

Relational Cost Analysis for Functional-Imperative Programs (Appendix)

Feb 2019

Contents

1	Syntax	3
2	Logical relation	18
3	Example revisited	58
3.1	InPlaceMap	58
3.2	MergeSort	63
3.3	Naive String Search	69
3.4	Boolean Or	74
3.5	Boolean Or, two implementations	76
3.6	Insertion sort	78
3.7	Cooley Tukey FFT algorithm	83
3.8	Square and multiply (SAM)	87
3.9	Constant-time comparison	88
4	Bidirectional type checking	89
5	Experimental Evaluation	96

List of Figures

1	Syntax of Values & Expressions	3
2	Operational Semantics: (1) standard, (2) forcing evaluation.	4
3	Binary to unary type erasure	5
4	sort environment well formedness	5
5	location environment well-formedness	5
6	Assertion well-formedness	5
7	Constraint well-formedness	6
8	Sorting rules	6
9	Well-formedness of unary types	7
10	Well-formedness of relational types	8
11	Heap well-formedness	8
12	Unary typing judgment, part (1)	9
13	Unary typing judgment, part (2)	10
14	Relational typing judgment, part (1)	11
15	Relational typing judgment, part (2)	12
16	Relational typing judgment, part (3)	13
17	Unary subtyping rules, part (1)	13
18	Unary subtyping rules, part (2)	14
19	Relational subtyping rules, part (1)	14
20	Relational subtyping rules, part (2)	15
21	Unary interpretation of types	16
22	Relational interpretation of types	17
23	Syntax of values and expressions in the bidirectional type checking version	89
24	Bidirectional algorithmic typing judgment explanation	90
25	Bidirectional algorithmic typing rules, part 1	91
26	Bidirectional algorithmic typing rules, part 2	92
27	Bidirectional algorithm-typing rules (alloc)	92
28	Bidirectional algorithm-typing rules (read)	93
29	Bidirectional algorithm-typing rules (update)	93
30	Bidirectional algorithm-typing rules (let)	94
31	Bidirectional algorithmic typing rules (return)	94
32	Bidirectional relational algorithmic subtyping rules	95
33	Bidirectional unary algorithmic subtyping rules	95

List of Theorems and Lemmas

1	Lemma (Monotonicity)	18
2	Lemma (Heap extension)	18
3	Lemma (Heap evaluation extension)	18
4	Lemma (Evaluation cost soundness)	18
5	Lemma (Well-formedness)	19
6	Lemma (Value Projection)	19
7	Lemma (Heap monotonicity)	19
8	Lemma (Value interpretation containment)	20
9	Lemma (Value evaluation)	21
10	Lemma (heap projection)	21
11	Lemma (heap subtyping)	21
12	Lemma (Subtyping soundness)	22
2.1	Theorem (Fundamental Theorem)	27
2.1.1	Corollary	27

1 Syntax

Unary types	$A ::= \text{int} \mid \text{int}[I] \mid \text{unit} \mid \overset{\text{exec}(L,U)}{\{P\} \exists \tilde{\gamma}. A \{Q\}} \mid \forall i :: S. A \mid A \longrightarrow A' \mid \text{Array}_\gamma[I] A \mid \text{list}[I] A \mid A_1 + A_2 \mid C \& A \mid C \supset A \mid \exists i :: S. A$
Relational types	$\tau ::= \text{int}_r \mid \text{unit}_r \mid \text{int}[I] \mid \overset{\text{diff}(D)}{\{P\} \exists \tilde{\gamma}. \tau \{Q\}} \mid \forall i :: S. \tau \mid \tau \longrightarrow \tau' \mid \text{Array}_\gamma[I] \tau \mid U(A_1, A_2) \mid \text{list}^\alpha[I] \tau \mid \square \tau \mid \tau_1 + \tau_2 \mid C \& \tau \mid C \supset \tau \mid \exists i :: S. \tau$
Index terms	$I, L, U, D, \alpha, \beta ::= i \mid b \mid n \mid r \mid I_1 + I_2 \mid I_1 * I_2 \mid I_1 - I_2 \mid \max(I_1, I_2) \mid \min(I_1, I_2) \mid \log_2(D) \mid \lfloor I \rfloor \mid \lceil I \rceil \mid \{I_i\}_{i \in K} \mid \beta \cup \beta \mid \beta \setminus \beta \mid \beta \cap \beta$
Sort	$S ::= \mathbb{R} \mid \mathbb{N} \mid \mathbb{B} \mid \mathbb{P} \mid \mathbb{L}$
Terms	$t ::= x \mid n \mid r \mid () \mid \lambda x. t \mid \text{fix } f(x). t \mid t_1 t_2 \mid \text{let } x = t_1 \text{ in } t_2 \mid \text{inl } t \mid \text{inr } t \mid \text{case } (t, x.t_1, y.t_2) \mid \Lambda. t \mid t [] \mid \text{pack } t \mid \text{unpack } t_1 \text{ as } x \text{ in } t_2 \mid \text{celim } t \mid \text{return } t \mid \text{let } \{x\} = t_1 \text{ in } t_2 \mid \text{alloc } t_1 t_2 \mid \text{read } t_1 t_2 \mid \text{updt } t_1 t_2 t_3$
Values	$v ::= n \mid l \mid r \mid () \mid \lambda x. t \mid \text{fix } f(x). t \mid \text{inl } v \mid \text{inr } v \mid \Lambda. t \mid \text{pack } v \mid \text{return } t \mid \text{alloc } t_1 t_2 \mid \text{updt } t_1 t_2 t_3 \mid \text{read } t_1 t_2 \mid \text{let } \{x\} = t_1 \text{ in } t_2$
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	$C ::= I_1 = I_2 \mid I_1 < I_2 \mid \neg C$
Constraint Env.	$\Phi ::= \top \mid C \wedge \Phi$
Assertions	$B, Q ::= \text{empty} \mid \gamma \rightarrow \beta \mid P \star Q$
Unary Heap	$H ::= [] \mid [l \rightarrow z] \mid H_1 \uplus H_2$
array	$z ::= [v_1, \dots, v_m]$
	$n \in \mathbb{N}, r \in \mathbb{R}, x \in \text{Var}, i \in i\text{Var}, \gamma \in i\text{Loc}, l \in d\text{Loc}$

Figure 1: Syntax of Values & Expressions

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

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

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

$$\frac{}{v \Downarrow^{0,0} v} \text{ E-V} \quad \frac{t \Downarrow^{c,k} v}{\text{inl } t \Downarrow^{c,k} \text{ inl } v} \text{ E-INL} \quad \frac{t \Downarrow^{c,k} v}{\text{inr } t \Downarrow^{c,k} \text{ inr } v} \text{ E-INR}$$

$$\frac{t \Downarrow^{c_1, k_1} \text{ inl } v \quad t_1[v/x] \Downarrow^{c_2, k_2} v_r}{\text{case } (t, x. t_1, y. t_2) \Downarrow^{c_1+c_2+c_{\text{case}}, k_1+k_2+1} v_r} \text{ E-CASEL} \quad \frac{t \Downarrow^{c_1, k_1} \text{ inr } v \quad t_2[v/y] \Downarrow^{c_2, k_2} v_r}{\text{case } (t, x. t_1, y. t_2) \Downarrow^{c_1+c_2+c_{\text{case}}, k_1+k_2+1} v_r} \text{ E-CASER}$$

$$\frac{t_1 \Downarrow^{c_1, k_1} \lambda x. t' \quad t_2 \Downarrow^{c_2, k_2} v \quad t'[v/x] \Downarrow^{c_3, k_3} v_1}{t_1 t_2 \Downarrow^{c_1+c_2+c_3+c_{\text{app}}, k_1+k_2+k_3+1} v_1} \text{ E-A}$$

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

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

$$\frac{t \Downarrow^{c,k} v}{\text{return } t; H \Downarrow_f^{c+c_{\text{ret}}, k+1} v; H} \text{ F-T}$$

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

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

$$\frac{t_1 \Downarrow^{c_1, k_1} l \quad t_2 \Downarrow^{c_2, k_2} n_2 \quad H(l)[n] = v}{\text{read } t_1 t_2; H \Downarrow_f^{c_1+c_2+c_{\text{read}}, k_1+k_2+1} v; H} \text{ F-R}$$

$$\frac{t_1 \Downarrow^{c_1, k_1} n \quad t_2 \Downarrow^{c_2, k_2} v \quad z = \overbrace{[v, \dots, v]}^n \quad l \text{ fresh}}{\text{alloc } t_1 t_2; H \Downarrow_f^{c_1+c_2+c_{\text{alloc}}, k_1+k_2+1} l; H, l \rightarrow z} \text{ F-L}$$

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

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

Figure 3: Binary to unary type erasure

$\boxed{\vdash \Delta}$

$$\frac{}{\vdash \emptyset} \text{SENV-EMPTY} \qquad \frac{\vdash \Delta \quad i \in \text{iVar} \quad S \in \text{Sort}}{\vdash \Delta, i :: S} \text{SENV-IV}$$

Figure 4: sort environment well formedness

$\boxed{\vdash \Sigma}$

$$\frac{}{\vdash \emptyset} \text{LENV-EMPTY} \qquad \frac{\vdash \Sigma \quad \gamma \in \text{iLoc}}{\vdash \Sigma, \gamma :: \mathbb{L}} \text{LENV-LV}$$

Figure 5: location environment well-formedness

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

$$\frac{}{\Sigma; \Delta \vdash \text{empty} \text{ wf}} \text{WF-AS-EMPTY} \qquad \frac{\Sigma; \Delta \vdash \gamma :: \mathbb{L} \quad \Sigma; \Delta \vdash \beta :: \mathbb{P}}{\Sigma; \Delta \vdash \gamma \rightarrow \beta \text{ wf}} \text{WF-AS-REFERTO} \\
\frac{\Sigma; \Delta \vdash P \text{ wf} \quad \Sigma; \Delta \vdash Q \text{ wf}}{\Sigma; \Delta \vdash P * Q \text{ wf}} \text{WF-AS-STAR}$$

Figure 6: Assertion well-formedness

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

$$\frac{\Delta \vdash I_1 :: S \quad \Delta \vdash I_2 :: S \quad S \in \{\mathbb{N}, \mathbb{R}\}}{\Delta \vdash I_1 = I_2 \quad wf} \text{WF-CS-EQ} \quad \frac{\Delta \vdash I_1 :: S \quad \Delta \vdash I_2 :: S \quad S \in \{\mathbb{N}, \mathbb{R}\}}{\Delta \vdash I_1 < I_2 \quad wf} \text{WF-CS-LT}$$

$$\frac{\Delta \vdash C \quad wf}{\Delta \vdash \neg C \quad wf} \text{WF-CS-NEG}$$

Figure 7: Constraint well-formedness

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

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

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

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

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

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

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

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

Figure 8: Sorting rules

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

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

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

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

$$\frac{\Sigma; \Delta; \Phi \vdash A \quad wf \quad \Sigma; \Delta \vdash \gamma :: \mathbb{L} \quad \Sigma; \Delta \vdash U :: \mathbb{R} \quad \Sigma; \Delta \vdash P \quad wf \quad \Sigma, \tilde{\gamma} : \tilde{\mathbb{L}}; \Delta \vdash Q \quad wf}{\Sigma; \Delta; \Phi \vdash \{P\} \exists \tilde{\gamma} : A \{Q\} \quad wf} \text{WF-A-MONAD}$$

$$\frac{\Sigma; \Delta; \Phi \vdash A \quad wf \quad \Sigma; \Delta; \Phi \vdash A' \quad wf \quad \Sigma; \Delta \vdash L :: \mathbb{R} \quad \Sigma; \Delta \vdash U :: \mathbb{R}}{\Sigma; \Delta; \Phi \vdash A \longrightarrow A' \quad wf} \text{WF-A-ARROW}$$

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

$$\frac{\Sigma; i :: S, \Delta; \Phi \vdash A \quad wf \quad \Sigma; i :: S, \Delta \vdash L :: \mathbb{R} \quad \Sigma; i :: S, \Delta \vdash U :: \mathbb{R}}{\Sigma; \Delta; \Phi \vdash \forall i :: S. A \quad wf} \text{WF-A-FORALL}$$

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

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

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

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

Figure 9: Well-formedness of unary types

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

$$\begin{array}{c}
\frac{}{\Sigma; \Delta; \Phi \vdash \emptyset \text{ wf}} \text{WF-R-GAMMA-EMPTYSET} \qquad \frac{\Sigma; \Delta; \Phi \vdash \Gamma \text{ wf} \quad \Sigma; \Delta; \Phi \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash \Gamma, x : \tau \text{ wf}} \text{WF-R-GAMMA} \\
\frac{}{\Sigma; \Delta; \Phi \vdash \text{unit}_r \text{ wf}} \text{WF-R-UNIT} \qquad \frac{}{\Sigma; \Delta; \Phi \vdash \text{int}_r \text{ wf}} \text{WF-R-INT} \\
\frac{\Sigma; \Delta; \Phi \vdash \tau_1 \text{ wf} \quad \Sigma; \Delta; \Phi \vdash \tau_2 \text{ wf}}{\Sigma; \Delta; \Phi \vdash \tau_1 + \tau_2 \text{ wf}} \text{WF-R-SUM} \\
\frac{\Sigma; \Delta \vdash P \text{ wf} \quad \Sigma; \Delta; \Phi \vdash \tau \text{ wf} \quad \Sigma; \Delta, \vec{\gamma} : \vec{\mathbb{L}} \vdash Q \text{ wf} \quad \Sigma; \Delta \vdash D :: \mathbb{R}}{\Sigma; \Delta; \Phi \vdash \{P\} \exists \vec{\gamma} : \tau \{Q\} \text{ wf}} \text{WF-R-MONAD} \\
\frac{\Sigma; \Delta; \Phi \vdash \tau \text{ wf} \quad \Sigma; \Delta; \Phi \vdash \tau' \text{ wf} \quad \Sigma; \Delta \vdash D :: \mathbb{R}}{\Sigma; \Delta; \Phi \vdash \tau \longrightarrow \tau' \text{ wf}} \text{WF-R-ARROW} \\
\frac{\Sigma; i :: S, \Delta; \Phi \vdash \tau \text{ wf} \quad \Sigma; i :: S, \Delta \vdash D :: \mathbb{R}}{\Sigma; \Delta; \Phi \vdash \forall i :: S. \tau \text{ wf}} \text{WF-R-FORALL} \qquad \frac{\Sigma; i :: S, \Delta; \Phi \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash \exists i :: S. \tau \text{ wf}} \text{WF-R-EXIST} \\
\frac{\Sigma; \Delta \vdash \gamma :: \mathbb{L} \quad \Sigma; \Delta \vdash I :: \mathbb{N} \quad \Sigma; \Delta; \Phi \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash \text{Array}_\gamma[I] \tau \text{ wf}} \text{WF-R-ARRAY} \\
\frac{\Sigma; \Delta; \Phi \vdash A_1 \text{ wf} \quad \Sigma; \Delta; \Phi \vdash A_2 \text{ wf}}{\Sigma; \Delta; \Phi \vdash U(A_1, A_2) \text{ wf}} \text{WF-R-UA} \\
\frac{\Sigma; \Delta \vdash \alpha :: \mathbb{N} \quad \Sigma; \Delta \vdash I :: \mathbb{N} \quad \Sigma; \Delta; \Phi \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash \text{list}^\alpha[I] \tau \text{ wf}} \text{WF-R-LIST} \\
\frac{\Delta \vdash C \text{ wf} \quad \Sigma; \Delta; \Phi \wedge C \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash C \& \tau \text{ wf}} \text{WF-R-C-AND} \\
\frac{\Delta \vdash C \text{ wf} \quad \Sigma; \Delta; \Phi \wedge C \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash C \supset \tau \text{ wf}} \text{WF-R-C-IMPLY} \qquad \frac{\Sigma; \Delta; \Phi \vdash \tau \text{ wf}}{\Sigma; \Delta; \Phi \vdash \square \tau \text{ wf}} \text{WF-R-SQUARE}
\end{array}$$

Figure 10: Well-formedness of relational types

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

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

Figure 11: Heap well-formedness

$$\boxed{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A}$$

$$\begin{array}{c}
\frac{}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 n : \text{int}} \text{U-INT} \qquad \frac{}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 n : \text{int}[n]} \text{U-I} \qquad \frac{\Sigma; \Delta; \Phi; \Omega, x : A \vdash_L^U t : B}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \lambda x. t : A \longrightarrow B} \text{U-ABS} \\
\\
\frac{\Omega(x) = A}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 x : A} \text{U-V} \qquad \frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : A \longrightarrow B \quad \text{exec}(L, U) \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : A}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2+L+L_{app}}^{U_1+U_2+U+U_{app}} t_1 t_2 : B} \text{U-A} \\
\\
\frac{}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 () : \text{unit}} \text{U-UNIT} \qquad \frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \text{return } t : \{P\} \exists \vec{\gamma}. A \{P\}} \text{U-T} \\
\\
\frac{P = P_1 \star P_2 \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \{P_1\} \exists \vec{\gamma}_1 : A \{Q_1 \star Q_2\} \quad \text{exec}(L, U) \quad \Sigma; \Delta, \vec{\gamma}_1; \Phi; \Omega, x : A \vdash_{L_2}^{U_2} t_2 : \{Q_1 \star P_2\} \exists \vec{\gamma}_2 : B \{Q\} \quad \text{exec}(L', U')}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2+L_l}^{U_1+U_2+U_l} \text{let } \{x\} = t_1 \text{ in } t_2 : \{P\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : B \{Q \star Q_2\}} \text{U-E} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : A \quad \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}\}} \text{U-L} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{Array}_\gamma[I] A \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_3}^{U_3} t_3 : A \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta; \Phi \models I' \in \beta \quad \Sigma; \Delta \vdash P \quad wf}{\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\}} \text{U-U} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{Array}_\gamma[I] A \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : \text{int}[I'] \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \text{read } t_1 t_2 : \{P\} \exists _ : A \{P\}} \text{U-R}
\end{array}$$

Figure 12: Unary typing judgment, part (1)

$$\begin{array}{c}
\frac{\Sigma; \Delta; \Phi; x : A, f : A \xrightarrow{\text{exec}(L,U)} B, \Omega \vdash_L^U t : B}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \text{fix } f(x). t : A \xrightarrow{\text{exec}(L,U)} B} \text{U-F} \quad \frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A_1 \quad \Delta; \Sigma; \Phi \vdash A_2 \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U \text{inl } t : A_1 + A_2} \text{U-INL} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A_2 \quad \Delta; \Sigma; \Phi \vdash A_1 \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U \text{inr } t : A_1 + A_2} \text{U-INR} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t : A_1 + A_2 \quad \Sigma; \Delta; \Phi; x : A_1, \Omega \vdash_{L_2}^{U_2} t_1 : A \quad \Sigma; \Delta; \Phi; y : A_2, \Omega \vdash_{L_2}^{U_2} t_2 : A}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2+L_c}^{U_1+U_2+U_c} \text{case } (t, x.t_1, y.t_2) : A} \text{U-CASE} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A \quad \Sigma; \Delta \models U \leq U' \quad \Sigma; \Delta \models L' \leq L}{\Sigma; \Delta; \Phi; \Omega \vdash_{L'}^{U'} t : A} \text{COST-TRANS} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A \quad \Sigma; \Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta \models U \leq U' \quad \Sigma; \Delta \models L' \leq L}{\Sigma; \Delta; \Phi; \Omega \vdash_{L'}^{U'} t : A'} \text{U-X} \\
\\
\frac{\Sigma; i :: S, \Delta; \Phi; \Omega \vdash_L^U t : A \quad i \notin FIV(\Phi; \Omega)}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \Lambda. t : \forall i :: S. A} \text{U-LAM} \quad \frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t : \forall i :: S. A \quad \Sigma; \Delta \vdash I :: S}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2}^{U_1+U_2} t [] : A\{I/i\}} \text{U-IAPP} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A\{I/i\} \quad \Sigma; \Delta \vdash I :: S}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U \text{pack } t : \exists i :: S. A} \text{U-PACK} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \exists i :: S. A \quad \Sigma; i :: S, \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : A_2 \quad i \notin FV(\Phi; \Omega; A_2, L_2, U_2)}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2}^{U_1+U_2} \text{unpack } t_1 \text{ as } x \text{ in } t_2 : A_2} \text{U-UNPACK} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : C \supset A \quad \Sigma; \Delta; \Phi \models C}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U \text{celim } t : A} \text{U-CELIM} \quad \frac{\Sigma; \Delta; \Phi \wedge C; \Omega \vdash_L^U t : A}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : C \supset A} \text{U-CIMPL} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : C \& A_1 \quad \Sigma; \Delta; \Phi \wedge C; x : A_1, \Omega \vdash_{L'}^{U'} t : A_2}{\Sigma; \Delta; \Phi; \Gamma \vdash_{L+L'}^{U+U'} \text{clet } t \text{ as } x \text{ in } t' : A_2} \text{U-ANDE} \\
\\
\frac{\Sigma; \Delta; \Phi \wedge C; \Omega \vdash_L^U t : A \quad \Sigma; \Delta; \Phi \wedge \neg C; \Omega \vdash_L^U t : A}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A} \text{U-SPLIT} \\
\\
\frac{\Sigma; \Delta; \Phi \models \perp \quad \vdash \Sigma; \Delta; \Phi \models \Omega \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : C \& A} \text{U-CONTRA} \quad \frac{\Sigma; \Delta; \Phi \wedge C; \Omega \vdash_L^U t : A \quad \Sigma; \Delta; \Phi \models C}{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : C \& A} \text{U-ANDI}
\end{array}$$

Figure 13: Unary typing judgment, part (2)

