\beta \neq 0$ because $S \cap [l,u] \neq \emptyset$

3.3 Naive String Search

Description: Given a long text string S :Array$_{\gamma_1}[N]$ int and a shorter substring W :Array$_{\gamma_2}[M]$ int to be searched in S. There is an int array P :Array$_{\gamma_3}[N]$ int which will store the matching result (1 as match, 0 as mismatch) for the same index as the text string. For example, for index 1, there is a match with W from index 1 in the text string S and $P[1] = 1$.

Helper function

Helper function will do the match process for one index m in s and update $p[m]$, i will start from 0 till the length of w (Q) if the prefix matches.

```
search = fix f (_).Λ.Λ.Λ.Λ.Λ.Λ.Λ.λs.λw.λm.λi.λl_w.λp.
           let {a} =read s (m + i) in
           let {b} =read w i in
              if (i + 1) == l_w then
                 if a == b then...(1)
                    updt p m 1
                 else
                    updt p m 0
              else
                 if a == b then...(2)
                    celim (f () [] [] [] [] [] [] []) w m (i + 1) l_w p
                 else
                    updt p m 0
```

$$\vdash \quad \mathsf{search} \quad : \quad \begin{array}{l} \mathrm{unit} \to \forall \gamma_1,\gamma_2,\gamma_3 : \mathbb{L}.\forall I,M,Q,N : \mathbb{N}.(I<Q<N \wedge M+I<N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \\ \mathsf{Array}_{\gamma_2}[Q]\ \mathrm{int} \to \mathrm{int}[M] \to \mathrm{int}[I] \to \mathrm{int}[Q] \to \mathsf{Array}_{\gamma_3}[N]\ \mathrm{int} \to \\ \{\gamma_1 \to \emptyset, \gamma_2 \to \emptyset, \gamma_3 \to \{M\}\}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \emptyset, \gamma_3 \to \{M\}\} \end{array}^{\mathrm{exec}(c_u+r,(Q-I)*r+c_u)}$$

Notice, one thing to mention is that in the unary type, in the precondition, $\gamma_3 \to \{M\}$, it means before helper is executed, it can only touch position M in p.

Where r is the cost of two reads and c_u is the cost for update, w has length Q, s,p have length N.

$$\vdash \mathsf{search} \ominus \mathsf{search} \lesssim 0 : \begin{array}{l} \mathrm{unit}_r \to \forall \gamma_1,\gamma_2,\gamma_3 : \mathbb{L}.\forall I,M,Q,N : \mathbb{N}.\forall \beta_2,\beta_3.(I<Q<N \wedge M+I<N) \subset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \\ \to \mathsf{Array}_{\gamma_2}[Q]\ \mathrm{int} \to \mathrm{int}[M] \to \mathrm{int}[I] \to \mathrm{int}[N] \to \mathrm{int}[Q] \to \mathsf{Array}_{\gamma_3}[N]\ \mathrm{int} \to \\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\} \end{array}^{\mathrm{diff}((Q-1-min(MIN(\beta_2 \cap [I,\infty)),Q-1))*r)}$$

where $MIN(beta : set)$ return the minimum element in the set $beta$.

First, we will apply rule R-Fix-Ext.

Set $\sigma = \mathrm{unit} \to \forall \gamma_1, \cdots \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}^{\mathrm{diff}((Q-1-min(MIN(\beta_2 \cap [I,\infty)),Q-1))*r)}$.

Set $A_1 = A_2 = \mathrm{unit} \to \forall \gamma_1\gamma_2,\gamma_3 : \mathbb{L}.\forall I,M,Q,N : \mathbb{N}.(I<Q<N \wedge M+I<N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to$
$\mathsf{Array}_{\gamma_2}[Q]\ \mathrm{int} \to \mathrm{int}[M] \to \mathrm{int}[I] \to \mathrm{int}[Q] \to \mathsf{Array}_{\gamma_3}[N]\ \mathrm{int} \to \{\gamma_3 \to \{M\}\}\ \exists\_.\mathrm{unit}\ \{\gamma_3 \to \{M\}\}^{\mathrm{exec}(c_u,(Q-I-1)*r+c_u)}$.

Set $\Gamma = \_ : \mathrm{unit}, f : \mathrm{unit} \xrightarrow{\mathrm{diff}(0)} \sigma, f\ U(A_1, A_2)$

$$\frac{\Gamma \vdash \Lambda.\ldots.\lambda s.\ldots. \ominus \Lambda.\ldots.\lambda s.\ldots. \lesssim 0 : \sigma \qquad |\Gamma|_1 \vdash_0^0 \mathsf{Fix}\, f(\_).\Lambda.\ldots.\lambda s.\ldots. : A_1 \qquad |\Gamma|_2 \vdash_0^0 \mathsf{Fix}\, f(\_).\Lambda.\ldots.\lambda s.\ldots. : A_2}{\Gamma \vdash \mathsf{Fix}\, f(\_).\Lambda.\ldots. \ominus \mathsf{Fix}\, f(\_).\Lambda.\ldots. \lesssim 0 : \mathrm{unit} \xrightarrow{\mathrm{diff}(0)} \sigma}\ \text{R-FIX-EXT}$$

We first show how do we get the unary type A_1.
Set $A_1 = \mathrm{unit} \to B_1$. So $B_1 = \forall \gamma_1\gamma_2,\gamma_3 : \mathbb{L}.\ldots.$, Set $\Omega = |\Gamma|_1 = \_ : \mathrm{unit}, f : A_1$.

unary type
First, we apply U-fix rule.

$$\frac{\Omega \vdash_0^0 \Lambda.\ldots.\lambda S.\ldots. : B_1}{\Omega \vdash_0^0 \mathsf{Fix}\, f(\_).\Lambda.\ldots. : A_1}\ \text{U-FIX}$$

Then, we apply rule U-ILam several times to introduce new index term into sort environment Δ.
Set $\Delta = \gamma_1 : \mathbb{L}, \gamma_2 : \mathbb{L}, \gamma_2 : \mathbb{L}, I : \mathbb{N}, M : \mathbb{N}, N : \mathbb{N}, Q : \mathbb{N}$.
To introduce the constraint, apply U-CIMPL. $\Phi = (I<Q<N \wedge M+I<N)$.
Apply U-Abs several times to eliminate all the lambdas, we introduce variables into the unary context,
$\Omega = s : \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int}, w : \mathsf{Array}_{\gamma_2}[Q]\ \mathrm{int}, m : \mathrm{int}[M], i : \mathrm{int}[I], l_w : \mathrm{int}[Q], p : \mathsf{Array}_{\gamma_3}[N]\ \mathrm{int}$. we will show:

$$\frac{\begin{array}{c} \{\gamma_3 \to \{M\}\} = \{\gamma_3 \to \{M\}\} \star \mathsf{empty} \qquad \vdash_0^0 \mathsf{read}\ s\ (m+i) : \{\gamma_3 \to \{M\}\}\ \exists\_.\mathrm{int}\ \{\gamma_3 \to \{M\}\}^{\mathrm{exec}(c_r,c_r)} \\ \Delta;\Phi;a : \mathrm{int}, \Omega \vdash_0^0 \mathsf{let}\,\{b\} = \mathsf{read}\ b\ (i) \cdots : \{\gamma_3 \to \{M\}\}\ \exists\_.\mathrm{int}\ \{\gamma_3 \to \{M\}\}^{\mathrm{exec}(c_r+c_u,c_r+(Q-I-1)*r+c_u)} \end{array}}{\Delta;\Phi;\Omega \vdash \mathsf{let}\,\{a\} = \mathsf{read}\ s\ (m+i) \ldots : \{\gamma_3 \to \{M\}\}\ \exists\_.\mathrm{unit}\ \{\gamma_3 \to \{M\}\}^{\mathrm{exec}(c_u+r,(Q-I)*r+c_u)}}\ \text{U-LET}$$

We apply U-Let again to introduce b into the unary environment. Set $\Omega' = a : \text{int}, b : \text{int}, \Omega$.
We need to show the unary type of the first if term.

$$\frac{\Delta;\Phi;\Omega' \vdash_0^0 \text{if } (a == b)(1) : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \mathbb{N}\} \quad \Delta;\Phi;\Omega' \vdash_0^0 \text{if } (a == b)(2) : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \quad \vdash_0^0 i+1 == l_w : \text{bool}}{\Delta;\Phi;\Omega' \vdash_0^0 \text{if } (i+1) == l_w \text{ then if } (a == b)(1) \text{ elseif } (a == b)(2) : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\}} \text{ U-CASE}$$

For if (2):

$$\frac{\Delta;\Phi;\Omega' \vdash_0^0 \text{celim}\,(f\ ()\ []\ []\ []\ []\ []\ []\ [])\ w\ m\ (i+1)\ l_w\ p : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \quad \Delta;\Phi;\Omega' \vdash_0^0 \text{updt}\ p\ m\ 0 : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \quad \vdash_0^0 a == b : \text{bool}}{\Delta;\Phi;\Omega' \vdash_0^0 \text{if } (a == b) \text{ then celim}\,(f\ ()\ []\ []\ []\ []\ []\ []\ [])\ w\ m\ (i+1)\ l_w\ p \text{ else updt}\ p\ m\ 0 : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\}} \text{ U-CASE}$$

By applying U-app, U-Iapp, U-celim, we get the type for celim $(f\ ()\ []\ []\ []\ []\ []\ []\ [])\ w\ m\ (i+1)\ l_w\ p$ is $\{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\}$
Along with the constraint $I+1 < Q < N \wedge M+I+1 < N$.
For the other branch, the term updt $p\ m\ 0$. We use **U-X** and subtyping rule **S-A-MONAD**.

$$\frac{\Delta;\Phi;\Omega' \vdash_0^0 \text{updt}\ p\ m\ 0 : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \quad \Delta;\Phi \vDash \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \sqsubseteq \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\} \quad \Delta \vDash 0 \le 0 \quad \Delta \vDash 0 \le 0}{\Delta;\Phi;\Omega' \vdash_0^0 \text{updt}\ p\ m\ 0 : \{\gamma_3 \to \{M\}\}^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \exists\_.\text{int}\ \{\gamma_3 \to \{M\}\}} \text{ U-EXEC}$$

For if (1), it is similar by using U-exec and subtyping rule **S-A-MONAD** to make its type consistent with if (2).

Relational type

There are two cases we need to consider for relational types. We first apply some simple relational rules R-ILAM and R-ABS. Subscription 1 and 2 are used for the two runs. We apply R-CIMPL to introduce the constraints into the constraint environment Φ.

Set Z as the body of the function helper. where $C = (I < Q < N \wedge M + I < N)$ and
$\tau = \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}^{\text{diff}((Q-1-min(MIN(\beta_2 \cap [I,\infty)),Q-1))*r)} \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}$.

To show the type of part Z, we will use the rule R-SPLIT first.

$$\frac{\Sigma;\Delta;\Phi \wedge C \wedge I \notin \beta_2;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,1) \quad \Sigma;\Delta;\Phi \wedge C \wedge I \in \beta_2;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,2)}{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau} \text{ R-SPLIT}$$

Case 1 $I \notin \beta_2$

Because $I \notin \beta_2 \Rightarrow b_1 = b_2$. We know $a_1 = a_2, m_1 = m_2$ and $i_1 = i_2$. The two runs will go to the same path,, in the rest part of the proof, we think $\Phi' = \Phi \wedge (I < Q < N \wedge M + I < N) \wedge i \notin \beta_2$. After two reads, we will reach a if conditional.

R-READ

$$\frac{\Delta;\Phi';\Gamma \vdash a \ominus a \lesssim 0 : \text{Array}_\gamma[n]\ \tau \quad \Delta;\Phi';\Gamma \vdash k \ominus k \lesssim 0 : \text{int}[i] \quad \Delta;\Phi' \vDash i \le n \quad \Delta;\vdash \text{empty} \quad wf}{\Delta;\Phi';\Gamma \vdash \text{read}\ S\ (m+i) \ominus \text{read}\ S\ (m+i) \lesssim 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}^{\text{diff}(0)} \exists\_.\text{int}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}}$$

Set R as the outer if if$(i+1 == l_w)$then (1) else (2).
Set V as let b =read W i in R.

$$\Delta;\Phi \vdash i+1 == l_w \ominus i+1 == l_w \lesssim 0 : \text{unit} + \text{unit}$$

$$\frac{\Delta;\Phi;\Gamma \vdash (1) \ominus (1) \lesssim^{\text{diff}((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)} 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\} \quad \Delta;\Phi;\Gamma \vdash (2) \ominus (2) \lesssim^{\text{diff}((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)} 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}}{\Delta;\Phi;\Gamma \vdash R \ominus R \lesssim^{\text{diff}((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)} 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}}\ \text{R-CASE}$$

$$\frac{\Delta;\Phi;\Gamma \vdash \text{read } W\ i \ominus \text{read } W\ i \lesssim^{\text{diff}(0)} 0 : \{\gamma_2 \to \beta_2\}\ \exists\_ : \text{int}\ \{\gamma_2 \to \beta_2\} \qquad P_2 = \text{empty}}{\Delta;\Phi;\Gamma \vdash V \ominus V \lesssim^{\text{diff}((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)} 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}}\ \text{R-LET}$$

$$\frac{\Delta;\Phi;\Gamma \vdash \text{read } S\ (m+i) \ominus \text{read } S\ (m+i) \lesssim^{\text{diff}(0)} 0 : \{\gamma_1 \to \emptyset\}\ \exists\_ : \text{int}\ \{\gamma_1 \to \emptyset\} \qquad P_2 = \text{empty}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim^{\text{diff}((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)} 0 : \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}}\ \text{R-LET}$$

(a) The relative cost for (1) is 0, because the cost of update is the same. We use sub-typing rule **S-RM** to make the types of two branches consistent.

(b)The relative cost of (2) depends on its first branch , which is a recursive call of the f function. We know that the relative cost is $(Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r$ by using relational typing rules R-APP and R-Iapp . Because we already know $i \notin \beta$, it is easy to infer that $MIN(\beta_2\cap[I,\infty)) = MIN(\beta_2\cap[I+1,\infty))$.

Case 2 $I \in \beta_2$.

When $i \in \beta_2$, b_1 may not be the same as b_2, which means the two runs may go to different branches when reaching the if conditional. So we will switch to unary typing. From **R-S**, we know the relative cost of helper is $(Q-I-1)*r$, the type is still $\{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{$ with diff$((Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1))*r)$.
Because $i \in \beta_2$, $(Q-1-min(MIN(\beta_2\cap[I,\infty)),Q-1)) = (Q-I-1)$.

We get the unary type of function helper from the context $|\Gamma|$ and apply U-IAPP and U-APP to get the unary type of the body:

$$\{\gamma_3 \to \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}$$

By applying unary subtyping rule **S-UM**, we know the following unary typing used in **R-S**.

$$\frac{\Delta;\Phi;|\Gamma|_1 \vdash_0^0 (1) : \{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\} \quad \Delta;\Phi;|\Gamma|_2 \vdash_0^0 (1) : \{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\}}{\Delta;\Phi;\Gamma \vdash (1) \ominus (1) \lesssim 0 : \quad U(\{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\}, \{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\})}\ \text{R-S}$$

$$\models U(\{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\}, \{\gamma_3 \to^{\text{exec}(c_u,(Q-I-1)*r+c_u)} \{M\}\}\ \exists\_.\text{unit}\ \{\gamma_3 \to \{M\}\}) \sqsubseteq$$

$$\{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3\}\ \exists\_.U(\text{unit},\text{unit})^{\text{diff}((Q-I-1)*r)}\ \{\gamma_1 \to \emptyset, \gamma_2 \to \beta_2, \gamma_3 \to \beta_3 \cup \{M\}\}$$

By subtyping rule **S-R-UNIT**, we get the relational type.

NSS function

```
NSS = fix F (S). λW.λm.λl_s.λl_w.λP.
        if (m + l_w) ≤ l_s then
          let {_} = (celim search S W m 0 l_s l_w P) in
          F(S) W (m+1) l_s l_w P
        else
          return ()
```

$$\vdash \mathsf{NSS} \ominus \mathsf{NSS} \lesssim 0 : \begin{array}{l} \forall M.\forall \gamma_1,\gamma_2,\gamma_3 : \mathbb{L}.\forall Q,N:\mathbb{N}.(Q \le N \wedge M+Q \le N) \subset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \rightarrow \\ \mathsf{Array}_{\gamma_2}[Q]\ \mathrm{int} \rightarrow \mathrm{int}[M] \rightarrow \mathrm{int}[N] \rightarrow \mathrm{int}[Q] \rightarrow \mathsf{Array}_{\gamma_3}[N]\ \mathrm{int} \rightarrow \\ \overset{\mathrm{diff}((Q-1-min(MIN(\beta_2\cap[0,\infty),Q-1))*r*(N-M-Q))}{\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\}}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup [M,N]\} \end{array}$$

For the type of NSS, we first apply R-FIX. Then we apply R-ILAM and R-ABS multiple times. Then after we introduce the constraints $(Q \le N \wedge M+Q \le N)$ into the constraint environment Φ, the cost comes from the first branch.
Set K as (celim helper $S\ W\ m\ 0\ l_s\ l_w\ P$) , Set Q as $F(S)\ W\ (m+1)\ l_s\ l_w\ P$.
Set X as let _ = (celim helper $S\ W\ m\ 0\ l_s\ l_w\ P$) in
$F(S)\ W\ (m+1)\ l_s\ l_w\ P$.

R-LET

$$\frac{\begin{array}{c} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} = \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} * \mathsf{empty} \\ \Delta;\Phi;\Gamma \vdash K \ominus K \lesssim 0 : \overset{\mathrm{diff}((Q-1-min(MIN(\beta_2\cap[0,\infty),Q-1))*r)}{\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\}}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\} \\ \Delta;\Phi;\Gamma \vdash Q \ominus Q \lesssim 0 : \overset{\mathrm{diff}((Q-1-min(MIN(\beta_2\cap[0,\infty),Q-1))*r*(N-(M+1)-Q))}{\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\} \cup [M+1,N]\} \end{array}}{\Delta;\Phi;\Gamma \vdash X \ominus X \lesssim 0 : \overset{\mathrm{diff}((Q-1-min(MIN(\beta_2\cap[0,\infty),Q-1))*r*(N-M-Q))}{\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\}}\ \exists\_.\mathrm{unit}\ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup [M,N]\}}$$

3.4 Boolean Or

```
BoolOr = fix f (_).Λ.Λ.Λ.Λ.λs. λm.λl_s.
           if m < l_s then
              let {a} =read s m in
                  if a then
                     return true
                  else
                     (celim f () [] [] [] []) s (m+1) l_s
           else
              return false
```

$$\vdash \quad \mathsf{BoolOr} \quad : \quad \begin{array}{l} \mathrm{unit} \rightarrow \forall \gamma_1 : \mathbb{L}.\forall M,N:\mathbb{N}.(M \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{bool} \rightarrow \\ \mathrm{int}[M] \rightarrow \mathrm{int}[N] \rightarrow \\ \overset{\mathrm{exec}(3,(N-M)*3+1)}{\{\gamma_1 \rightarrow \emptyset\}}\ \exists\_.\mathrm{bool}\ \{\gamma_1 \rightarrow \emptyset\} \end{array}$$

$$\vdash \mathsf{BoolOr} \ominus \mathsf{BoolOr} \lesssim 0 : \begin{array}{l} \mathrm{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}.\forall M,N:\mathbb{N}.\forall \beta_1.(M \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ U(\mathrm{bool},\mathrm{bool}) \\ \rightarrow \mathrm{int}[M] \rightarrow \mathrm{int}[N] \rightarrow \\ \overset{\mathrm{diff}((N-1-min(MIN(\beta_1\cap[M,\infty)),N))*3+1)}{\{\gamma_1 \rightarrow \beta_1\}}\ \exists\_.U(\mathrm{bool},\mathrm{bool})\ \{\gamma_1 \rightarrow \beta_1\} \end{array}$$

For the relational type, we show the type derivation on case analysis using R-Split rule.