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

$$\begin{array}{c}
\frac{}{\Sigma; \Delta; \Phi; \Gamma \vdash n \ominus n \lesssim 0 : \text{int}_r} \text{R-INT} \quad \frac{}{\Sigma; \Delta; \Phi; \Gamma \vdash n \ominus n \lesssim 0 : \text{int}_r[n]} \text{R-I} \quad \frac{}{\Sigma; \Delta; \Phi; \Gamma \vdash 0 \ominus 0 \lesssim 0 : \text{unit}_r} \text{R-UNIT} \\
\\
\frac{\Gamma(x) = \tau}{\Sigma; \Delta; \Phi; \Gamma \vdash x \ominus x \lesssim 0 : \tau} \text{R-V} \quad \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{\text{diff}(D)} \sigma} \text{R-ABS} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \tau \xrightarrow{\text{diff}(D)} \sigma \quad \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} \text{R-APP} \\
\\
\frac{\Sigma; \Delta; \Phi; x : \tau, f : \tau \xrightarrow{\text{diff}(D)} \sigma, \Gamma \vdash t_1 \ominus t_2 \lesssim D : \sigma}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{Fix } f(x). t_1 \ominus \text{Fix } f(x). t_2 \lesssim D : \tau \xrightarrow{\text{diff}(D)} \sigma} \text{R-FIX} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t' \lesssim D : \tau_1 \quad \Sigma; \Delta; \Phi \vdash \tau_2 \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{inl } t \ominus \text{inl } t' \lesssim D : \tau_1 + \tau_2} \text{R-INL} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t' \lesssim D : \tau_2 \quad \Sigma; \Delta; \Phi \vdash \tau_1 \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{inr } t \ominus \text{inr } t' \lesssim D : \tau_1 + \tau_2} \text{R-INR} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t' \lesssim D_1 : \tau_1 + \tau_2 \quad \Sigma; \Delta; \Phi; \Gamma, x : \tau_1 \vdash t_1 \ominus t'_1 \lesssim D_2 : \tau \quad \Sigma; \Delta; \Phi; \Gamma, y : \tau_2 \vdash t_2 \ominus t'_2 \lesssim D_2 : \tau}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{case } (t, x. t_1, y. t_2) \ominus \text{case } (t', x. t'_1, y. t'_2) \lesssim D_1 + D_2 : \tau} \text{R-CASE} \\
\\
\frac{\Sigma; \Delta; \Phi; |\Gamma|_1 \vdash_{L_1}^{U_1} t_1 : A_1 \quad \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)} \text{R-S} \\
\\
\frac{\Sigma; \Delta; \Phi; \Omega \vdash t \ominus t \lesssim D : \tau \quad \Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; \Delta; \Phi \models D \leq D'}{\Sigma; \Delta; \Phi; \Gamma \vdash t \ominus t \lesssim D' : \tau'} \text{R-EXEC} \\
\\
\frac{\Sigma; \Delta; \Phi \wedge C; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \quad \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} \text{R-P} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \quad \forall x \in \text{dom}(\Gamma). \Sigma; \Delta; \Phi \models \Gamma(x) \sqsubseteq \square \Gamma(x)}{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim 0 : \tau} \text{NC} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \{I/i\} \quad \Sigma; \Delta \vdash I :: S}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{pack } t_1 \ominus \text{pack } t_2 \lesssim D : \exists i :: S. \tau} \text{R-PACK} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \exists i :: S. \tau_1 \quad \Sigma; i :: S, \Delta; \Phi; x : \tau_1, \Gamma \vdash t_2 \ominus t'_2 \lesssim D_1 + D_2 : \tau_2 \quad i \notin FV(\Phi; \Gamma; \tau_2)}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{unpack } t_1 \text{ as } x \text{ in } t_2 \ominus \text{unpack } t'_1 \text{ as } x \text{ in } t'_2 \lesssim D_1 + D_2 : \tau_2} \text{R-UNPACK}
\end{array}$$

Figure 14: Relational typing judgment, part (1)

$$\begin{array}{c}
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{return } t_1 \ominus \text{return } t_2 \lesssim 0 : \{P\} \exists _ . \tau \{P\}} \text{R-T} \\
\\
\frac{P = P_1 \star P_2 \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \{P_1\} \exists \vec{\gamma}_1 . \tau \{Q_1 \star Q_2\} \quad \Sigma; \Delta; \vec{\gamma}_1 : \vec{L}; \Phi; \Gamma, x : \tau \vdash t_2 \ominus t'_2 \lesssim D_2 : \{Q_1 \star P_2\} \exists \vec{\gamma}_2 . \sigma \{Q\}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{let } \{x\} = t_1 \text{ in } t_2 \ominus \text{let } \{x\} = t'_1 \text{ in } t'_2 \lesssim D_1 + D_2 : \{P\} \exists \vec{\gamma}_1 \vec{\gamma}_2 : \sigma \{Q \star Q_2\}} \text{R-LET} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \tau \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{alloc } t_1 t_2 \ominus \text{alloc } t'_1 t'_2 \lesssim 0 : \{P\} \exists \gamma . \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\}} \text{R-ALLOC} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \square \tau \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{alloc } t_1 t_2 \ominus \text{alloc } t'_1 t'_2 \lesssim 0 : \{P\} \exists \gamma . \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \emptyset\}} \text{R-ALLOC-BOX} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Delta \models I' \leq I \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 t_2 \ominus \text{read } t'_1 t'_2 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \tau \{P \star \gamma \rightarrow \beta\}} \text{R-R} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta; \Phi \models I' \not\leq \beta \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 t_2 \ominus \text{read } t'_1 t'_2 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \square \tau \{P \star \gamma \rightarrow \beta\}} \text{R-RB} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_3 \ominus t'_3 \lesssim D_3 : \tau \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{updt } t_1 t_2 t_3 \ominus \text{updt } t'_1 t'_2 t'_3 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \text{unit}_r \{P \star \gamma \rightarrow \beta \cup \{I'\}\}} \text{R-U} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_3 \ominus t'_3 \lesssim D_3 : \square \tau \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{updt } t_1 t_2 t_3 \ominus \text{updt } t'_1 t'_2 t'_3 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \text{unit}_r \{P \star \gamma \rightarrow \beta \setminus \{I'\}\}} \text{R-UB}
\end{array}$$

Figure 15: Relational typing judgment, part (2)

$$\begin{array}{c}
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \quad i \notin FV(\Phi; \Gamma)}{\Sigma; \Delta; \Phi; \Gamma \vdash \Lambda t_1 \ominus \Lambda t_2 \lesssim 0 : \forall i :: S. \tau} \text{diff}(D) \quad \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \forall i :: S. \tau \quad \Sigma; \Delta \vdash I :: S}{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 [] \ominus t_2 [] \lesssim D + D' [I/i] : \tau \{I/i\}} \text{diff}(D') \quad \text{R-ILAM} \quad \text{R-IAPP} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : C \supset \tau \quad \Sigma; \Delta; \Phi \models C}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{celim } t_1 \ominus \text{celim } t_2 \lesssim 0 : \tau} \text{R-CELIM} \quad \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} \text{R-CIMPL} \\
\\
\frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : C \& \tau_1 \quad \Sigma; \Delta; \Phi \wedge C; x : \tau_1, \Gamma \vdash t'_1 \ominus t'_2 \lesssim D' : \tau_2}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{clet } t_1 \text{ as } x \text{ in } t'_1 \ominus \text{clet } t_2 \text{ as } x \text{ in } t'_2 \lesssim D + D' : \tau_2} \text{R-ANDE} \\
\\
\frac{\Sigma; \Delta; \Phi \wedge C; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau \quad \Sigma; \Delta; \Phi \models C}{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : C \& \tau} \text{R-ANDI} \\
\\
\frac{\Sigma; \Delta; \Phi; x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(D)} \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 D : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2} \text{R-FIX-EXT}
\end{array}$$

Figure 16: Relational typing judgment, part (3)

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

$$\begin{array}{c}
\frac{\Sigma; \Delta; \Phi \models A'_1 \sqsubseteq A_1 \quad \Sigma; \Delta; \Phi \models A_2 \sqsubseteq A'_2 \quad \Sigma; \Delta; \Phi \models L' \leq L \quad \Sigma; \Delta; \Phi \models U \leq U'}{\Sigma; \Delta; \Phi \models A_1 \xrightarrow{\text{exec}(L,U)} A_2 \sqsubseteq A'_1 \xrightarrow{\text{exec}(L',U')} A'_2} \text{S-A-EXEC} \\
\\
\frac{\Sigma; i :: S, \Delta; \Phi \models A' \sqsubseteq A \quad \Sigma; i :: S, \Delta; \Phi \models L' \leq L \quad \Sigma; \Delta; \Phi \models U \leq U' \quad i \notin FV(\Phi)}{\Sigma; \Delta; \Phi \models \forall i :: S. A \sqsubseteq \forall i :: S. A'} \text{S-A-FORALL} \\
\\
\frac{\Sigma; \Delta; \Phi \models U \leq U' \quad \Sigma; \Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta; \Phi \models L' \leq L \quad \Sigma, \tilde{\gamma}_1; \Delta; \Phi \models Q' \sqsubseteq Q \quad \Sigma; \Delta; \Phi \models \tilde{\gamma}_1 \sqsubseteq \tilde{\gamma}_2}{\Sigma; \Delta; \Phi \models \{P\} \exists \tilde{\gamma}_1 : A \{Q\} \sqsubseteq \{P'\} \exists \tilde{\gamma}_2 : A' \{Q'\}} \text{S-A-MONAD} \\
\\
\frac{\Sigma; \Delta; \Phi \models A_1 \sqsubseteq A'_1 \quad \Sigma; \Delta; \Phi \models A_2 \sqsubseteq A'_2}{\Sigma; \Delta; \Phi \models A_1 + A_2 \sqsubseteq A'_1 + A'_2} \text{S-UM} \quad \frac{\Sigma; \Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta; \Phi \models I = I'}{\Sigma; \Delta; \Phi \models \text{Array}_\gamma [I] A \sqsubseteq \text{Array}_\gamma [I'] A'} \text{S-A-ARRAY} \\
\\
\frac{\Sigma; \Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta; \Phi \models I = I'}{\Sigma; \Delta; \Phi \models \text{list}[I] A \sqsubseteq \text{list}[I'] A'} \text{S-A-LIST} \quad \frac{\Sigma; i :: S, \Delta; \Phi \models A \sqsubseteq A' \quad i \notin FV(\Phi)}{\Sigma; \Delta; \Phi \models \exists i :: S. A \sqsubseteq \exists i :: S. A'} \text{S-A-EXIST}
\end{array}$$

Figure 17: Unary subtyping rules, part (1)

$$\begin{array}{c}
\frac{\Sigma; f\Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta; \Phi \wedge C \models C'}{\Sigma; \Delta; \Phi \models C \& A \sqsubseteq C \& A'} \text{S-A-CAND} \quad \frac{\Sigma; \Delta; \Phi \models A \sqsubseteq A' \quad \Sigma; \Delta; \Phi \wedge C' \models C}{\Sigma; \Delta; \Phi \models C \supset A \sqsubseteq C \supset A'} \text{S-A-CIMPL} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models A \sqsubseteq A} \text{S-A-REFL} \quad \frac{\Sigma; \Delta; \Phi \models A_1 \sqsubseteq A_2 \quad \Sigma; \Delta; \Phi \models A_2 \sqsubseteq A_3}{\Sigma; \Delta; \Phi \models A_1 \sqsubseteq A_3} \text{S-A-TRAN}
\end{array}$$

Figure 18: Unary subtyping rules, part (2)

$\Sigma; \Delta; \Phi \models \tau_1 \sqsubseteq \tau_2$

$$\begin{array}{c}
\frac{}{\Sigma; \Delta; \Phi \models \text{int}_r \sqsubseteq \square U(\text{int}, \text{int})} \text{S-R-INT-BOX} \quad \frac{}{\Sigma; \Delta; \Phi \models \text{int}_r [I] \sqsubseteq \square U(\text{int}, \text{int})} \text{S-R-INT-I-BOX} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models \square U(\text{int}, \text{int}) \sqsubseteq \square \text{int}_r} \text{S-R-UINT-BOX} \quad \frac{}{\Sigma; \Delta; \Phi \models \text{unit} \sqsubseteq \square \text{unit}} \text{S-R-UNIT-BOX} \\
\\
\frac{\Sigma; \Delta; \Phi \models \tau'_1 \sqsubseteq \tau_1 \quad \Sigma; \Delta; \Phi \models \tau_2 \sqsubseteq \tau'_2 \quad \Sigma; \Delta; \Phi \models c \leq c'}{\Sigma; \Delta; \Phi \models \tau_1 \xrightarrow{\text{diff}(c)} \tau_2 \sqsubseteq \tau'_1 \xrightarrow{\text{diff}(c')} \tau'_2} \text{S-R-DIFF} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models \square(\tau_1 \xrightarrow{\text{diff}(c)} \tau_2) \sqsubseteq \square \tau_1 \xrightarrow{\text{diff}(0)} \square \tau_2} \text{S-R-BOX-DIFF} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models U(A_1 \xrightarrow{\text{exec}(L,U)} A_2, A'_1 \xrightarrow{\text{exec}(L',U')} A'_2) \sqsubseteq U(A_1, A'_1) \xrightarrow{\text{diff}(U-L')} U(A_2, A'_2)} \text{S-R-EXECDIFF} \\
\\
\frac{\Sigma; i :: S, \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; i :: S, \Delta; \Phi \models c \leq c' \quad i \notin FV(\Phi)}{\Sigma; \Delta; \Phi \models \forall i :: S. \tau \sqsubseteq \forall i :: S. \tau'_1} \text{S-R-FORALL-DIFF} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models \square(\forall i :: S. \tau) \sqsubseteq \forall i :: S. \square \tau} \text{S-R-FORALL-BOX} \quad \frac{}{\Sigma; \Delta; \Phi \models U(\text{unit}, \text{unit}) \sqsubseteq \text{unit}} \text{S-R-UNIT} \\
\\
\frac{}{\Sigma; \Delta; \Phi \models U(\forall i :: S. A, \forall i :: S. A') \sqsubseteq \forall i :: S. U(A, A')} \text{S-R-FORALL-U} \\
\\
\frac{\Sigma; \Delta; \Phi \models \tau_1 \sqsubseteq \tau'_1 \quad \Sigma; \Delta; \Phi \models \tau_2 \sqsubseteq \tau'_2}{\Sigma; \Delta; \Phi \models \tau_1 + \tau_2 \sqsubseteq \tau'_1 + \tau'_2} \text{S-R-SUM} \quad \frac{}{\Sigma; \Delta; \Phi \models \square \tau_1 + \square \tau_2 \sqsubseteq \square(\tau_1 + \tau_2)} \text{S-R-SUM-BOX} \\
\\
\frac{\Sigma; i :: S, \Delta; \Phi \models \tau \sqsubseteq \tau' \quad i \notin FV(\Phi)}{\Sigma; \Delta; \Phi \models \exists i :: S. \tau \sqsubseteq \exists i :: S. \tau'} \text{S-R-EXIST} \quad \frac{}{\Sigma; \Delta; \Phi \models \exists i :: S. \square \tau \sqsubseteq \square(\exists i :: S. \tau)} \text{S-R-EXIST-BOX}
\end{array}$$

Figure 19: Relational subtyping rules, part (1)

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

$$\begin{array}{c}
\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; \Delta; \Phi \models I = I' \quad \Sigma; \Delta; \Phi \models \gamma = \gamma'}{\Sigma; \Delta; \Phi \models \text{Array}_{\gamma}[I] \tau \sqsubseteq \text{Array}_{\gamma'}[I'] \tau'} \text{S-R-ARRAY} \\
\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; \Delta; \Phi \models I = I' \quad \Sigma; \Delta; \Phi \models \alpha \leq \alpha'}{\Sigma; \Delta; \Phi \models \text{list}^\alpha[I] A \sqsubseteq \text{list}^{\alpha'}[I'] A'} \text{S-R-LIST} \\
\frac{\Sigma; \Delta; \Phi \models A_1 \sqsubseteq A'_1 \quad \Sigma; \Delta; \Phi \models A_2 \sqsubseteq A'_2}{\Sigma; \Delta; \Phi \models U(A_1, A_2) \sqsubseteq U(A'_1, A'_2)} \text{S-R-UA} \\
\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; \Delta; \Phi \wedge C \models C'}{\Sigma; \Delta; \Phi \models C \& \tau \sqsubseteq C \& \tau'} \text{S-R-C-AND} \quad \frac{}{\Sigma; \Delta; \Phi \models C \& \square \tau \sqsubseteq \square(C \& \tau)} \text{S-R-C-AND-BOX} \\
\frac{}{\Sigma; \Delta; \Phi \models U(C \supset A, C \supset A') \sqsubseteq C \supset U(A, A')} \text{S-R-CIMPL-U} \\
\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Sigma; \Delta; \Phi \wedge C' \models C}{\Sigma; \Delta; \Phi \models C \supset \tau \sqsubseteq C \supset \tau'} \text{S-R-C-IMPL} \\
\frac{}{\Sigma; \Delta; \Phi \models \square(C \supset \tau) \sqsubseteq C \supset \square \tau} \text{S-R-C-IMPL-BOX} \quad \frac{}{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau} \text{S-R-REFL} \\
\frac{}{\Sigma; \Delta; \Phi \models \tau \sqsubseteq U(|\tau|_1, |\tau|_2)} \text{S-R-W} \quad \frac{}{\Sigma; \Delta; \Phi \models \square \tau \sqsubseteq \tau} \text{S-R-T} \\
\frac{\Sigma; \Delta; \Phi \models \tau_1 \sqsubseteq \tau_2 \quad \Sigma; \Delta; \Phi \models \tau_2 \sqsubseteq \tau_3}{\Sigma; \Delta; \Phi \models \tau_1 \sqsubseteq \tau_3} \text{S-R-TRAN} \\
\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Delta; \Phi \models c \sqsubseteq c' \quad \Delta; \Phi \models P' \sqsubseteq P \quad \Delta; \Phi \models \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2 \quad \Sigma, \vec{\gamma}_1 : \vec{\Gamma}; \Delta; \Phi \models Q \sqsubseteq Q'}{\Sigma; \Delta; \Phi \models \{P\} \exists \vec{\gamma}_1 : \tau \{Q\} \sqsubseteq \{P'\} \exists \vec{\gamma}_2 : \tau' \{Q'\}} \text{S-RM} \\
\frac{}{\Sigma; \Delta; \Phi \models U(\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}_2 : A_1 \{Q_2\}) \sqsubseteq \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\}} \text{S-RUM} \\
P \sqsubseteq P' \Leftrightarrow P = \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \beta_2, \dots, \gamma_n \rightarrow \beta_n\} \\
P' = \{\gamma_1 \rightarrow \beta'_1, \gamma_2 \rightarrow \beta'_2, \dots, \gamma_n \rightarrow \beta'_n\} \\
\forall i \in \{1, \dots, n\}. \beta_i \sqsubseteq \beta'_i
\end{array}$$

Figure 20: Relational subtyping rules, part (2)

$\llbracket \text{int} \rrbracket_{g,k}$	$= \{n \mid n \in \mathbb{N}\}$
$\llbracket \text{unit} \rrbracket_{g,k}$	$= \{\emptyset\}$
$\llbracket \text{bool} \rrbracket_{g,k}$	$= \{b \mid b \in \mathbb{B}\}$
$\llbracket \text{int}[I] \rrbracket_{g,k}$	$= \{n \mid I = n\}$
$\llbracket \text{Array}_\gamma[I] A \rrbracket_{g,k}$	$= \{I \mid I = n \wedge g(\gamma) = (l, n, A)\}$
$\llbracket A \xrightarrow{\text{exec}(L,U)} A' \rrbracket_{g,k}$	$= \{\text{fix } f(x).t \mid \forall k' < k, \forall g' \supseteq g, \forall v.v \in \llbracket A \rrbracket_{g',k'} \Rightarrow$ $t[v/x][\text{fix } f(x).t/f] \in \llbracket A' \rrbracket_{g',k'}^{E,(L,U)}\}$ $\cup \{\lambda x.t \mid \forall k' < k, \forall g' \supseteq g, \forall v.v \in \llbracket A \rrbracket_{g',k'} \Rightarrow t[v/x] \in \llbracket A' \rrbracket_{g',k'}^{E,(L,U)}\}$
$\llbracket \{P\} \exists \vec{\gamma}. A \{Q\} \rrbracket_{g,k}$	$= \left\{ v \mid \forall g_1 \supseteq g, \forall k_1 \leq k, \forall k_2 < k_1, c, \forall H. (H \models_{g_1, k_1} P \wedge v; H \Downarrow_f^{c, k_2} Q) \right.$ $\quad \Rightarrow \exists g_2 \supseteq g_1, H_1, v_1, \vec{\gamma}. (v; H \Downarrow_f^{c, k_2} v_1; H_1) \wedge L \leq c \leq U \wedge H_1 \models_{g_2, k_1 - k_2} Q$ $\quad \wedge v_1 \in \llbracket A \rrbracket_{g_2, k_1 - k_2} \wedge (\exists n. P = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\}$ $\quad \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j] \neq H_1[l_i][j] \Rightarrow j \in T_i) \left. \right\}$
$\llbracket \forall i :: S. A \rrbracket_{g,k}$	$= \{\Lambda.t \mid \forall I. \vdash I :: S. t \in \llbracket A[I/i] \rrbracket_{g, k-1}^{E,(L[I/i], U[I/i])}\}$
$\llbracket \exists i :: S. A \rrbracket_{g,k}$	$= \{\text{pack } v \mid \exists I. \vdash I :: S. v \in \llbracket A[I/i] \rrbracket_{g, k-1}\}$
$\llbracket A_1 + A_2 \rrbracket_{g,k}$	$= \{\text{inl } v \mid v \in \llbracket A_1 \rrbracket_{g,k}\} \cup \{\text{inr } v \mid v \in \llbracket A_2 \rrbracket_{g,k}\}$
$\llbracket C \supset A \rrbracket_{g,k}$	$= \{v \mid \not\models C \wedge v \in \llbracket A \rrbracket_{g, k-1}\}$
$\llbracket C \& A \rrbracket_{g,k}$	$= \{v \mid \models C \wedge v \in \llbracket A \rrbracket_{g, k-1}\}$
$\llbracket A \rrbracket_{g,k}^{E,(L,U)}$	$= \{t \mid \forall v, \forall k' \leq k. t \Downarrow_f^{c, k'} v \Rightarrow v \in \llbracket A \rrbracket_{g, k-k'} \wedge L \leq c \leq U\}$
$\llbracket \cdot \rrbracket_{g,k}$	$= \{\emptyset\}$
$\llbracket \Omega, x : A \rrbracket_{g,k}$	$= \{(\sigma[v/x]) \mid \sigma \in \llbracket \Omega \rrbracket_{g,k} \wedge v \in \llbracket A \rrbracket_{g,k}\}$
where	
$H \models_{g,k} \text{empty}$	iff true
$H \models_{g,k} (\gamma \rightarrow \beta)$	iff $\exists l, A, n : g(\gamma) = (l, A, n)$ $\wedge (\forall i \leq n. (H(l)[i]) \in \llbracket A \rrbracket_{g, k-1})$
$H \models_{g,k} (P * Q)$	iff $\exists H', H'', g', g'' :$ $(H = H' \uplus H'') \wedge (g = g' \uplus g'')$ $\wedge (H' \models_{g', k} p) \wedge (H'' \models_{g'', k} q)$
$v; H \Downarrow_f^{c, k}$	$\triangleq \exists v', H'. v; H \Downarrow_f^{c, k} v'; H'$

Figure 21: Unary interpretation of types

$$\begin{aligned}
\langle \text{int}_r \rangle_{G,k} &= \{(n_1, n_2) \mid n_1 = n_2\} \\
\langle \text{unit}_r \rangle_{G,k} &= \{(0, 0)\} \\
\langle \text{int}_r [I] \rangle_{G,k} &= \{(n, n) \mid I = n\} \\
\langle \text{Array}_\gamma [I] \tau \rangle_{G,k} &= \{(l_1, l_2) \mid I = n \wedge G(\gamma) = (l_1, l_2, \tau, n)\} \\
\langle \tau_1 \xrightarrow{\text{diff}(D)} \tau_2 \rangle_{G,k} &= \{(\text{fix } f(x).t_1, \text{fix } f(x).t_2) \mid \forall G' \supseteq G, k' \leq k-1, \forall v_1, v_2. (v_1, v_2) \in \langle \tau_1 \rangle_{G',k'} \\
&\Rightarrow (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 \langle \tau_2 \rangle_{G',k'}^{E,D}\} \\
&\wedge \forall j. (\text{fix } f(x).t_1 \in \llbracket \tau_1 \rrbracket_{G_1,j} \xrightarrow{\text{exec}(0,\infty)} \llbracket \tau_2 \rrbracket_{G_1,j} \wedge (\text{fix } f(x).t_2 \in \llbracket \tau_1 \rrbracket_{G_2,j} \xrightarrow{\text{exec}(0,\infty)} \llbracket \tau_2 \rrbracket_{G_2,j})) \\
\langle \{P\} \exists \tilde{\gamma}. \tau \{Q\} \rangle_{G,k} &= \left\{ (v_1, v_2) \mid \forall G_1 \supseteq G, k_1 \leq k, k_2 < k_1, k_3, c_1, c_2, H_1, H_2. \left((H_1, H_2) \models_{G_1, k_1} P \wedge v_1; H_1 \Downarrow_f^{c_1, k_2} \right. \right. \\
&\wedge v_2; H_2 \Downarrow_f^{c_2, k_3} \left. \Rightarrow \exists G_2 \supseteq G_1, H'_1, H'_2, v'_1, v'_2, c_1, c_2, \tilde{\gamma}. \right. \\
&\quad \left. \left(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 \right. \right. \\
&\quad \left. \left. (H'_1, H'_2) \models_{G_2, k_1 - k_2} Q \wedge (v'_1, v'_2) \in \langle \tau \rangle_{G_2, k_1 - k_2} \wedge c_1 - c_2 \leq D \right) \right\} \\
\langle U(A_1, A_2) \rangle_{G,k} &= \{(v_1, v_2) \mid \forall k', v_1 \in \llbracket A_1 \rrbracket_{G_1, k'}, v_2 \in \llbracket A_2 \rrbracket_{G_2, k'}\} \\
\langle \forall i :: S. \tau \rangle_{G,k} &= \{(\Lambda. t_1, \Lambda. t_2) \mid \forall I. \vdash I :: S. (t_1, t_2) \in \langle \tau [I/i] \rangle_{G, k-1}^{E, D[I/i]} \\
&\forall j. t_1 \in \llbracket \tau \rrbracket_{G_1, j}^{E, (0, \infty)} \wedge t_2 \in \llbracket \tau \rrbracket_{G_2, j}^{E, (0, \infty)}\} \\
\langle \exists i :: S. \tau \rangle_{G,k} &= \{(\text{pack } v_1, \text{pack } v_2) \mid \exists I. \vdash I :: S. (v_1, v_2) \in \langle \tau [I/i] \rangle_{G, k-1}\} \\
\langle \tau_1 + \tau_2 \rangle_{G,k} &= \{(\text{inl } v_1, \text{inl } v_2) \mid (v_1, v_2) \in \langle \tau_1 \rangle_{G, k-1}\} \cup \{(\text{inr } v_1, \text{inr } v_2) \mid (v_1, v_2) \in \langle \tau_2 \rangle_{G, k-1}\} \\
\langle \Box \tau \rangle_{G,k} &= \{(v, v) \mid (v, v) \in \langle \tau \rangle_{G, k}\} \\
\langle C \supset \tau \rangle_{G,k} &= \{(v_1, v_2) \mid \not\models C \vee (v_1, v_2) \in \langle \tau \rangle_{G, k-1}\} \\
\langle C \& \tau \rangle_{G,k} &= \{(v_1, v_2) \mid \models C \wedge (v_1, v_2) \in \langle \tau \rangle_{G, k-1}\} \\
\langle \tau \rangle_{G,k}^{E,D} &= \{(t_1, t_2) \mid \forall k_1 \leq k, v_1 v_2. (t_1 \Downarrow_f^{c_1, k_1} v_1 \wedge t_2 \Downarrow_f^{c_2, k_2} v_2) \Rightarrow \\
&(v_1, v_2) \in \langle \tau \rangle_{G, k-k_1} \wedge c_1 - c_2 \leq D\} \\
\langle \cdot \rangle_{G,k} &= \{\emptyset\} \\
\langle \Gamma \rangle_{G,k} &= \{(\sigma_1, \sigma_2) \mid \forall x \in \text{dom}(\Gamma). \forall \tau. x : \tau \in \Gamma. (\sigma_1(x), \sigma_2(x)) \in \langle \tau \rangle_{G, k}\}
\end{aligned}$$

where

$$\begin{aligned}
(H_1, H_2) \models_{G,k} \text{empty} &\text{ iff } \text{true} \\
(H_1, H_2) \models_{G,k} (\gamma \mapsto \beta) &\text{ iff } \exists l_1, l_2, \tau, n : G(\gamma) = (l_1, l_2, \tau, n) \\
&\wedge (\forall i \leq n. (H_1(l_1)[i], H_2(l_2)[i]) \in \langle \tau \rangle_{G, k-1}) \\
&\wedge (\forall i \leq n. (H_1(l_1)[i] \neq H_2(l_2)[i] \Rightarrow i \in \beta)) \\
(H_1, H_2) \models_{G,k} (P * Q) &\text{ iff } \exists H'_1, H''_1, H'_2, H''_2, G', G'' : \\
&(H_1 = H'_1 \uplus H''_1) \wedge (H_2 = H'_2 \uplus H''_2) \wedge (G = G' \uplus G'') \\
&\wedge ((H'_1, H'_2) \models_{G',k} P) \wedge ((H''_1, H''_2) \models_{G'',k} Q) \\
G(\gamma) = (l_1, l_2, \tau, n), & G_1(\gamma) = (l_1, \llbracket \tau \rrbracket_1, n), G_2(\gamma) = (l_2, \llbracket \tau \rrbracket_2, n)
\end{aligned}$$

Figure 22: 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 $k' \leq k$ and $G \subseteq G'$, then $\llbracket \tau \rrbracket_{G,k} \subseteq \llbracket \tau \rrbracket_{G',k'}$.
2. If $k' \leq k$ and $g \subseteq g'$, then $\llbracket A \rrbracket_{g,k} \subseteq \llbracket A \rrbracket_{g',k'}$.
3. If $k' \leq k$ and $G \subseteq G'$, then $\llbracket \tau \rrbracket_{G,k}^{E,D} \subseteq \llbracket \tau \rrbracket_{G',k'}^{E,D}$.
4. If $k' \leq k$ and $g \subseteq g'$, then $\llbracket A \rrbracket_{g,k}^{E,(L,U)} \subseteq \llbracket A \rrbracket_{g',k'}^{E,(L,U)}$.
5. If $k' \leq k$ and $G \subseteq G'$, then $\llbracket \Gamma \rrbracket_{G,k} \subseteq \llbracket \Gamma \rrbracket_{G',k'}$.
6. If $k' \leq k$ and $g \subseteq g'$, then $\llbracket \Omega \rrbracket_{g,k} \subseteq \llbracket \Omega \rrbracket_{g',k'}$.

Proof. (1,3) and (2,4) are proved simultaneously by induction on τ and induction hypothesis,(5,6) follows from (1,2). \square

Lemma 2 (Heap extension). If $(H_1, H_2) \models_{G,k} P$, then for any H'_1, H'_2 , $(H_1 \uplus H'_1, H_2 \uplus H'_2) \models_{G,k} P$.

Proof. It is simply proved by case analysis on P . \square

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. \square

Lemma 4 (Evaluation cost soundness).

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

Proof. (1) is directly proved from the definition of unary expression interpretation. If $\vdash_L^U t : \tau$, then $t \in \llbracket \tau \rrbracket_{G,k}^{E,(L,U)}$, unfold the definition, we get : $L \leq c \leq U$ and $t \Downarrow^c v$.

(2) $t \in \llbracket \{P\} \exists \gamma : A \{Q\} \rrbracket_{G,k}^{E,(L,U)}$, $t \Downarrow^{c_1} v'$ and $v' \Downarrow_f^{c_2} v$ unfold the definition, we know $c = c_1 + c_2$ and $L \leq c_1 \leq U$ and $L' \leq c_2 \leq U'$. It is proved.

(3) $t \in \llbracket \{P_1\}_{L_1}^{U_1} \exists \gamma_1 : \{P_2\}_{L_2}^{U_2} \exists \gamma_2 : \dots \{Q_2\} \{Q_1\} \rrbracket_{G,k}^{E,(L,U)}$. Unfold the definition, we can get : $L + L_1 + L_2 + \dots L_n \leq c_1 + c_2 + \dots + c_n \leq U + U_1 + U_2 + \dots + U_n$. 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 , $\llbracket \tau \rrbracket_{G,k}^{E,D}$, unfold the definition, $c_1 - c_2 \leq D$ 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 $\langle \{P\} \exists \gamma : \tau \{Q\} \rangle_{G,k}^{E,D}$, unfold the definition, $c_1 - c_2 \leq D + D'$ is proved.

(6) From the fundamental theory, If $\vdash t_1 \ominus t_2 \lesssim D : \{P_1\}^{\text{diff}(D_1)} \exists \gamma_1 : \{P_2\}^{\text{diff}(D_2)} \exists \gamma_2 : \dots \{Q_2\} \{Q_1\}$, we know closed terms (t_1, t_2) in the interpretation of nested relational type τ , $\langle \tau \rangle_{G,k}^{E,D}$, unfold the definition level by level, $c_1 - c_2 \leq D + D_1 + D_2 \dots + D_n$ is proved. □

Lemma 5 (Well-formedness).

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

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

Lemma 6 (Value Projection).

1. If $(v_1, v_2) \in \langle \tau \rangle_{G,k}$, then $\forall k. v_1 \in \llbracket \tau|_1 \rrbracket_{G_1,k}$ and $v_2 \in \llbracket \tau|_2 \rrbracket_{G_2,k}$.
2. If $(\sigma_1, \sigma_2) \in \langle \Gamma \rangle_{G,k}$, then $\forall k. \sigma_1 \in \langle \llbracket \Gamma|_1 \rrbracket_{G_1,k} \rangle$ and $\sigma_2 \in \langle \llbracket \Gamma|_2 \rrbracket_{G_2,k} \rangle$.

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

Lemma 7 (Heap monotonicity).

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

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

Assume $k' \leq k$ 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 $H \vDash_{G',k'} P$

Do case analysis on P , there are three cases:

case 1: $P = \text{empty}$

It is vacuously true that $H \vDash_{g',k'} \text{empty}$.

case 2: $P = \gamma \rightarrow \beta$

From $H_1, H_2 \vDash_{G,k} \gamma \rightarrow \beta$, we know:

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

$$\forall i \leq n. (H_1(l_1)[i], H_2(l_2)[i]) \in \langle \tau \rangle_{G,k-1} \tag{2}$$

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

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

case 3: $P = P_1 \star P_2$

Suppose $\exists n. P = \{\gamma_i \rightarrow \beta_i\}$ for $i \in [1, \dots, n]$.

Unfold the definition of $H_1, H_2 \models_{G,k} \{\gamma_i \rightarrow \beta_i\}$ for $i \in [1, \dots, n]$.

We know there exists $H_1 = (h_1^1) \uplus (h_1^2) \uplus \dots \uplus (h_1^n)$ and $H_2 = (h_2^1) \uplus (h_2^2) \uplus \dots \uplus (h_2^n)$ and $G = G_1 \uplus G_2 \uplus \dots \uplus G_n$

so that $h_1^i, h_2^i \models_{G_i,k} \gamma_i \rightarrow \beta_i$ for any i .

Since $G' \supseteq G$, we know there exists $G'_i \supseteq G_i$ s.t. $G' = G'_1 \uplus G'_2 \uplus \dots \uplus G'_n \wedge \forall i. G'_i \supseteq G_i$

Use the conclusion of case 2, we know:

$\forall i \in [1, n]. h_1^i, h_2^i \models_{G'_i,k'} \gamma_i \rightarrow \beta_i$.

proof of statement (2)

Assume $k' \leq k$ and $g' \supseteq g$, and we have

$$H \models_{g,k} P$$

. TS: $H \models_{g',k'} P$

Unfold the definition of $H \models_{g',k'} P$

Case analysis on P , there are three cases:

case 1: $P = \text{empty}$

It is vacuously true that $H \models_{g',k'} \text{empty}$.

case 2: $P = \gamma \rightarrow \beta$

From $H \models_{g,k} \gamma$, we know:

$$g(\gamma) = (l, A, n) \tag{1}$$

$$\forall i \leq n. H(l)[i] \in \llbracket A \rrbracket_{g,k-1} \tag{2}$$

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

case 3: $P = P * Q$

Assume $p = \{\gamma_i\}$ for $i \in [1, \dots, n]$.

From the definition of $H \models_{g,k} \{\gamma_1, \gamma_2, \dots, \gamma_n\}$, we know

$\exists H = (h_1^1) \uplus (h_1^2) \uplus \dots \uplus (h_1^n)$ and $g = g_1 \uplus g_2 \uplus \dots \uplus g_n$ s.t. $h_1^i \models_{g_i,k} \gamma_i \rightarrow \beta_i$ for any i .

Since $g' \supseteq g$, we know: $\exists g'_1, g'_2, \dots, g'_n. g' = g'_1 \uplus g'_2 \uplus \dots \uplus g'_n \wedge \forall i. g'_i \supseteq g_i$.

By the conclusion of case 2. we know: $h_1^i \models_{g'_i,k'} \gamma_i$ for any i .

This completes the proof of statement (2). □

Lemma 8 (Value interpretation containment).

1. If $(v_1, v_2) \in \langle \tau \rangle_{G,k}$, then $(v_1, v_2) \in \langle \tau \rangle_{G,k}^{E,0}$
2. If $v \in \llbracket A \rrbracket_{g,k}$, then $\forall t \geq 0. v \in \llbracket A \rrbracket_{g,k}^{E,(0,t)}$

Proof. (1) Assume $(v_1, v_2) \in \langle \tau \rangle_{G,k}^{(*)}$

TS: $(v_1, v_2) \in \langle \tau \rangle_{G,k}^{E,0}$

Following the definition of $\langle \tau \rangle_{G,k}^{E,0}$, we get: $v_1 \Downarrow^0 v_1$ and $v_1 \Downarrow^0 v_1$ and $0 - 0 \leq 0$ and $(v_1, v_2) \in \langle \tau \rangle_{G,k-0}$. This completes the proof. □

Proof. Assume $v \in \llbracket A \rrbracket_{g,k}^{(*)}$

TS: $\forall t \geq 0. v \in \llbracket A \rrbracket_{g,k}^{E,(0,t)}$ Following the definition of $\llbracket A \rrbracket_{g,k}^{E,(0,t)}$. We know $v \Downarrow^0 v$ and $v \in \llbracket A \rrbracket_{g,k}$. 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 $(H_1, H_2) \models_{G,k} \gamma_i \rightarrow \beta_i, i \in (1, n)$, then $\forall T, T'. H_1 \models_{G|_1, k} \gamma_i \rightarrow T_i$ and $H_2 \models_{G|_2, k} \gamma_i \rightarrow T'_i$*

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

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

$$G(\gamma) = (l_1, l_2, \tau, n) \quad (1)$$

$$\forall i \leq n. (h_i(l_1)[i], h'_i(l_2)[i]) \in \llbracket \tau \rrbracket_{G, k-1} \quad (2)$$

From (1), we know:

$$G|_1(\gamma) = (l_1, \gamma, n) \quad (3)$$

$$G|_2(\gamma) = (l_2, \gamma, n) \quad (4)$$

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

$$\forall i \leq n. (H_1(l_1)[i]) \in \llbracket \tau|_1 \rrbracket_{G|_1, k-1} \quad (5)$$

$$\forall i \leq n. (H_2(l_2)[i]) \in \llbracket \tau|_2 \rrbracket_{G|_2, k-1} \quad (6)$$

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

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

Proof. Assume we have:

$$H_1, H_2 \models_{G,k} \delta P$$

We want to show :

$$H_1, H_2 \models_{G,k} \delta P'$$

Suppose $\exists n. \delta P = \{\gamma_i \rightarrow \beta_i\}$ for $i \in [1, \dots, n]$.

Unfold the definition of $H_1, H_2 \models_{G,k} \{\gamma_i \rightarrow \beta_i\}$ for $i \in [1, \dots, n]$.

We know there exists $H_1 = (h_1^1) \uplus (h_1^2) \uplus \dots \uplus (h_1^n)$ and $H_2 = (h_2^1) \uplus (h_2^2) \uplus \dots \uplus (h_2^n)$ and $G = G_1 \uplus G_2 \uplus \dots \uplus G_n$ so that $h_1^i, h_2^i \models_{G_i, k} \gamma_i \rightarrow \beta_i$ for any i .

Unfold the definition of $\Delta; \Phi \models \delta P \subseteq \delta P'$, we know:

$$\delta P' = \{\gamma_i \rightarrow \beta'_i\}, i \in [1, \dots, n]$$

STS: $\forall i \in [1, \dots, n]. \exists x_1^i, x_2^i \models_{G_i, k} \gamma_i \rightarrow \beta'_i$ where $H_1 = (x_1^1) \uplus (x_1^2) \uplus \dots \uplus (x_1^n)$ and $H_2 = (x_2^1) \uplus (x_2^2) \uplus \dots \uplus (x_2^n)$

Pick i . Choose $x_1^i = h_1^i$ and $x_2^i = h_2^i$

Unfold the definition of $h_1^i, h_2^i \models_{G_i, k} \gamma_i \rightarrow \beta_i$.

we have: $G_i(\gamma_i) = (l_1^i, l_2^i, \tau^i, n)$ and $\forall m \leq n. (h_1^i(l_1^i[m]) \neq h_2^i(l_2^i[m]) \Rightarrow m \in \beta_i$

Since we know $\beta_i \subseteq \beta'_i$, we get:

$$\forall i \in [1, \dots, n]. G_i(\gamma_i) = (l_1^i, l_2^i, \tau^i, n) \wedge \forall m \leq n. (x_1^i(m) \neq x_2^i(m)) \Rightarrow m \in \beta'_i$$

For any $i \in [1, \dots, n]$, choose $x_1^i = h_1^i$ and $x_2^i = h_2^i$, we know: $x_1^i, x_2^i \models_{G_i, k} \gamma_i \rightarrow \beta'_i$

We also know $H_1 = (x_1^1) \uplus (x_1^2) \uplus \dots \uplus (x_1^n)$ and $H_2 = (x_2^1) \uplus (x_2^2) \uplus \dots \uplus (x_2^n)$ from assumption.

This completes the proof of the lemma. □

- Lemma 12** (Subtyping soundness). 1. If $\Delta; \Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $(v, v') \in \langle \delta \tau \rangle_{G,k}$, then $(v, v') \in \langle \delta \tau' \rangle_{G,k}$.
2. If $\Delta; \Phi \models A \sqsubseteq A'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $(v) \in \llbracket \delta A \rrbracket_{g,k}$, then $(v) \in \llbracket \delta A' \rrbracket_{g,k}$.
3. If $\Delta; \Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $(t, t') \in \langle \delta \tau \rangle_{G,k}^{E,D}$ and $D \leq D'$, then $(t, t') \in \langle \delta \tau' \rangle_{G,k}^{E,D'}$.
4. If $\Delta; \Phi \models A \sqsubseteq A'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $t \in \llbracket \delta A \rrbracket_{g,k}^{E,(L,U)}$ and $L' \leq L$ and $U \leq U'$, then $t \in \llbracket \delta A' \rrbracket_{g,k}^{E,(L',U')}$.
5. If $\Delta; \Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $\forall i \in \{1, 2\}. (v) \in \llbracket \delta \tau|_i \rrbracket_{G_i,k}$, then $v \in \llbracket \delta \tau' \rrbracket_{G_i,k}$.
6. If $\Delta; \Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $\forall i \in \{1, 2\}. (e) \in \llbracket \delta \tau|_i \rrbracket_{G_i,k}^{E,(L,U)}$ and $L' \leq L$ and $U \leq U'$, then $e \in \llbracket \delta \tau' \rrbracket_{G_i,k}^{E,(L',U')}$.

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 $(t, t') \in \langle \delta \tau \rangle_{G,k}^{E,D}$ and $D \leq D'$.

TS: $(t, t') \in \langle \delta \tau' \rangle_{G,k}^{E,D'}$.

Assume that

1. $t \Downarrow^{s,c} v$
2. $t' \Downarrow^{s',c'} v'$
3. $s < k$

Unfold the assumption $(t, t') \in \langle \delta \tau \rangle_{G,k}^{E,D}$ using assumption (1)(2)(3), we know

$$c - c' \leq D \quad (\text{a})$$

$$(v, v') \in \langle \delta \tau \rangle_{G,k-s} \quad (\text{b})$$

we conclude:

$c - c' \leq D'$ from assumption $D \leq D'$

By IH1 on (b), we know $(v, v') \in \langle \delta \tau' \rangle_{G,k-s}$ □

Proof. Proof of statement (4) Assume $\Delta; \Phi \models A \sqsubseteq A'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $e \in \llbracket \delta A \rrbracket_{g,k}^{E,(L,U)}$ and $L' \leq L$ and $U \leq U'$

TS: $t \in \llbracket \delta A \rrbracket_{g,k}^{E,(L,U)}$

Assume:

1. $t \Downarrow^{s,c}$
2. $s < k$

Unfold the assumption $t \in \llbracket \delta A \rrbracket_{g,k}^{E,(L,U)}$, we know:

$$L \leq c \leq U \quad (\text{a})$$

$$v \in \llbracket \delta A \rrbracket_{g,k-s} \quad (\text{b})$$

We conclude: $L' \leq c \leq U'$ from assumption.