Set Z as the body of the function. where $C = (M \le N)$ and $\tau = \overset{\mathrm{diff}((N-1-min(MIN(\beta_1\cap[M,\infty),N))*3+1)}{\{\gamma_1 \rightarrow \beta_1\}}\ \exists\_.U(\mathrm{bool},\mathrm{bool})\ \{\gamma_1 \rightarrow \beta_1\}$.

To show the type of part Z, we will use the rule R-SPLIT first.

$$\frac{\begin{array}{l} \Sigma;\Delta;\Phi \wedge C \wedge M \notin \beta_1;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,1) \\ \Sigma;\Delta;\Phi \wedge C \wedge M \in \beta_1;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,2) \end{array}}{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau}\ \text{R-SPLIT}$$

Case 1 $M \notin \beta_1$

Because $M \notin \beta_1$. We know the two runs will go to the same path and we use relational typing rules to obtain the relational type of the body Z. After applying the rule R-Case, we need to show the type of the two branches respectively. Set the first branch as (1), the second (return false) as (2).

$$\frac{\Delta;\Phi \vdash a \ominus a \lesssim 0 : \text{unit} + \text{unit} \qquad \Delta;\Phi;\Gamma \vdash (1) \ominus (1) \lesssim 0 : \{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool},\text{bool})\ \{\gamma_1 \to \beta_1\}^{\text{diff}((N-1-min(MIN(\beta_1 \cap [M,\infty)),N))*3+1)} \quad \Delta;\Phi;\Gamma \vdash (2) \ominus (2) \lesssim 0 : \{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool},\text{bool})\ \{\gamma_1 \to \beta_1\}^{\text{diff}((N-1-min(MIN(\beta_1 \cap [M,\infty)),N))*3+1)}}{\Delta;\Phi;\Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool},\text{bool})\ \{\gamma_1 \to \beta_1\}^{\text{diff}((N-1-min(MIN(\beta_1 \cap [M,\infty)),N))*3+1)}}\ \text{R-CASE}$$

(a) The relative cost for (2) is 0, because return false causes no relative cost. We use sub-typing rule **S-RM** to make the types of two branches consistent.

(b)The relative cost of (1) depends on its second branch , which is a recursive call of the f function. We know that the relative cost is $(N - min(MIN(\beta_1 \cap [M,\infty)), N)) * r$ by using relational typing rules R-APP and R-Iapp . Because we already know $M \notin \beta_1$, it is easy to infer that $MIN(\beta_1 \cap [M,\infty)) = MIN(\beta_1 \cap [M+1,\infty))$.

Case 2 $M \in \beta_1$.

When $M \in \beta_1$, the two runs may go to different branches when reaching the if conditional. So we will switch to unary typing. From **R-S**, we know the relative cost of the function is $(N-M-1)*3+1$, the type is still $\{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool},\text{bool})\ \{\gamma_1 \to \beta_1\}^{\text{diff}((N-1-min(MIN(\beta_1 \cap [M,\infty)),N))*3+1)}$ for the reason that under the assumption of $M \in \beta_1$, $(N-1-min(MIN(\beta_1 \cap [M,\infty)), N)) = (N-1-M)$.

We get the unary type of function body from the context $|\Gamma|$ and apply U-IAPP and U-APP to get the unary type of the body:

$$\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)}$$

By applying unary subtyping rule **S-UM**, we know the following unary typing used in **R-S**.

$$\frac{\Delta;\Phi;|\Gamma|_1 \vdash_0^0 (1) : \{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)} \qquad \Delta;\Phi;|\Gamma|_2 \vdash_0^0 (1) : \{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)}}{\Delta;\Phi;\Gamma \vdash (1) \ominus (1) \lesssim 0 : \quad U(\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)}, \{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)})}\ \text{R-S}$$

$$\models U(\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)}, \{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}\ \{\gamma_1 \to \emptyset\}^{\text{exec}(3,(N-M)*3+1)}) \sqsubseteq$$

$$\{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool},\text{bool})\ \{\gamma_1 \to \beta_1\}^{\text{diff}((N-1-M)*3+1)}$$

3.5 Boolean Or, two implementations

We have two implementaions for the above Boolean or functions.

$$
\begin{array}{l}
\mathsf{fix}\,\mathsf{BoolOr1}\,(\_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda s.\ \lambda m.\lambda l_s.\\
\quad \mathsf{if}\ m < l_s\ \mathsf{then}\\
\quad\quad \mathsf{let}\ \{a\} = \mathsf{read}\ s\ m\ \mathsf{in}\\
\quad\quad\quad \mathsf{if}\ a\ \mathsf{then}\\
\quad\quad\quad\quad \mathsf{return}\ true\\
\quad\quad\quad \mathsf{else}\\
\quad\quad\quad\quad (\mathsf{celim}\ \mathsf{BoolOr1}\,()\,[]\,[]\,[]\,[])\ s\,(m+1)\ l_s\\
\quad \mathsf{else}\\
\quad\quad \mathsf{return}\ false
\end{array}
$$

The unary type of BoolOr1 as follows.

$$
\vdash \quad \mathsf{BoolOr1} \quad : \quad
\begin{array}{l}
\mathrm{unit} \to \forall \gamma_1 : \mathbb{L}.\forall M, N : \mathbb{N}.(M \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{bool} \to \\
\mathrm{int}[M] \to \mathrm{int}[N] \to \{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}^{\mathrm{exec}(3,(N-M)*3+1)}\ \{\gamma_1 \to \emptyset\}
\end{array}
$$

The following BoolOr2 is another implenmentation of BoolOr.

$$
\begin{array}{l}
\mathsf{fix}\,\mathsf{BoolOr2}\,(\_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda s.\ \lambda m.\lambda l_s.\\
\quad \mathsf{if}\ m < l_s\ \mathsf{then}\\
\quad\quad \mathsf{let}\ \{a\} = \mathsf{read}\ s\ m\ \mathsf{in}\\
\quad\quad\quad \mathsf{if}\ (\mathsf{celim}\ \mathsf{BoolOr2}\,()\,[]\,[]\,[]\,[])\ s\,(m+1)\ l_s)\ \mathsf{then}\\
\quad\quad\quad\quad \mathsf{return}\ true\\
\quad\quad\quad \mathsf{else}\\
\quad\quad\quad\quad \mathsf{return}\ a\\
\quad \mathsf{else}\\
\quad\quad \mathsf{return}\ false
\end{array}
$$

With its unary type presented below.

$$
\vdash \quad \mathsf{BoolOr2} \quad : \quad
\begin{array}{l}
\mathrm{unit} \to \forall \gamma_1 : \mathbb{L}.\forall M, N : \mathbb{N}.(M \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{bool} \to \\
\mathrm{int}[M] \to \mathrm{int}[N] \to \\
\{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}^{\mathrm{exec}(3*(N-M)+1,3*(N-M)+1)}\ \{\gamma_1 \to \emptyset\}
\end{array}
$$

We can obtain the relational type with relative cost 0.

$$
\vdash \mathsf{BoolOr1} \ominus \mathsf{BoolOr2} \lesssim 0 : \quad
\begin{array}{l}
\mathrm{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall M, N : \mathbb{N}.\forall \beta_1.(M \le N) \supset U(\mathsf{Array}_{\gamma_1}[N]\ \mathrm{bool}, \mathsf{Array}_{\gamma_1}[N]\ \mathrm{bool}) \\
\to U(\mathrm{int}[M], \mathrm{int}[M]) \to U(\mathrm{int}[N], \mathrm{int}[N]) \to \{\gamma_1 \to \beta_1\}\ \exists\_.U(\mathrm{bool}, \mathrm{bool})^{\mathrm{diff}(0)}\ \{\gamma_1 \to \beta_1\}
\end{array}
$$

The derivation mainly relies on the switch rules.

$$
\frac{\begin{array}{c}
\Delta;\Phi;|\Gamma|_1 \vdash^0_0 \mathsf{BoolOr1} : A \to^{\mathrm{exec}(3,(N-M)*3+1)} \{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}\ \{\gamma_1 \to \emptyset\} \\
\Delta;\Phi;|\Gamma|_2 \vdash^0_0 \mathsf{BoolOr2} : A \to^{\mathrm{exec}((N-M)*3+1,(N-M)*3+1)} \{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}\ \{\gamma_1 \to \emptyset\}
\end{array}}{\begin{array}{c}
\Delta;\Phi;\Gamma \vdash \mathsf{BoolOr1} \ominus \mathsf{BoolOr2} \lesssim 0 : \\
U(A \to^{\mathrm{exec}(3,(N-M)*3+1)} \{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}\ \{\gamma_1 \to \emptyset\}, A \to^{\mathrm{exec}((N-M)*3+1,(N-M)*3+1)} \{\gamma_1 \to \emptyset\}\ \exists\_.\mathrm{bool}\ \{\gamma_1 \to \emptyset\})
\end{array}}\ \text{R-S}
$$

We set A to be $\text{unit} \to \forall \gamma_1 : \mathbb{L}.\forall M, N : \mathbb{N}.(M \leq N) \supset \mathsf{Array}_{\gamma_1}[N]\ \text{bool} \to \text{int}[M] \to \text{int}[N] \to$.

We set A_1 to be $\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}^{\text{exec}(3,(N-M)*3+1)}\ \{\gamma_1 \to \emptyset\}$. set A_2 to be $\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}^{\text{exec}((N-M)*3+1,(N-M)*3+1)}\ \{\gamma_1 \to \emptyset\}$.
We set τ to be $\text{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall M, N : \mathbb{N}.\forall \beta_1.(M \leq N) \supset U(\mathsf{Array}_{\gamma_1}[N]\ \text{bool}, \mathsf{Array}_{\gamma_1}[N]\ \text{bool}) \to U(\text{int}[M], \text{int}[M]) \to U(\text{int}[N], \text{int}[N])$.
We set τ' to be $\{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool}, \text{bool})^{\text{diff}(0)}\ \{\gamma_1 \to \beta_1\}$.

Use the subtyping rules **S-R-FORALL-U**, **S-R-CIMPL-U** and **S-R-EXECDIFF**, we can show that

$$\models U(A \to A_1, A \to A_2) \sqsubseteq \tau \to U(A_1, A_2)$$

Next, using the monadic sutyping **S-RUM**, we can show that

$$\models U(\{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}^{\text{exec}(3,(N-M)*3+1)}\ \{\gamma_1 \to \emptyset\}, \{\gamma_1 \to \emptyset\}\ \exists\_.\text{bool}^{\text{exec}((N-M)*3+1,(N-M)*3+1)}\ \{\gamma_1 \to \emptyset\}) \sqsubseteq$$
$$\{\gamma_1 \to \beta_1\}\ \exists\_.U(\text{bool}, \text{bool})^{\text{diff}(0)}\ \{\gamma_1 \to \beta_1\}$$

Finally, using the ruele **R-EXEC**, we can derive the relational type shown above.

3.6 Insertion sort

```
fix ISort (_).Λ.Λ.Λ.Λ.λs.λi.λl_s.
  if i < l_s then
    let {a} =read s i in
    let {b} = celim(insert() [] [] [] [] []) s a 0 i in
        celim(ISort() [] [] [] [] [])s(i + 1)l_s
  else
    return ()
```

$$\vdash \quad \mathsf{ISort} \quad : \quad \begin{array}{l} \text{unit} \to \forall \gamma_1 : \mathbb{L}.\forall N, I : \mathbb{N}.(I \leq N) \supset \mathsf{Array}_{\gamma_1}[N]\ \text{int} \to \\ \text{int}[I] \to \text{int}[N] \to \\ \text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1)) \\ \qquad \{\gamma_1 \to \mathbb{N}\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \mathbb{N}\} \end{array}$$

where minimal cost obtained when the array is already sorted in the ascending order and the maximal cost corresponds to a sorted array in the descending order.

$$\vdash \mathsf{ISort} \ominus \mathsf{ISort} \lesssim 0 : \begin{array}{l} \text{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall N, I : \mathbb{N}.\forall \beta_1.(I \leq N) \supset \mathsf{Array}_{\gamma_1}[N]\ \text{int} \\ \to \text{int}[I] \to \text{int}[N] \to \\ \qquad \text{diff}(\frac{N*(N+1)-k*(k+1)}{2}) \\ \{\gamma_1 \to \beta_1\}\ \exists\_.\text{unit}_r\ \{\gamma_1 \to \mathbb{N}\} \end{array}$$

where $k = max(I, min(MIN(\beta_1), N))$.

$$
\begin{array}{l}
\mathsf{fix\ insert}\ (\_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda s.\lambda a.\lambda idx.\lambda i.\\
\quad \mathsf{let}\ \{b\} = \mathsf{read}\ s\ idx\ \mathsf{in}\\
\quad\quad \mathsf{if}\ a \geq b\ \mathsf{then}\\
\quad\quad\quad \mathsf{celim}(insert()\,[]\,[]\,[]\,[]\,[])\ s\ a\ (idx+1)\ i\\
\quad\quad \mathsf{else}\\
\quad\quad\quad \mathsf{let}\ \_ = \mathsf{celim}(\mathsf{shift}\,()\,[]\,[]\,[]\,[])\ s\ idx\ (i-1)\ \mathsf{in}\\
\quad\quad\quad \mathsf{updt}\ s\ idx\ a
\end{array}
$$

$$
\vdash \quad \mathsf{insert} \quad : \quad
\begin{array}{l}
\mathrm{unit} \to \forall \gamma_1 : \mathbb{L}.\forall N, A, IDX, I : \mathbb{N}.(IDX \leq N \wedge I \leq N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \\
\mathrm{int} \to \mathrm{int}[IDX] \to \mathrm{int}[I] \to \\
\underset{\mathrm{exec}(I-IDX+1,2*(I-IDX)+2)}{} \\
\{\gamma_1 \to \mathbb{N}\}\ \exists \_.\mathrm{unit}\ \{\gamma_1 \to \mathbb{N}\}
\end{array}
$$

where the minimal cost consists of one update operation and (I-IDX) times read operations when the value 'a' inserted is already the biggest element in the sequence ranging from [idx, i]. the maximal cost consists of 2*(I-IDX) coming from shift and one read and update operation in addition when the inserted value 'a' is the smallest element in the sequence.

$$
\vdash \mathsf{insert} \ominus \mathsf{insert} \lesssim 0 : \quad
\begin{array}{l}
\mathrm{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall N, A, IDX, I : \mathbb{N}.\forall \beta_1.(IDX \leq N \wedge I \leq N \wedge \beta_1 \cap [IDX, I] = \emptyset) \\
\supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \mathrm{int}[M] \to \mathrm{int}[N] \to \\
\underset{\mathrm{diff}(0)}{} \\
\{\gamma_1 \to \beta_1\}\ \exists \_.\mathrm{unit}_r\ \{\gamma_1 \to \beta_1\}
\end{array}
$$

The relational cost of insert function is 0 when the premise $\beta_1 \cap [IDX, I] = \emptyset$ holds, which means in the range [IDX, I], the elements of arrays on the two runs are exactly the same. Considering we are inserting the same value 'a' (s[I]), it always goes into the same path and hence incurs no relative cost.

$$
\begin{array}{l}
\mathsf{fix\ shift}\ (\_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda s.\lambda idx.\lambda i.\\
\quad \mathsf{if}\ idx \leq i\ \mathsf{then}\\
\quad\quad \mathsf{let}\ \{c\} = \mathsf{read}\ s\ i\ \mathsf{in}\\
\quad\quad \mathsf{let}\ \{\_\} = \mathsf{updt}\ s\ i+1\ c\ \mathsf{in}\\
\quad\quad\quad \mathsf{celim}\ (\mathsf{shift}\,()\,[]\,[]\,[]\,[])\ s\ idx\ (i-1)\\
\quad \mathsf{else}\\
\quad\quad \mathsf{return}\,()
\end{array}
$$

$$
\vdash \quad \mathsf{shift} \quad : \quad
\begin{array}{l}
\mathrm{unit} \to \forall \gamma_1 : \mathbb{L}.\forall N, IDX, I : \mathbb{N}.(IDX \leq I \wedge I < N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \\
\mathrm{int}[IDX] \to \mathrm{int}[I] \to \\
\underset{\mathrm{exec}(2*(I-IDX+1),2*(IDX-I+1)}{} \\
\{\gamma_1 \to \mathbb{N}\}\ \exists \_.\mathrm{unit}\ \{\gamma_1 \to \mathbb{N}\}
\end{array}
$$

where cost 2 comes from one read and one update operation.

$$
\vdash \mathsf{shift} \ominus \mathsf{shift} \lesssim 0 : \quad
\begin{array}{l}
\mathrm{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall N, IDX, I : \mathbb{N}.(IDX \leq I \wedge I < N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \\
\to \mathrm{int}[IDX] \to \mathrm{int}[I] \to \\
\underset{\mathrm{diff}(0)}{} \\
\{\gamma_1 \to \beta_1\}\ \exists \_.\mathrm{unit}_r\ \{\gamma_1 \to \beta_1 \cup [ID]\}
\end{array}
$$

Let us look at the derivation of the relational type of ISort.

$$
\begin{array}{l}
\mathsf{fix}\ \mathsf{ISort}\ (\_).\Lambda.\Lambda.\Lambda.\Lambda.\lambda s.\lambda i.\lambda l_s. \\
\quad \mathsf{if}\ i < l_s\ \mathsf{then} \\
\quad\quad \mathsf{let}\ \{a\} = \mathsf{read}\ s\ i\ \ \mathsf{in} \\
\quad\quad \mathsf{let}\ \{b\} = \mathsf{celim}(insert()\ []\ []\ []\ []\ [])\ s\ a\ 0\ i\ \mathsf{in} \\
\quad\quad\quad \mathsf{celim}(\mathsf{ISort}()\ []\ []\ []\ [])s(i+1)l_s \\
\quad \mathsf{else} \\
\quad\quad \mathsf{return}\ ()
\end{array}
$$

$$
\vdash \quad \mathsf{ISort} \quad : \quad
\begin{array}{l}
\mathsf{unit} \to \forall \gamma_1 : \mathbb{L}.\forall N, I : \mathbb{N}.(I \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathsf{int} \to \\
\mathsf{int}[I] \to \mathsf{int}[N] \to \quad \{\gamma_1 \to \mathbb{N}\}\ \exists\_.\mathsf{unit}\ \{\gamma_1 \to \mathbb{N}\}
\end{array}
^{\mathrm{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}
$$

$$
\vdash \mathsf{ISort} \ominus \mathsf{ISort} \lesssim 0 : \quad
\begin{array}{l}
\mathsf{unit}_r \to \forall \gamma_1 : \mathbb{L}.\forall N, I : \mathbb{N}.\forall \beta_1.(I \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathsf{int} \\
\to \mathsf{int}[I] \to \mathsf{int}[N] \to \{\gamma_1 \to \beta_1\}\ \exists\_.\mathsf{unit}_r\ \{\gamma_1 \to \mathbb{N}\}
\end{array}
^{\mathrm{diff}(\frac{N*(N+1)-k*(k+1)}{2})}
$$

where $k = max(I, min(MIN(\beta_1), N))$.

We first apply the fix-ext rule to store the unary information of ISort into the context. After several application of R-ILam and R-Abs rules as well as the rule R-CImpl, we introduce the index variables, variables and the constraint to the left side of the typing judgment. We will focus on the relational type of the body of the function ISort.

Set Z as the body of the function. where $C = (I \le N)$ and
$\tau = ^{\mathrm{diff}(\frac{N*(N+1)-k*(k+1)}{2})}\{\gamma_1 \to \beta_1\}\ \exists\_.\mathsf{unit}_r\ \{\gamma_1 \to \mathbb{N}\}$.

To show the type of part Z, we will use the rule R-SPLIT first.

$$
\frac{\Sigma;\Delta;\Phi \wedge C \wedge \beta_1 \cap [0,I] = \emptyset;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,1) \quad \Sigma;\Delta;\Phi \wedge C \wedge \beta_1 \cap [0,I] \neq \emptyset;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau \ldots (Case\,2)}{\Sigma;\Delta;\Phi \wedge C;\Gamma \vdash Z \ominus Z \lesssim 0 : \tau}\ \text{R-SPLIT}
$$

Case 1 $\beta_1 \cap [0, I] = \emptyset$

We know the two runs will go to the same path and we use relational typing rules to obtain the relational type of the body Z. After applying the rule R-Case, we need to show the type of the two branches respectively. Set the first branch as (1), the second (return ()) as (2), Φ' to be the updated constraint environment.

$$
\frac{\Delta;\Phi' \vdash i < l_s \ominus i < l_s \lesssim 0 : \mathsf{unit} + \mathsf{unit} \qquad \Delta;\Phi';\Gamma \vdash (1) \ominus (1) \lesssim 0 : \tau \qquad \Delta;\Phi';\Gamma \vdash (2) \ominus (2) \lesssim 0 : \tau}{\Delta;\Phi';\Gamma \vdash Z \ominus Z \lesssim 0 : \tau}\ \text{R-CASE}
$$

(a) The relative cost for (2) is 0, return () as a unit operation causes no relative cost. We use subtyping rule **S-RM** to make the types of two branches consistent.

(b)The relative cost of (1) depends on the insert function as well as the recursive call of ISort.

$$\begin{array}{c}\text{Set (3) as let } b = \text{celim}(insert()\,[]\,[]\,[]\,[]\,[])\ s\ a\ 0\ i \text{ in}\\ \text{celim}(ISort()\,[]\,[]\,[]\,[])s(i+1)l_s\end{array}$$

$$\dfrac{\Delta;\Phi';\Gamma \vdash insert\ldots \ominus insert\ldots \lesssim 0 : \{\gamma_1 \to \beta_1\}\ \overset{\text{diff}(0)}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \beta_1\} \quad \Delta;\Phi';\Gamma \vdash ISort\ldots \ominus ISort\ldots \lesssim 0 : \tau[k/k']}{\Delta;\Phi';\Gamma \vdash (3) \ominus (3) \lesssim 0 : \tau}\ \text{R-LET}$$

where $k' = max(I+1, min(MIN(\beta_1), N))$.
Because in this case, we know that $\beta_1 \cap [0,I] = \emptyset \Rightarrow k = k'$, and the relative cost of insert is 0 and the relative cost of (3) comes from the recursive call. Because we already know $k' = k$, it is easy to infer that $\tau[k/k'] = \tau$.

Case 2 $\beta_1 \cap [0,I] \neq \emptyset$.
The two runs may go to different branches when inserting one element into the array. So we will switch to unary typing.

We show the unary type of (3) first:

$$\dfrac{\Delta;\Phi;\Omega \vdash^{c_1}_{c_1} insert\cdots : \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(I+1,2I+2)}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\} \quad \Delta;\Phi;\Omega \vdash^{c_2}_{c_2} ISort\cdots : \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+2)*(I+3)}{2},(2N+1)*(N+1)-(2I+5)*(I+2))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\}}{\Delta;\Phi;\Omega \vdash^{c_1+c_2+1}_{c_1+c_2+1} (3) : \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\}}\ \text{U-LET}$$

From **R-S**, we know the relative cost of the function ISort is $\frac{N*(N+1)-I*(I+1)}{2}$. Because we know that $\beta_1 \cap [0,I] \neq \emptyset \Rightarrow k = I$, the type is still $\{\gamma_1 \to \beta_1\}\ \overset{\text{diff}(\frac{N*(N+1)-k*(k+1)}{2})}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \mathbb{N}\}$.

By applying unary subtyping rule **S-UM**, we know the following unary typing used in **R-S**.

$$\dfrac{\Delta;\Phi;|\Gamma|_1 \vdash^{c_1+c_2+1}_{c_1+c_2+1} (3) : \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\} \quad \Delta;\Phi;|\Gamma|_2 \vdash^{c_1+c_2+1}_{c_1+c_2+1} (3) : \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\}}{\Delta;\Phi;\Gamma \vdash (3) \ominus (3) \lesssim 0 : U(\overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \to \mathbb{N}\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \mathbb{N}\}}, \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \to \mathbb{N}\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \mathbb{N}\}})}\ \text{R-S}$$

$$\models U(\overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \to \mathbb{N}\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \mathbb{N}\}}, \overset{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2},(2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \to \mathbb{N}\}\ \exists\_.\text{unit}\ \{\gamma_1 \to \mathbb{N}\}}) \sqsubseteq \overset{\text{diff}(\frac{N*(N+1)-I*(I+1)}{2})}{\{\gamma_1 \to \beta_1\}\ \exists\_.\text{unit}_r\ \{\gamma_1 \to \mathbb{N}\}}$$

3.7 Cooley Tukey FFT algorithm

$$
\begin{array}{l}
\mathsf{fix\ sp}\,(\_).\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda n.\lambda i.\lambda y.\lambda pr.\\
\quad \mathsf{if}\ i < (n/2)\ \mathsf{then}\\
\quad\quad \mathsf{let}\ \{a\} = \mathsf{read}\ x\ (pr + 2 * i)\ \mathsf{in}\\
\quad\quad \mathsf{let}\ \{b\} = \mathsf{read}\ x\ (pr + 2 * i + 1)\ \mathsf{in}\\
\quad\quad \mathsf{let}\ \{\_\} = \mathsf{updt}\ y\ (pr + i)\ a\ \mathsf{in}\\
\quad\quad \mathsf{let}\ \{\_\} = \mathsf{updt}\ y\ (pr + i + n/2)\ b\ \mathsf{in}\\
\quad\quad\quad \mathsf{celim}(\mathsf{sp}()\ []\ []\ []\ []\ []\ [])\ x\ n\ (i+1)\ y\ pr\\
\quad \mathsf{else}\\
\quad\quad \mathsf{return}\ ()
\end{array}
$$

$$
\vdash\ \mathsf{sp}\ :\ \begin{array}{l}
\mathrm{unit} \to \forall \gamma_1, \gamma_2 : \mathbb{L}.\forall M, N, I, PR : \mathbb{N}.(I \le (M/2) \wedge (M + PR) < N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \\
\mathrm{int}[M] \to \mathrm{int}[I] \to \mathsf{Array}_{\gamma_2}[N]\ \mathrm{int} \to \mathrm{int}[PR] \to \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}\ \overset{\mathrm{exec}(4*(M/2-I),4*(M/2-I))}{\exists\_.\mathrm{unit}}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}
\end{array}
$$

$$
\vdash \mathsf{sp} \ominus \mathsf{sp} \lesssim 0 :\ \begin{array}{l}
\mathrm{unit}_r \to \forall \gamma_1 \gamma_2 : \mathbb{L}.\forall M, N, I, PR : \mathbb{N}.\forall \beta_1.(I \le (M/2) \wedge (M + PR) < N) \\
\supset \mathsf{Array}_{\gamma_1}[N]\ U(\mathrm{int}, \mathrm{int}) \to \mathrm{int}[M] \to \mathrm{int}[I] \to \mathsf{Array}_{\gamma_2}[N]\ U(\mathrm{int}, \mathrm{int}) \\
\to \mathrm{int}[PR] \to \{\gamma_1 \to \beta_1, \gamma_2 \to \mathbb{N}\}\ \overset{\mathrm{diff}(0)}{\exists\_.\mathrm{unit}_r}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}
\end{array}
$$

$$
\begin{array}{l}
\mathsf{fix\ cp}\,(\_).\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda y.\lambda l.\lambda u.\\
\quad \mathsf{if}\ l \le u\ \mathsf{then}\\
\quad\quad \mathsf{let}\ \{a\} = \big(\ \mathsf{read}\ x\ l\ \big)\ \mathsf{in}\\
\quad\quad \mathsf{let}\{\_\} = \big(\ \mathsf{updt}\ y\ l\ a\ \big)\ \mathsf{in}\\
\quad\quad \big(\mathsf{celim}(\mathsf{cp}()\ []\ []\ []\ []\ []\ [])\ x\ y\ (l+1)\ u\big)\\
\quad \mathsf{else\ return}()
\end{array}
$$

$$
\vdash\ \mathsf{cp}\ :\ \begin{array}{l}
\mathrm{unit} \to \forall \gamma_1, \gamma_2.\forall L, U, N.(L \le U \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ \mathrm{int} \to \mathsf{Array}_{\gamma_2}[N]\ \mathrm{int} \to \\
\mathrm{int}[L] \to \mathrm{int}[U] \to \\
\{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}\ \overset{\mathrm{exec}(2*(U-L+1),2*(U-L+1))}{\exists\_.\mathrm{unit}}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}
\end{array}
$$

$$
\vdash \mathsf{cp} \ominus \mathsf{cp} \lesssim 0 :\ \begin{array}{l}
\mathrm{unit}_r \to \forall \gamma_1, \gamma_2, \beta_1.\forall L, U, N.(L \le U \le N) \supset \mathsf{Array}_{\gamma_1}[N]\ U(\mathrm{int}, \mathrm{int}) \to \mathsf{Array}_{\gamma_2}[N]\ U(\mathrm{int}, \mathrm{int}) \\
\to \mathrm{int}[L] \to \mathrm{int}[U] \to \{\gamma_1 \to \beta_1, \gamma_2 \to \mathbb{N}\}\ \overset{\mathrm{diff}(0)}{\exists\_.\mathrm{unit}_r}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}
\end{array}
$$

separate

$$
\begin{array}{l}
\mathsf{fix\ separate}\,(\_).\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda n.\lambda y.\lambda pr.\\
\quad \mathsf{let}\ \_ = \mathsf{celim}(\mathsf{sp}()\ []\ []\ []\ []\ []\ [])\ x\ n\ 0\ y\ pr\ \mathsf{in}\\
\quad\quad \mathsf{celim}(\mathsf{cp}()\ []\ []\ []\ []\ []\ [])\ y\ x\ pr\ (n+pr)
\end{array}
$$

$$\vdash \text{ separate } : \begin{array}{l} \text{unit} \to \forall \gamma_1, \gamma_2. \forall M, N, PR. (M + PR < N) \supset \text{Array}_{\gamma_1}[N]\ \text{int} \to \\ \text{int}[M] \to \text{Array}_{\gamma_2}[N]\ \text{int} \to \text{int}[PR] \to \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\}\ \overset{\text{exec}(4*M, 4*M)}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\} \end{array}$$

$$\vdash \text{separate} \ominus \text{seperate} \lesssim 0 : \begin{array}{l} \text{unit}_r \to \forall \gamma_1, \gamma_2, \beta_1. \forall M, N, PR. (M + PR < N) \supset \text{Array}_{\gamma_1}[N]\ U(\text{int}, \text{int}) \to \\ \text{int}[M] \to \text{Array}_{\gamma_2}[N]\ U(\text{int}, \text{int}) \to \text{int}[PR] \to \\ \{\gamma_1 \to \beta_1, \gamma_2 \to \mathbb{N}\}\ \overset{\text{diff}(0)}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\} \end{array}$$

```
fix loop (_).Λ.Λ.Λ.Λ.Λ.λk.λn.λx.λpr.
  if k < (n/2) then
    let {e} =read x (k + pr) in
    let {o} =read x (k + pr + n/2) in
    let w = e^{-2*PI*k/n} in
    let {_} =updt x (k + pr) (e + w * o) in
    let {_} =updt x (k + pr + n/2) (e − w * o) in
        celim(loop() [] [] [] [] []) (k + 1) n x pr
  else
    return ()
```

$$\vdash \text{ loop } : \begin{array}{l} \text{unit} \to \forall \gamma_1. \forall K, M, N, PR. (PR + M < N) \supset \text{int}[K] \to \text{int}[M] \to \text{Array}_{\gamma_1}[N]\ \text{int} \to \\ \text{int}[PR] \to \{\gamma_1 \to \mathbb{N}\}\ \overset{\text{exec}(4*(M-K), 4*(M-K))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}\} \end{array}$$

$$\vdash \text{loop} \ominus \text{loop} \lesssim 0 : \begin{array}{l} \text{unit}_r \to \forall \gamma_1, \beta_1. \forall K, M, N, PR. (PR + M < N) \supset \text{int}[K] \to \text{int}[M] \to \\ \text{Array}_{\gamma_1}[N]\ U(\text{int}, \text{int}) \to \text{int}[PR] \to \{\gamma_1 \to \beta_1\}\ \overset{\text{diff}(0)}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \mathbb{N}\} \end{array}$$

```
fix FFT (_).Λ.Λ.Λ.Λ.Λ.λx.λy.λn.λpr.
  if 2 ≤ n then
    let {_} = celim(separate () [] [] [] [] []) x n y pr in
    let {_} = celim(FFT () [] [] [] [] []) x y (n/2) pr in
    let {_} = celim(FFT () [] [] [] [] []) x y (n/2) (pr + n/2) in
        celim(loop() [] [] [] [] []) 0 n x pr
  else
    return ()
```

$$\vdash \text{ FFT } : \begin{array}{l} \text{unit} \to \forall \gamma_1, \gamma_2. \forall M, N, PR. (PR + M < N) \supset \text{Array}_{\gamma_1}[N]\ \text{int} \to \text{Array}_{\gamma_2}[N]\ \text{int} \to \\ \text{int}[M] \to \text{int}[PR] \to \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N},\}\ \overset{\text{exec}(8*M*\log(M), 8*M*\log(M))}{\exists\_.\text{unit}}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N},\} \end{array}$$

$$\vdash \text{FFT} \ominus \text{FFT} \lesssim 0 : \begin{array}{l} \text{unit}_r \to \forall \gamma_1, \gamma_2, \beta_1. \forall M, N, PR. (PR + M < N) \supset \text{Array}_{\gamma_1}[N]\ U(\text{int}, \text{int}) \to \\ \text{Array}_{\gamma_2}[N]\ U(\text{int}, \text{int}) \to \text{int}[M] \to \text{int}[PR] \to \{\gamma_1 \to \beta_1, \gamma_2 \to \mathbb{N}\}\ \overset{\text{diff}(0)}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \mathbb{N}, \gamma_2 \to \mathbb{N}\} \end{array}$$

3.8 Square and multiply (SAM)

This examples implements the square and multiply algorithm for the positive power of the a number. It bases on the observation that $x^m = (x^2)^{\frac{m}{2}}$ if m is even and $x^m = x * (x^2)^{\frac{m}{2}}$ if m is odd.

$$
\begin{array}{l}
\mathsf{fix}\ \mathsf{sam}\ ().\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda a.\lambda i.\lambda n. \\
\quad \mathsf{if}\ i \le (n-1)\ \mathsf{then} \\
\quad\quad \mathsf{let}\ \{b\} = \mathsf{read}\ a\ i\ \mathsf{in} \\
\quad\quad \mathsf{let}\ \{r\} = \mathsf{celim}(\mathsf{sam}\ ()\ []\ []\ []\ []\ []\ [])\ x\ a\ (i+1)\ n\ \mathsf{in} \\
\quad\quad\quad \mathsf{return}(\mathsf{func}\ x\ b\ r) \\
\quad \mathsf{else} \\
\quad\quad \mathsf{if}\ x = 0\ \mathsf{return}\ 1 \\
\quad\quad \mathsf{return}\ x
\end{array}
$$

where func is a function which depends on the value of b (1 or 0), returns the result $x*1$ if $b=1$, returns 1 otherwise. The relatonal type is derived by split the cases on the assumption $I \in \beta$ and on the two cases, func will generates different costs based on the assumption whether $I \in \beta$ or not.

$$
\vdash \mathsf{sam} \ominus \mathsf{sam} \lesssim 0 : \begin{array}{l} \mathrm{unit}_r \to \forall \gamma_1.\forall I,N,X,\beta.(I<N) \supset \mathrm{int}[X] \to \mathsf{Array}_{\gamma_1}[N]\ U(\mathrm{bool},\mathrm{bool}) \to \\ \mathrm{int}[I] \to \mathrm{int}[N] \to \{\gamma_1 \to \beta\}\ \overset{\mathrm{diff}(\beta \cap [I,N])}{\exists\_}.U(\mathrm{int})\ \{\gamma_1 \to \beta\} \end{array}
$$

function func

$$
\begin{array}{l}
\mathsf{fix}\ \mathsf{func}\ (x).\lambda r.\lambda b. \\
\quad \mathsf{if}\ b\ \mathsf{then} \\
\quad\quad \mathsf{return}\ x * r * r \\
\quad \mathsf{else} \\
\quad\quad \mathsf{return}\ r * r
\end{array}
$$

$$
\vdash \mathsf{func} \ominus \mathsf{func} \lesssim 0 : \ U(\mathrm{int}) \to U(\mathrm{int}) \to \mathrm{int}_r \overset{\mathrm{diff}(0)}{\longrightarrow} U(\mathrm{int})
$$

$$
\vdash \mathsf{func} \ominus \mathsf{func} \lesssim 0 : \ U(\mathrm{int}) \to U(\mathrm{int}) \to U(\mathrm{int}) \overset{\mathrm{diff}(1)}{\longrightarrow} U(\mathrm{int})
$$

3.9 Constant-time comparison

$$
\begin{array}{l}
\mathsf{fix}\ \mathsf{comp}\ ().\lambda l_1.\lambda l_2.\lambda i.\lambda n. \\
\quad \mathsf{if}\ i < n\ \mathsf{then} \\
\quad\quad \mathsf{let}\ \{a\} = \mathsf{read}\ l_1\ i\ \mathsf{in} \\
\quad\quad \mathsf{let}\ \{b\} = \mathsf{read}\ l_2\ i\ \mathsf{in} \\
\quad\quad \mathsf{return}\ \mathsf{boolAnd}\big(\mathsf{celim}(comp()[][][])l_1\ l_2\ (i+1)\ n, \mathsf{eq}(a,b)\big) \\
\quad \mathsf{else} \\
\quad\quad \mathsf{return}\ true
\end{array}
$$

where function boolAnd has the type $(U(\mathrm{bool}) \times U(\mathrm{bool}) \overset{\mathrm{diff}(0)}{\longrightarrow} U(\mathrm{bool})$, and eq has type $(U(\mathrm{int}) \times U(\mathrm{int}) \overset{\mathrm{diff}(0)}{\longrightarrow} U(\mathrm{bool})$
.

$$\vdash \mathsf{comp} \ominus \mathsf{comp} \lesssim 0 : \begin{array}{l} \mathrm{unit}_r \to \forall \gamma_1, \gamma_2. \forall I, N, \beta_1, \beta_2. (I \geq 0) \supset \mathsf{Array}_{\gamma_1}[N]\ U(\mathrm{int}, \mathrm{int}) \to \mathsf{Array}_{\gamma_2}[N]\ U(\mathrm{int}, \mathrm{int}) \to \\ \mathrm{int}[I] \to \mathrm{int}[N] \to \{\gamma_1 \to \beta_1, \gamma_2 \to \beta_2\}\ \exists \_. \overset{\mathrm{diff}(0)}{U}(\mathrm{bool})\ \{\gamma_1 \to \beta_1, \gamma_2 \to \beta_2\} \end{array}$$

When we assume the two arrays has the same length N, start comparing the two arrays from the same index i on the two runs. No relative cost is generated because the execution paths across the two runs are the same.

3.10 Loop Unswitching

In the next example, we will have a look at one technique applied in compiler optimization, known as "loop unswitching". In this example, we show our system can provide a more precise relative cost than the standard worst-case/best-case analysis when dealing with two programs that are not structurally similar.

We propose a function loop in Figure 25, which operates a simple loop over an input array A, from index m, to the end of the array specified by the length of the array n of type int[N]. Inside the loop, there is an if conditional if $b \ldots$, whose then branch reads the value in the array at index m and then does some pure computation $f\ h$ on the value h just read before going into the next iteration. For simplicity, we let the else branch do nothing, return unit.