By IH2 with the first premise on (b), we get: $v \in \llbracket \delta A' \rrbracket_{g,k-s}$ □

Proof. Proof of statement (6) Assume $\Delta; \Phi \models \tau \sqsubseteq \tau'$ and $\vdash \delta : \Delta$ and $\models \Phi \delta$ and $\forall i \in \{1, 2\}. (e) \in \llbracket \delta \tau|_i \rrbracket_{G_i,k}^{E,(L,U)}$ and $L' \leq L$ and $U \leq U'$

TS: $t \in \llbracket \delta \tau' \rrbracket_{G_i,k}^{E,(L,U)}$

Assume:

1. $t \Downarrow^{s,c}$
2. $s < k$

Unfold the assumption $t \in \llbracket \delta \tau' \rrbracket_{G|i,k}^{E,(L,U)}$, we know:

$$L \leq c \leq U \quad (\text{a})$$

$$v \in \llbracket \delta \tau' \rrbracket_{G|i,k-s} \quad (\text{b})$$

We conclude: $L' \leq c \leq U'$ from assumption.

By IH5 with the first premise on (b), we get: $v \in \llbracket \delta \tau' \rrbracket_{G|i,k-s}$ \square

Proof of statement (1)

The proof is by induction on the subtyping derivation, we will detail some of the most interesting cases.

Case

$$\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Delta; \Phi \models D \leq D' \quad \Delta; \Phi \models P' \subseteq P \quad \Sigma, \vec{\gamma}_1 : \vec{L}; \Delta; \Phi \models Q \subseteq Q' \quad \Sigma \models \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2}{\Sigma; \Delta; \Phi \models \{P\} \exists \vec{\gamma}_1 : \tau \{Q\} \sqsubseteq \{P'\} \exists \vec{\gamma}_2 : \tau' \{Q'\}} \text{s-RM}$$

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

$$(v_1, v_2) \in \llbracket \{\delta P\} \exists \vec{\gamma}_1 : \delta \tau \{ \delta Q \} \rrbracket_{\delta G, k}^{\text{diff}(\delta D)} \quad (1)$$

We want to show

$$(v_1, v_2) \in \llbracket \{\delta P'\} \exists \vec{\gamma}_2 : \delta \tau' \{ \delta Q' \} \rrbracket_{\delta G, k}^{\text{diff}(\delta D')} \quad (2)$$

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

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

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

$$v_1; H_1 \downarrow_f^{c_1, k_2} v'_1; H'_1 \wedge v_2; H_2 \downarrow_f^{c_2, k_2} v'_2; H'_2 \wedge (H'_1, H'_2) \models_{G_2, k_1 - k_2} \delta Q \wedge (v'_1, v'_2) \in \llbracket \delta \tau \rrbracket_{G_2, k_1 - k_2} \wedge c_1 - c_2 \leq \delta D$$

Since $\Sigma, \vec{\gamma}_1 : \vec{L}; \Delta; \Phi \models Q \subseteq Q'$ 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 $\Delta; \Phi \models D \leq D'$

By IH on the first premise, we also get

$(v'_1, v'_2) \in \llbracket \delta \tau' \rrbracket_{G_2, k_1 - k_2}$ and $c_1 - c_2 \leq \delta D'$ respectively.

So we can derive Equation (2), and this completes the proof of case-sub-r-monad.

s-RUM

$$\Sigma; \Delta; \Phi \models U (\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}_2 : A_1 \{Q_2\}) \sqsubseteq \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\}$$

Pick G, k, v, v' .

Assume $\vdash \delta : \Delta$ and $\models \Phi \delta$, we have:

$$(v, v') \in \llbracket \delta U (\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}_2 : A_2 \{Q_2\}) \rrbracket_{G, k}^{\text{exec}(L,U), \text{exec}(L',U')} \quad (\text{a})$$

TS: $(v, v') \in \llbracket \delta \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\} \rrbracket_{G, k}^{\text{diff}(U-L')}$

Unfold the definition of $\llbracket \delta \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\} \rrbracket_{G, k}^{\text{diff}(U-L')}$.

Pick $G' \supseteq G, k' \leq k, k'' < k, H_1, H_2$.

Assume

$$(H_1, H_2) \models_{G', k'} \gamma_i \rightarrow \beta_i \quad (1)$$

$$H_1; v \downarrow_f^{k''} \wedge H_2; v' \not\Downarrow_f^f \quad (2)$$

STS: $\exists G'' \supseteq G, H'_1, H'_2, v_1, v'_1$

$$H_1; v \downarrow_f^{c_1, k''} H'_1; v_1 \quad (3)$$

$$H_2; v' \downarrow_f^{c_2} H'_2; v'_1 \quad (4)$$

$$H'_1, H'_2 \models_{G'', k'-k''} \gamma_i \rightarrow \beta_i \cup T_i \cup T'_i \quad (5)$$

$$(v_1, v'_1) \in \llbracket U(A_1, A_2) \rrbracket_{G'', k'-k''} \quad (6)$$

From (a), unfold its definition of $(\delta U (\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}'_1 : A_2 \{Q_2\}))_{G, k}$

$$\forall k', v \in \llbracket \{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\} \rrbracket_{G|_1, k'} \quad (b)$$

$$\forall k', v' \in \llbracket \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}'_2 : A_2 \{Q_2\} \rrbracket_{G|_2, k'} \quad (c)$$

Pick k' as k .

Unfold its definition in (b).

Show we already got: $G'|_1 \supseteq G|_1$ and $k' \leq k$ and $k'' < k'$ and $H_1; v \downarrow_f^{k''}$ and $H_1 \models_{G'|_1, k'} \gamma_i \rightarrow T_i$ holds by using Lemma (heap projection) on (1).

Derive from the definition, we know:

$\exists g''_1 \supseteq G'|_1, H'_1, v_1$.

$$H_1; v \downarrow_f^{c_1, k''} H'_1; v_1 \quad (d)$$

$$H'_1 \models_{g''_1, k'-k''} Q_1 \quad (e)$$

$$v_1 \in \llbracket A_1 \rrbracket_{g''_1, k'-k''} \quad (f)$$

$$\forall i. g''_1(\gamma_i) = (l_i, A, m) \Rightarrow H_1(l)[n] \neq H'_1(l)[n] \Rightarrow n \in T_i \quad (*)$$

Unfold its definition in (c).

Show we already got: $G'|_2 \supseteq G|_2$ and $k' \leq k$ and $k'' < k'$ and $H_2; v \downarrow_f$ and $H_2 \models_{G'|_2, k'} \gamma_i \rightarrow T'_i$ 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 $k_1 \leq k'$. We choose $k'' = k_1$. we get $H_2; v \downarrow_f^{k''}$

Derive from the definition, we know:

$\exists g''_2 \supseteq G'|_2, H'_2, v'_1$.

$$H_2; v \downarrow_f^{c_2, k''} H'_2; v'_1 \quad (g)$$

$$H'_2 \models_{g''_2, k'-k''} Q_2 \quad (h)$$

$$v'_1 \in \llbracket A_2 \rrbracket_{g''_2, k'-k''} \quad (i)$$

$$\forall i. g''_2(\gamma_i) = (l_i, A, m) \Rightarrow H_2(l)[n] \neq H'_2(l)[n] \Rightarrow n \in T'_i \quad (**)$$

(3),(4) already proved.

STS1: $H'_1, H'_2 \models_{G'', k'-k''} \gamma_i \rightarrow \beta_i \cup T_i \cup T'_i$

From (e), we assume

$$H'_1 = h_1 \uplus h_2 \uplus \dots \uplus h_n \quad (7)$$

$$H'_2 = h'_1 \uplus h'_2 \uplus \dots \uplus h'_n \quad (8)$$

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 $H'_1, H'_2 \models_{G,k} \gamma_i \rightarrow \beta_i \cup T_i \cup T'_i$. $\forall n. H'_1(l_i)[n] \neq H'_2(l_i)[n] \Rightarrow n \in \beta_i \cup T_i \cup T'_i$ 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.

$$\frac{\Delta; \Phi \models L' \leq L \quad \Delta; \Phi \models U \leq U' \quad \frac{\Delta; \Phi \models A \sqsubseteq A' \quad \Delta; \Phi \models P \sqsubseteq P' \quad \Sigma, \vec{\gamma}_1; \Delta; \Phi \models Q' \sqsubseteq Q \quad \vec{\gamma}_1 \subseteq \vec{\gamma}_2}{\Sigma; \Delta; \Phi \models \{P\} \exists \vec{\gamma}_1 : A \{Q\} \sqsubseteq \{P'\} \exists \vec{\gamma}_2 : A' \{Q'\}} \text{S-A-MONAD}}{\text{exec}(L,U) \quad \text{exec}(L',U')}$$

Pick g,k,v.

Assume $\vdash \delta : \Delta$ and $\models \Phi \delta$, we know

$$v \in \{P\} \exists \vec{\gamma}_1 : A \{Q\} \quad (a)$$

TS:

$$v \in \{P'\} \exists \vec{\gamma}_2 : A' \{Q'\} \quad (b)$$

Unfold the definition of $\{P\} \exists \vec{\gamma}_1 : A \{Q\}$, we pick $g' \supseteq g, k' \leq k, k'' < k', H$

Assume

$$H \models_{g',k'} P \quad (1)$$

$$H; v \downarrow_f^{k''} \quad (2)$$

We know: $\exists g'' \supseteq g, H', v_1$

$$H; v \downarrow_f^{c_1, k''} H'; v_1 \quad (3)$$

$$H' \models_{g'', k' - k''} Q \quad (4)$$

$$v_1 \in \llbracket A \rrbracket_{g'', k' - k''} \quad (5)$$

$$P = \gamma_i \rightarrow T_i \Rightarrow \forall i. g''(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T_i \quad (*)$$

Pick g'', H', v_1 .

By IH 2 on the first premise $\Delta; \Phi \models A \sqsubseteq A'$ with (5), we know:

$$v_1 \in \llbracket A' \rrbracket_{g'', k' - k''} \quad (6)$$

TS: $v \in \{P'\} \exists \vec{\gamma}_2 : A' \{Q'\}$, unfold its definition.

we know: $H \models_{g',k'} P'$ from (1) and $P \supseteq P'$ because P and P' have the same γ_i . together with (2)

STS:

$$H; v \Downarrow_f^{c_1, k''} H'; v_1 \quad (c)$$

$$H' \models_{g'', k' - k''} Q' \quad (d)$$

$$v_1 \in \llbracket A' \rrbracket_{g'', k' - k''} \quad (e)$$

$$P' = \gamma_i \rightarrow T'_i \Rightarrow \forall i. g'_1(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T'_i \quad (**)$$

(c) is proved by assumption (3).

(e) is proved by (6).

(d) is proved by (4) and $Q' \supseteq Q$ (**). 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

$$\Sigma; \Delta; \Phi \models U \left(\frac{\text{exec}(L, U)}{\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}_2 : A_1 \{Q_2\}} \frac{\text{exec}(L', U')}{\{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\}} \text{diff}(U-L') \right)$$

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

$$v \in \llbracket \delta U \left(\frac{\text{exec}(L, U)}{\{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\}, \{\gamma_i \rightarrow T'_i\} \exists \vec{\gamma}_1 : A_2 \{Q_2\}} \right) \rrbracket_{|G|_i, k} \quad (a)$$

TS:

$$v \in \llbracket \delta \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : U(A_1, A_2) \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\} \rrbracket_{|G|_i, k} \quad (b)$$

Choose $i=1$.

From (a), we know:

$$v \in \llbracket \delta \{\gamma_i \rightarrow T_i\} \exists \vec{\gamma}_1 : A_1 \{Q_1\} \rrbracket_{|G|_1, k} \quad (c)$$

From (b),

STS:

$$v \in \llbracket \delta \{\gamma_i \rightarrow \beta_i\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : A_1 \{\gamma_i \rightarrow \beta_i \cup T_i \cup T'_i\} \rrbracket_{|G|_1, k} \quad (d)$$

Assume $g' \supseteq G|_1, k' \leq k, k'' < k'$. we assume

$$H \models_{g', k'} \gamma_i \rightarrow \mathbb{N} \quad (1)$$

$$H; v \Downarrow_f^{k''} \quad (2)$$

STS: $\exists g'' \supseteq g', H', v'$

$$H; v \Downarrow_f^{k'', c_1} H'; v' \quad (3)$$

$$H' \models_{g'', k' - k''} \gamma_i \rightarrow \mathbb{N} \quad (4)$$

$$v' \in \llbracket A_1 \rrbracket_{g'', k' - k''} \quad (5)$$

$$P' = \gamma_i \rightarrow \mathbb{N} \Rightarrow \forall i. g'_1(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in N \quad (**)$$

From (c), unfold its definition, we know (2) and $H \models_{g',k'} \gamma_i \rightarrow T_i$ holds from (1). Then we get:
 $\exists g'' \supseteq g', H', v'$

$$H; v \Downarrow_f^{k'',c_1} H'; v' \quad (e)$$

$$H' \models_{g'',k'-k''} Q_1 \quad (f)$$

$$v' \in \llbracket A_1 \rrbracket_{g'',k'-k''} \quad (g)$$

$$P = \gamma_i \rightarrow T_i \Rightarrow \forall i. g'_1(\gamma_i) = (l_i, A, m) \Rightarrow H(l_i)[n] \neq H'(l_i)[n] \Rightarrow n \in T_i \quad (**)$$

(3) is proved by (e). (4) is proved by (f) because Q_1 contains at least as many γ_i as its precondition $\gamma_i \rightarrow T_i$. (*) is proved by (**) because $T_i \supseteq \mathbb{N}$ This completes the proof of this case.

$$\frac{\Sigma; \Delta; \Phi \models \tau \sqsubseteq \tau' \quad \Delta; \Phi \models D \leq D' \quad \Delta; \Phi \models P' \subseteq P \quad \Sigma, \vec{\gamma}_1 : \vec{L}; \Delta; \Phi \models Q \subseteq Q' \quad \Sigma \models \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2}{\Sigma; \Delta; \Phi \models \{P\} \exists \vec{\gamma}_1 : \tau \{Q\} \sqsubseteq \{P'\} \exists \vec{\gamma}_2 : \tau' \{Q'\}} \text{S-RM}$$

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

$$v \in \llbracket \{P\} \exists \vec{\gamma}_1 : \tau \{Q\} \mid i \rrbracket_{|G|_i, k} \quad (a)$$

TS:

$$v \in \llbracket \{\delta \{P'\} \exists \vec{\gamma}_2 : \tau' \{Q'\} \mid i \rrbracket_{|G|_i, k} \quad (b)$$

Choose $i=1$.

From (a), we know:

$$v \in \llbracket \{\delta \{P\}_1\} \exists \vec{\gamma}_1 : \tau_1 \{Q\}_1 \rrbracket_{|G|_1, k} \quad (c)$$

From (c), STS:

$$v \in \llbracket \{\delta \{P'\}_1\} \exists \vec{\gamma}_2 : \tau'_1 \{Q'\}_1 \rrbracket_{|G|_1, k} \quad (d)$$

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 $\models \delta \Phi$ and $(\sigma_1, \sigma_2) \in \langle \delta \Gamma \rangle_{G, k}$, then $(\delta \sigma_1 t_1, \delta \sigma_2 t_2) \in \langle \delta \tau \rangle_{G, k}^{E, (\delta D)}$.
2. If $\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A$ and $\vdash \delta : \Delta$ and $\models \delta \Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $\sigma \in \langle \delta \Omega' \rangle_{g, k}$, then $(\delta \sigma t) \in \llbracket \delta A \rrbracket_{g, k}^{E, (\delta L, \delta U)}$.
3. If $\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D : \tau$ and $\vdash \delta : \Delta$ and $\models \delta \Phi$, then for $i \in \{1, 2\}$. if there exists Γ'_i s.t. $FV(t_i) \subseteq \text{dom}(\Gamma'_i)$ and $\Gamma'_i \subseteq \Gamma$ and $\sigma_i \in \llbracket \delta \Gamma'_i \mid i \rrbracket_{|G|_i, k}$, then $\delta \sigma_i t_i \in \llbracket \delta \tau_i \rrbracket_{|G|_i, k}^{E, (0, \infty)}$.

Corollary 2.1.1. If $\forall k. \bullet \vdash t_1 \ominus t_2 \lesssim n : \tau$, then $(t_1, t_2) \in \langle \tau \rangle_{G, k}^E$.

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

Proof by induction on typing derivation:

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

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

TS: $(\delta \sigma_1 \text{fix } f(x).t_1, \delta \sigma_2 \text{fix } f(x).t_2) \in \langle \delta(\tau_1 \xrightarrow{\text{diff}(D)} \tau_2) \rangle_{G,k}^{E,D}$

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

STS: $(\delta \sigma_1 \text{fix } f(x).t_1, \delta \sigma_2 \text{fix } f(x).t_2) \in \langle \delta \tau_1 \xrightarrow{\text{diff}(\delta D)} \delta \tau_2 \rangle_{G,k}$

Set $F = \delta \sigma_1 \text{fix } f(x).t_1, F' = \delta \sigma_2 \text{fix } f(x).t_2$,

We want to prove a general statement

$$\forall m \leq k. (F, F') \in \langle \delta \tau_1 \xrightarrow{\text{diff}(\delta D)} \delta \tau_2 \rangle_{G,m}$$

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: $\forall k' < m. (v_1, v_2) \in \langle \tau_1 \rangle_{G,k'} \Rightarrow \dots$
 $k' \leq -1$, it is vacuous by definition.

subsubcase 2: STS: $\forall j. F \in \llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,j}} \wedge F' \in \llbracket \delta \tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_2 \rrbracket_{G_{2,j}}$
Pick j .

1. STS 1: $F \in \llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,j}}$
We prove the more general statement

$$\forall m' \leq j. F \in \llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,m'}} \quad (2.1)$$

By subinduction on m' .

There are two csases:

1.1. $m'=0$.

Unfold the definition of $\llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,m'}}$, there is no non-negative $k' < m'(0)$, it is vacuously true.

1.2. $m'=m''+1$.

By sub-IH on m''

$$F \in \llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,m''}} \quad (2.2)$$

STS: $F \in \llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,m'}}$

Unfold the definition of $\llbracket \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1 \rrbracket_{G_{1,m''+1}}$.

Pick $j'' < m'' + 1, g' \supseteq G_{1,j''}$ and assume that $v \in \llbracket \delta \tau_1|_1 \rrbracket_{g',j''}$.

STS: $\delta \sigma_1 t_1[v/x][\text{fix } f(x).t_1/f] \in \llbracket \tau_2|_1 \rrbracket_{g',j''}^{E,(0,\infty)}$

This follows by IH 3 on the premise instantiated with

$\sigma_1[v/x][\text{fix } f(x).t_1/f] \in \llbracket \delta(x : \tau_1|_1, f : \delta \tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} \delta \tau_2|_1, |\Gamma_1|) \rrbracket_{g',j''}$ which holds because

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

ii. $\sigma_1 \in \llbracket \delta \Gamma|_1 \rrbracket_{g',j''}$ by Lemma 6 and then Lemma 1.

iii. $v \in \llbracket \delta \tau_1|_1 \rrbracket_{g',j''}$ from the assumption.

- iv. $F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{g',j''}$ by Lemma 1 on (3.2).
2. STS 2: $F' \in \llbracket |\delta\tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_2 \rrbracket_{G|_2,j}$
 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: $\forall k' < m. (v_1, v_2) \in \langle \tau_1 \rangle_{G,k} \Rightarrow \dots$

Assume $G' \supseteq G, k' \leq (m-1)$. Pick v_1, v_2 s.t. $(v_1, v_2) \in \langle \delta\tau_1 \rangle_{G',k'}$.

STS: $(t_1[v_1/x][\text{fix } f(x).t_1/f], t_2[v_2/x][\text{fix } f(x).t_2/f]) \in \langle \delta\tau_2 \rangle_{G',k'}^{E,(\delta D)}$

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 $(\sigma'_1, \sigma'_2) \in \langle (\Gamma, x : \tau, f : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2)\delta \rangle_{G',k'}$ which holds because

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

subsubcase 2: $\forall j. F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G|_1,j} \wedge F' \in \llbracket |\delta\tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_2 \rrbracket_{G|_2,j}$
 which is already proved above

This completes the proof of this case.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2, \Gamma, f : U(A_1, A_2) \vdash t_1 \ominus t_2 \lesssim D : \tau_2 \quad \Sigma; \Delta; \Phi; |\Gamma|_1 \vdash_0^0 \text{Fix } f(x).t_1 : A_1 \quad \Sigma; \Delta; \Phi; |\Gamma|_2 \vdash_0^0 \text{Fix } f(x).t_2 : A_2}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{Fix } f(x).t_1 \ominus \text{Fix } f(x).t_2 \lesssim D : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2} \text{R-FIX-EXT}$$

Assume that $\vdash \delta : \Delta$ and $\models \Phi\delta$ and $(\sigma_1, \sigma_2) \in \langle \delta\Gamma \rangle_{G,k}$.

TS: $(\text{fix } f(x).t_1\delta\sigma_1, \text{fix } f(x).t_2\delta\sigma_2) \in \langle \delta(\tau_1 \xrightarrow{\text{diff}(D)} \tau_2) \rangle_{G,k}^{E,D}$

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

STS: $(\text{fix } f(x).t_1\delta\sigma_1, \text{fix } f(x).t_2\delta\sigma_2) \in \langle \delta\tau_1 \xrightarrow{\text{diff}(\delta D)} \delta\tau_2 \rangle_{G,k}$

Set $F = \text{fix } f(x).t_1\delta\sigma_1, F' = \text{fix } f(x).t_2\delta\sigma_2$,

We want to prove a general statement

$$\forall m \leq k. (F, F') \in \langle \delta\tau_1 \xrightarrow{\text{diff}(\delta D)} \delta\tau_2 \rangle_{G,m}$$

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: $\forall k' < m. (v_1, v_2) \in \langle \tau_1 \rangle_{G,k'} \Rightarrow \dots$
 $k' \leq -1$, it is vacuous by definition.

subsubcase 2: STS: $\forall j. F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G|_1,j} \wedge F' \in \llbracket |\delta\tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_2 \rrbracket_{G|_2,j}$
 Pick j .

1. STS 1: $F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,j}$
 We prove the more general statement

$$\forall m' \leq j. F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,m'} \quad (2.3)$$

By subinduction on m' .
 There are two csases:

- 1.1. $m'=0$.

Unfold the definition of $\llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,m'}$, there is no non-negative $k' < 0(m')$, it is vacuously true.

- 1.2. $m'=m''+1$.