Actually, this program can be transformed a little bit, by pulling out the if conditional from the function body, as we can see in the right part of Figure 25, we call it loopOp, which is the optimized version of loop. We suppose the original loop and the optimized loopOp operates on the same array A, has the identical pure computation f inside the loop, and shares the boolean input b in two runs. Intuitively, we think the relative cost is upper bounded by N when we count one unit cost for elimination forms because the first one checks b at each iteration while the optimized one checks only once. When people use the worst-case/best-case analysis, they obtain the following unary types at first.

$$\vdash \lambda f.\lambda b.\mathsf{loop} : \begin{array}{l} (A \to \mathrm{unit}) \to \mathrm{bool} \to \forall \gamma_1, N, M. \\ \mathsf{Array}_{\gamma_1}[N]\ A \to \mathrm{int}[M] \to \mathrm{int}[N] \to \{\gamma_1 \to \emptyset\}\ \overset{\mathrm{exec}(1,3*(N-M)+1)}{\exists \_.\mathrm{unit}}\ \{\gamma_1 \to \emptyset\} \end{array}$$

$$\vdash \lambda f.\lambda b.\mathsf{loopOp} : \begin{array}{l} (A \to \mathrm{unit}) \to \mathrm{bool} \to \forall \gamma_1, N, M. \\ \mathsf{Array}_{\gamma_1}[N]\ A \to \mathrm{int}[M] \to \mathrm{int}[N] \to \{\gamma_1 \to \emptyset\}\ \overset{\mathrm{exec}(1,2*(N-M)+1)}{\exists \_.\mathrm{unit}}\ \{\gamma_1 \to \emptyset\} \end{array}$$

In the above unary type, both precondition and postcondition have location g referring to the empty set. It is consistent with our assumption that f does some pure computation, hence indicates the arrays remain intact within the execution of either loop or loopOp, which requires no writing permission. With the help of the subtyping rule **s-rum**, we obtain the relative cost, which is $3 * (N - M)$. When we start the loop from the beginning, which means $M = 0$, then the relative cost is bounded by $3 * N$, instead of our intuition N.

On the other hand, we can do better if we use the relational analysis, in particular, the relational asynchronous rule **r-e-case** in Figure **??**. Remember, we use if t then t_1 else t_2 as the syntactic sugar for the construct case $(t, x.t_1, y.t_2)$. To be precise, we can have a simplified asynchronous rule **r-e-if** from **r-e-case**.

$$\frac{\Sigma; \Delta; \Phi; |\Gamma|_2 \vdash_{L_1}^{U_1} t' : \mathrm{bool} \qquad \Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t_1' \lesssim D_2 : \tau \qquad \Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t_2' \lesssim D_2 : \tau}{\Sigma; \Delta; \Phi; \Gamma \vdash t \ominus \mathsf{if}\ t'\ \mathsf{then}\ t_1' \mathsf{else}\ t_2' \lesssim D_2 - L_1 - c_{case} : \tau}\ \textbf{r-e-if}$$

The above asynchronous rule allows us to compare loop with respect to the inner recursive function loop′ inside loopOp. When we compare the body of loop and loop′, when we type the then branch of the first if conditional if $m < n \ldots$, we come across the structurally dissimilar part, the red part in loop and the green part part in loop′ in Figure 25. In this turn, we use a similar asynchronous rule **r-case-e** in Figure **??**. In particular, we want to avoid comparing the else branch in the red part return () with the green part, when we know the conditional b agree in two runs. To this end, we can refine our

```
fix loop (A). λm.λn.                    loopOp = if b then
  if m < n then                                  fix loop′ (A). λm.λn.
    if b then                                      if m < n then
      let {h} =read A m  in ;                        let {h} =read A m  in ;
      let _ = f h  in                                let _ = f h  in
      loop A (m+1) n                                 loop′ A (m+1) n
    else                                           else
      return ()                                      return ()
  else                                           else
    return ()                                      λA.λm.λn.return ()
```

Figure 25: Loop Unswitching code

boolean type bool to carry its value as we do for the integer type, bool[B] for unary boolean type and bool$_r$[B] for the relational one, where $B \in \{\text{true}, \text{false}\}$ so that the erasure operation over the relational type gives $|\text{bool}_r[B]|_i = \text{bool}[B]$. Now, we can have more fine-grained asynchronous rules from the aforementioned rule **e-if**.

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t' : \text{bool}[\text{true}] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t_1' \lesssim D_2 : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus \text{if } t' \text{ then } t_1' \text{else } t_2' \lesssim D_2 - L_1 - c_{case} : \tau} \textbf{ r-e-if-t}$$

$$\frac{\Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_1}_{L_1} t' : \text{bool}[\text{false}] \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t \ominus t_2' \lesssim D_2 : \tau}{\Sigma;\Delta;\Phi;\Gamma \vdash t \ominus \text{if } t' \text{ then } t_1' \text{else } t_2' \lesssim D_2 - L_1 - c_{case} : \tau} \textbf{ r-e-if-f}$$

With all the possible extensions, we are able to provide the relational type of the two programs as follows:

$$\vdash \lambda f.\lambda b.\text{loop} \ominus \lambda f.\lambda b.\text{loopOp} \lesssim 0 : \begin{array}{l} (U(A) \to \text{unit}) \to \forall B :: \{\text{true}, \text{false}\}.\text{bool}_r[B] \\ \xrightarrow{\text{diff}(-1)} \forall \gamma_1, N, M.\ \text{Array}_{\gamma_1}[N]\ U(A) \to \text{int}[M] \\ \to \text{int}[N] \to \{\gamma_1 \to \emptyset\}\ \overset{\text{diff}(N-M)}{\exists\_.\text{unit}_r}\ \{\gamma_1 \to \emptyset\} \end{array}$$

Notice, the negative cost -1 in the type comes from checking b at the beginning in the optimized version. And the relative cost embedded in the monadic type reflects the difference between the red and green parts for every iteration.

4 Bidirectional type checking

Terms $t ::= x \mid l \mid n \mid r \mid () \mid \Lambda.t \mid t[] \mid \lambda x.t \mid t\ u \mid \text{return}\ t \mid \text{let}\ \{x\} = t\ \text{in}\ t_1 \mid$
$\text{alloc}\ t\ t \mid \text{alloc}_\square\ t\ t \mid \text{updt}\ t\ t\ t \mid \text{update}_\square\ t\ t\ t \mid \text{read}\ t\ t \mid \text{read}_\square\ t\ t \mid \text{inl}\ t \mid \text{inr}\ t \mid \text{case}\ (t, x.t_1, y.t_2) \mid$
$\text{pack}\ t \mid \text{fix}\ f(x).t \mid \text{unpack}\ t_1\ \text{as}\ x\ \text{in}\ t_2 \mid \text{celim}\ t$
$\texttt{switch}\ t \mid \texttt{NC}\ t \mid \texttt{split}\ t\ \text{with}\ C \mid \texttt{contra}\ t \mid \text{FIXEXT}\ t\ \text{with}\ A$

Values $v ::= n \mid l \mid r \mid () \mid \lambda x.t \mid \Lambda.t \mid \text{return}\ t \mid \text{alloc}\ t\ t \mid \text{alloc}_\square\ t\ t \mid \text{updt}\ t\ t\ t \mid \text{update}_\square\ t\ t\ t \mid \text{read}\ t\ t \mid \text{read}_\square\ t\ t \mid$
$\text{inl}\ v \mid \text{inr}\ v \mid \text{let}\ \{x\} = t\ \text{in}\ t_1 \mid \text{pack}\ v \mid \Lambda.t \mid \text{fix}\ f(x).t$

Figure 26: Syntax of values and expressions in the ArelCore

Terms $t ::= x \mid l \mid n \mid r \mid () \mid \Lambda.t \mid t\,[] \mid \lambda x.t \mid t\,u \mid \mathsf{return}\,t \mid \mathsf{let}\,\{x\} = t\,\mathsf{in}\,t_1 \mid$
$\mathsf{alloc}\,t\,t \mid \mathsf{alloc}_\square\,t\,t \mid \mathsf{updt}\,t\,t\,t \mid \mathsf{update}_\square\,t\,t\,t \mid \mathsf{read}\,t\,t \mid \mathsf{read}_\square\,t\,t \mid \mathrm{inl}\,t \mid \mathrm{inr}\,t \mid \mathrm{case}\,(t, x.t_1, y.t_2) \mid$
$\mathsf{pack}\,t \mid \mathsf{fix}\,f(x).t \mid \mathsf{unpack}\,t_1\,\mathsf{as}\,x\,\mathsf{in}\,t_2 \mid \mathsf{celim}\,t$
$\mathtt{switch}\,t \mid \mathtt{NC}\,t \mid \mathtt{split}\,t\,\mathrm{with}\,C \mid \mathtt{contra}\,t \mid (t : \tau, D) \mid (t : A, L, U) \mid \mathsf{FIXEXT}\,t\,\mathrm{with}\,A$

Values $v ::= n \mid l \mid r \mid () \mid \lambda x.t \mid \Lambda.t \mid \mathsf{return}\,t \mid \mathsf{alloc}\,t\,t \mid \mathsf{alloc}_\square\,t\,t \mid \mathsf{updt}\,t\,t\,t \mid \mathsf{update}_\square\,t\,t\,t \mid \mathsf{read}\,t\,t \mid \mathsf{read}_\square\,t\,t \mid$
$\mathrm{inl}\,v \mid \mathrm{inr}\,v \mid \mathsf{let}\,\{x\} = t\,\mathsf{in}\,t_1 \mid \mathsf{pack}\,v \mid \Lambda.t \mid \mathsf{fix}\,f(x).t$

Figure 27: Syntax of values and expressions in the biArel

$\boxed{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow t_1^* \ominus t_2^* \lesssim D : \tau}$ Expressions $t_1 \ominus t_2$ are embedded into $t_1^* \ominus t_1^*$ with the relational type τ and the relational cost D.

$\boxed{\Sigma;\Delta;\Phi_a;\Omega \vdash_L^U t \rightsquigarrow t^* : A}$ Expression t is embedded into $t*$ with the unary type A and the minimum and maximum execution costs L and U, respectively.

General rules

$$\frac{\Sigma;\Delta;\Phi_a;|\Gamma| \vdash_{L_1}^{U_1} t_1 \rightsquigarrow t_1^* : A \qquad \Sigma;\Delta;\Phi_a;|\Gamma| \vdash_{L_2}^{U_2} t_2 \rightsquigarrow t_2^* : A}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow \mathtt{switch}\ t_1^* \ominus \mathtt{switch}\ t_2^* \lesssim U_1 - L_2 : U\ A}\ \textbf{e-switch}$$

Subsumption

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_L^U t \rightsquigarrow t^* : A \qquad \Delta;\Phi_a \models A \sqsubseteq A' \qquad \Delta;\Phi_a \models L' \leq L \qquad \Delta;\Phi_a \models U \leq U'}{\Delta;\Phi_a;\Omega \vdash_{k'}^{t'} e \rightsquigarrow e^* : A'}\ \textbf{e-u-sub}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow {t_1}^* \ominus {t_2}^* \lesssim D : \tau \qquad \Delta;\Phi_a \models \tau \sqsubseteq \tau' \qquad t' = \mathtt{coerce}_{\tau,\tau'} \qquad \Delta;\Phi_a \models D \leq D'}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow t'\ t_1^* \ominus t'\ t_2^* \lesssim D' : \tau'}\ \textbf{e-r-sub}$$

Variables x

$$\frac{\Omega(x) = A}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^0 x \rightsquigarrow x : A}\ \textbf{e-u-var} \qquad \frac{\Gamma(x) = \tau}{\Sigma;\Delta;\Phi_a;\Gamma \vdash x \ominus x \rightsquigarrow x \ominus x \lesssim 0 : \tau}\ \textbf{e-r-var}$$

alloc

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_{L_1}^{U_1} t_1 \rightsquigarrow t_1^* : \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash_{L_2}^{U_2} t_2 \rightsquigarrow t_2^* : A \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^0 \mathsf{alloc}\ t_1\ t_2 \rightsquigarrow \mathsf{alloc}\ t_1^*\ t_2^* : \{P\}\ \overset{\mathrm{exec}(L_1+L_2+L_a,U_1+U_2+U_a)}{\exists \gamma.\ \mathsf{Array}_\gamma[I]\ A}\ \{P \star \gamma \to \mathbb{N}\}}\ \textbf{e-u-alloc}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathrm{int}[I] \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \tau \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \rightsquigarrow \mathsf{alloc}\ t_1^*\ t_2^* \ominus \mathsf{alloc}\ t_1'^*\ t_2'^* \lesssim 0 : \{P\}\ \overset{\mathrm{diff}(D_1+D_2)}{\exists \gamma.\ \mathsf{Array}_\gamma[I]\ \tau}\ \{P \star \gamma \to \mathbb{N}\}}\ \textbf{e-r-alloc}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathrm{int}[I] \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \Box\tau \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \rightsquigarrow \mathsf{alloc}_\Box\ t_1^*\ t_2^* \ominus \mathsf{alloc}_\Box\ t_1'^*\ t_2'^* \lesssim 0 : \{P\}\ \overset{\mathrm{diff}(D_1+D_2)}{\exists \gamma.\ \mathsf{Array}_\gamma[I]\ \tau}\ \{P \star \gamma \to \emptyset\}}\ \textbf{e-r-allocb}$$

read

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U_1}_{L_1} t_1 \rightsquigarrow t_1^* : \mathsf{Array}_\gamma[I]\ A \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash^{U_2}_{L_2} t_2 \rightsquigarrow t_2^* : \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{0}_{0} \mathsf{read}\ t_1\ t_2 \rightsquigarrow \mathsf{read}\ t_1^*\ t_2^* : \overset{\mathrm{exec}(L_1+L_2+L_r,U_1+U_2+U_r)}{\{P\}\ \exists\gamma.A\ \{P\}}}\ \textbf{e-u-read}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \rightsquigarrow \mathsf{read}\ t_1^*\ t_2^* \ominus \mathsf{read}\ t_1'^*\ t_2'^* \lesssim 0 : \overset{\mathrm{diff}(D_1+D_2)}{\{P\}\ \exists\gamma.\tau\ \{P\}}}\ \textbf{e-r-read}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \le I \qquad I' \notin \beta \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \rightsquigarrow \mathsf{read}\ t_1^*\ t_2^* \ominus \mathsf{read}\ t_1'^*\ t_2'^* \lesssim 0 : \overset{\mathrm{diff}(D_1+D_2)}{\{P \star \gamma \mapsto \beta\}\ \exists\gamma.\square\tau\ \{P \star \gamma \mapsto \beta\}}}\ \textbf{e-r-readb}$$

update

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U_1}_{L_1} t_1 \rightsquigarrow t_1^* : \mathsf{Array}_\gamma[I]\ A \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash^{U_2}_{L_2} t_2 \rightsquigarrow t_2^* : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash^{U_3}_{L_3} t_3 \rightsquigarrow t_3^* : A \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Delta;\Phi_a \models I' \in \beta}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{0}_{0} \mathsf{updt}\ t_1\ t_2\ t_3 \rightsquigarrow \mathsf{updt}\ t_1^*\ t_2^*\ t_3^* : \overset{\mathrm{exec}(L_1+L_2+L_3+L_u,U_1+U_2+U_3+U_u)}{\{P \star \gamma \mapsto \beta\}\ \exists\gamma.A\ \{P \star \gamma \mapsto \beta\}}}\ \textbf{e-u-updt}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \rightsquigarrow {t_3}^* \ominus {t_3'}^* \lesssim D_3 : \tau \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \rightsquigarrow \mathsf{updt}\ t_1^*\ t_2^*\ t_3^* \ominus \mathsf{updt}\ t_1'^*\ t_2'^*\ t_3'^* \lesssim 0 : \overset{\mathrm{diff}(D_1+D_2+D_3)}{\{P \star \gamma \mapsto \beta\}\ \exists\gamma.\tau\ \{P \star \beta \cup \{I'\}\}}}\ \textbf{e-r-updt}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \rightsquigarrow {t_3}^* \ominus {t_3'}^* \lesssim D_3 : \square\tau \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \rightsquigarrow \mathsf{update}_\square\ t_1^*\ t_2^*\ t_3^* \ominus \mathsf{update}_\square\ t_1'^*\ t_2'^*\ t_3'^* \lesssim 0 : \overset{\mathrm{diff}(D_1+D_2+D_3)}{\{P \star \gamma \mapsto \beta\}\ \exists\gamma.\tau\ \{P \star \beta \setminus \{I'\}\}}}\ \textbf{e-r-updtb}$$

return

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U}_{L} t \rightsquigarrow t^* : A}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{0}_{0} \mathsf{return}\ t \rightsquigarrow \mathsf{return}\ t^* : \overset{\mathrm{exec}(L,U)}{\{P\}\ \exists\gamma.A\ \{P\}}}\ \textbf{e-u-ret}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t \ominus t' \rightsquigarrow t^* \ominus {t'}^* \lesssim D_3 : \tau \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{return}\ t \ominus \mathsf{return}\ t' \rightsquigarrow \mathsf{return}\ t^* \ominus \mathsf{return}\ t'^* \lesssim 0 : \overset{\mathrm{diff}(D)}{\{P\}\ \exists\gamma.\tau\ \{P\}}}\ \textbf{e-r-ret}$$

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

$$\frac{\Delta;\psi_a \models I = I'}{\Delta;\psi_a;\Phi_a \models \mathrm{int}[I] \equiv \mathrm{int}[I']}\ \textbf{eq-int-I} \qquad \frac{\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \qquad \models \gamma = \gamma' \qquad \Delta;\psi_a \models I = I'}{\Delta;\psi_a;\Phi_a \models \mathsf{Array}_{\gamma}[I]\ \tau \equiv \mathsf{Array}_{\gamma'}[I']\ \tau'}\ \textbf{eq-r-array}$$

$$\frac{\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \qquad \Delta;\psi_a;\Phi_a \models P' \equiv P \qquad \Delta;\psi_a;\Phi_a \models Q \equiv Q' \qquad \models \vec{\gamma}_1 = \vec{\gamma}_2}{\Delta;\psi_a;\Phi_a \models \{P\}\ \exists \vec{\gamma}_1 : \tau\ \{Q\} \equiv \{P'\}\ \exists \vec{\gamma}_2 : \tau'\ \{Q'\}}\ \textbf{eq-r-monad}$$

$$\frac{\Delta;\psi_a;\Phi_a \models \tau_1 \equiv \tau_1' \qquad \Delta;\psi_a;\Phi_a \models \tau_2 \equiv \tau_2' \qquad \Delta;\psi_a \models D = D'}{\Delta;\psi_a;\Phi_a \models \tau_1 \xrightarrow{\mathrm{diff}(D)} \tau_2 \equiv \tau_1' \xrightarrow{\mathrm{diff}(D')} \tau_2'}\ \textbf{eq-r-fun}$$

Figure 28: Relational type equivalence rules

$$\frac{\Delta;\psi_a;\Phi_a \models P \equiv P' \Rightarrow \Phi \qquad \Delta;\psi_a \models \beta_1 \doteq \beta_2}{\Delta;\psi_a;\Phi_a \models P \star \gamma \rightarrow \beta_1 \equiv P' \star \gamma \rightarrow \beta_2}\ \textbf{eq-r-predicate}$$

$$\frac{}{\Delta;\psi_a;\Phi_a \models \varnothing \equiv \varnothing}\ \textbf{eq-r-predicate-empty}$$

Figure 29: Predicate equivalence rules

$\boxed{\text{bind}}$

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_{L_1}^{U_1} t_1 \rightsquigarrow t_1^* : \{P\}\ \exists^{\mathrm{exec}(L,U)} \vec{\gamma}_1.A\ \{P'\} \qquad \Sigma;\Delta,\vec{\gamma}_1;\Phi_a;\Omega,x:A \vdash_{L_2}^{U_2} t_2 \rightsquigarrow t_2^* : \{P'\}\ \exists^{\mathrm{exec}(L',U')} \vec{\gamma}_2.A'\ \{Q\}}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^{\mathrm{exec}(L+L'+L_1+L_2+L_l,U+U'+U_1+U_2+U_l)} \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \rightsquigarrow \mathsf{let}\ \{x\} = t_1^*\ \mathsf{in}\ t_2^* : \{P\}\ \exists \_.A'\ \{Q\}}\ \textbf{e-u-bind}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \{P\}\ \exists^{\mathrm{diff}(D)} \vec{\gamma}_1.\tau\ \{P'\}\ D_1 \\ \Sigma;\Delta,\vec{\gamma}_1 : \vec{\mathbb{L}};\Phi_a;\Gamma,x:\tau \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \{P'\}\ \exists^{\mathrm{diff}(D')} \vec{\gamma}_2.\tau'\ \{Q\}\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \rightsquigarrow \mathsf{let}\ \{x\} = t_1^*\ \mathsf{in}\ t_2^* \ominus \mathsf{let}\ \{x\} = t_1'^*\ \mathsf{in}\ t_2^* \lesssim 0 : \{P\}\ \exists^{\mathrm{diff}(D_1+D_2+D+D')} \_.\tau'\ \{Q\}}\ \textbf{e-r-bind}$$

$\boxed{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U}_{L} t :^{c} A}$ the unary core typing rules.

$\boxed{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \lesssim D :^{c} \tau}$ the relational core typing rules.

General rules

$$\frac{\Delta;\Phi_a;|\Gamma| \vdash^{U_1}_{L_1} t_1 :^{c} A \qquad \Delta;\Phi_a;|\Gamma| \vdash^{U_2}_{L_2} t_2 :^{c} A}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathtt{switch}\ t_1 \ominus \mathtt{switch}\ t_2 \lesssim U_1 - L_2 :^{c} U\ A}\ \textbf{c-switch}$$

Unary Subtyping

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Omega \vdash^{U}_{L} t :^{c} A \qquad \Delta;\Phi_a \models A \sqsubseteq A' \\ \Delta;\Phi_a \models L' \leq L \qquad \Delta;\Phi_a \models U \leq U'\end{array}}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U'}_{L'} t :^{c} A'}\ \textbf{c-}\sqsubseteq$$

Binary Subeffecting

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t \ominus t' \lesssim D :^{c} \tau \qquad \Delta;\Phi_a \models \tau \equiv \tau' \\ \Delta;\Phi_a \models D \leq D'\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t \ominus t' \lesssim D' :^{c} \tau'}\ \textbf{c-r-}\equiv$$

Alloc

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U_1}_{L_1} t_1 :^{c} \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash^{U_2}_{L_2} t_2 :^{c} A \qquad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{0}_{0} \mathsf{alloc}\ t_1\ t_2 :^{c}_{\mathrm{exec}(L_1+L_2+L_a,U_1+U_2+U_a)} \{P\}\ \exists\gamma.\ \mathsf{Array}_{\gamma}[I]\ A\ \{P \star \gamma \to \mathbb{N}\}}\ \textbf{c-alloc}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^{c} \mathrm{int}[I] \quad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^{c} \tau \quad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \lesssim 0 :^{c}_{\mathrm{diff}(D_1+D_2)} \{P\}\ \exists\gamma.\ \mathsf{Array}_{\gamma}[I]\ \tau\ \{P \star \gamma \to \mathbb{N}\}}\ \textbf{c-r-alloc}$$

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^{c} \mathrm{int}[I] \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^{c} \square\tau \qquad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}_{\square}\ t_1\ t_2 \ominus \mathsf{alloc}_{\square}\ t_1'\ t_2' \lesssim 0 :^{c}_{\mathrm{diff}(D_1+D_2)} \{P\}\ \exists\gamma.\ \mathsf{Array}_{\gamma}[I]\ \tau\ \{P \star \gamma \to \emptyset\}}\ \textbf{c-r-allocB}$$

read

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^{U_1}_{L_1} t_1 :^{c} \mathsf{Array}_{\gamma}[I]\ A \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash^{U_2}_{L_2} t_2 :^{c} \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \leq I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Omega \vdash^{0}_{0} \mathsf{read}\ t_1\ t_2 :^{c}_{\mathrm{exec}(L_1+L_2+L_r,U_1+U_2+U_r)} \{P\}\ \exists\_.A\ \{P\}}\ \textbf{c-read}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^{c} \mathsf{Array}_{\gamma}[I]\ \tau \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^{c} \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \leq I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \lesssim 0 :^{c}_{\mathrm{diff}(D_1+D_2)} \{P\}\ \exists\_.\tau\ \{P\}}\ \textbf{c-r-read}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^{c} \mathsf{Array}_{\gamma}[I]\ \tau \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^{c} \mathrm{int}[I'] \qquad \Delta;\Phi_a \models I' \leq I \qquad I' \notin \beta \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}_{\square}\ t_1\ t_2 \ominus \mathsf{read}_{\square}\ t_1'\ t_2' \lesssim 0 :^{c}_{\mathrm{diff}(D_1+D_2)} \{P \star \gamma \mapsto \beta\}\ \exists\_.\square\tau\ \{P \star \gamma \mapsto \beta\}}\ \textbf{c-r-readb}$$

update

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_{L_1}^{U_1} t_1 :^c \mathsf{Array}_\gamma[I]\ A \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash_{L_2}^{U_2} t_2 :^c \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Omega \vdash_{L_3}^{U_3} t_3 :^c A \qquad \Delta;\Phi_a \vDash I' \le I \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Delta;\Phi_a \vDash I' \in \beta}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^{0} \mathsf{updt}\ t_1\ t_2\ t_3 :^c \overset{\mathrm{exec}(L_1+L_2+L_3+L_u, U_1+U_2+U_3+U_u)}{\{P \star \gamma \to \beta\}\ \exists \_.\mathrm{unit}\ \{P \star \gamma \to \beta\}}}\ \text{U-U}$$

c-update

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 :^c \tau \qquad \Delta;\Phi_a \vDash I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \lesssim 0 :^c \overset{\mathrm{diff}(D_1+D_2+D_3)}{\{P \star \gamma \mapsto \beta\}\ \exists \_.\mathrm{unit}_r\ \{P \star \gamma \mapsto \beta \cup \{I'\}\}}}$$

c-r-update

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathrm{int}[I'] \qquad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 :^c \square\tau \qquad \Delta;\Phi_a \vDash I' \le I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{update}_\square\ t_1\ t_2\ t_3 \ominus \mathsf{update}_\square\ t_1'\ t_2'\ t_3' \lesssim 0 :^c \overset{\mathrm{diff}(D_1+D_2+D_3)}{\{P \star \gamma \mapsto \beta\}\ \exists \_.\mathrm{unit}_r\ \{P \star \gamma \mapsto \beta \setminus \{I'\}\}}}$$

c-r-updateb

return

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_L^U t :^c A \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^0 \mathsf{return}\ t :^c \overset{\mathrm{exec}(L,U)}{\{P\}\ \exists\gamma.A\ \{P\}}}$$

c-return

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \lesssim D :^c \tau \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Sigma;\Delta \vdash Q \quad wf \qquad \Sigma;\Delta \vDash P \subseteq Q}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{return}\ t_1 \ominus \mathsf{return}\ t_2 \lesssim 0 :^c \overset{\mathrm{diff}(D)}{\{P\}\ \exists \_.\tau\ \{Q\}}}$$

c-r-return

bind

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash_{L_1}^{U_1} t_1 :^c \overset{\mathrm{exec}(L,U)}{\{P\}\ \exists\vec{\gamma}_1.A\ \{P'\}} \qquad \Sigma;\Delta,\vec{\gamma}_1;\Phi_a;\Omega,x:A \vdash_{L_2}^{U_2} t_2 :^c \overset{\mathrm{exec}(L',U')}{\{P'\}\ \exists\vec{\gamma}_2.A'\ \{Q\}}}{\Sigma;\Delta;\Phi_a;\Omega \vdash_0^0 \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 :^c \overset{\mathrm{exec}(L+L'+L_1+L_2+L_l, U+U'+U_1+U_2+U_l)}{\{P\}\ \exists\vec{\gamma}_1,\vec{\gamma}_2.A'\ \{Q\}}}$$

c-bind

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \overset{\mathrm{diff}(D)}{\{P\}\ \exists\vec{\gamma}_1.\tau\ \{P'\}} \qquad \Sigma;\Delta,\vec{\gamma}_1:\vec{\mathbb{L}};\Phi_a;\Gamma,x:\tau \vdash t_2 \ominus t_2' \lesssim D_2 :^c \overset{\mathrm{diff}(D')}{\{P'\}\ \exists\vec{\gamma}_2.\sigma\ \{Q\}}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \lesssim 0 :^c \overset{\mathrm{diff}(D+D'+D_1+D_2)}{\{P\}\ \exists\vec{\gamma}_1\vec{\gamma}_2.\sigma\ \{Q\}}}$$

c-r-bind

$\boxed{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \downarrow \tau, D \Rightarrow \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, the relational typing context Γ, terms t_1 and t_2 check against the input relational type τ and the relative cost D and generates the constraint Φ.

$\boxed{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \uparrow \tau \Rightarrow [\psi], D, \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, the relational typing context Γ, terms t_1 and t_2 synthesize the output relational type τ and the output relative cost D and generates the constraint Φ with all the new generated variables in ψ.

$\boxed{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A, L, U \Rightarrow \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, the unary typing context Ω, term t checks against the input unary type A and its upper bound and lower bound of the execution cost specified by L and U, and generates the constraint Φ.

$\boxed{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \uparrow A \Rightarrow [\psi], L, U, \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, the unary typing context Ω, terms t synthesizes the output unary type τ and the its upper bound and lower bound of the execution cost and generates the constraint Φ with all the new generated variables in ψ.

$\boxed{\Sigma;\Delta;\Psi_a;\Phi_a \models^{\mathsf{A}} A_1 \sqsubseteq A_2 \Rightarrow \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, checks whether A_1 is subtype of A_2 and generates constraints Φ

$\boxed{\Sigma;\Delta;\Psi_a;\Phi_a \models \tau_1 \equiv \tau_2 \Rightarrow \Phi}$ Under the location environment Σ, the index variable environment Δ, the existential variable context Ψ_a, the current constraint environment Φ_a, checks whether τ_1 is equivalent to τ_2 and generates constraints Φ

Figure 30: Bidirectional algorithmic typing judgment explanation

constant integer n

$$\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathtt{n} \uparrow \mathrm{int} \Rightarrow [.], 0, 0, \top \ \textbf{alg-u-n-}\uparrow \qquad \Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathtt{n} \ominus \mathtt{n} \uparrow \mathrm{int}_r \Rightarrow [.], 0, \top \ \textbf{alg-r-n-}\uparrow$$

fix

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a; f : A_1 \xrightarrow{\mathrm{exec}(L',U')} A_2, x : A_1, \Omega \vdash e \downarrow A_2, L', U' \Rightarrow \Phi}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathrm{fix}\ f(x).e \downarrow A_1 \xrightarrow{\mathrm{exec}(L',U')} A_2, L, U \Rightarrow \Phi \wedge L \doteq 0 \wedge 0 \doteq U}\ \textbf{alg-u-fix-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a; f : \tau_1 \xrightarrow{\mathrm{diff}(t')} \tau_2, x : \tau_1, \Gamma \vdash t \ominus t' \downarrow \tau_2, D' \Rightarrow \Phi}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathrm{fix}\ f(x).t \ominus \mathrm{fix}\ f(x).t' \downarrow \tau_1 \xrightarrow{\mathrm{diff}(D')} \tau_2, D \Rightarrow \Phi \wedge 0 \doteq D}\ \textbf{alg-r-fix-}\downarrow$$

variable x

$$\frac{\Omega(x) = A}{\Delta;\psi_a;\Phi_a;\Omega \vdash x \uparrow A \Rightarrow [.], 0, 0, \top}\ \textbf{alg-u-var-}\uparrow \qquad \frac{\Gamma(x) = \tau}{\Delta;\psi_a;\Phi_a;\Gamma \vdash x \ominus x \uparrow \tau \Rightarrow [.], 0, \top}\ \textbf{alg-r-var-}\uparrow$$

switch

$$\frac{\Delta;\psi_a;\Phi_a;|\Gamma| \vdash t_1 \uparrow A \Rightarrow [\psi_1], \_, U_1, \Phi_1 \qquad \Delta;\psi_a;\Phi_a;|\Gamma| \vdash t_2 \uparrow A \Rightarrow [\psi_2], L_2, \_, \Phi_2}{\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathtt{switch}\ t_1 \ominus \mathtt{switch}\ t_2 \uparrow U\,A \Rightarrow [\psi_1, \psi_2], U_1 - L_2, \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-switch}\uparrow$$

$$\frac{\begin{array}{c} L_1, U_1, L_2, U_2 \in \mathrm{fresh}(\mathbb{R}) \\ \Delta; U_1, L_1, \psi_a;\Phi_a;|\Gamma| \vdash t_1 \downarrow A, L_1, U_1 \Rightarrow \Phi_1 \qquad \Delta; U_2, L_2, \psi_a;\Phi_a;|\Gamma| \vdash t_2 \downarrow A, L_2, U_2 \Rightarrow \Phi_2 \end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathtt{switch}\ t_1 \ominus \mathtt{switch}\ t_2 \downarrow D, U\,A \Rightarrow \exists L_1, U_1 :: \mathbb{R}.(\Phi_1 \wedge \exists L_2, U_2 :: \mathbb{R}.\Phi_2 \wedge U_1 - L_2 \doteq D)}\ \textbf{alg-r-switch}\downarrow$$

split

$$\frac{\begin{array}{c} \Sigma;\Delta;\psi_a;C \wedge \Phi_a;\Omega \vdash t_1 \downarrow A, L, U \Rightarrow \Phi_1 \\ \Sigma;\Delta;\psi_a;\neg C \wedge \Phi_a;\Omega \vdash t_1 \downarrow A, L, U \Rightarrow \Phi_2 \qquad \Delta \vdash C\ \mathtt{wf} \end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathtt{split}\ (t_1)\ \mathrm{with}\ C \downarrow A, L, U \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2}\ \textbf{alg-u-split}\downarrow$$

$$\frac{\begin{array}{c} \Sigma;\Delta;\psi_a;C \wedge \Phi_a;\Gamma \vdash t_1 \ominus t_1' \downarrow \tau, D \Rightarrow \Phi_1 \\ \Sigma;\Delta;\psi_a;\neg C \wedge \Phi_a;\Gamma \vdash t_1 \ominus t_1' \downarrow \tau, D \Rightarrow \Phi_2 \qquad \Delta \vdash C\ \mathtt{wf} \end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathtt{split}\ (t_1)\ \mathrm{with}\ C \ominus \mathtt{split}\ (t_1')\ \mathrm{with}\ C \downarrow \tau, D \Rightarrow C \to \Phi_1 \wedge \neg C \to \Phi_2}\ \textbf{alg-r-split}\downarrow$$

contra

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models \bot}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathtt{contra}\ t \downarrow A, L, U \Rightarrow \top}\ \textbf{alg-u-contra}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models \bot}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathtt{contra}\ t \ominus \mathtt{contra}\ t' \downarrow \tau, D \Rightarrow \top}\ \textbf{alg-r-contra}\downarrow$$

Figure 31: Bidirectional algorithmic typing rules, part 1

↑↓

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \uparrow A' \Rightarrow [\psi], L', U', \Phi_1 \qquad \Sigma;\Delta;\psi,\psi_a;\Phi_a \models^{\mathsf{A}} A' \sqsubseteq A \Rightarrow \Phi_2}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A, L, U \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2 \wedge L' \le L \wedge U \le U'}\ \textbf{alg-}\uparrow\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \uparrow \tau' \Rightarrow [\psi], D', \Phi_1 \qquad \Sigma;\Delta;\psi,\psi_a;\Phi_a \models \tau' \equiv \tau \Rightarrow \Phi_2}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \exists(\psi).\Phi_1 \wedge \Phi_2 \wedge D' \le D}\ \textbf{alg-r-}\uparrow\downarrow$$

annotation

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A, L, U \Rightarrow \Phi \qquad \Delta;\Phi_a \vdash^{\mathsf{A}} A\ \mathtt{wf} \qquad \mathrm{FIV}(A, L, U) \in \Delta}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash (t : A, L, U) \uparrow A \Rightarrow [\cdot], L, U, \Phi}\ \textbf{alg-u-anno-}\uparrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \Phi \qquad \Delta;\Phi_a \vdash \tau\ \mathtt{wf} \qquad \mathrm{FIV}(\tau, D) \in \Delta}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash (t : \tau, D) \ominus (t' : \tau, D) \uparrow \tau \Rightarrow [\cdot], D, \Phi}\ \textbf{alg-r-anno-}\uparrow$$

Figure 32: Bidirectional algorithmic typing rules, part 2

Alloc algo

$$\frac{\begin{array}{c}\Sigma;\Delta \vdash L_1, U_1, L_2, U_2 \in \mathrm{fresh}(\mathbb{R}) \\ \Sigma;\Delta;L_1,U_1,\psi_a;\Phi_a;\Omega \vdash t_1 \downarrow \mathrm{int}[I], L_1, U_1 \Rightarrow \Phi_1 \qquad \Sigma;\Delta;L_2,U_2,\psi_a;\Phi_a;\Omega \vdash t_2 \downarrow A, L_2, U_2 \Rightarrow \Phi_2 \\ \Phi = \Phi_2 \wedge L_1 + L_2 + L_a \doteq L \wedge U_1 + U_2 + U_a \doteq U \qquad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{alloc}\, t_1\, t_2 \downarrow \{P\}\ \exists\gamma : \mathsf{Array}_\gamma[I]\ A\ \{P \star \gamma \to \mathbb{N}\}, 0, 0 \Rightarrow \exists L_1, U_1 :: \mathbb{R}.(\Phi_1 \wedge \exists L_2, U_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-u-alc-}\downarrow^{\mathrm{exec}(L,U)}$$

$$\frac{\begin{array}{c}D_1, D_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_1,\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \downarrow \mathrm{int}[I], D_1 \Rightarrow \Phi_1 \qquad \Sigma;\Delta;D_2,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \downarrow \tau, D_2 \Rightarrow \Phi_2 \\ \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \qquad \Sigma \vdash \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{alloc}\, t_1\, t_2 \ominus \mathsf{alloc}\, t_1'\, t_2' \downarrow \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \mathbb{N}\}, 0 \Rightarrow \exists D_1 :: \mathbb{R}.\Phi_1 \wedge (\exists D_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-r-alc-}\downarrow^{\mathrm{diff}(D)}$$

$$\frac{\begin{array}{c}D_1, D_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_1,\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \downarrow \mathrm{int}[I], D_1 \Rightarrow \Phi_1 \qquad \Sigma;\Delta;D_2,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \downarrow \Box\tau, D_2 \Rightarrow \Phi_2 \\ \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \qquad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{alloc}_\Box\, t_1\, t_2 \ominus \mathsf{alloc}_\Box\, t_1'\, t_2' \downarrow \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\}, 0 \Rightarrow \exists D_1 :: \mathbb{R}.\Phi_1 \wedge (\exists D_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-r-alcB-}\downarrow^{\mathrm{diff}(D)}$$

Figure 33: Bidirectional algorithm-typing rules (alloc)