By sub-IH on m''

$$F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,m''} \quad (2.4)$$

STS: $F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,m'}$

Unfold the definition of $\llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,m''+1}$.

Pick $j'' < m'' + 1, g' \supseteq G_1$ and assume that $v \in \llbracket |\delta\tau_1|_1 \rrbracket_{g',j''}$.

STS: $\delta\sigma_1 t_1 [v/x] [\text{fix } f(x). t_1 / f] \in \llbracket |\tau_2|_1 \rrbracket_{g',j''}^{E,(0,\infty)}$

This follows by IH 3 on the premise instantiated with

$\sigma_1 [v/x] [\text{fix } f(x). t_1 / f] \in \llbracket \delta | (x : \tau_1, f : \delta\tau_1 \xrightarrow{\text{diff}(D)} \delta\tau_2, f : U(A_1, A_2), \Gamma_1) |_1 \rrbracket_{g',j''} \Rightarrow$
 $\sigma_1 [v/x] [\text{fix } f(x). t_1 / f] \in \llbracket \delta (x : |\tau_1|_1, f : A_1, |\Gamma_1|_1) \rrbracket_{g',j''}$ which holds because

- i. $FV(t_1) \subseteq \text{dom}(x : \tau_1, f : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2, f : U(A_1, A_2), \Gamma)$ using Lemma 5.
- ii. $\sigma_1 \in \llbracket \delta \Gamma |_1 \rrbracket_{g',j''}$ by Lemma 6 and then Lemma 1.
- iii. $v \in \llbracket |\delta\tau_1|_1 \rrbracket_{g',j''}$ from assumption.
- iv. $F \in \llbracket \delta A_1 \rrbracket_{g',j''}$ by IH2 on the second premise of FIX-EXT instantiated with $\Omega' = |\Gamma|_1$ to show $F \in \llbracket \delta A_1 \rrbracket_{G_1,k}$. Then use Lemma 1.

2. STS 2: $F' \in \llbracket |\delta\tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_2 \rrbracket_{G_2,j}$

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: $\forall k' < m. (v_1, v_2) \in \langle \tau_1 \rangle_{G,k} \Rightarrow \dots$

Assume $G' \supseteq G, k' \leq (m-1)$. Pick v_1, v_2 s.t. $(v_1, v_2) \in \langle \delta\tau_1 \rangle_{G',k'}$.

STS: $(t_1 [v_1/x] [\text{fix } f(x). t_1 / f], t_2 [v_2/x] [\text{fix } f(x). t_2 / f]) \in \langle \delta\tau_2 \rangle_{G',k'}^{E,(\delta D)}$

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 $(\sigma'_1, \sigma'_2) \in \langle (\Gamma, x : \tau, f : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2, f : U(A_1, A_2)) \delta \rangle_{G',k'}$ which holds because

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

subsubcase 2: $\forall j. F \in \llbracket |\delta\tau_1|_1 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_1 \rrbracket_{G_1,j} \wedge F' \in \llbracket |\delta\tau_1|_2 \xrightarrow{\text{exec}(0,\infty)} |\delta\tau_2|_2 \rrbracket_{G_2,j}$
 which is already proved above

This proof is the same as case Fix.

$$\text{CASE} \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \tau \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{alloc } t_1 t_2 \ominus \text{alloc } t'_1 t'_2 \lesssim 0 : \{P\} \exists \gamma. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\}} \text{R-ALLOC}$$

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

TS: $(\delta \sigma_1(\text{alloc } t_1 t_2), \delta \sigma_2(\text{alloc } t'_1 t'_2)) \in (\{P\} \exists \tilde{\gamma}. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\})_{G,k}^{E,0}$

By unfolding the definition of $(\tau)_{G,k}^E$

STS: $(\delta \sigma_1(\text{alloc } t_1 t_2), \delta \sigma_2(\text{alloc } t'_1 t'_2)) \in (\{P\} \exists \gamma. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\})_{G,k}$

Unfold the definition of $(\{P\} \exists \gamma. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\})_{G,k}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k$, $k'' < k', k'''$.

Assume

$$\delta \sigma_1(\text{alloc } t_1 t_2); H_1 \Downarrow_f^{k''} \tag{1}$$

$$\delta \sigma_2(\text{alloc } t'_1 t'_2); H_2 \Downarrow_f^{k'''} \tag{2}$$

Because t_1, t_2 are sub terms of $\delta \sigma \text{ alloc } t_1 t_2$, t'_1, t'_2 are sub terms of $\delta \sigma \text{ alloc } t'_1 t'_2$
From (1)(2), we get

$$\delta \sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta \sigma_1 t_2 \Downarrow_f^{k_2} \tag{3}$$

$$\delta \sigma_2 t'_1 \Downarrow_f \wedge \delta \sigma_2 t'_2 \Downarrow_f \tag{4}$$

From (3),(4), we get:

$$\exists v_1, v_2, k_1, k_2. \delta \sigma_1 t_1 \Downarrow^{k_1, c_1} v_1 \wedge \delta \sigma_1 t_2 \Downarrow^{k_2, c_2} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_2 + c_{\text{alloc}} \tag{a}$$

$$\exists v'_1, v'_2, k'_1, k'_2. \delta \sigma_2 t'_1 \Downarrow^{k'_1, c'_1} v'_1 \wedge \delta \sigma_2 t'_2 \Downarrow^{k'_2, c'_2} v'_2 \wedge k''' = k'_1 + k'_2 + 1 \wedge c' = c'_1 + c'_2 + c_{\text{alloc}} \tag{b}$$

From (a),(b) and the evaluation rules E-alloc we get:

$$\exists l. (\text{alloc } t_1 t_2) \delta \sigma; H_1 \Downarrow_f^{k'', c} l; H_1, l \rightarrow [v_2, \dots v_2] \tag{c}$$

$$\exists l'. (\text{alloc } t'_1 t'_2) \delta \sigma; H_2 \Downarrow_f^{k''', c'} l'; H_2, l' \rightarrow [v'_2, \dots v'_2] \tag{d}$$

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

$$(\delta \sigma_1 t_1, \delta \sigma_2 t'_1) \in (\text{int}_r [I])_{G', k''}^{E, D_1} \tag{*}$$

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

$$(v_1, v'_1) \in (\text{int}_r [I])_{G', k'' - k_1} \Rightarrow v_1 = v'_1 = I \wedge c_1 - c'_1 \leq D_1 \tag{ih1}$$

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

$$(\delta \sigma_1 t_2, \delta \sigma_2 t'_2) \in (\tau)_{G', k' - k_1}^{E, D_2} \tag{**}$$

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

$$(v_2, v'_2) \in (\tau)_{G', k' - k_1 - k_2} = (\tau)_{G', k' - k'' + 1} \wedge c_2 - c'_2 \leq D_2 \tag{ih2}$$

Let us assume:

$$G'' = G'[r \rightarrow (l, l', \tau, D)] \quad (\text{e})$$

$$H_1' = H_1, l \rightarrow [v_2, v_2, \dots, v_2] \quad (\text{f})$$

$$H_2' = H_2, l' \rightarrow [v_2', v_2', \dots, v_2'] \quad (\text{g})$$

we want to show 3 cases.

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

TS2 $(H_1', H_2') \models_{G'', k' - k''} P \star \gamma \rightarrow \mathbb{N}$
 $G''(\gamma) = (l, l', \tau, D), H_1'' = l \rightarrow [v_2, v_2, \dots, v_2], H_2'' = l' \rightarrow [v_2', v_2', \dots, v_2']$

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 $\forall i \leq I, H_1''(l)[i] \neq H_2''(l')[i] \rightarrow i \in \beta = \mathbb{N}$
it is trivial

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

This proof is complete.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_1' \lesssim D_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t_2' \lesssim D_2 : \square\tau \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{alloc } t_1 \ t_2 \ominus \text{alloc } t_1' \ t_2' \lesssim 0 : \{P\} \exists \gamma. \text{Array}_\gamma[I] \ \square\tau \ \{P \star \gamma \rightarrow \emptyset\}} \text{R-ALLOC-BOX}^{\text{diff}(D_1+D_2)}$$

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

TS: $(\delta \sigma_1(\text{alloc } t_1 \ t_2), \delta \sigma_2(\text{alloc } t_1' \ t_2')) \in (\{P\} \exists \gamma. \text{Array}_\gamma[I] \ \tau \ \{P \star \gamma \rightarrow \emptyset\})_{G, k}^{E, 0}$

By unfolding the definition of $(\tau)_{G, k}^E$

STS: $(\delta \sigma_1(\text{alloc } t_1 \ t_2), \delta \sigma_2(\text{alloc } t_1' \ t_2')) \in (\{P\} \exists \gamma. \text{Array}_\gamma[I] \ \tau \ \{P \star \gamma \rightarrow \emptyset\})_{G, k}^{\text{diff}(D_1+D_2)}$

Unfold the definition of $(\{P\} \exists \gamma. \text{Array}_\gamma[I] \ \tau \ \{P \star \gamma \rightarrow \emptyset\})_{G, k}^{\text{diff}(D_1+D_2)}$

Pick $G' \supseteq G, H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$\delta \sigma_1(\text{alloc } t_1 \ t_2); H_1 \Downarrow_f^{k''} \quad (1)$$

$$\delta \sigma_2(\text{alloc } t_1' \ t_2'); H_2 \Downarrow_f^{k'''} \quad (2)$$

Because t_1, t_2 are sub terms of $\delta \sigma \text{ alloc } t_1 \ t_2, t_1', t_2'$ are sub terms of $\delta \sigma \text{ alloc } t_1' \ t_2'$
From (1)(2), we get

$$\delta \sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta \sigma_1 t_2 \Downarrow_f^{k_2} \quad (3)$$

$$\delta \sigma_2 t_1' \Downarrow_f \wedge \delta \sigma_2 t_2' \Downarrow_f \quad (4)$$

From (3),(4), we get:

$$\exists v_1, v_2, k_1, k_2. \delta \sigma_1 t_1 \Downarrow^{k_1, c_1} v_1 \wedge \delta \sigma_1 t_2 \Downarrow^{k_2, c_2} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_2 + c_{\text{alloc}} \quad (\text{a})$$

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

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

$$\exists l. (\text{alloc } t_1 \ t_2) \delta \sigma; H_1 \Downarrow_f^{k'', c} l; H_1, l \rightarrow [v_2, \dots, v_2] \quad (\text{c})$$

$$\exists l'. (\text{alloc } t_1' \ t_2') \delta \sigma; H_2 \Downarrow_f^{c'} l'; H_2, l' \rightarrow [v_2', \dots, v_2'] \quad (\text{d})$$

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

$$(\delta\sigma_1 t_1, \delta\sigma_2 t'_1) \in \langle \text{int}_r [I] \rangle_{G', k''}^{E, D_1} \quad (*)$$

Unfold the definition of $\langle \text{int}_r [I] \rangle_{G', k''}^E$ and using (a) and $k_1 \leq k''$, we know:

$$(v_1, v'_1) \in \langle \text{int}_r [I] \rangle_{G', k'' - k_1} \Rightarrow v_1 = v'_1 = I \wedge c_1 - c'_1 \leq D_1 \quad \text{ih1}$$

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

$$(\delta\sigma_1 t_2, \delta\sigma_2 t'_2) \in \langle \tau \rangle_{G', k' - k_1}^{E, D_2} \quad (**)$$

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

$$(v_2, v'_2) \in \langle \Box\tau \rangle_{G', k' - k_1 - k_2} = \langle \Box\tau \rangle_{G', k' - k'' + 1} \wedge c_2 - c'_2 \leq D_2 \Rightarrow v_2 = v'_2 \quad \text{ih2}$$

Let us assume:

$$G'' = G'[r \rightarrow (l, l', \tau, D)] \quad (\text{e})$$

$$H'_1 = H_1, l \rightarrow [v_2, v_2, \dots, v_2] \quad (\text{f})$$

$$H'_2 = H_2, l' \rightarrow [v'_2, v'_2, \dots, v'_2] \quad (\text{g})$$

we want to show 3 cases.

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

TS2 $(H'_1, H'_2) \models_{G'', k' - k''} P \star \gamma \rightarrow \phi$
 $G''(\gamma) = (l, l', \tau, D), H''_1 = l \rightarrow [v_2, v_2, \dots, v_2], H''_2 = l' \rightarrow [v'_2, v'_2, \dots, v'_2]$

STS1 $\forall i \leq I, (H''_1(l)[i], H''_2(l')[i]) \in \langle \tau \rangle_{G'', k' - k'' - 1}$
It is proved by using ih2 and Lemma 1.

STS2 $\forall i \leq I, H'_1(l)[i] \neq H'_2(l')[i] \rightarrow i \in \emptyset$
We know that $\forall i \leq I, H_1(l)[i] = v_2, H'_1(l')[i] = v'_2$ 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 $c - c' \leq (D_1 + D_2)$ from ih1 and ih2.

This proof is complete.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim D_1 : \tau_1 \xrightarrow{\text{diff}(D)} \tau_2 \quad \Sigma; \Delta; \Phi; \Gamma \vdash u_1 \ominus u_2 \lesssim D_2 : \tau_1}{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 u_1 \ominus t_2 u_2 \lesssim D + D_1 + D_2 : \tau_2} \text{R-APP}$$

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

TS: $(\delta\sigma_1(t_1 u_1), \delta\sigma_2(t_2 u_2)) \in \langle \delta\tau_2 \rangle_{G, k}^{E, (D + D_1 + D_2)}$.

Pick $k' \leq k, v_1, v_2$.

Assume

$$\delta\sigma_1(t_1 u_1) \Downarrow^{c, k'} v_1 \wedge \delta\sigma_2(t_2 u_2) \Downarrow^{c'} v_2 \quad (\text{a})$$

STS: $(v_1, v_2) \in \langle \delta\tau_2 \rangle_{G, k - k'} \wedge c - c' \leq (D + D_1 + D_2)$.

From (a) and the evaluation rule R-App, we know:

$$\begin{aligned} & \exists \dots (\delta\sigma_1 t_1) \Downarrow^{k_1, c_1} \text{fix } f(x). t'_1 \wedge (\delta\sigma_1 u_1) \Downarrow^{k'_1, c'_1} v'_1 \wedge \\ & t'_1 [\text{fix } f(x). t'_1 / f] [v'_1 / x] \Downarrow^{k''_1, c''_1} v_1 \wedge k' = k_1 + k'_1 + k''_1 + 1 \wedge c = c_1 + c'_1 + c''_1 + c_{app} \quad (1) \end{aligned}$$

$$\exists \dots (\delta\sigma_2 t_2) \Downarrow^{c_2} \text{fix } f(x). t'_2 \wedge (\delta\sigma_2 u_2) \Downarrow^{c'_2} v'_2 \wedge t'_2 [\text{fix } f(x). t'_2 / f] [v'_2 / x] \Downarrow^{c''_2} v_2 \wedge c = c_2 + c'_2 + c''_2 + c_{app} \quad (2)$$

By IH on the first premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta \Gamma \rangle_{G,k}$, we get:

$$(\delta \sigma_1 t_1, \delta \sigma_2 t_2) \in \langle \delta \tau_1 \longrightarrow \delta \tau_2 \rangle_{G,k}^{E, D_1} \quad (b)$$

By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta \Gamma \rangle_{G,k}$, we get:

$$(\delta \sigma_1 u_1, \delta \sigma_2 u_2) \in \langle \delta \tau_1 \rangle_{G,k}^{E, D_2} \quad (c)$$

Unfold the definition of (b), because $k_1 < k$, from (1),(1), we get:

$$(\text{fix } f(x). t'_1, \text{fix } f(x). t'_2) \in \langle \delta \tau_1 \longrightarrow \delta \tau_2 \rangle_{G, k-k_1}^{\text{diff}(D)} \wedge c_1 - c_2 \leq D_1 \quad (3)$$

Unfold the definition of (c), because $k'_1 < k$, from (1),(1), we get:

$$(v'_1, v'_2) \in \langle \delta \tau_1 \rangle_{G, k-k'_1} \wedge c_1 - c_2 \leq D_1 \quad (4)$$

We want to show $(v_1, v_2) \in \langle \delta \tau_2 \rangle_{G, k-k', D}$ and $c'_1 - c'_2 \leq D$.

Considering both the third statements in (1),(2), it is suffice to show:

STS: $(t'_1[\text{fix } f(x). t'_1 / f][v'_1 / x], t'_2[\text{fix } f(x). t'_2 / f][v'_2 / x]) \in \langle \sigma \rangle_{G, k-k'+k'_1}^E$

Use Lemma 1 on (3) with $k - k' + k'_1 \leq k - k_1$. we get:

$$(\text{fix } f(x). t'_1, \text{fix } f(x). t'_2) \in \langle \delta \tau_1 \longrightarrow \delta \tau_2 \rangle_{G, k-k'+k'_1}^{\text{diff}(D)} \quad (5)$$

Unfold (5).

Because $G \supseteq G$ and $(k - k' + k'_1) \leq k - k_1 - 1$

By using Lemma 1 on (4) and $(k - k' + k'_1) \leq k - k'_1$, we get :

$$(v'_1, v'_2) \in \langle \delta \tau_1 \rangle_{G, k-k'+k'_1} \quad (6)$$

Using (6), we know:

$$(t'_1[\text{fix } f(x). t'_1 / f][v'_1 / x], t'_2[\text{fix } f(x). t'_2 / f][v'_2 / x]) \in \langle \delta \tau_2 \rangle_{G, k-k'+k'_1}^{E, D} \quad (7)$$

Unfold (7) and we know: $c'_1 - c'_2 \leq D$

This completes the proof of case r-app.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Delta \models I' \leq I \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 t_2 \ominus \text{read } t'_1 t'_2 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists_{-} \tau \{P \star \gamma \rightarrow \beta\}} \text{R-R}^{\text{diff}(D_1+D_2)}$$

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

TS: $(\delta \sigma_1(\text{read } t_1 t_2), \delta \sigma_2(\text{read } t'_1 t'_2)) \in \langle \delta \{P \star \gamma \rightarrow \beta\} \exists_{-} \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}^{E, 0}$

Because $\text{read } t_1 t_2$ is value.

STS: $(\delta \sigma_1(\text{read } t_1 t_2), \delta \sigma_2(\text{read } t'_1 t'_2)) \in \langle \delta \{P \star \gamma \rightarrow \beta\} \exists_{-} \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}^{\text{diff}(D_1+D_2)}$

Unfold the definition of $\langle \delta \{P \star \gamma \rightarrow \beta\} \exists_{-} \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G', k'} P \star \gamma \rightarrow \beta \quad (1)$$

$$\delta \sigma_1(\text{read } t_1 t_2); H_1 \Downarrow_f^{k''} \quad (2)$$

$$\delta \sigma_2(\text{read } t'_1 t'_2); H_2 \Downarrow_f^{k''} \quad (3)$$

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

$$\delta\sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta\sigma_1 t_2 \Downarrow_f^{k_2} \quad (4)$$

$$\delta\sigma_2 t'_1 \Downarrow_f^{k'_1} \wedge \delta\sigma_2 t'_2 \Downarrow_f^{k'_2} \quad (5)$$

From (4),(5), we get:

$$\exists l, n, k_1, k_2. \delta\sigma_1 t_1 \Downarrow^{k_1, c_1} l \wedge \delta\sigma_1 t_2 \Downarrow^{k_2, c_1} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c'_1 + c_{read} \quad (a)$$

$$\exists l', n', k'_1, k'_2. \delta\sigma_2 t'_1 \Downarrow^{k'_1, c_2} l' \wedge \delta\sigma_2 t'_2 \Downarrow^{k'_2, c_2} n' \wedge k''' = k'_1 + k'_2 + 1 \wedge c' = c_2 + c'_2 + c_{read} \quad (b)$$

From (a),(b) and the evaluation rule we get:

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

$$\exists v'. \delta\sigma_2(\text{read } t'_1 t'_2); H_2 \Downarrow_f^{k''', c'} v'; H_2 \wedge H_2(l')[n'] = v' \quad (d)$$

By IH on the first premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta\Gamma \rangle_{G', k'}$ by lemma 1. we get:

$$(\delta\sigma_1 t_1, \delta\sigma_2 t'_1) \in \langle \text{Array}_\gamma[I] \delta\tau \rangle_{G', k'}^{E, D_1} \quad (6)$$

Unfold (6), since $k_1 \leq k'$ and $\delta\sigma_1 t_1 \Downarrow^{k_1} l$ and $\delta\sigma_2 t'_1 \Downarrow l'$, we know

$$(l, l') \in \langle \text{Array}_\gamma[I] \delta\tau \rangle_{G', k' - k_1} \wedge c_1 - c_2 \leq D_1 \quad (e)$$

From (e), we know :

$$G'(\gamma) = (l, l', \tau, D) \quad (7)$$

By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta\Gamma \rangle_{G', k'}$ by lemma 1. we get:

$$(\delta\sigma_1 t_2, \delta\sigma_2 t'_2) \in \langle \text{int}_r[I'] \rangle_{G', k'}^E \quad (8)$$

Unfold (8), since $k_2 \leq k'$, we know

$$(n, n') \in \langle \text{int}_r[I'] \rangle_{G', k' - k_2} \Rightarrow n = n' = I' \wedge c'_1 - c'_2 \leq D_2 \quad (f)$$

Let us assume:

$$G' = G' \quad (g)$$

STS1: $(H_1, H_2) \models_{G', k' - k''} P \star \gamma \rightarrow \beta$

By Lemma 7 and (1), this is proved.

STS2: $(v, v') \in \langle \tau \rangle_{G', k' - k''}$

from (c),(d). we know $H_1(l)[n] = v$ and $H_2(l')[n'] = v'$

based on (e) and (f), we know

Unfold (1), we know:

$$\forall i \leq n, (H_1(l)(i), H_2(l')(i)) \in \langle \tau \rangle_{G', k' - 1}$$

s.t we know

$$(H_1(l)[n], H_2(l')[n']) \in \langle \tau \rangle_{G', k' - 1}$$

Because $k' - k'' \leq k' - 1$,

By Lemma 1, We get: $(v, v') \in \langle \tau \rangle_{G', k' - k''}$

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

This completes the proof of case read.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta \models I' \leq I \quad \Sigma; \Delta; \Phi \models I' \notin \beta \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 \ t_2 \ominus \text{read } t'_1 \ t'_2 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \square \tau \{P \star \gamma \rightarrow \beta\}} \text{R-RB}$$

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

TS: $(\delta \sigma_1(\text{read } t_1 \ t_2), \delta \sigma_2(\text{read } t'_1 \ t'_2)) \in \langle \delta \{P \star \gamma \rightarrow \beta\} \exists _ . \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}^{E,0}$

Because $\text{read } t_1 \ t_2$ is value.

STS: $(\delta \sigma_1(\text{read } t_1 \ t_2), \delta \sigma_2(\text{read } t'_1 \ t'_2)) \in \langle \delta \{P \star \gamma \rightarrow \beta\} \exists _ . \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}$

Unfold the definition of $\langle \delta \{P \star \gamma \rightarrow \beta\} \exists _ . \tau \{P \star \gamma \rightarrow \beta\} \rangle_{G,k}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G',k'} P \star \gamma \rightarrow \beta \quad (1)$$

$$\delta \sigma_1(\text{read } t_1 \ t_2); H_1 \Downarrow_f^{k''} \quad (2)$$

$$\delta \sigma_2(\text{read } t'_1 \ t'_2); H_2 \Downarrow_f^{k''} \quad (3)$$

Because t_1, t_2 are sub terms of $\text{read } t_1 \ t_2$, t'_1, t'_2 are sub terms of $\text{read } t'_1 \ t'_2$

From (2)(3), we get

$$\delta \sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta \sigma_1 t_2 \Downarrow_f^{k_2} \quad (4)$$

$$\delta \sigma_2 t'_1 \Downarrow_f^{k'_1} \wedge \delta \sigma_2 t'_2 \Downarrow_f^{k'_2} \quad (5)$$

From (4),(5), we get:

$$\exists l, n, k_1, k_2. \delta \sigma_1 t_1 \Downarrow_f^{k_1, c_1} l \wedge \delta \sigma_1 t_2 \Downarrow_f^{k_2, c'_1} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c'_1 + c_{read} \quad (a)$$

$$\exists l', n', k'_1, k'_2. \delta \sigma_2 t'_1 \Downarrow_f^{k'_1, c_2} l' \wedge \delta \sigma_2 t'_2 \Downarrow_f^{k'_2, c'_2} n' \wedge k''' = k'_1 + k'_2 + 1 \wedge c' = c_2 + c'_2 + c_{read} \quad (b)$$

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

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

$$\exists v'. H_2; \delta \sigma_2(\text{read } t'_1 \ t'_2) \Downarrow_f^{k''', c'} v'; H_2 \wedge H_2(l')[n'] = v' \quad (d)$$

By IH on the first premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta \Gamma \rangle_{G',k'}$ by lemma 1. we get:

$$(\delta \sigma_1 t_1, \delta \sigma_2 t'_1) \in \langle \text{Array}_\gamma[I] \delta \tau \rangle_{G',k'}^{E, D_1} \quad (6)$$

Unfold (6), since $k_1 \leq k'$ and $\delta \sigma_1 t_1 \Downarrow_f^{k_1} l$ and $\delta \sigma_2 t'_1 \Downarrow_f^{k'_1} l'$, we know

$$(l, l') \in \langle \text{Array}_\gamma[I] \delta \tau \rangle_{G',k'-k_1} \wedge c_1 - c_2 \leq D_1 \quad (e)$$

From (e), we know :

$$G'(\gamma) = (l, l', \tau, I) \quad (7)$$

By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in \langle \delta \Gamma \rangle_{G',k'}$ by lemma 1. we get:

$$(\delta \sigma_1 t_2, \delta \sigma_2 t'_2) \in \langle \delta \text{int}_\tau[I'] \rangle_{G',k'}^E \quad (8)$$

Unfold (8), since $k_2 \leq k'$, we know

$$(n, n') \in (\text{int}_r[I'])_{G', k' - k_2} \Rightarrow n = n' = I' \wedge c'_1 - c'_2 \leq D_2 \quad (\text{f})$$

Let us assume:

$$G' = G' \quad (\text{g})$$

STS1: $(H_1, H_2) \models_{G', k' - k''} P \star \gamma \rightarrow \beta$

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), we know

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: $(v, v') \in (\tau)_{G', k' - k''}$, consider the premise $i \notin \beta$, we know $v = v'$, so that we prove $(v, v') \in (\Box\tau)_{G', k' - k''}$.

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

This completes the proof of case read-box.

$$\text{CASE} \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_3 \ominus t'_3 \lesssim D_3 : \tau \quad \Delta; \Phi \models I' \leq I \quad \Sigma; \Delta \vdash P \quad w_f}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{updt } t_1 \ t_2 \ t_3 \ominus \text{updt } t'_1 \ t'_2 \ t'_3 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{P \star \gamma \rightarrow \beta \cup \{I'\}\}} \text{R-U}$$

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

TS: $(\delta\sigma_1(\text{updt } t_1 \ t_2 \ t_3), \delta\sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3)) \in (\{\gamma \rightarrow \beta\} \exists_- . \text{unit } \{\gamma \rightarrow \beta \cup \{I'\}\})_{G, k}^{E, 0}$

Because $\text{updt } t_1 \ t_2 \ t_3$ is value.

STS: $(\delta\sigma_1(\text{updt } t_1 \ t_2 \ t_3), \delta\sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3)) \in (\{\gamma \rightarrow \beta\} \exists_- . \text{unit } \{\gamma \rightarrow \beta \cup \{I'\}\})_{G, k}$

Unfold the definition of $(\{\gamma \rightarrow \beta\} \exists_- . \text{unit } \{\gamma \rightarrow \beta \cup \{I'\}\})_{G, k}$.

Pick $G' \ni G, H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G', k'} P \star \gamma \rightarrow \beta \wedge H_1 = H_{p1} \uplus H_{l1} \wedge H_{l1} = l \rightarrow [v, \dots, v] \quad (1)$$

$$\wedge H_2 = H_{p2} \uplus H_{l2} \wedge H_{l2} = l' \rightarrow [v', \dots, v'] \quad (1)$$

$$\delta\sigma_1(\text{updt } t_1 \ t_2 \ t_3); H_1 \Downarrow_f^{k''} \quad (2)$$

$$\delta\sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3); H_2 \Downarrow_f^{k'''} \quad (3)$$

Because t_1, t_2, t_3 are sub terms of $\text{updt } t_1 \ t_2 \ t_3$, t'_1, t'_2, t'_3 are sub terms of $\text{updt } t'_1 \ t'_2 \ t'_3$
From (2)(3), we get

$$\delta\sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta\sigma_1 t_2 \Downarrow_f^{k_2} \wedge \delta\sigma_1 t_3 \Downarrow_f^{k_3} \quad (4)$$

$$\delta\sigma_2 t'_1 \Downarrow_f^{k'_1} \wedge \delta\sigma_2 t'_2 \Downarrow_f^{k'_2} \wedge \delta\sigma_2 t'_3 \Downarrow_f^{k'_3} \quad (5)$$

From (4),(5), we get:

$$\exists l, n, v, k_1, k_2, k_3. \delta\sigma_1 t_1 \Downarrow^{k_1, c_1} l \wedge \delta\sigma_1 t_2 \Downarrow^{k_2, c'_1} n \wedge \delta\sigma_1 t_3 \Downarrow^{k_3, c''_1} v \wedge$$

$$k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c'_1 + c''_1 + c_{update} \quad (\text{a})$$

$$\exists l', n', v', k'_1, k'_2, k'_3. \delta\sigma_2 t'_1 \Downarrow^{k'_1, c_2} l' \wedge \delta\sigma_2 t'_2 \Downarrow^{k'_2, c'_2} n' \wedge \delta\sigma_2 t'_3 \Downarrow^{k'_3, c''_2} v' \wedge c' = c_2 + c'_2 + c''_2 + c_{update} \quad (\text{b})$$

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

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

$$\text{updt } t'_1 \ t'_2 \ t'_3; H_2 \Downarrow^{k''', c'} (); H_2(l')[n'] \leftarrow v' \quad (\text{d})$$

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

$$(\delta\sigma_1 t_1, \delta\sigma_2 t'_1) \in \langle \text{Array}_\gamma[I] \ \delta\tau \rangle_{G', k'}^{E, D_1} \quad (\text{6})$$

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

$$(\delta\sigma_1 t_2, \delta\sigma_2 t'_2) \in \langle \text{int}_r[I] \rangle_{G', k'}^{E, D_2} \quad (\text{7})$$

$$(\delta\sigma_1 t_3, \delta\sigma_2 t'_3) \in \langle \tau \rangle_{G', k'}^{E, D_3} \quad (\text{8})$$

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

$$(l, l') \in \langle \text{Array}_\gamma[I] \ \tau \rangle_{G', k' - k_1} \Rightarrow G'(\gamma) = (l, l', \tau, I) \wedge c_1 - c_2 \leq D_1 \quad (\text{e})$$

$$(n, n') \in \langle \text{int}_r[I] \rangle_{G', k' - k_2} \Rightarrow n = n' = I \wedge c'_1 - c'_2 \leq D_2 \quad (\text{f})$$

$$(v, v') \in \langle \tau \rangle_{G', k' - k_3} \wedge c''_1 - c''_2 \leq D_3 \quad (\text{g})$$

Let us assume:

$$G'' = G' \quad (\text{9})$$

$$H'_1 = H_1(l)[n] \leftarrow v \wedge H'_2 = H_2(l')[n'] \leftarrow v' \quad (\text{10})$$

STS1: $(H'_1, H'_2) \models_{G', k' - k''} P \star \gamma \rightarrow \beta \cup \{I'\}$

Unfold (1) and consider (e), we know

$$\forall i \leq I. (H_{11}(l)[i], H_{12}(l')[i]) \in \langle \tau \rangle_{G', k' - 1} \quad (\text{11})$$

$$\forall i \leq I. (H_{11}(l)[i] \neq H_{12}(l')[i]) \Rightarrow i \in \beta \quad (\text{12})$$

Since (e), we need to prove two sub cases.

subgoal 1: $\forall i \leq I. (H'_{11}(l)[i], H'_{12}(l')[i]) \in \langle \tau \rangle_{G', k' - k''}$

when $i=n$,

we prove $(H'_{11}(l)[i], H'_{12}(l')[i]) \in \langle \tau \rangle_{G', k' - k_3}$ from (g) and (10), and then use lemma 1.

when $i \neq n$,

proved by using lemma 1 on (11)

subgoal 2: $\forall i \leq I. (H'_{11}(l)[i] \neq H'_{12}(l')[i]) \Rightarrow i \in \beta \cup \{I'\}$

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: $(0, 0) \in \langle \text{unit}_r \rangle_{G', k' - k''}$

It is proved by unfolding the definition.

STS3 $c - c' \leq D_1 + D_2 + D_3$.

It is proved by (e),(f),(g).

This completes the proof of case update.

$$\text{CASE} \frac{\Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma [I] \tau \quad \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Delta; \Phi; \Gamma \vdash t_3 \ominus t'_3 \lesssim D_3 : \square \tau \quad \Delta; \Phi \models I' \leq I \quad \Sigma; \Delta \vdash P \quad wf}{\Delta; \Phi; \Gamma \vdash \text{updt } t_1 \ t_2 \ t_3 \ominus \text{updt } t'_1 \ t'_2 \ t'_3 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{P \star \gamma \rightarrow \beta \setminus \{I'\}\}} \text{R-UB}$$

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

TS: $(\delta \sigma_1(\text{updt } t_1 \ t_2 \ t_3), \delta \sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3)) \in \langle \{P \star \gamma \rightarrow \beta\} \exists_- \text{unit} \{P \star \gamma \rightarrow \beta \setminus \{I'\}\} \rangle_{G,k}^{E,0}$

Because $\text{updt } t_1 \ t_2 \ t_3$ is value.

STS: $(\delta \sigma_1(\text{updt } t_1 \ t_2 \ t_3), \delta \sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3)) \in \langle \{P \star \gamma \rightarrow \beta\} \exists_- \text{unit} \{P \star \gamma \rightarrow \beta \setminus \{I'\}\} \rangle_{G,k}^{\text{diff}(D_1+D_2+D_3)}$

Unfold the definition of $\langle \{P \star \gamma \rightarrow \beta\} \exists_- \text{unit} \{P \star \gamma \rightarrow \beta \setminus \{I'\}\} \rangle_{G,k}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G',k'} P \star \gamma \rightarrow \beta \wedge H_1 = H_{p1} \uplus H_{l1} \wedge H_{l1} = l \rightarrow [v, \dots, v] \wedge H_2 = H_{p2} \uplus H_{l2} \wedge H_{l2} = l' \rightarrow [v', \dots, v'] \quad (1)$$

$$\delta \sigma_1(\text{updt } t_1 \ t_2 \ t_3); H_1 \Downarrow_f^{k''} \quad (2)$$

$$\delta \sigma_2(\text{updt } t'_1 \ t'_2 \ t'_3); H_2 \Downarrow_f^{k'''} \quad (3)$$

Because t_1, t_2, t_3 are sub terms of $\text{updt } t_1 \ t_2 \ t_3$, t'_1, t'_2, t'_3 are sub terms of $\text{updt } t'_1 \ t'_2 \ t'_3$
From (2)(3), we get

$$\delta \sigma_1 t_1 \Downarrow_f^{k_1} \wedge \delta \sigma_1 t_2 \Downarrow_f^{k_2} \wedge \delta \sigma_1 t_3 \Downarrow_f^{k_3} \quad (4)$$

$$\delta \sigma_2 t'_1 \Downarrow_f^{k'_1} \wedge \delta \sigma_2 t'_2 \Downarrow_f^{k'_2} \wedge \delta \sigma_2 t'_3 \Downarrow_f^{k'_3} \quad (5)$$

From (4),(5), we get:

$$\exists l, n, v, k_1, k_2, k_3. \delta \sigma_1 t_1 \Downarrow^{k_1, c_1} l \wedge \delta \sigma_1 t_2 \Downarrow^{k_2, c'_1} n \wedge \delta \sigma_1 t_3 \Downarrow^{k_3, c''} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c'_1 + c'' + c_{\text{update}} \quad (a)$$

$$\exists l', n', v', k'_1, k'_2, k'_3. \delta \sigma_2 t'_1 \Downarrow^{k'_1, c_2} l' \wedge \delta \sigma_2 t'_2 \Downarrow^{k'_2, c'_2} n' \wedge \delta \sigma_2 t'_3 \Downarrow^{k'_3, c''} v' \wedge c' = c_2 + c'_2 + c'' + c_{\text{update}} \quad (b)$$

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

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

$$\text{updt } t'_1 \ t'_2 \ t'_3; H_2 \Downarrow^{k''', c'} (); H_2(l')[n'] \leftarrow v' \quad (d)$$

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

$$(\delta \sigma_1 t_1, \delta \sigma_2 t'_1) \in \langle \text{Array}_\gamma [I] \delta \tau \rangle_{G',k'}^{E, D_1} \quad (6)$$

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

$$(\delta \sigma_1 t_2, \delta \sigma_2 t'_2) \in \langle \text{int}_r [I] \rangle_{G',k'}^{E, D_2} \quad (7)$$

$$(\delta \sigma_1 t_3, \delta \sigma_2 t'_3) \in \langle \square \delta \tau \rangle_{G',k'}^{E, D_3} \quad (8)$$

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

$$(l, l') \in \langle \text{Array}_\gamma [I] \tau \rangle_{G', k' - k_1} \Rightarrow G'(\gamma) = (l, l', \tau, I) \wedge c_1 - c_2 \leq D_1 \quad (\text{e})$$

$$(n, n') \in \langle \text{int}_r [I] \rangle_{G, k' - k_2} \Rightarrow n = n' = I \wedge c'_1 - c'_2 \leq D_2 \quad (\text{f})$$

$$v = v' \wedge (v, v') \in \langle \tau \rangle_{G', k' - k_3} c''_1 - c''_2 \leq D_3 \quad (\text{g})$$

Let us assume:

$$G'' = G' \quad (\text{9})$$

$$H'_1 = H_1(l)[n] \leftarrow v \wedge H'_2 = H_2(l')[n'] \leftarrow v' \quad (\text{10})$$

STS1: $(H'_1, H'_2) \models_{G', k' - k''} P \star \gamma \rightarrow \beta / \{I'\}$

Unfold (1) and consider (e), we know

$$\forall i \leq I. (H_{l_1}(l)[i], H_{l_2}(l')[i]) \in \langle \delta \tau \rangle_{G', k' - 1} \quad (\text{11})$$

$$\forall i \leq I. (H_{l_1}(l)[i] \neq H_{l_2}(l')[i]) \Rightarrow i \in \beta \quad (\text{12})$$

Since (e), we need to prove two sub cases.

subgoal 1: $\forall i \leq I. (H'_{l_1}(l)[i], H'_{l_2}(l')[i]) \in \langle \tau \rangle_{G', k' - k''}$

when $i=n$,

we prove $(H'_{l_1}(l)[i], H'_{l_2}(l')[i]) \in \langle \tau \rangle_{G', k' - k_3}$ from (g) and (10), and then use lemma 1.

when $i \neq n$,

proved by using lemma 1 on (11)

subgoal 2: $\forall i \leq I. (H'_{l_1}(l)[i] \neq H'_{l_2}(l')[i]) \Rightarrow i \in \beta / \{I'\}$

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: $(0, 0) \in \langle \text{unit}_r \rangle_{G', k' - k''}$

It is proved by unfolding the definition.

STS3 $c - c' \leq D_1 + D_2 + D_3$.

It is proved by (e),(f),(g).

This completes the proof of case update-box.

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

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

TS: $(\delta \sigma_1(\text{return } t_1), \delta \sigma_2(\text{return } t_2)) \in \langle \{P\} \exists_{\tau} \{P\} \rangle_{G, k}^{E, 0, \text{diff}(D)}$

Because return t is value,

STS: $(\delta \sigma_1(\text{return } t_1), \delta \sigma_2(\text{return } t_2)) \in \langle \{P\} \exists_{\tau} \{P\} \rangle_{G, k}^{\text{diff}(D)}$

Unfold the definition of $\langle \{P\} \exists_{\tau} \{P\} \rangle_{G, k}^{\text{diff}(D)}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k, k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G', k'} P \quad (\text{1})$$

$$\delta \sigma_1(\text{return } t_1); H_1 \Downarrow_f^{k''} \quad (\text{2})$$

$$\delta \sigma_2(\text{return } t_2); H_2 \Downarrow_f^{k'''} \quad (\text{3})$$

Because t_1 is the subterm of return t_1 , From (2)(3), We know:

$$\delta\sigma_1 t_1 \Downarrow^{k_1} \quad (4)$$

$$\delta\sigma_2 t_2 \Downarrow_f^{k'''} \quad (5)$$

From (4),(5), we know:

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

$$\exists v_2, k'_1, \delta\sigma_2 t_2 \Downarrow^{k'_1, c_2} v_2 \wedge k''' = k'_1 + 1 \wedge c' = c_2 + c_{ret} \quad (b)$$

From (a),(b) and the evaluation rule R-ret:

$$H_1; \text{return } t_1 \Downarrow_f^{k'', c} v_1; H_1 \quad (c)$$

$$H_2; \text{return } t_2 \Downarrow_f^{k''', c'} v_2; H_2 \quad (d)$$

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

$$(\delta\sigma_1 t_1, \delta\sigma_2 t_2) \in \langle \delta\tau \rangle_{G', k'}^{E, D} \quad (6)$$

Unfold (6), we get:

$$(v_1, v_2) \in \langle \delta\tau \rangle_{G', k' - k_1} \quad (7)$$

Let us assume:

$$G'' = G' \quad (8)$$

$$H'_1 = H'_1 \wedge H'_2 = H'_2 \leftarrow v' \quad (9)$$

STS1: $(H'_1, H'_2) \models_{G', k' - k''} P$
it is trivially true.

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

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

This completes the proof of case r-ret.

$$\text{CASE } \frac{\Delta; \Phi; \Gamma \vdash t_1 \ominus t_2 \lesssim_{D_1} \{P_1\} \exists \vec{\gamma}_1. \tau_1 \{Q_1 \star Q_2\} \quad \Delta, \vec{\gamma}_1 : \vec{\mathbb{L}}; \Phi; \Gamma, x : \tau_1 \vdash t'_1 \ominus t'_2 \lesssim_{D_2} \{Q_1 \star P_2\} \exists \vec{\gamma}_2. \tau_2 \{Q\}}{\Delta; \Phi; \Gamma \vdash \text{let } \{x\} = t_1 \text{ in } t'_1 \ominus \text{let } \{x\} = t_2 \text{ in } t'_2 \lesssim_{D_1 + D_2} \{P\} \exists \vec{\gamma}_1 \vec{\gamma}_2 : \tau_2 \{Q \star Q_2\}} \text{R-LET}^{\text{diff}(D+D')}$$

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

TS: $(\delta\sigma_1(\text{let } \{x\} = t_1 \text{ in } t'_1), \delta\sigma_2(\text{let } \{x\} = t_2 \text{ in } t'_2)) \in \langle \delta \{P\} \exists \vec{\gamma}_1 \vec{\gamma}_2 : \tau_2 \{Q \star Q_2\} \rangle_{G, k}^{E, (D_1 + D_2)}$

Because $\text{let } \{x\} = t_1 \text{ in } t'_1$ is value.

STS: $(\delta\sigma_1(\text{let } \{x\} = t_1 \text{ in } t'_1), \delta\sigma_2(\text{let } \{x\} = t_2 \text{ in } t'_2)) \in \langle \delta \{P\} \exists \vec{\gamma}_1 \vec{\gamma}_2 : \tau_2 \{Q \star Q_2\} \rangle_{G, k}^{\text{diff}(D+D')}$

Unfold the definition of $\langle \delta \{P\} \exists \vec{\gamma}_1 \vec{\gamma}_2 : \tau_2 \{Q \star Q_2\} \rangle_{G, k}^{\text{diff}(D+D')}$.

Pick $G' \supseteq G$, $H_1, H_2, k' \leq k$, $k'' < k', k'''$.