Read algo

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t_1 \uparrow \mathsf{Array}_\gamma[I]\ A \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Omega \vdash t_2 \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \\ \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \qquad \Phi = \Phi_2 \wedge L_1 + L_2 + c_r \doteq L \wedge U_1 + U_2 + c_r \doteq U \qquad P = P' \star \gamma \rightarrow \_ \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{read}\ t_1\ t_2 \downarrow \{P\}\ \overset{\mathsf{exec}(L,U)}{\exists \_ : A}\ \{P\}, 0, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).\Phi))}\ \textbf{alg-u-read-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow \mathsf{Array}_\gamma[I]\ \tau \Rightarrow [\psi_1], D_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\ P = P' \star \gamma \rightarrow \_ \qquad \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \qquad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \downarrow \{P\}\ \overset{\mathsf{diff}(D)}{\exists \_ . \tau}\ \{P\}, 0 \Rightarrow \exists(\psi_1).\Phi_1 \wedge (\exists(\psi_2).\Phi)}\ \textbf{alg-r-read-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow \mathsf{Array}_\gamma[I]\ \tau \Rightarrow [\psi_1], D_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\ P = P' \star \gamma \rightarrow \beta \qquad \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \wedge \neg(I' \in \beta) \qquad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{read}_\square\ t_1\ t_2 \ominus \mathsf{read}_\square\ t_1'\ t_2' \downarrow \{P\}\ \overset{\mathsf{diff}(D)}{\exists \_ . \tau}\ \{P\}, 0 \Rightarrow \exists(\psi_1).\Phi_1 \wedge (\exists(\psi_2).\Phi)}\ \textbf{alg-r-readB-}\downarrow$$

Figure 34: Bidirectional algorithm-typing rules (read)

Update algo

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t_1 \uparrow \mathsf{Array}_\gamma[I]\ A \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Omega \vdash t_2 \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \\ \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \wedge I' \in \beta \qquad L_3, U_3 \in \mathsf{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;L_3,U_3,\psi_2,\psi_1,\psi_a;\Phi_a;\Omega \vdash t_3 \downarrow A, L_3, U_3 \Rightarrow \Phi_3 \\ \Phi = \Phi_3 \wedge L_1 + L_2 + L_3 + c_r \doteq L \wedge U_1 + U_2 + U_3 + c_r \doteq U \qquad P = P' \star \gamma \rightarrow \beta \qquad \Sigma;\Delta \vdash P' \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{update}\ t_1\ t_2\ t_3 \downarrow \{P\}\ \overset{\mathsf{exec}(L,U)}{\exists \_ : \mathsf{unit}}\ \{P\}, 0, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists L_3, U_3 :: \mathbb{R}.\Phi)))}\ \textbf{alg-u-updt-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow \mathsf{Array}_\gamma[I]\ \tau \Rightarrow [\psi_1], D_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\ \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \wedge \beta' = \beta \cup \{I'\} \qquad D_3 \in \mathsf{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_3,\psi_2,\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \downarrow \tau, D_3 \Rightarrow \Phi_3 \\ P = P' \star \gamma \rightarrow \beta \qquad Q = P' \star \gamma \rightarrow \beta' \qquad \Phi = \Phi_2 \wedge D_1 + D_2 + D_3 \doteq D \qquad \Sigma;\Delta \vdash P' \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{update}\ t_1\ t_2\ t_3 \ominus \mathsf{update}\ t_1'\ t_2'\ t_3' \downarrow \{P\}\ \overset{\mathsf{diff}(D)}{\exists \_ . \mathsf{unit}}\ \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi))}\ \textbf{alg-r-updt-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow \mathsf{Array}_\gamma[I]\ \tau \Rightarrow [\psi_1], D_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow \mathsf{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\ \Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \wedge \beta' = \beta / \{I'\} \qquad D_3 \in \mathsf{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_3,\psi_2,\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \downarrow \square\tau, D_3 \Rightarrow \Phi_3 \\ P = P' \star \gamma \rightarrow \beta \qquad Q = P' \star \gamma \rightarrow \beta' \qquad \Phi = \Phi_2 \wedge D_1 + D_2 + D_3 \doteq D \qquad \Sigma;\Delta \vdash P' \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{update}_\square\ t_1\ t_2\ t_3 \ominus \mathsf{update}_\square\ t_1'\ t_2'\ t_3' \downarrow \{P\}\ \overset{\mathsf{diff}(D)}{\exists \_ . \mathsf{unit}}\ \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi))}\ \textbf{alg-r-updtB-}\downarrow$$

Figure 35: Bidirectional algorithm-typing rules (update)

Bind algo

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t_1 \uparrow^{\mathrm{exec}(L,U)} \{P\}\ \exists\vec{\gamma}_1 : A_1\ \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \qquad L_2, U_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;L_2,U_2,\psi_1,\psi_a;\Phi_a;\Omega \vdash t_2 \downarrow^{\mathrm{exec}(L',U')} \{P'\}\ \exists\vec{\gamma}_2 : A_2\ \{Q\}, L_2, U_2 \Rightarrow \Phi_2 \qquad \Phi = \Phi_2 \wedge L + L' + L_1 + L_2 + c_l \doteq L'' \wedge U + U' + U_1 + U_2 + c_l \doteq U''}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{let_m}\{x\} = t_1 \text{ in } t_2 \downarrow^{\mathrm{exec}(L'',U'')} \{P\}\ \exists\_ : A_2\ \{Q\}, 0, 0 \Rightarrow \exists(\psi_1).\Phi_1 \wedge (\exists L_2, U_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-u-bind-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow^{\mathrm{diff}(D)} \{P\}\ \exists\vec{\gamma}_1 : \tau_1\ \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1 \qquad D_2 \in \mathrm{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_2,\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \downarrow^{\mathrm{diff}(D')} \{P'\}\ \exists\vec{\gamma}_2 : \tau_2\ \{Q\}, D_2 \Rightarrow \Phi_2 \qquad \Phi = \Phi_2 \wedge D_1 + D_2 + D + D' \doteq D''}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{let}\ \{x\} = t_1 \text{ in } t_2 \ominus \mathsf{let}\ \{x\} = t_1' \text{ in } t_2' \downarrow^{\mathrm{diff}(D'')} \{P\}\ \exists\_.\tau_2\ \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists D_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-r-bind-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t_1 \uparrow^{\mathrm{exec}(L,U)} \{P\}\ \exists\vec{\gamma}_1 : A_1\ \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Omega \vdash t_2 \uparrow^{\mathrm{exec}(L',U')} \{Q'\}\ \exists\vec{\gamma}_2 : A_2\ \{Q\} \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \qquad P' = Q'}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{let_m}\{x\} = t_1 \text{ in } t_2 \uparrow^{\mathrm{exec}(L+L'+L_1+L_2+c_l,U+U'+U_1+U_2+c_l)} \{P\}\ \exists\_ : A_2\ \{Q\} \Rightarrow [\psi_1,\psi_2], 0, 0, \Phi_1 \wedge \Phi_2}\ \textbf{alg-u-bind-}\uparrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow^{\mathrm{diff}(D)} \{P\}\ \exists\vec{\gamma}_1 : \tau_1\ \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1 \qquad \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow^{\mathrm{diff}(D')} \{Q'\}\ \exists\vec{\gamma}_2 : \tau_2\ \{Q\} \Rightarrow [\psi_2], D_2, \Phi_2 \qquad Q' = P'}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{let_m}\{x\} = t_1 \text{ in } t_2 \ominus \mathsf{let_m}\{x\} = t_1' \text{ in } t_2' \uparrow^{\mathrm{diff}(D+D'+D_1+D_2)} \{P\}\ \exists\_.\tau_2\ \{Q\} \Rightarrow [\psi_1,\psi_2], 0, \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-bind-}\uparrow$$

Figure 36: Bidirectional algorithm-typing rules (let)

Return algo

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A, L, U \Rightarrow \Phi \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{return}\ t \downarrow^{\mathrm{exec}(L,U)} \{P\}\ \exists\_ : A_2\ \{P\}, 0, 0 \Rightarrow \Phi}\ \textbf{alg-u-ret-}\downarrow$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \Phi \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Sigma;\Delta \vdash Q \quad wf \qquad \Sigma;\Delta \vDash P \subseteq Q}{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{return}\ t \ominus \mathsf{return}\ t' \downarrow^{\mathrm{diff}(D)} \{P\}\ \exists\_.\tau\ \{Q\}, 0 \Rightarrow \Phi}\ \textbf{alg-r-ret-}\downarrow$$

Figure 37: Bidirectional algorithmic typing rules (return)

Algorithmic Reltional type equivalence

$$\frac{}{\Sigma;\Delta;\psi_a;\Phi_a \models \text{int}[I] \equiv \text{int}[I'] \Rightarrow I \doteq I'}\ \textbf{alg-r-int-I}$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \qquad \Sigma \vDash \gamma = \gamma'}{\Sigma;\Delta;\psi_a;\Phi_a \models \mathsf{Array}_{\gamma}[I]\ \tau \equiv \mathsf{Array}_{\gamma'}[I']\ \tau' \Rightarrow I \doteq I' \wedge \Phi}\ \textbf{alg-r-array}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\psi_a;\Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1 \\ \Sigma;\Delta;\psi_a;\Phi_a \models P' \equiv P \Rightarrow \Phi_2 \qquad \Sigma;\Delta;\psi_a;\Phi_a \models Q \equiv Q' \Rightarrow \Phi_3 \qquad \Sigma \vDash \vec{\gamma}_1 = \vec{\gamma}_2\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a \models \{P\}\ \overset{\text{diff}(D)}{\exists \vec{\gamma}_1 : \tau}\ \{Q\} \equiv \{P'\}\ \overset{\text{diff}(D)}{\exists \vec{\gamma}_2 : \tau'}\ \{Q'\} \Rightarrow D \doteq D' \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_3}\ \textbf{alg-r-monad}$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models \tau_1 \equiv \tau_1' \Rightarrow \Phi_1 \qquad \Sigma;\Delta;\psi_a;\Phi_a \models \tau_2 \equiv \tau_2' \Rightarrow \Phi_2}{\Sigma;\Delta;\psi_a;\Phi_a \models \tau_1 \overset{\text{diff}(D)}{\longrightarrow} \tau_2 \equiv \tau_1' \overset{\text{diff}(D')}{\longrightarrow} \tau_2' \Rightarrow D \doteq D' \wedge \Phi_1 \wedge \Phi_2}\ \textbf{alg-r-fun}$$

Figure 38: algorithmic relational equivalence rules

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models P \equiv P' \Rightarrow \Phi}{\Sigma;\Delta;\psi_a;\Phi_a \models P \star \gamma \rightarrow \beta_1 \equiv P' \star \gamma \rightarrow \beta_2 \Rightarrow \Phi \wedge \beta_1 \doteq \beta_2}\ \textbf{predicate}$$

$$\frac{}{\Sigma;\Delta;\psi_a;\Phi_a \models \emptyset \equiv \emptyset \Rightarrow \top}\ \textbf{predicate-empty}$$

Figure 39: algorithmic predicate equivalence

Algo Unary subtyping

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} A_1' \sqsubseteq A_1 \Rightarrow \Phi_1 \qquad \Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} A_2 \sqsubseteq A_2' \Rightarrow \Phi_2}{\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} A_1 \overset{\text{exec}(L,U)}{\longrightarrow} A_2 \sqsubseteq A_1' \overset{\text{exec}(L',U')}{\longrightarrow} A_2' \Rightarrow \Phi_1 \wedge \Phi_2 \wedge L' < L \wedge U < U'}\ \textbf{alg-u-fun}$$

$$\frac{\begin{array}{c}\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} A \sqsubseteq A' \Rightarrow \Phi_1 \\ \Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} P' \sqsubseteq P \Rightarrow \Phi_2 \qquad \Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} Q \sqsubseteq Q' \Rightarrow \Phi_3 \qquad \Sigma \vDash \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} \{P\}\ \overset{\text{exec}(k,t)}{\exists \vec{\gamma}_1 : A}\ \{Q\} \sqsubseteq \{P'\}\ \overset{\text{exec}(k',t')}{\exists \vec{\gamma}_2 : A'}\ \{Q'\} \Rightarrow \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \wedge k' \leq k \wedge t \leq t'}\ \textbf{alg-u-monad}$$

$$\frac{\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} P \sqsubseteq P' \Rightarrow \Phi}{\Sigma;\Delta;\psi_a;\Phi_a \models^{\mathsf{A}} P \star \gamma \rightarrow \beta_1 \sqsubseteq P' \star \gamma \rightarrow \beta_2 \Rightarrow \Phi \wedge \beta_1 \sqsubseteq \beta_2}\ \textbf{u-predicate}$$

Figure 40: Bidirectional unary algorithmic subtyping rules

$$
\begin{array}{lcl}
|.| & : & \text{Expression} \to \text{Expression} \\
 & & \\
|\mathtt{n}| & = & \mathtt{n} \\
|()| & = & () \\
|x| & = & x \\
|\mathsf{FIXEXT}\ t \text{ with } U(A_1, A_2)| & = & \mathsf{FIXEXT}\ |t| \text{ with } U(A_1, A_2) \\
|\text{fix } f(x).e| & = & \text{fix } f(x).|t| \\
|t_1\ t_2| & = & |t_1|\,|t_2| \\
\vdots & & \\
|(t : A, L, U)| & = & |t| \\
|(t : \tau, D)| & = & |t|
\end{array}
$$

Figure 41: Annotation erasure

4.1 Metatheory

Lemma 1 (Embedding of Binary Subtyping in ARel)
If $\Delta;\Phi \models \tau \sqsubseteq \tau'$ then $\exists t \in \mathsf{ArelCore}$ such that $\Delta;\Phi;\cdot \vdash t \ominus t \lesssim 0 :^c \tau \xrightarrow{\mathrm{diff}(0)} \tau'$.

Proof. Proof is by induction on the subtyping derivation. We denote the witness t of type $\tau \xrightarrow{\mathrm{diff}(0)} \tau'$ as $\mathtt{coerce}_{\tau,\tau'}$ for clarity.

$$\frac{\Sigma;\Delta;\Phi \models \tau \sqsubseteq \tau' \quad \Delta;\Phi \models D \leq D' \qquad \Sigma,\vec{\gamma}_1;\Delta;\Phi \models Q \sqsubseteq Q' \qquad \Sigma;\Delta;\Phi \models P' \sqsubseteq P \qquad \vec{\gamma}_1 \subseteq \vec{\gamma}_2}{\Sigma;\Delta;\Phi \models \{P\}\ \exists \vec{\gamma}_1.\tau^{\mathrm{diff}(D)}\ \{Q\} \sqsubseteq \{P'\}\ \exists \vec{\gamma}_2.\tau'^{\mathrm{diff}(D')}\ \{Q'\}}\ \textbf{S-RM}$$

Case
By IH on $(\star)$, $\exists \mathtt{coerce}_{\tau,\tau'}\ .\ \ i :: S,\Delta;\Phi;\cdot \vdash \mathtt{coerce}_{\tau,\tau'} \ominus \mathtt{coerce}_{\tau,\tau'} \lesssim 0 :^c \tau \xrightarrow{\mathrm{diff}(0)} \tau'$
We can construct the following derivation where $t = \lambda x.\mathsf{let}\ \{y\} = (x)\ \mathsf{in}\ \mathsf{return}\,\mathtt{coerce}_{\tau,\tau'}\ y$ using the **c-r-≡** ,**c-r- bind** and **c-r-return** rules.

$$\Delta;\Phi;\cdot \vdash t \ominus t \lesssim 0 :^c (\{P\}\ \exists \vec{\gamma}_1.\tau^{\mathrm{diff}(D)}\ \{Q\}) \xrightarrow{\mathrm{diff}(0)} (\{P'\}\ \exists \vec{\gamma}_2.\tau'^{\mathrm{diff}(D')}\ \{Q'\})$$

$$\frac{\beta'_i = \beta_i \cup T_i \cup T'_i}{\Sigma;\Delta;\Phi \models U(\{\gamma_i \rightarrowtail T_i\}\ \exists \vec{\gamma}'_1.A_1^{\mathrm{exec}(L,U)}\ \{Q_1\},\{\gamma_i \rightarrowtail T'_i\}\ \exists \vec{\gamma}'_2.A_2^{\mathrm{exec}(L',U')}\ \{Q_2\}) \sqsubseteq \{\gamma_i \rightarrowtail \beta_i\}\ \exists \vec{\gamma}_1,\vec{\gamma}_2.U(A_1,A_2)^{\mathrm{diff}(U-L')}\ \{\gamma_i \rightarrowtail \beta'_i\}}\ \textbf{S-RUM}$$

Case
We can construct the following derivation where $t = \lambda x.\mathsf{let}\ \{y\} = (\mathtt{switch}\ x)\ \mathsf{in}\ \mathsf{return}\ y$ using the **c-switch** ,**c-r- bind** and **c-r-return** rules.

$$\Delta;\Phi;\cdot \vdash t \ominus t \lesssim 0 :^c (U(\{\gamma_i \rightarrowtail T_i\}\ \exists \vec{\gamma}'_1.A_1^{\mathrm{exec}(L,U)}\ \{Q_1\},\{\gamma_i \rightarrowtail T'_i\}\ \exists \vec{\gamma}'_2.A_2^{\mathrm{exec}(L',U')}\ \{Q_2\})) \xrightarrow{\mathrm{diff}(0)} (\{\gamma_i \rightarrowtail \beta_i\}\ \exists \vec{\gamma}_1,\vec{\gamma}_2.U(A_1,A_2)^{\mathrm{diff}(U-L')}\ \{\gamma_i \rightarrowtail \beta'_i\})$$

□

Lemma 2 (Reflexivity of Predicate Equivalence in ARel)
$\Delta;\psi_a;\Phi_a \models P \equiv P \Rightarrow \Phi$ and $\Delta;\psi_a;\Phi_a \models \Phi$.

Proof. By induction on the definiton of the predicate P. □

Lemma 3 (Reflexivity of Algorithmic Binary Type Equivalence in ARel)
$\Delta;\psi_a;\Phi_a \models \tau \equiv \tau \Rightarrow \Phi$ and $\Delta;\psi_a;\Phi_a \models \Phi$.

Proof. By induction on the binary type. □

Lemma 4 (Reflexivity of Predicate Subtyping in ARel)
$\Delta;\Phi_a \models^{\mathsf{A}} P \sqsubseteq P \Rightarrow \Phi$ and $\Delta;\Phi_a \models \Phi$.

Proof. By induction on the definiton of the predicate P. □

Lemma 5 (Reflexivity of Unary Algorithmic Subtyping in ARel)
$\Delta;\Phi_a \models^{\mathsf{A}} A \sqsubseteq A \Rightarrow \Phi$ and $\Delta;\Phi_a \models \Phi$.

Proof. By induction on the unary type. □

Lemma 6 (Transitivity of Unary Algorithmic Subtyping in ARel)
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 7 (Soundness of the Algorithmic Unary Subtyping in ARel)
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 $\Delta \rhd \theta_a : \psi_a$ 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 8 (Completeness of the Unary Algorithmic Subtyping in ARel)
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 9 (Soundness of the Algorithmic Predicate Equality in ARel)
Assume that

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

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

Proof. By induction on the algorithmic predicate equivalence derivation. □

Theorem 10 (Soundness of the Algorithmic Binary Type Equality in ARel)
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 s.t $\Delta \rhd \theta_a : \psi_a$ 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 11 (Completeness of the Algorithmic Predicate Equivalence in ARel)
Assume that $\Delta; \Phi_a \models P \equiv P'$. Then $\exists \Phi$. such that $\Delta; \Phi_a \models P \equiv P' \Rightarrow \Phi$ and $\Delta; \Phi_a \models \Phi$.

Proof. By induction on the predicate equivalence derivation. □

Theorem 12 (Completeness of the Binary Algorithmic Type Equivalence in ARel)
Assume that $\Sigma; \Delta; \Phi_a \models \tau' \equiv \tau$. 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 13 (Soundness of ArelCore & Type Preservation of Embedding)
The following holds.