Assume

$$(H_1, H_2) \models_{G', k'} P \quad (1)$$

$$\delta\sigma_1(\text{let } \{x\} = t_1 \text{ in } t'_1); H_1 \Downarrow_f^{k''} \quad (2)$$

$$\delta\sigma_2(\text{let } \{x\} = t_2 \text{ in } t'_2); H_2 \Downarrow_f^{k'''} \quad (3)$$

Because t_1 is the subterm of $\text{let } \{x\} = t_1 \text{ in } t'_1$, From (2), we know:

$$\exists v_1, v'_1, v''_1, v'''_1, k_1, k_2, k_3, k_4, H'_1, H''_1. \delta\sigma_1 t_1 \Downarrow^{k_1, c_1} v_1 \wedge v_1; H_1 \Downarrow_f^{k_2, c'_1} v'_1; H'_1 \\ \delta\sigma_1 t'_1[v'_1/x] \Downarrow^{k_3, c_3} v''_1 \wedge v''_1; H'_1 \Downarrow_f^{k_4, c_4} v'''_1; H''_1 \wedge k'' = k_1 + k_2 + k_3 + k_4 + 1 \wedge c = c_1 + c_2 + c_3 + c_4 + c_{let} \quad (\text{a})$$

Similarly, from (3), we get:

$$\exists v_2, v'_2, v''_2, v'''_2, k'_1, k'_2, k'_3, k'_4, H'_2, H''_2. \delta\sigma_1 t_2 \Downarrow^{k'_1, c'_2} v_2 \text{ and } v_2; H_2 \Downarrow_f^{k'_2, c'_2} v'_2; H'_2 \text{ and} \\ \delta\sigma_1 t'_2[v'_2/x] \Downarrow^{k'_3, c'_2} v''_2 \text{ and } v''_2; H'_2 \Downarrow_f^{k'_4, c''_2} v'''_2; H''_2 \wedge k'' = k'_1 + k'_2 + k'_3 + k'_4 + 1 \wedge c' = c_2 + c'_2 + c''_2 + c'''_2 + c_{let} \quad (\text{b})$$

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

$$\exists H_a, H_c, H'_a, H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v_1; H_a \Downarrow_f v'_1; H'_a \wedge H'_a = H1_{Q1} \uplus H1_{Q2} \\ \wedge v''_1; H1_{Q1} \uplus H_c \Downarrow_f v'''_1; H1_Q \wedge H''_1 = H1_Q \uplus H1_{Q2} \quad (\text{c})$$

$$\exists H_a, H_c, H'_a, H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v_1; H_a \Downarrow_f v'_1; H'_a \wedge H'_a = H1_{Q1} \uplus H1_{Q2} \\ \wedge v''_1; H1_{Q1} \uplus H_c \Downarrow_f v'''_1; H1_Q \wedge H''_1 = H1_Q \uplus H1_{Q2} \quad (\text{d})$$

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

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

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: $c - c' \leq (D_1 + D_2 + D + D')$

By IH on the second premise instantiated with $(\sigma_1, \sigma_2) \in (\delta\Gamma)_{G', k'}$, we know:

$$(\delta\sigma_1 t_1, \delta\sigma_2 t_2) \in (\{P_1\} \exists \tilde{\gamma}_1. \tau \{Q_1 \star Q_2\})_{G', k'}^{diff(D), E, D_1} \quad (4)$$

Unfold the definition of $(\{P_1\} \exists \tilde{\gamma}_1. \tau \{Q_1 \star Q_2\})_{G', k'}^{diff(D), E, D_1}$, use (a), we get:

$$(v_1, v_2) \in (\{P_1\} \exists \tilde{\gamma}_1. \tau_1 \{Q_1 \star Q_2\})_{G', k' - k_1}^{diff(D)} \wedge c_1 - c_2 \leq D_1 \quad (\text{e})$$

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:

$$v_1; H_a \Downarrow_f^{c'_1} v'_1; H'_a \wedge v_2; H_b \Downarrow_f^{c'_2} v'_2; H'_b \wedge H'_a, H'_b \vDash Q_1 \star Q_2$$

$$(v'_1, v'_2) \in (\tau_1)_{G', k' - k_1} \wedge c'_1 - c'_2 \leq D \quad (\text{k})$$

By IH on the third premise, we know:

$$(\delta\sigma_1 v''_1, \delta\sigma_2 v''_2) \in (\{Q_1 \star P_2\} \exists \tilde{\gamma}_1. \tau \{Q\})_{G', k'}^{diff(D'), E, D_2} \quad (5)$$

Unfold the definition of $(\{Q_1 \star P_2\} \exists \tilde{\gamma}_1. \tau_2 \{Q\})_{G', k'}^{diff(D), E, D_2}$, use (a), we get:

$$(v_1, v_2) \in (\{Q_1 \star P_2\} \exists \tilde{\gamma}_1. \tau_2 \{Q\})_{G', k' - k_1}^{diff(D')} \wedge c''_1 - c''_2 \leq D_2 \quad (\text{f})$$

We have

$$(H'_a, H'_b) \models_{G_f, k_f} Q_1 \star Q_2 \Rightarrow H1_{Q1}, H2_{Q1} \models Q_1 \wedge H1_{Q2}, H2_{Q2} \models Q_2 \wedge H_c, H_d \models P_2$$

$$H1_{Q1} \uplus H_c; v_1''' \Downarrow_f \wedge H2_{Q1} \uplus H_d; v_2''' \Downarrow_f$$

Unfold the definition (f), we get:

$$H1_{Q1} \uplus H_c; v_1''' \Downarrow_f^{c_1'''} H1_Q; v_1''' \wedge H2_{Q1} \uplus H_d; v_2''' \Downarrow_f^{c_2'''} H2_Q; v_2''' \wedge H1_Q, H2_Q \models Q$$

$$(v_1''', v_2''') \in \langle \tau_2 \rangle_{G', k' - k_1} \wedge c_1''' - c_2''' \leq D' \quad (\text{g})$$

STS1: Use Lemma 2, we know that $H1_Q \uplus H1_{Q2}, H2_Q \uplus H2_{Q2} \models Q \star Q_2$, which proves the
 STS2: it is proved from (g).

STS3: it is proved from (e),(k),(f),(g).

This completes the proof of let case.

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

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

TS: $(\delta\sigma_1(t_1), \delta\sigma_2(t_2)) \in \langle \delta U(A_1, A_2) \rangle_{G, k, \delta(U_1 - L_2)}^{E, (\delta(U_1 - L_2))}$

Unfold the definition.

Assume:

$$(a) \delta\sigma_1 t_1 \Downarrow^{k_1, c} v_1$$

$$(b) \delta\sigma_2 t_2 \Downarrow^{k_2, c'} v_2$$

$$(c) k_1 < k$$

TS 1: $(v_1, v_2) \in \langle U(\delta A_1, \delta A_2) \rangle_{G, k - k_1}$

TS 2: $c - c' \leq U_1 - L_2$

We prove TS 1 first.

Unfold its definition,

STS: $\forall k'. v_1 \in \langle \delta A_1 \rangle_{G1, k'}$ and $v_2 \in \langle \delta A_2 \rangle_{G2, k'}$

Pick k' .

By IH 2 on the first premise using

1. $FV(t_1) \subseteq \text{dom}(\delta|\Gamma|_1)$ using Lemma 6.
2. $\models \delta\Phi$
3. $\forall k. \sigma_1 \in \langle \delta|\Gamma|_1 \rangle_{G1, k}$

We get:

$$\forall k. \delta\sigma_1 t_1 \in \llbracket \delta A_1 \rrbracket_{G1, k}^{E, (\delta L_1, \delta U_1)} \quad (1)$$

Unfold the definition of (1), choose $k = k_1 + k'$. we know

$$v_1 \in \llbracket \delta A_1 \rrbracket_{G1, k'} \wedge \delta L_1 \leq c \leq \delta U_1 \quad (a)$$

By IH 2 on the second premise using

1. $FV(t_2) \subseteq \text{dom}(\delta|\Gamma|_2)$ using Lemma 6.

2. $\models \delta\Phi$

3. $\forall k. \sigma_2 \in \langle \delta|\Gamma|_2 \rangle_{G|_2, k}$

We get:

$$\forall k. \delta\sigma_2 t_2 \in \llbracket \delta A_2 \rrbracket_{G|_2, k}^{E, (\delta L_2, \delta U_2)} \quad (2)$$

Unfold the definition of (1), choose $k = k_2 + k'$. we know

$$v_2 \in \llbracket \delta A_2 \rrbracket_{G|_2, k'} \wedge \delta L_2 \leq c' \leq \delta U_2 \quad (b)$$

(a),(b) finished the proof of STS1.

(a),(b) finished the proof of STS2.

This completes the proof of switch case.

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

$$\text{CASE } \frac{\xi; \Delta; \Phi; x : A, f : A \xrightarrow{\text{exec}(L,U)} B, \Omega \vdash_L^U t : B}{\Delta; \Phi; \Omega \vdash_0^0 \text{fix } f(x).t : A \xrightarrow{\text{exec}(L,U)} B} \text{ U-F}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta\Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \llbracket \delta\Omega' \rrbracket_{g,k}$

TS: $(\delta\sigma \text{fix } f(x).t) \in \llbracket \delta A \xrightarrow{\text{exec}(L,U)} B \rrbracket_{g,k}^{E,(0,0)}$

Because $\text{fix } f(x).e$ is value.

STS: $(\delta\sigma \text{fix } f(x).t) \in \llbracket \delta A \xrightarrow{\text{exec}(L,U)} B \rrbracket_{g,k}$

We prove the more general statement

$$\forall k' \leq k. \delta\sigma(\text{fix } f(x).t) \in \llbracket \delta A \xrightarrow{\text{exec}(L,U)} \delta B \rrbracket_{g,k'} \quad (1)$$

by sub-induction on k' . There are two cases:

subcase 1: $k'=0$

there is no non-negative k such that $k \leq 0$, the goal is vacuously true.

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

By sub-IH, we know:

$$\delta\sigma(\text{fix } f(x).t) \in \llbracket \delta A \xrightarrow{\text{exec}(L,U)} \delta B \rrbracket_{g,k''} \quad (2)$$

STS: $(\text{fix } f(x).\delta\sigma e) \in \llbracket \delta A \xrightarrow{\text{exec}(L,U)} \delta B \rrbracket_{g,k''+1}$

Pick $j < k''+1$ and assume that $v \in \llbracket \delta A \rrbracket_{g,j}$

STS: $\sigma[v/x][\text{fix } f(x).e/f] \in \llbracket \delta B \rrbracket_{g,j}^{E,(\delta L, \delta U)}$

This follows by IH on the premise instantiated with

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

This completes the proof of unary fix case.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{int}[I] \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : A \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \text{alloc } t_1 t_2 : \{P\} \exists \gamma : \text{Array}_\gamma[I] A \{P \star \gamma \rightarrow \mathbb{N}\}} \text{ U-ALLOC}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta\Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \llbracket \delta\Omega' \rrbracket_{g,k}$

TS: $(\delta\sigma \text{alloc } t_1 t_2) \in \llbracket \{P\} \exists \gamma : \text{Array}_\gamma[I] A \{P \star \gamma \rightarrow \mathbb{N}\} \rrbracket_{g,k}^{E,(0,0)}$

Because $\text{alloc } t_1 t_2$ is value.

STS: $(\delta\sigma \text{alloc } t_1 t_2) \in \llbracket \{P\} \exists \gamma : \text{Array}_\gamma[I] A \{P \star \gamma \rightarrow \mathbb{N}\} \rrbracket_{g,k}$

Unfold the definition of $\llbracket \{P\} \exists \gamma : \text{Array}_\gamma [I] A \{P\gamma \rightarrow \mathbb{N}\} \rrbracket_{g,k}$.

Pick $g' \supseteq g$, $k' \leq k$, $k'' < k'$ and H .

Assume $H \models_{g',k'} P$ 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 $\delta\sigma \text{ alloc } t_1 t_2$, we get

$$\delta\sigma t_1 \Downarrow_f^{k_1} \wedge \delta\sigma t_2 \Downarrow_f^{k_2} \quad (1)$$

From (1), we get:

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

From (a) and evaluation rules we get:

$$\exists l. \delta\sigma(\text{alloc } t_1 t_2); H \Downarrow_f^{k'', c} l; H[l \rightarrow [v_2, \dots, v_2]] \quad (b)$$

By IH on the first premise instantiated with $\sigma \in \llbracket \delta\Omega' \rrbracket_{g,k''}$ using lemma 1 and $FV(t_1) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$, we get:

$$(\delta\sigma t_1) \in \llbracket \text{int}_r [I] \rrbracket_{g',k''}^{E, (\delta L_1, \delta U_1)} \quad (*)$$

Unfold the definition of $\llbracket \text{int}[I] \rrbracket_{g',k''}^{E, (\delta L_1, \delta U_1)}$ and using (a) and $k_1 \leq k''$, we know:

$$(v_1) \in \llbracket \text{int}[I] \rrbracket_{G', k'' - k_1} \Rightarrow v_1 = I \wedge \delta L_1 \leq c_1 \leq \delta U_1 \quad \text{IH1}$$

By IH on the second premise instantiated with $(\sigma) \in \llbracket \delta\Gamma \rrbracket_{g',k' - k_1}$ using lemma 1 and $FV(t_2) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$, we get:

$$(\delta\sigma t_2) \in \llbracket \delta A \rrbracket_{g',k' - k_1}^{E, (\delta L_2, \delta U_2)} \quad (**)$$

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

$$(v_2) \in \llbracket \delta A \rrbracket_{g',k' - k_1 - k_2} = \llbracket \delta A \rrbracket_{g',k' - k'' + 1} \wedge \delta L_2 \leq c_2 \leq \delta U_2 \quad \text{IH2}$$

Let us assume:

$$g'' = g'[r \rightarrow (l, A, D)] \quad (c)$$

$$H' = H, [l \rightarrow [v_2, v_2, \dots, v_2]] \quad (d)$$

we want to show 4 cases.

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

TS2 $(H') \models_{g'',k' - k''} p \star \gamma \rightarrow \mathbb{N}$
we know that $g''(\gamma) = (l, A, D)$, $H' = H \uplus H_l \wedge H_l = l \rightarrow [v_2, v_2, \dots, v_2]$

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

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

TS4 $\exists n. P = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\} \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i \}$
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.

$$\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{Array}_\gamma [I] A \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : \text{int}[I'] \quad \models I' \leq I \quad \Sigma; \Delta \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)} \text{read } t_1 t_2 : \frac{\{P\} \exists_- : A \{P\}}{\{P\} \exists_- : A \{P\}}} \text{U-R}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta \Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \langle \delta \Omega' \rangle_{g,k}$

TS: $(\delta \sigma \text{ read } t_1 t_2) \in \llbracket \frac{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)}{\{P\} \exists_- : A \{P\}} \rrbracket_{g,k}^{E,(0,0)}$

Because $\text{read } t_1 t_2$ is value.

STS: $(\delta \sigma \text{ read } t_1 t_2) \in \llbracket \frac{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)}{\{P\} \exists_- : A \{P\}} \rrbracket_{g,k}$

Unfold the definition of $\llbracket \frac{\text{exec}(L_1+L_2+L_r, U_1+U_2+U_r)}{\{P\} \exists_- : A \{P\}} \rrbracket_{g,k}$.

Pick $g' \supseteq g$, $k' \leq k$, $k'' < k'$ and H.

Assume $H \models_{g',k'} \gamma$ and $H; \text{read } t_1 t_2 \Downarrow^{k''}$

From $H; \text{read } t_1 t_2 \Downarrow^{k''}$, because t_1, t_2 are sub terms of $\delta \sigma \text{ read } t_1 t_2$, we get

$$\delta \sigma t_1 \Downarrow_f^{k_1} \wedge \delta \sigma t_2 \Downarrow_f^{k_2} \quad (1)$$

From (1), we get:

$$\exists l, n, v, k_1, k_2. \delta \sigma t_1 \Downarrow^{k_1, c_1} l \wedge \delta \sigma t_2 \Downarrow^{k_2, c_2} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c_2 + c_{\text{read}} \wedge H(l)[n] = v \quad (a)$$

From (a) and evaluation rules we get:

$$\delta \sigma (\text{read } t_1 t_2); H \Downarrow_f^{k'', c} v; H \quad (b)$$

By IH on the first premise instantiated with $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g',k'}$ by lemma 1 and $FV(t_1) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$. we get:

$$(\delta \sigma t_1) \in \llbracket \text{Array}_\gamma [I] A \rrbracket_{g',k'}^{E, \delta(L_1, U_1)} \quad (6)$$

Unfold (6), since $k_1 \leq k'$ and $\delta \sigma t_1 \Downarrow^{k_1, c_1} l$, we know

$$l \in \llbracket \text{Array}_\gamma [I] A \rrbracket_{g', k' - k_1} \wedge \delta L_1 \leq c_1 \leq \delta U_1 \quad (e)$$

From (e), we know :

$$g'(\gamma) = (l, A, I) \quad (7)$$

By IH on the second premise instantiated with $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g',k'}$ by lemma 1 and $FV(t_2) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$. we get:

$$(\delta \sigma t_2) \in \llbracket \text{int}[I'] \rrbracket_{g',k'}^{E, \delta(L_2, U_2)} \quad (8)$$

Unfold (8), since $k_2 \leq k'$, we know

$$n \in \llbracket \text{int}[I'] \rrbracket_{g', k' - k_2} \Rightarrow n = I' \wedge \delta L_2 \leq c_2 \leq \delta U_2 \quad (f)$$

Let us assume:

$$g' = g' \quad (g)$$

STS1: $H \models_{g', k' - k''} P$

proved by assumption $H \models_{g', k'} P$

STS2: $(v) \in \llbracket A \rrbracket_{g', k' - k''}$

from (a). we know $H_1(l)[n] = v$

From $H \models_{g', k' - k''} \gamma$ we know:

$\forall i \leq n, (H_1(l)(i)) \in \llbracket A \rrbracket_{g', k' - 1}$

s.t we know

$(H_1(l)[n]) \in \llbracket A \rrbracket_{g', k' - 1}$

Because $k' - k'' \leq k' - 1$,

By Lemma 1, We get: $(v) \in \llbracket A \rrbracket_{g', k' - k''}$

STS3: $\delta(L_1 + L_2 + L_r) \leq c \leq \delta(U_1 + U_2 + U_r)$

by the fact that $\delta L_r \leq c_{read} \leq \delta U_r$, it is proved by (e)(f).

STS4: $\exists n. P = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\} \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$

H is not changed, $H=H'$. it is trivially true.

This completes the proof of case read.

$$\frac{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t_1 : \text{Array}[I] \ A \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_2}^{U_2} t_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Omega \vdash_{L_3}^{U_3} t_3 : A \quad \Delta; \Phi \models I' \leq I \quad \Sigma; \Delta; \vdash P \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^{\text{exec}(L_1+L_2+L_3+L_u, U_1+U_2+U_3+U_u)} \text{updt } t_1 \ t_2 \ t_3 : \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit } \{P \star \gamma \rightarrow \beta\}} \text{U-U}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta \Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g,k}$

TS: $(\delta \sigma \text{ updt } t_1 \ t_2 \ t_3) \in \llbracket \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit } \{P \star \gamma \rightarrow \beta\} \rrbracket_{g,k}^{E,(0,0)}$

Because read $t_1 \ t_2$ is value.

STS: $(\delta \sigma \text{ updt } t_1 \ t_2 \ t_3) \in \llbracket \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit } \{P \star \gamma \rightarrow \beta\} \rrbracket_{g,k}$

Unfold the definition of $\llbracket \{P \star \gamma \rightarrow \beta\} \exists_- : \text{unit } \{P \star \gamma \rightarrow \beta\} \rrbracket_{g,k}$.

Pick $g' \supseteq g$, $k' \leq k$, $k'' < k'$ and H.

Assume $H \models_{g',k'} P \star \gamma \rightarrow \beta$ and $H; \text{updt } t_1 \ t_2 \ t_3 \Downarrow^{k''} H = H_p \uplus H_l \wedge H_p \models P \wedge H_l \models \gamma \rightarrow \beta$

From $H; \text{updt } t_1 \ t_2 \ t_3 \Downarrow^{k''}$, because t_1, t_2 are sub terms of $\delta \sigma \text{ updt } t_1 \ t_2 \ t_3$, we get

$$\delta \sigma t_1 \Downarrow_f^{k_1} \wedge \delta \sigma t_2 \Downarrow_f^{k_2} \wedge \delta \sigma t_3 \Downarrow_f^{k_3} \quad (1)$$

From (1), we get:

$$\exists l, n, v, k_1, k_2, k_3. \delta \sigma t_1 \Downarrow^{k_1, c_1} l \wedge \delta \sigma t_2 \Downarrow^{k_2, c'_1} n \wedge \delta \sigma t_3 \Downarrow^{k_3, c''_1} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c'_1 + c''_1 + c_{update} \quad (a)$$

From (a) and evaluation rules we get:

$$\delta \sigma(\text{updt } t_1 \ t_2 \ t_3); H \Downarrow_f^{k'', c} (); H[l][n] \leftarrow v \quad (b)$$

By IH on first premise instantiated with $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g',k'}$ by lemma 1 and $FV(t_2) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$.

We get:

$$(\delta \sigma t_1) \in \llbracket \text{Array}_\gamma[I] \ A \rrbracket_{G',k'}^{E, \delta(L_1, U_1)} \quad (2)$$

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

$$(\delta \sigma t_2) \in \llbracket \text{int}[I] \rrbracket_{g',k'}^{E, \delta(L_2, U_2)} \quad (3)$$

$$(\delta \sigma t_3) \in \llbracket A \rrbracket_{g',k'}^{E, \delta(L_3, U_3)} \quad (4)$$

From (2),(3),(4), we know:

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

$$n \in \llbracket \text{int}[I] \rrbracket_{g',k'-k_2} \Rightarrow n = I \wedge \delta L_2 \leq c'_1 \leq \delta U_2 \quad (d)$$

$$v \in \llbracket A \rrbracket_{g',k'-k_3} \wedge \delta L_3 \leq c''_1 \leq \delta U_3 \quad (e)$$

Let us assume:

$$g'' = g' \quad (5)$$

$$H'_1 = H(l)[n] \leftarrow v = H_p + H_l(l)[n] \leftarrow v \wedge H'_1 = H_l(l)[n] \leftarrow v \quad (6)$$

STS1: $(H'_1) \models_{g',k'-k''} P \star \gamma \rightarrow \beta$
 Unfold $H \models_{g',k'} P \star \gamma \rightarrow \beta$ and consider (c), we know

$$\forall i \leq I. (H_i(l)[i]) \in \llbracket A \rrbracket_{g',k'-1} \quad (7)$$

we need to prove:

subgoal 1: $\forall i \leq I. (H'_i(l)[i]) \in \llbracket A \rrbracket_{g',k'-k''}$
 when $i=n$,
 we prove $(H'_i(l)[i]) \in \llbracket A \rrbracket_{g',k'-k_3}$ from (e) and (6), and then use lemma 1.
 when $i \neq n$,
 proved by using lemma 1 on (7)

STS2: $(0) \in \llbracket \text{unit} \rrbracket_{g',k'-k''}$

It is proved by unfolding the definition.

STS3: $\delta(L_1 + L_2 + L_3 + L_u) \leq c \leq \delta(U_1 + U_2 + U_3 + U_u)$
 by the fact that $\delta L_u \leq c_{update} \leq \delta U_u$, it is proved by (c)(d)(e).