1. If $\Sigma; \Delta; \Phi; \Omega \vdash^{U}_{L} t \rightsquigarrow t^* : A$, then $\Sigma; \Delta; \Phi; \Omega \vdash^{U}_{L} t^* :^{c} A$ and $\Sigma; \Delta; \Phi; \Omega \vdash^{U}_{L} t : A$.
2. If $\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \rightsquigarrow t_1^* \ominus t_2^* \lesssim D : \tau$, then $\Sigma; \Delta; \Phi; \Gamma \vdash t_1^* \ominus t_2^* \lesssim D :^{c} \tau$ and $\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$.

Proof. Proof is by simultaneous induction on the embedding derivations. The proof follows from the embedding rules.

Proof of Theorem 13.2:

Case $\dfrac{\Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_1}_{L_1} t_1 \rightsquigarrow t_1^* : A \ (\star) \qquad \Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_2}_{L_2} t_2 \rightsquigarrow t_2^* : A \ (\diamond)}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow \mathtt{switch}\ t_1^* \ominus \mathtt{switch}\ t_2^* \lesssim U_1 - L_2 : U\ A}$ **e-switch**

By Theorem 13.1 on ($\star$), we get $\Delta;\Phi;|\Gamma| \vdash^{U_1}_{L_1} t_1^* :^c A_1 \ (\star\star)$.

By Theorem 13.1 on ($\diamond$), we get $\Delta;\Phi;|\Gamma| \vdash^{U_2}_{L_2} t_2^* :^c A_2 \ (\diamond\diamond)$.

Then, we conclude as follows:

$$\frac{\Delta;\Phi_a;|\Gamma| \vdash^{U_1}_{L_1} t_1 :^c A \qquad \Delta;\Phi_a;|\Gamma| \vdash^{U_2}_{L_2} t_2 :^c A}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathtt{switch}\ t_1 \ominus \mathtt{switch}\ t_2 \lesssim U_1 - L_2 :^c U\ A}\ \textbf{c-switch}$$

.

Case $\dfrac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow {t_1}^* \ominus {t_2}^* \lesssim D : \tau \ (\star) \\ \Delta;\Phi_a \models \tau \sqsubseteq \tau' \ (\diamond) \qquad t' = \mathtt{coerce}_{\tau,\tau'} \qquad \Delta;\Phi_a \models D \leq D' \ (\dagger)\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow t'\ t_1^* \ominus t'\ t_2^* \lesssim D' : \tau'}$ **e-r-sub**

By Theorem 13.2 on ($\star$), $\Delta;\Phi;\Gamma \vdash t_1^* \ominus t_2^* \lesssim D :^c \tau \ (\star\star)$.

By Lemma 1 using ($\diamond$), we know that $\Delta;\Phi;\cdot \vdash t' \ominus t' \lesssim 0 :^c \tau \xrightarrow{\mathrm{diff}(0)} \tau' \ (\diamond\diamond)$.

By applying **c-r-app** rule and ($\star\star$) and ($\diamond\diamond$), we get $\Delta;\Phi;\Gamma \vdash t'\ t_1^* \ominus t'\ t_2^* \lesssim D :^c \tau' \ (\spadesuit)$.

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

Then, we conclude as follows:

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t \ominus t' \lesssim D :^c \tau \ (\spadesuit) \qquad \Delta;\Phi_a \models \tau \equiv \tau' \ (\spadesuit\spadesuit) \\ \Delta;\Phi_a \models D \leq D' \ (\dagger)\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t \ominus t' \lesssim D' :^c \tau'}\ \textbf{c-r-}\equiv$$

Case $\dfrac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathbf{int}[I] \ (\star) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \tau \ (\diamond) \qquad \gamma\ \mathbf{fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \rightsquigarrow \mathsf{alloc}\ t_1^*\ t_2^* \ominus \mathsf{alloc}\ t_1'^*\ t_2'^* \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \rightarrow \mathbb{N}\}}$ **e-r-alloc**

By Theorem 13.2 on ($\star$), $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathrm{int}[I] \ (\star\star)$.

By Theorem 13.2 on ($\diamond$), we know that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \tau \ (\diamond\diamond)$.

By applying **c-r-alloc** rule and ($\star\star$) and ($\diamond\diamond$), we get:

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathrm{int}[I] \quad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \tau \quad \gamma\ \mathrm{fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 :^c \{P\}\ \exists\gamma.\ \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \rightarrow \mathbb{N}\}}\ \textbf{c-r-alloc}$$

Case $\dfrac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \ (\star) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathbf{int}[I'] \ (\diamond) \qquad \Delta;\Phi_a \models I' \leq I \qquad I' \notin \beta \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \rightsquigarrow \mathsf{read}\ t_1^*\ t_2^* \ominus \mathsf{read}\ t_1'^*\ t_2'^* \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P \star \gamma \mapsto \beta\}\ \exists\gamma.\Box\tau\ \{P \star \gamma \mapsto \beta\}}$ **e-r-readb**

By Theorem 13.2 on ($\star$), $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau \ (\star\star)$.

By Theorem 13.2 on ($\diamond$), we know that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathrm{int}[I'] \ (\diamond\diamond)$.

By applying **c-r-readb** rule and ($\star\star$) and ($\diamond\diamond$), we get:

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau\ (\star\star) \quad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathsf{int}[I']\ (\diamond\diamond) \quad \Delta;\Phi_a \models I' \le I \quad I' \notin \beta \quad \Sigma;\Delta \vdash P\ \ wf}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}_\square\ t_1\ t_2 \ominus \mathsf{read}_\square\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 :^c \{P \star \gamma \mapsto \beta\}\ \exists\_.\square\tau\ \{P \star \gamma \mapsto \beta\}}\ \textbf{c-r-readb}$$

$$\textbf{Case}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau\ (\star) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathbf{int}[I']\ (\diamond) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \rightsquigarrow {t_3}^* \ominus {t_3'}^* \lesssim D_3 : \square\tau\ (\dagger) \quad \Delta;\Phi_a \models I' \le I \quad \Sigma;\Delta \vdash P\ \ wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{updt}\ t_1\ t_2\ t_3 \ominus \mathsf{updt}\ t_1'\ t_2'\ t_3' \rightsquigarrow \mathsf{update}_\square\ t_1^*\ t_2^*\ t_3^* \ominus \mathsf{update}_\square\ t_1'^*\ t_2'^*\ t_3'^* \overset{\mathrm{diff}(D_1+D_2+D_3)}{\lesssim} 0 : \{P \star \gamma \mapsto \beta\}\ \exists\gamma.\tau\ \{P \star \beta \setminus \{I'\}\}}\ \textbf{e-r-u}$$

By Theorem 13.2 on $(\star)$, $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau$ $(\star\star)$.

By Theorem 13.2 on $(\diamond)$, we know that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathsf{int}[I']$ $(\diamond\diamond)$.

By Theorem 13.2 on $(\dagger)$, we know that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 :^c \square\tau$ $(\dagger\dagger)$.

By applying **c-r-updateb** rule and $(\star\star)$, $(\diamond\diamond)$ and $(\dagger,\dagger)$, we get:

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \mathsf{Array}_\gamma[I]\ \tau\ (\star\star) \quad \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 :^c \mathsf{int}[I']\ (\diamond\diamond) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \lesssim D_3 :^c \square\tau\ (\dagger\dagger) \quad \Delta;\Phi_a \models I' \le I \quad \Sigma;\Delta \vdash P\ \ wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{update}_\square\ t_1\ t_2\ t_3 \ominus \mathsf{update}_\square\ t_1'\ t_2'\ t_3' \overset{\mathrm{diff}(D_1+D_2+D_3)}{\lesssim} 0 :^c \{P \star \gamma \mapsto \beta\}\ \exists\_.\mathrm{unit}_r\ \{P \star \gamma \mapsto \beta \setminus \{I'\}\}}\ \textbf{c-r-updateb}$$

□

Theorem 14 (Completeness of ArelCore)
The following holds.

1. If $\Sigma;\Delta;\Phi;\Omega \vdash^U_L t : A$ then, $\exists\, t^*$ such that $\Sigma;\Delta;\Phi;\Omega \vdash^U_L t \rightsquigarrow t^* : A$.
2. If $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ then, $\exists\, t_1^*$ and $\exists\, t_2^*$ such that $\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow t_1^* \ominus t_2^* \lesssim D : \tau$.

Proof. Proof is by simultaneous induction on the typing derivations.

Proof of Theorem 14.1:

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi;\Omega \vdash^U_L t : A\ (\star) \quad \Sigma;\Delta;\Phi \models A \sqsubseteq A' \quad \Sigma;\Delta \models U \le U' \quad \Sigma;\Delta \models L' \le L}{\Sigma;\Delta;\Phi;\Omega \vdash^{U'}_{L'} t : A'}\ \textbf{u-x}$$

By Theorem 14.1 on $(\star)$, we get $\exists\, t^*$ such that $\Sigma;\Delta;\Phi_a;\Omega \vdash^U_L t \rightsquigarrow t^* : A$ $(\star\star)$.

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

$$\frac{\Sigma;\Delta;\Phi_a;\Omega \vdash^U_L t \rightsquigarrow t^* : A\ (\star\star) \quad \Delta;\Phi_a \models A \sqsubseteq A' \quad \Delta;\Phi_a \models L' \le L \quad \Delta;\Phi_a \models U \le U'}{\Delta;\Phi_a;\Omega \vdash^{t'}_{k'} e \rightsquigarrow e^* : A'}\ \textbf{e-u-sub}$$

Proof of Theorem 14.2:

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi;|\Gamma|_1 \vdash^{U_1}_{L_1} t_1 : A_1\ (\star) \quad \Sigma;\Delta;\Phi;|\Gamma|_2 \vdash^{U_2}_{L_2} t_2 : A_2\ (\diamond)}{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_2 \lesssim U_1 - L_2 : U(A_1, A_2)}\ \textbf{r-s}$$

By Theorem 14.1 on ($\star$), we get $\exists\, t_1^*$ such that $\Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_1}_{L_1} t_1 \rightsquigarrow t_1^* : A_1 \quad (\star\star)$.
By Theorem 14.1 on ($\diamond$), we get $\exists\, t_2^*$ such that $\Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_2}_{L_2} t_2 \rightsquigarrow t_2^* : A_2 \quad (\star\star)$.
By **e-switch** rule using ($\star\star$) and ($\diamond\diamond$) ,we conclude as follows.

$$\frac{\Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_1}_{L_1} t_1 \rightsquigarrow t_1^* : A_1 \;\; (\star\star) \qquad \Sigma;\Delta;\Phi_a;|\Gamma| \vdash^{U_2}_{L_2} t_2 \rightsquigarrow t_2^* : A_2 \;\; (\diamond\diamond)}{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \rightsquigarrow \mathtt{switch}\ t_1^* \ominus \mathtt{switch}\ t_2^* \lesssim U_1 - L_2 : U(A_1, A_2)}\ \textbf{e-switch}$$

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathbf{int}[I] \;\; (\star) \qquad \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \Box\tau \;\; (\diamond) \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists\gamma.\, \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\}}\ \textbf{r-alloc-box}$$

By Theorem 14.2 on ($\star$), we get $\exists\, t_1^*$ such that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathrm{int}[I] \quad (\star\star)$.
By Theorem 14.2 on ($\diamond$), we get $\exists\, t_2^*$ such that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \Box\tau \;\; (\diamond\diamond)$.
By **e-r-allocb** rule using ($\star\star$) and ($\diamond\diamond$) ,we conclude as follows.

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathrm{int}[I] \;\; (\star\star) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \Box\tau \;\; (\diamond\diamond) \qquad \gamma \text{ fresh} \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{alloc}\ t_1\ t_2 \ominus \mathsf{alloc}\ t_1'\ t_2' \rightsquigarrow \mathsf{alloc}_\Box\ t_1^*\ t_2^* \ominus \mathsf{alloc}_\Box\ t_1'^*\ t_2'^* \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists\gamma.\, \mathsf{Array}_\gamma[I]\ \tau\ \{P \star \gamma \to \emptyset\}}\ \textbf{e-r-allocb}$$

$$\textbf{Case}\ \frac{\begin{array}{c}\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \\ \Sigma;\Delta;\Phi;\Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \mathbf{int}[I'] \qquad \Delta;\Phi \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P \star \gamma \to \beta\}\ \exists\_.\tau\ \{P \star \gamma \to \beta\}}\ \textbf{r-r}$$

By Theorem 14.2 on ($\star$), we get $\exists\, t_1^*$ such that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \quad (\star\star)$.
By Theorem 14.2 on ($\diamond$), we get $\exists\, t_2^*$ such that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \quad (\diamond\diamond)$.
By **e-r-read** rule using ($\star\star$) and ($\diamond\diamond$) ,we conclude as follows.

$$\frac{\begin{array}{c}\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \mathsf{Array}_\gamma[I]\ \tau \;\; (\star\star) \\ \Sigma;\Delta;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \mathrm{int}[I'] \;\; (\diamond\diamond) \qquad \Delta;\Phi_a \models I' \le I \qquad \Sigma;\Delta \vdash P \quad wf\end{array}}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{read}\ t_1\ t_2 \ominus \mathsf{read}\ t_1'\ t_2' \rightsquigarrow \mathsf{read}\ t_1^*\ t_2^* \ominus \mathsf{read}\ t_1'^*\ t_2'^* \overset{\mathrm{diff}(D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists\gamma.\tau\ \{P\}}\ \textbf{e-r-read}$$

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi;\Gamma \vdash t_1 \ominus t_1' \overset{\mathrm{diff}(D)}{\lesssim} D_1 : \{P\}\ \exists\vec{\gamma}_1.\tau\ \{P'\} \;\; (\star) \qquad \Sigma;\Delta,\vec{\gamma}_1 : \vec{\mathbb{L}};\Phi;\Gamma, x : \tau \vdash t_2 \ominus t_2' \overset{\mathrm{diff}(D')}{\lesssim} D_2 : \{P'\}\ \exists\vec{\gamma}_2.\sigma\ \{Q\} \;\; (\diamond)}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \overset{\mathrm{diff}(D+D'+D_1+D_2)}{\lesssim} 0 : \{P\}\ \exists\vec{\gamma}_1\vec{\gamma}_2 : \sigma\ \{Q\}}\ \textbf{r-bind}$$

By Theorem 14.2 on ($\star$), we get $\exists\, t_1^*$ such that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \overset{\mathrm{diff}(D)}{\lesssim} D_1 : \{P_1\}\ \exists\vec{\gamma}_1.\tau\ \{Q_1 \star Q_2\}$
$D_1 \quad (\star\star)$.
By Theorem 14.2 on ($\diamond$), we get $\exists\, t_2^*$ such that $\Sigma;\Delta,\vec{\gamma}_1 : \vec{\mathbb{L}};\Phi_a;\Gamma, x : \tau \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \overset{\mathrm{diff}(D')}{\lesssim} D_2 : \{Q_1 \star P_2\}\ \exists\vec{\gamma}_2.\tau'\ \{Q\} \;\; (\diamond\diamond)$.
By **e-r-bind** rule using ($\star\star$) and ($\diamond\diamond$) ,we conclude as follows.

$$\frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \rightsquigarrow {t_1}^* \ominus {t_1'}^* \lesssim D_1 : \{P\}\ \overset{\mathrm{diff}(D)}{\exists \vec{\gamma}_1.\tau}\ \{P'\}\ D_1\ (\star\star) \quad \Sigma;\Delta,\vec{\gamma}_1:\vec{\mathbb{L}};\Phi_a;\Gamma,x:\tau \vdash t_2 \ominus t_2' \rightsquigarrow {t_2}^* \ominus {t_2'}^* \lesssim D_2 : \{P'\}\ \overset{\mathrm{diff}(D')}{\exists \vec{\gamma}_2.\tau'}\ \{Q\}\ (\diamond\diamond)}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \rightsquigarrow \mathsf{let}\ \{x\} = t_1^*\ \mathsf{in}\ t_2^* \ominus \mathsf{let}\ \{x\} = t_1'^*\ \mathsf{in}\ t_2^* \lesssim 0 :\ \overset{\mathrm{diff}(D_1+D_2+D+D')}{\{P\}\ \exists\_.\tau'}\ \{Q\}}\ \textbf{e-r-bind}$$

□

Theorem 15 (Invariant of the Algorithmic Typechecking)
We have the following.

1. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A,L,U \Rightarrow \Phi$ and $\mathrm{FIV}(\Phi_a,\Omega;A,L,U) \subseteq dom(\Delta,\psi_a)$. Then $\mathrm{FIV}(\Phi) \subseteq dom(\Delta;\psi_a)$.
2. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \uparrow A \Rightarrow [\psi],L,U,\Phi$ and $\mathrm{FIV}(\Phi_a,\Omega) \subseteq dom(\Delta,\psi_a)$. Then $\mathrm{FIV}(A,L,U,\Phi) \subseteq dom(\Delta,\psi;\psi_a)$.
3. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \downarrow \tau,D \Rightarrow \Phi$ and $\mathrm{FIV}(\Phi_a,\Gamma,\tau,D) \subseteq dom(\Delta,\psi_a)$.Then $\mathrm{FIV}(\Phi) \subseteq dom(\Delta;\psi_a)$.
4. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \uparrow \tau \Rightarrow [\psi],D,\Phi$ and $\mathrm{FIV}(\Phi_a,\Gamma) \subseteq dom(\Delta,\psi_a)$.Then $\mathrm{FIV}(\tau,D,\Phi) \subseteq dom(\Delta;\psi;\psi_a)$.

Theorem 16 (Soundness of the Algorithmic Typechecking in ARel)
We have the following.

1. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A,L,U \Rightarrow \Phi$ and
   1.1. $\mathrm{FIV}(\Phi_a,\Omega,A,L,U) \subseteq dom(\Delta,\psi_a)$
   1.2. $\Sigma;\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$ is provable for some θ_a such that $\Delta \rhd \theta_a : \psi_a$ is derivable

   Then $\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U[\theta_a]}_{L[\theta_a]} |t| :^c A[\theta_a]$.
2. Assume that $\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t \uparrow A \Rightarrow [\psi],L,U,\Phi$ and
   2.1. $\mathrm{FIV}(\Phi_a,\Omega) \subseteq dom(\Delta,\psi_a)$
   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 $\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U[\theta\,\theta_a]}_{L[\theta\,\theta_a]} |t| :^c A[\theta\,\theta_a]$.
3. Assume that $\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \downarrow \tau,D \Rightarrow \Phi$ and
   3.1. $\mathrm{FIV}(\Phi_a,\Gamma,\tau,D) \subseteq dom(\Delta,\psi_a)$
   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 $\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t| \ominus |t'| \lesssim D[\theta_a] :^c \tau[\theta_a]$.
4. Assume that $\Delta;\psi_a;\Phi_a;\Gamma \vdash t \ominus t' \uparrow \tau \Rightarrow [\psi],D,\Phi$ and
   4.1. $\mathrm{FIV}(\Phi_a,\Gamma) \subseteq dom(\Delta,\psi_a)$
   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 $\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t| \ominus |t'| \lesssim D[\theta\,\theta_a] :^c \tau[\theta\,\theta_a]$.

Proof. Statements (1—4) follow from simultaneous structural induction on the algorithmic typing derivations. We present several cases below.

Proof of Theorem 16.1:

Case $\dfrac{\Delta;\psi_a;\Phi_a;\Omega \vdash t \downarrow A, L, U \Rightarrow \Phi\ (\diamond) \qquad \Sigma;\Delta \vdash P \quad wf}{\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{return}\ t \downarrow \overset{\mathsf{exec}(L,U)}{\{P\}\ \exists\_ : A\ \{P\}}, 0, 0 \Rightarrow \Phi}$ **alg-u-ret-↓**

TS: $\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U[\theta_a]}_{L[\theta_a]} |\ \mathsf{return}\ t\ | :^c \overset{\mathsf{exec}(L[\theta_a],U[\theta_a])}{\{P\}\ \exists\_ : A\ \{P\}}\ [\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Omega, (\overset{\mathsf{exec}(L,U)}{\{P\}\ \exists\_ : A\ \{P\}}), L, U) \subseteq dom(\Delta, \psi_a)$ $(\star)$

$\Delta;\Phi_a[\theta_a] \models (\Phi)[\theta_a]$ $(\star\star)$

Using $(\star)$, $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta;\Phi_a[\theta_a] \models \Phi[\theta_a]$

By Theorem 16.1 on $(\diamond)$ using $(\star)$ and b), we can show that

$$\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U[\theta_a]}_{L[\theta_a]} |t| :^c A[\theta_a] \tag{4.1}$$

Using rule **c-ret** rule, we get

$\dfrac{\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{u[\theta_a]}_{L[\theta_a]} |t| :^c A[\theta_a] \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^0_0 |\ \mathsf{return}\ t\ | :^c \overset{\mathsf{exec}(L[\theta_a],U[\theta_a])}{\{P\}\ \exists\gamma.A\ \{P\}}}$ **c-return**

Proof of Theorem 16.2:

Case $\dfrac{\begin{array}{c}\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash t_1 \uparrow \overset{\mathsf{exec}(L,U)}{\{P\}\ \exists\vec{\gamma}_1 : A_1\ \{P'\}} \Rightarrow [\psi_1], L_1, U_1, \Phi_1\ (\dagger) \\ \Sigma;\Delta;\psi_1,\psi_a;\Phi_a;\Omega, x : A_1 \vdash t_2 \uparrow \overset{\mathsf{exec}(L',U')}{\{Q'\}\ \exists\vec{\gamma}_2 : A_2\ \{Q\}} \Rightarrow [\psi_2], L_2, U_2, \Phi_2\ (\spadesuit) \qquad P' = Q'\end{array}}{\Sigma;\Delta;\psi_a;\Phi_a;\Omega \vdash \mathsf{let_m}\{x\} = t_1\ \mathsf{in}\ t_2 \uparrow \overset{\mathsf{exec}(L+L'+L_1+L_2+C_L, U+U'+U_1+U_2+c_l)}{\{P\}\ \exists\_ : A_2\ \{Q\}} \Rightarrow [\psi_1,\psi_2], 0, 0, \Phi_1 \wedge \Phi_2}$ **alg-u-bind-↑**

TS: $\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{(U_1+U_2+c_l)[\theta\theta_a]}_{(L_1+L_2+c_l)[\theta\theta_a]} |\ \mathsf{let_m}\{x\} = t_1\ \mathsf{in}\ t_2\ | :^c \overset{\mathsf{exec}((L+L')[\theta\theta_a],(U+U')[\theta\theta_a])}{\{P\}\ \exists\_ : A\ \{P\}}\ [\theta\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Omega) \subseteq dom(\Delta, \psi_a)$ $(\star)$

$\Delta;\Phi_a[\theta_a] \models (\Phi_1 \wedge \Phi_2)[\theta\theta_a]$ $(\star\star)$

From $(\star\star)$, we know : $\Delta;\Phi_a[\theta_a] \models (\Phi_1)[\theta\theta_a]$ $(\diamond)$ and $\Delta;\Phi_a[\theta_a] \models (\Phi_2)[\theta\theta_a]$ $(\diamond\diamond)$ By Theorem 16.2 on $(\dagger)$ using $(\star)$ and $(\diamond)$, we can show that

$\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U_1[\theta\theta_a]}_{L_1[\theta\theta_a]} |t_1| :^c \{P\}\ \overset{\mathsf{exec}(L,U)}{\exists\vec{\gamma}_1 : A_1}\ \{P'\}[\theta\theta_a]$ $(\dagger\dagger)$.

From Theorem 15, we know that $\mathrm{FIV}(A_1) \subseteq dom(\psi_1)$ which implies $\mathrm{FIV}(\Phi_a, \Omega, x : A_1) \subseteq dom(\Delta, (\psi_a, \psi_1))$ $(\heartsuit)$

.

By Theorem 16.2 on $(\spadesuit)$ using $(\heartsuit)$ and $(\diamond\diamond)$, we can show that

$\Sigma;\Delta;\Phi_a[\theta_a];\Omega, x : A_1[\theta_a] \vdash^{U_2[\theta\theta_a]}_{L_2[\theta\theta_a]} |t_2| :^c \{Q'\}\ \overset{\mathsf{exec}(L',U')}{\exists\vec{\gamma}_2 : A_2}\ \{Q\}[\theta\theta_a]$ $(\spadesuit\spadesuit)$.

$$\dfrac{\begin{array}{c}\Sigma;\Delta;\Phi_a[\theta_a];\Omega[\theta_a] \vdash^{U_1[\theta\theta_a]}_{L_1[\theta\theta_a]} |t_1| :^c \{P\}\ \overset{\mathsf{exec}(L,U)}{\exists\vec{\gamma}_1 : A_1}\ \{P'\}[\theta\theta_a]\ (\dagger\dagger) \\ \Sigma;\Delta;\Phi_a[\theta_a];\Omega, x : A_1[\theta_a] \vdash^{U_2[\theta\theta_a]}_{L_2[\theta\theta_a]} |t_2| :^c \{Q'\}\ \overset{\mathsf{exec}(L',U')}{\exists\vec{\gamma}_2 : A_2}\ \{Q\}[\theta\theta_a]\ (\spadesuit\spadesuit)\end{array}}{\Sigma;\Delta;\Phi_a;\Omega \vdash^0_0 \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 :^c \overset{\mathsf{exec}((L+L'+L_1+L_2+c_l)[\theta\theta_a],(U+U'+U_1+U_2+c_l)[\theta\theta_a])}{\{P\}\ \exists\vec{\gamma}_1, \vec{\gamma}_2.A_2\ \{Q\}}}$$ **c-bind**

Proof of Theorem 16.3:

$$\textbf{Case}\ \frac{\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \uparrow \mathsf{Array}_\gamma[I]\ \tau \Rightarrow [\psi_1], D_1, \Phi_1\ (a) \qquad \Sigma;\Delta;\psi_a,\psi_1;\Phi_a;\Gamma \vdash t_2 \ominus t_2' \uparrow \mathbf{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2\ (b)}{\Sigma;\Delta;\psi_a;\Phi_a \vDash I' \leq I \wedge \beta' = \beta/\{I'\}} \qquad D_3 \in \mathbf{fresh}(\mathbb{R}) \qquad \Sigma;\Delta;D_3,\psi_2,\psi_1,\psi_a;\Phi_a;\Gamma \vdash t_3 \ominus t_3' \downarrow \square\tau, D_3 \Rightarrow \Phi_3\ (c)$$
$$P = P' \star \gamma \rightarrow \beta \qquad Q = P' \star \gamma \rightarrow \beta' \qquad \Phi = \Phi_2 \wedge D_1 + D_2 + D_3 \doteq D \qquad \Sigma;\Delta \vdash P' \quad wf$$
$$\textbf{alg-r-u}$$
$$\Sigma;\Delta;\psi_a;\Phi_a;\Gamma \vdash \mathsf{update}_\square\ t_1\ t_2\ t_3 \ominus \mathsf{update}_\square\ t_1'\ t_2'\ t_3' \downarrow \{P\}\ \exists\_.\overset{\mathrm{diff}(D)}{\mathbf{unit}}\ \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi)))$$

TS: $\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |\,\mathsf{update}_\square\ t_1\ t_2\ t_3\,| \ominus |\,\mathsf{update}_\square\ t_1'\ t_2'\ t_3'\,| \lesssim D[\theta_a] :^c \{P\}\ \exists\_.\overset{\mathrm{diff}(D)}{\mathsf{unit}}\ \{Q\}[\theta_a]$.

By the main assumptions, we have $\mathrm{FIV}(\Phi_a, \Gamma, (\{P\}\ \exists\_.\overset{\mathrm{diff}(D)}{\mathsf{unit}}\ \{Q\}), D) \subseteq dom(\Delta, \psi_a)$ $(\star)$

$\Delta;\Phi_a[\theta_a] \vDash (\exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi))))[\theta_a]$ $(\star\star)$

By Theorem 15 on $(\star)$ and the first premise, then we get $\mathrm{FIV}(\mathsf{Array}_\gamma[I]\ \tau, D_1, \Phi_1) \subseteq dom(\Delta, \psi_a, \psi_1)$ (1).

Similarly on the second premise, we have $\mathrm{FIV}(\mathrm{int}[I'], D_2, \Phi_2) \subseteq dom(\Delta, \psi_a, \psi_1, \psi_2)$ (2)

Using $(\star)$, (1), (2), $(\star\star)$'s derivation must be in a form such that we have

a) $\Delta \vdash d_3 :: \mathbb{R}$

b) $\Delta;\Phi_a[\theta_a] \vDash \Phi_1[\theta_a]$

c) $\Delta;\Phi_a[\theta_a] \vDash \Phi_2[\theta_a]$

d) $\Delta;\Phi_a[\theta_a] \vDash \Phi_3[\theta_a, D_3 \mapsto d_3]$

e) $\Delta;\Phi_a[\theta_a] \vDash (D_1[\theta_a] + D_2[\theta_a] + d_3) \doteq D[\theta_a]$

By Theorem 16.4 on (a) using (1) and b), we know:

$\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_1| \ominus |t_1'| \lesssim D_1[\theta\,\theta_a] :^c \mathsf{Array}_\gamma[I]\ \tau[\theta\,\theta_a]$ $(\star)$

By Theorem 16.4 on (b) using (2) and c), we know:

$\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_2| \ominus |t_2'| \lesssim D_2[\theta\,\theta_a] :^c \mathrm{int}[I'][\theta\,\theta_a]$ $(\diamond)$

By Theorem 16.3 on (c) using (1)(2) and d), we know:

$\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_3| \ominus |t_3'| \lesssim d_3 :^c \square\tau[\theta_a]$ $(\dagger)$

Using the above statements and the premises, we can conclude using the rule **c-r-updateb**.

$$\frac{\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_1| \ominus |t_1'| \lesssim D_1[\theta\,\theta_a] :^c \mathsf{Array}_\gamma[I]\ \tau[\theta\,\theta_a]\ (\star) \qquad \Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_2| \ominus |t_2'| \lesssim D_2[\theta\,\theta_a] :^c \mathrm{int}[I'][\theta\,\theta_a]\ (\diamond) \qquad \Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |t_3| \ominus |t_3'| \lesssim d_3 :^c \square\tau[\theta_a]\ (\dagger) \qquad \Delta;\Phi_a \vDash I' \leq I \qquad \Sigma;\Delta \vdash P \quad wf}{\Sigma;\Delta;\Phi_a[\theta_a];\Gamma[\theta_a] \vdash |\,\mathsf{update}_\square\ t_1\ t_2\ t_3\,| \ominus |\,\mathsf{update}_\square\ t_1'\ t_2'\ t_3'\,| \lesssim 0 :^c \{P \star \gamma \mapsto \beta\}\ \exists\_.\overset{\mathrm{diff}(D[\theta_a])}{\mathsf{unit}_r}\ \{P \star \gamma \mapsto \beta \setminus \{I'\}\}}\ \textbf{c-r-updateb}$$

□

Theorem 17 (Completeness of the Algorithmic Typechecking in Arel)
We have the following.

1. Assume that $\Sigma;\Delta;\Phi_a;\Omega \vdash^U_L t :^c A$. Then, $\exists t'$ such that
   - 1.1. $\Sigma;\Delta;\cdot;\Phi_a;\Omega \vdash t' \downarrow A, L, U \Rightarrow \Phi$
   - 1.2. $\Delta;\Phi_a \vDash \Phi$
   - 1.3. $|t'| = t$

2. Assume that $\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \lesssim D :^c \tau$. Then, $\exists t_1', t_2'$ such that

   2.1. $\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1' \ominus t_2' \downarrow \tau, D \Rightarrow \Phi$

   2.2. $\Delta;\Phi_a \models \Phi$

   2.3. $|t_1'| = t_1$ and $|t_2'| = t_2$

Proof. Proof is by simultaneous induction on the ArelCore typing derivations.

Proof of Theorem 17.2:

Case $\dfrac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_2 \lesssim D :^c \tau \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Sigma;\Delta \vdash Q \quad wf \qquad \Sigma;\Delta \models P \subseteq Q}{\Sigma;\Delta;\Phi_a;\Gamma \vdash \mathsf{return}\ t_1 \ominus \mathsf{return}\ t_2 \lesssim 0 :^c \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\_.\tau\ \{Q\}}$ **c-r-return**

By IH on the first premise, we know that exists t_1', t_2' such that:

$\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1' \ominus t_2' \downarrow \tau, D \Rightarrow \Phi$ $(\star)$

$\Delta;\Phi_a \models \Phi$ $(\diamond)$

$|t_1'| = t_1$ and $|t_2'| = t_2$ $(\dagger)$

By using $(\star)$ and the premises, we can conclude using the algorithmic return rule.

$$\frac{\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1' \ominus t_2' \downarrow \tau, D \Rightarrow \Phi \qquad \Sigma;\Delta \vdash P \quad wf \qquad \Sigma;\Delta \vdash Q \quad wf \qquad \Sigma;\Delta \models P \subseteq Q}{\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash \mathsf{return}\ t_1' \ominus \mathsf{return}\ t_2' \downarrow\ \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\_.\tau\ \{Q\}, 0 \Rightarrow \Phi}\ \textbf{alg-r-ret-}\downarrow$$

$$\textbf{Case}\ \frac{\Sigma;\Delta;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \lesssim D_1 :^c \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\} \qquad \Sigma;\Delta,\vec{\gamma}_1 :: \vec{\mathbb{L}};\Phi_a;\Gamma, x:\tau \vdash t_2 \ominus t_2' \lesssim D_2 :^c \{P'\}\ \overset{\mathrm{diff}(D')}{\exists}\vec{\gamma}_2.\sigma\ \{Q\}}{\Sigma;\Delta;\Phi;\Gamma \vdash \mathsf{let}\ \{x\} = t_1\ \mathsf{in}\ t_2 \ominus \mathsf{let}\ \{x\} = t_1'\ \mathsf{in}\ t_2' \lesssim 0 :^c \{P\}\ \overset{\mathrm{diff}(D+D'+D_1+D_2)}{\exists}\vec{\gamma}_1\vec{\gamma}_2.\sigma\ \{Q\}}\ \textbf{c-r-bind}$$

By IH on the first premise, we know that exists $t_1^*, t_1'^*$ such that:

$\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1^* \ominus t_1'^* \downarrow \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}, D_1 \Rightarrow \Phi_1$ $(\star)$

$\Delta;\Phi_a \models \Phi_1$ $(\diamond)$

$|t_1^*| = t_1$ and $|t_1'^*| = t_1'$ $(\dagger)$

Choose $t_1^* = (t_1 :\{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}, D_1)$ and $t_1'^* = (t_1' :\{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}, D_1)$.

By the rule **alg-r-anno-**$\uparrow$, we have:

$$\frac{\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1 \ominus t_1' \downarrow\ \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}, D_1 \Rightarrow \Phi_1\ (\star) \qquad \Delta;\Phi_a \vdash \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}\ \mathtt{wf} \qquad \mathrm{FIV}(\{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\}, D_1) \in \Delta}{\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_1^* \ominus t_1'^* \uparrow \{P\}\ \overset{\mathrm{diff}(D)}{\exists}\vec{\gamma}_1.\tau\ \{P'\} \Rightarrow [\cdot], D_1, \Phi}\ \textbf{alg-r-anno-}\uparrow$$

By IH on the second premise, we know that exists $t_2^*, t_2'^*$ such that:

$\Sigma;\Delta;\cdot;\Phi_a;\Gamma \vdash t_2^* \ominus t_2'^* \downarrow \{P'\}\ \overset{\mathrm{diff}(D')}{\exists}\vec{\gamma}_2.\sigma\ \{Q\}, D_2 \Rightarrow \Phi_2$ $(\star\star)$

$\Delta;\Phi_a \models \Phi_2$ $(\diamond\diamond)$

$|t_2^*| = t_2$ and $|t_2'^*| = t_2'$ $(\dagger\dagger)$

By $(\star\star)$, we can show that for $D_2' \in \text{fresh}(\mathbb{R})$ where $\Phi_2' = \Phi_2 \wedge D_2 \doteq D_2'$

$$\Sigma; \Delta; D_2', \cdot; \Phi_a; \Gamma \vdash t_2 \ominus t_2' \downarrow \{P'\}\ \exists \vec{\gamma}_2 \overset{\text{diff}(D')}{:} \tau_2\ \{Q\}, D_2' \Rightarrow \Phi_2' \tag{4.2}$$

By the algorithmic rule **alg-r-let-↓**, we can conclude that:

$$\frac{\Sigma; \Delta; \cdot; \Phi_a; \Gamma \vdash t_1^* \ominus t_1'^* \uparrow \{P\}\ \exists \vec{\gamma}_1 \overset{\text{diff}(D)}{:} \tau_1\ \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1 \qquad D_2' \in \text{fresh}(\mathbb{R}) \qquad \Sigma; \Delta; D_2'; \Phi_a; \Gamma \vdash t_2 \ominus t_2' \downarrow \{P'\}\ \exists \vec{\gamma}_2 \overset{\text{diff}(D')}{:} \tau_2\ \{Q\}, D_2' \Rightarrow \Phi_2' \qquad \Phi = \Phi_2' \wedge D_1 + D_2' + D + D' \doteq D''}{\Sigma; \Delta; \cdot; \Phi_a; \Gamma \vdash \text{let } \{x\} = t_1^* \text{ in } t_2 \ominus \text{let } \{x\} = t_1'^* \text{ in } t_2' \downarrow \{P\}\ \exists \_ \overset{\text{diff}(D'')}{:} \tau_2\ \{Q\}, 0 \Rightarrow \Phi_1 \wedge (\exists D_2 :: \mathbb{R}.\Phi)}\ \textbf{alg-r-bind-}\downarrow$$

In this case, we know that there exists
$t = \text{let } \{x\} = (t_1 : \{P\}\ \exists \vec{\gamma}_1 \overset{\text{diff}(D)}{.} \tau\ \{P'\}, D_1) \text{ in } t_2$ and $t' = \text{let } \{x\} = (t_1' : \{P\}\ \exists \vec{\gamma}_1 \overset{\text{diff}(D)}{.} \tau\ \{P'\}, D_1) \text{ in } t_2'$.

□

5 Experimental Evaluation

Table 1: Summary of experimental results

Benchmark	LOC	#TYP	#ESF	TC (s)	TC-SMT (s)	TF-SMT (s)
mapi(1)	12	2	0	0.802	1.051	0.01
mapi(2)	19	3	1	1.247	0.994	0.02
boolOr	48	8	3	1.574	1.131	2.38
separate	36	8	0	1.351	2.148	0.01
loop	23	5	0	1.167	2.114	0.01
FFT	66	17	0	2.591	4.268	0.01
Search	62	10	3	3.753	4.430	6.56
NSS	94	12	3	4.158	4.413	10.03
shift	14	3	0	0.660	1.394	0.01
insert	22	6	0	1.001	3.019	0.01
iSort	134	12	3	2.897	6.181	10.70
merge(1)	29	8	0	2.203	2.232	0.01
merge(2)	64	11	2	3.231	0.349	0.02
sam	19	4	1	0.946	0.083	0.02
comp	20	3	0	1.138	0.112	0.01

Benchmark	LOC	# TYP	typechecking time	SMT time
`map`	13	2	0.494s	5.632s
`merge`	24	8	1.528s	11.937s
`NSS`	24	2	1.028s	5.632s
`boolOr`	13	2	0.499s	0.044s
`insert`	17	3	0.375s	0.039s
`iSort`	11	3	0.280s	0.034s
`shift`	12	2	0.489s	0.045s

Table 2: Unary experiments