STS4: $\exists n. P = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\} \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
 Because $H'_1 = H(l)[n] \leftarrow v = H_p + H_l(l)[n] \leftarrow v \wedge H'_l = H_l(l)[n] \leftarrow v$, When $l_i \neq l$, $H(l_i) = H'(l_i)$, it is true. When $l_i = l$, only at position n so that $H(l)[n] \neq H(l)[n]$, we need to show $n \notin \beta$, which is the assumption from the rule.

This completes the proof of case unary update.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Omega \vdash_L^U t : A \quad \Delta; \Phi; \Omega \vdash p \quad wf}{\Sigma; \Delta; \Phi; \Omega \vdash_0^0 \text{return } t : \{P\} \exists \gamma. A \{P\}} \text{U-T} \quad \frac{}{\text{exec}(L,U)}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta \Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \langle \delta \Omega' \rangle_{g,k}$

TS: $(\delta \sigma \text{return } t) \in \llbracket \{P\} \exists \gamma. A \{P\} \rrbracket_{g,k}^{E,(0,0)}$

Because $\text{return } t$ is value.

STS: $(\delta \sigma \text{return } t) \in \llbracket \{P\} \exists \gamma. A \{P\} \rrbracket_{g,k}^{\text{exec}(L,U)}$

Unfold the definition of $\llbracket \{P\} \exists \gamma. A \{P\} \rrbracket_{g,k}^{\text{exec}(L,U)}$.

Pick $g' \supseteq g$, $H_1, k' \leq k$, $k'' < k'$.

Assume

$$H_1 \models_{g',k'} P \quad (1)$$

$$\delta \sigma(\text{return } t); H_1 \Downarrow_f^{k''} \quad (2)$$

Because t is the subterm of $\text{return } t$, From (2), We know:

$$\delta \sigma_1 t \Downarrow^{k''} \quad (3)$$

$$(2.5)$$

From (3), we know:

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

From (a) and the evaluation rule U-ret:

$$\exists H'_1. \text{return } t; H_1 \Downarrow_f^{k'', c} v_1; H'_1 \quad (b)$$

By IH on the premise instantiated with $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g',k'}$ by lemma 1. We get:

$$(\delta \sigma t_1) \in \llbracket \delta A \rrbracket_{G',k'}^{E,\delta(L,U)} \quad (5)$$

Unfold (5), we get:

$$v_1 \in \llbracket \delta A \rrbracket_{g',k'-k_1} \wedge \delta L_1 \leq c_1 \leq \delta U_1 \quad (6)$$

Let us assume:

$$g'' = g' \quad (7)$$

$$H'_1 = H_1 \quad (8)$$

STS1: $H'_1 \models_{g',k'-k''} P$
it is from assumption by using lemma 1.

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

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

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

This completes the proof of case unary return.

$$\frac{\begin{array}{c} P = P_1 \star P_2 \\ \Sigma; \Delta; \Phi; \Omega \vdash_{L_1}^{U_1} t : \{P_1\} \exists \vec{\gamma}_1 : A \{Q_1 \star Q_2\} \quad \Sigma; \Delta, \vec{\gamma}_1; \Phi; \Omega, x : A \vdash_{L_2}^{U_2} u : \{Q_1 \star P_2\} \exists \vec{\gamma}_2 : B \{Q\} \end{array}}{\Sigma; \Delta; \Phi; \Omega \vdash_{L_1+L_2+L_l}^{U_1+U_2+U_l} \text{let } \{x\} = t \text{ in } u : \{P\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : B \{Q \star Q_2\}} \text{U-E}$$

Assume that $\vdash \delta : \Delta$ and $\models \delta \Phi$ and there exists Ω' . s.t. $FV(t) \subseteq \text{dom}(\Omega')$ and $\Omega' \subseteq \Omega$ and $(\sigma) \in \llbracket \delta \Omega' \rrbracket_{g,k}$

TS: $(\delta \sigma_1(\text{let } \{x\} = t_1 \text{ in } t'_1)) \in \llbracket \delta \{P\} \exists \vec{\gamma}_1, \vec{\gamma}_2 : B \{Q \star Q_2\} \rrbracket_{g,k}^{(E,(U_1+U_2,L_1+L_2))}$

Unfold the definition .

Pick $g' \supseteq g$, $H_1, k' \leq k$, $k'' < k'$.

Assume

$$H_1 \models_{G',k'} P \quad (1)$$

$$\delta \sigma_1(\text{let } \{x\} = t_1 \text{ in } t'_1); H_1 \Downarrow_f^{k''} \quad (2)$$

Because t_1 is the subterm of $\text{let } \{x\} = t_1 \text{ in } t'_1$, From (2), We know:

$$\begin{aligned} \exists v_1, v'_1, v''_1, v'''_1, k_1, k_2, k_3, k_4, H'_1, H''_1. \delta \sigma_1 t_1 \Downarrow_f^{k_1, c_1} v_1 \text{ and } H_1; v_1 \Downarrow_f^{k_2, c_2} H'_1; v'_1 \text{ and} \\ \delta \sigma_1 t'_1[v'_1/x] \Downarrow_f^{k_3, c_3} v''_1 \text{ and } H'_1; v''_1 \Downarrow_f^{k_4, c_4} H''_1; v'''_1 \text{ and} \\ k'' = k_1 + k_2 + k_3 + k_4 + 1 \wedge c = c_1 + c_2 + c_3 + c_4 + c_{let} \end{aligned} \quad (a)$$

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

$$\begin{aligned} \exists H_a, H_c, H'_a, H1_{Q1}, H1_{Q2}, H1_Q, s.t. H_1 = H_a \uplus H_c \wedge v_1; H_a \Downarrow_f v'_1; H'_a \wedge H'_a = H1_{Q1} \uplus H1_{Q2} \\ \wedge v''_1; H1_{Q1} \uplus H_c \Downarrow_f v'''_1; H1_Q \wedge H''_1 = H1_Q \uplus H1_{Q2} \end{aligned} \quad (c)$$

From (1) : $H \models_{g',k'} P_1 \star P_2$, we assume that

$$\begin{aligned} \exists g_a, g_b : (H_1 = H_a \uplus H_c) \\ (g' = g_a \uplus g_c) \text{ and } (H_a) \models_{g_a, k'} P_1 \text{ and } (H_c) \models_{g_c, k'} P_2 \end{aligned} \quad (*)$$

STS1: $(H_1'') \models_{g'', k' - k''} Q \star Q_2$

STS2: $(v_1''') \in \langle B \rangle_{G'', k' - k''}$

STS3: $(L + L') \leq c_f \leq (U + U') \wedge L_1 + L_2 + L_{let} \leq c_p \leq U_1 + U_2 + U_{let}$

ST4: $P = \gamma_i \rightarrow \beta_i \forall i. g''(\gamma_i) = (l_i, m, A) \rightarrow H_1(l_i)[n] \neq H_1''(l_i)[n] \Rightarrow n \in \beta_i$

By IH on the second premise instantiated with $(\sigma_1) \in \langle \delta\Omega \rangle_{g', k'}$, we know:

$$(\delta\sigma_1 t_1) \in \langle \{P_1\} \exists \vec{\gamma}_1 : A \{Q_1 \star Q_2\} \rangle_{g', k'}^{E, (L_1, U_1)} \quad (4)$$

Unfold the definition ,use (a), we get:

$$(v_1) \in \langle \{P_1\} \exists \vec{\gamma}_1 : A \{Q_1 \star Q_2\} \rangle_{g', k'} \wedge L_1 \leq c_1 \leq U_1 \wedge P_1 = \gamma_i \rightarrow \beta_i \Rightarrow \forall i. g''(\gamma_i) = (l_i, m, A) \rightarrow H_1(l_i)[n] \neq H_1'(l_i)[n] \Rightarrow n \in \beta_i \quad (e)$$

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

$$(H_a \models_{g_f, k_f} P_1 \wedge H_a; v_1 \Downarrow_f \wedge H_b; v_2 \Downarrow_f$$

Unfold the definition, we get:

$$H_a; v_1 \Downarrow_f^{c_2} H'_a; v'_1 \wedge H'_a \models Q_1 \star Q_2$$

$$(v'_1) \in \langle A \rangle_{g', k' - k_1} \wedge L \leq c_2 \leq U \quad (k)$$

By IH on the third premise, we know:

$$(\delta\sigma_1 v''_1) \in \langle \{Q_1 \star P_2\} \exists \vec{\gamma}_1 . B \{Q\} \rangle_{G', k'}^{E, D_2} \quad (5)$$

Unfold the definition, we get:

$$v_1 \in \langle \{Q_1 \star P_2\} \exists \vec{\gamma}_1 . \tau_2 \{Q\} \rangle_{g', k' - k_1} \wedge l_2 \leq c_3 \leq U_2 \wedge Q_1 \uplus P_2 = \gamma_i \rightarrow \beta_i \Rightarrow \forall i. g''(\gamma_i) = (l_i, m, A) \rightarrow H_1'(l_i)[n] \neq H_1''(l_i)[n] \Rightarrow n \in \beta_i \quad (f)$$

We have

$$(H'_a) \models_{g_f, k_f} Q_1 \star Q_2 \Rightarrow H1_{Q_1} \models Q_1 \wedge H1_{Q_2} \models Q_2 \wedge H_c \models P_2$$

$$H1_{Q_1} \uplus H_c; v_1''' \Downarrow_f$$

Unfold the definition (f), we get:

$$H1_{Q_1} \uplus H_c; v_1''' \Downarrow_f^{c_4} H1_Q; v_1'''' \wedge H1_Q \models Q$$

$$(v_1'''') \in \langle B \rangle_{g', k' - k_1} \wedge L' \leq c_4 \leq U' \quad (g)$$

STS1: Use Lemma 2, we know that $H1_Q \uplus H1_{Q_2} \models Q \star Q_2$, 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 $\models \delta \Phi$. Then for $i \in \{1, 2\}$. if there exists Γ'_i s.t. $FV(e_i) \subseteq \text{dom}(\Gamma'_i), \Gamma'_i \subseteq \Gamma$ and $(\sigma_i) \in \llbracket \delta \Gamma'_i \uparrow_i \rrbracket_{G_{|i,k}}$, then $\delta \sigma_i e_i \in \llbracket \delta \tau \uparrow_i \rrbracket_{G_{|i,k}}^{E, (0, \infty)}$
Because it is similar for the two case, we only show the case when $i=1$.

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

Assume that $\models \Phi \delta$ and there exists Γ' s.t. $FV(\text{fix } f(x). t_1) \subseteq \text{dom}(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in \llbracket \delta \Gamma' \uparrow_1 \rrbracket_{G_{|1,k}}$

TS: $(\text{fix } f(x). \delta \sigma t_1) \in \llbracket (|\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|) \rrbracket_{G_{|1,k}}^{E, (0, \infty)}$

By Lemma 31, STS: $(\text{fix } f(x). \delta \sigma t_1) \in \llbracket (|\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|) \rrbracket_{G_{|1,k}}^{\text{exec}(0, \infty)}$

We prove the more general statement:

$$\forall m' \leq k. (\text{fix } f(x). \delta \sigma t_1) \in \llbracket (|\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|) \rrbracket_{G_{|1,m'}}^{\text{exec}(0, \infty)} \quad (1)$$

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: $m' = m'' + 1 \leq k$

By sub-IH: $(\text{fix } f(x). \delta \sigma t_1) \in \llbracket (|\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|) \rrbracket_{G_{|1,m''}}^{\text{exec}(0, \infty)}$

STS: $(\text{fix } f(x). \delta \sigma t_1) \in \llbracket (|\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|) \rrbracket_{G_{|1,m''+1}}^{\text{exec}(0, \infty)}$

Unfold the definition. Pick $j \leq m'' + 1, g' \supseteq G_{|1}$.

Assume $v \in \llbracket \delta \tau_1 \uparrow_1 \rrbracket_{g', j}$

STS: $\delta \sigma t_1 [v/x] [\text{fix } f(x). t_1 / f] \in \llbracket \delta \tau_2 \uparrow_1 \rrbracket_{g', j}^{E, (0, \infty)}$

By IH 3 on the premise instantiated with $\sigma [v/x] [\text{fix } f(x). t_1 / f] \in \llbracket \delta(x : |\tau_1 \uparrow_1|, f : |\delta \tau_1 \uparrow_1| \longrightarrow |\delta \tau_2 \uparrow_1|, |\Gamma'|) \rrbracket_{g', j}^{\text{exec}(0, \infty)}$ which holds because

1. $\sigma \in \llbracket \delta \Gamma' \uparrow_1 \rrbracket_{g', j}$ by Lemma 1.
2. $v \in \llbracket \tau_1 \uparrow_1 \rrbracket_{g', j}$ from assumption.
3. $\text{fix } f(x). t_1 \in \llbracket \tau_1 \uparrow_1 \longrightarrow \tau_2 \uparrow_1 \rrbracket_{g', j}^{\text{exec}(0, \infty)}$ by Lemma 1 on the sub-IH conclusion.

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

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{int}[I] \quad \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \tau \quad \gamma \text{ fresh}}{\Delta; \Phi; \Gamma \vdash \text{alloc } t_1 t_2 \ominus \text{alloc } t'_1 t'_2 \lesssim 0 : \{P\} \exists \gamma. \text{Array}_\gamma [I] \tau \{P \star \gamma \rightarrow \mathbb{N}\}} \text{ALLOC}$$

Assume that $\models \Phi \delta$ and there exists Γ' s.t. $FV(\text{alloc } t_1 t_2) \subseteq \text{dom}(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in \llbracket \delta \Gamma' \uparrow_1 \rrbracket_{G_{|1,k}}$

TS: $\delta \sigma \text{ alloc } t_1 t_2 \in \llbracket \delta(\{|P|_1\} \exists \gamma. \text{Array}_\gamma [I] |\tau|_1 \{|P|_1 \star \gamma \rightarrow \mathbb{N}\}) \rrbracket_{G_{|1,k}}^{E, (0, \infty)}$

STS: $\delta \sigma \text{ alloc } t_1 t_2 \in \llbracket \delta(\{|P|_1\} \exists \gamma. \text{Array}_\gamma [I] |\tau|_1 \{|P|_1 \star \gamma \rightarrow \mathbb{N}\}) \rrbracket_{G_{|1,k}}^{\text{exec}(0, \infty)}$

Unfold the definition.

Pick $g' \supseteq G_{|1}, k' \leq k, k'' < k'$ and assume $H \models_{g', k'} |P|_1$ and $H; \text{alloc } t_1 t_2 \Downarrow_f^{k''}$

STS: $H; v \Downarrow_f^{k'',c} v', H'$ and $0 \leq c \leq \infty$ and $H' \models_{g'',k'-k''} \gamma$ and $v' \in \llbracket \text{Array}_\gamma[I] \mid \tau \mid_1 \rrbracket_{g'',k'-k''}$

From $H; \text{alloc } t_1 t_2 \Downarrow^{k''}$, because t_1, t_2 are sub terms of $\delta\sigma \text{ alloc } t_1 t_2$, we get

$$\delta\sigma t_1 \Downarrow_f^{k_1} \wedge \delta\sigma t_2 \Downarrow_f^{k_2} \quad (1)$$

From (1), we get:

$$\exists v_1, v_2, k_1, k_2. \delta\sigma t_1 \Downarrow^{k_1, c_1} v_1 \wedge \delta\sigma t_2 \Downarrow^{k_2, c'_1} v_2 \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c'_1 + c_{alloc} \quad (a)$$

From (a) and evaluation rules we get:

$$\exists l. H, (\text{alloc } t_1 t_2) \delta\sigma \Downarrow_f^{k'',c} l; H[l \rightarrow [v_2, \dots, v_2]] \quad (b)$$

By IH 3 on the first premise instantiated with $\sigma \in \llbracket \delta\Gamma' \mid_1 \rrbracket_{g, k''}$ using lemma 1 and $FV(t_1) \subseteq \text{dom}(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta\sigma t_1) \in \llbracket \delta\text{int}_r[I] \mid_1 \rrbracket_{g', k''}^{E, (0, \infty)} \quad (*)$$

Unfold the definition of $\llbracket \text{int}[I] \rrbracket_{g', k''}^{E, (0, \infty)}$ and using (a) and $k_1 \leq k''$, we know:

$$(v_1) \in \llbracket \text{int}[I] \rrbracket_{G', k'' - k_1} \Rightarrow v_1 = I \wedge 0 \leq c_1 \leq \infty \quad \text{IH1}$$

By IH 3 on the second premise instantiated with $(\sigma) \in \llbracket \delta\Gamma' \mid_1 \rrbracket_{g', k' - k_1}$ using lemma 1 and $FV(t_2) \subseteq \text{dom}(\Gamma' \mid_1)$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta\sigma t_2) \in \llbracket \delta\tau \mid_1 \rrbracket_{g', k' - k_1}^{E, (0, \infty)} \quad (**)$$

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

$$(v_2) \in \llbracket \delta\tau \mid_1 \rrbracket_{g', k' - k_1 - k_2} = \llbracket \delta\tau \mid_1 \rrbracket_{g', k' - k'' + 1} \wedge 0 \leq c'_1 \leq \infty \quad \text{IH2}$$

Let us assume:

$$g'' = g'[\gamma \rightarrow (l, \tau \mid_1, I)] \quad (c)$$

$$H'_1 = H_1, [l \rightarrow [v_2, v_2, \dots, v_2]] \quad (d)$$

we want to show 2 cases.

TS1 $(v' = l) \in \llbracket \text{Array}_\gamma[I] \mid \tau \mid_1 \rrbracket_{g'', k' - k''}$
it is proved by unfolding the definition and using the assumption (c).

TS2 $(H'_1) \models_{g'', k' - k''} |P|_1 \star \gamma \rightarrow \mathbb{N}$
we know that $g''(\gamma) = (l, \tau \mid_1, I)$, $H'_1 = H_1 \uplus H_l \wedge H_l = l \rightarrow [v_2, \dots, v_2]$, $H_1 \models |P|_1$ from assumption.

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

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

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

This proof is complete.

$$\text{CASE } \frac{\Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}_r[I'] \quad \Delta; \Phi \models I' \leq I}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 t_2 \ominus \text{read } t'_1 t'_2 \lesssim 0 : \{P\} \exists \tau \{P\}} \text{R-READ}$$

Assume that $\models \Phi\delta$ and there exists Γ' s.t. $FV(\text{read } t_1 t_2) \subseteq \text{dom}(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in \llbracket \delta\Gamma' \llbracket 1 \rrbracket \rrbracket_{G_1, k}$

TS: $\delta\sigma \text{ read } t_1 t_2 \in \llbracket \delta(\llbracket P \llbracket 1 \rrbracket \exists_{-} \cdot \llbracket \tau \llbracket 1 \rrbracket \llbracket P \llbracket 1 \rrbracket \rrbracket \rrbracket \rrbracket_{G_1, k}^{E, (0, \infty)}$

STS: $\delta\sigma \text{ alloc } t_1 t_2 \in \llbracket \delta(\llbracket P \llbracket 1 \rrbracket \exists_{-} \cdot \llbracket \tau \llbracket 1 \rrbracket \llbracket P \llbracket 1 \rrbracket \rrbracket \rrbracket \rrbracket_{G_1, k}^{E, (0, \infty)}$

Unfold the definition.

Pick $g' \supseteq G_1, k' \leq k, k'' < k'$ and assume $H \models_{g', k'} |P|_1$ and $H; \text{read } t_1 t_2 \Downarrow_f^{k''}$

STS: $H; \text{read } t_1 t_2 \Downarrow_f^{k'', c} v', H'$ and $0 \leq c \leq \infty$ and $H' \models_{g'', k'-k''} \gamma$ and $v' \in \llbracket \tau \llbracket 1 \rrbracket \rrbracket_{g'', k'-k''}$

From $\text{read } t_1 t_2; H \Downarrow_f^{k''}$, because t_1, t_2 are sub terms of $\delta\sigma \text{ read } t_1 t_2$, we get

$$\delta\sigma t_1 \Downarrow_f^{k_1} \wedge \delta\sigma t_2 \Downarrow_f^{k_2} \quad (1)$$

From (1), we get:

$$\exists l, n, v, k_1, k_2. \delta\sigma t_1 \Downarrow^{k_1, c_1} l \wedge \delta\sigma t_2 \Downarrow^{k_2, c'_1} n \wedge k'' = k_1 + k_2 + 1 \wedge c = c_1 + c'_1 + c_{read} \wedge H(l)[n] = v \quad (a)$$

From (a) and evaluation rules we get:

$$H; \delta\sigma(\text{read } t_1 t_2) \Downarrow_f^{k'', c} H; v \quad (b)$$

By IH 3 on the first premise instantiated with $\sigma \in \llbracket \delta\Gamma' \llbracket 1 \rrbracket \rrbracket_{g, k''}$ using lemma 1 and $FV(t_1) \subseteq \text{dom}(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta\sigma t_1) \in \llbracket \text{Array}_\gamma[I] \llbracket \tau \llbracket 1 \rrbracket \rrbracket_{g', k'}^{E, (0, \infty)} \quad (6)$$

Unfold (6), since $k_1 \leq k'$ and $\delta\sigma t_1 \Downarrow^{k_1, c_1} l$, we know

$$l \in \llbracket \text{Array}_\gamma[I] \llbracket \tau \llbracket 1 \rrbracket \rrbracket_{g', k'-k_1} \wedge 0 \leq c_1 \leq \infty \quad (e)$$

From (e), we know :

$$g'(\gamma) = (l, \llbracket \tau \llbracket 1 \rrbracket \rrbracket, I) \quad (7)$$

By IH 3 on the first premise instantiated with $\sigma \in \llbracket \delta\Gamma' \llbracket 1 \rrbracket \rrbracket_{g, k''}$ using lemma 1 and $FV(t_2) \subseteq \text{dom}(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta\sigma t_2) \in \llbracket \text{int}[I'] \rrbracket_{g', k'}^{E, (0, \infty)} \quad (8)$$

Unfold (8), since $k_2 \leq k'$, we know

$$n \in \llbracket \text{int}[I'] \rrbracket_{g', k'-k_2} \Rightarrow n = I' \wedge 0 \leq c'_1 \leq \infty \quad (f)$$

Let us assume:

$$g' = g' \quad (g)$$

STS1: $H \models_{g', k'-k''} |P|_1$

proved by Lemma 1 on assumption $H \models_{g', k'} |P|_1$

STS2: $(v' = v) \in \llbracket \tau \llbracket 1 \rrbracket \rrbracket_{g', k'-k''}$

from (a). we know $H_1(l)[n] = v$

From $H \models_{g', k'-k''} \gamma$ we know:

$\forall i \leq n, (H_1(l)(i)) \in \llbracket A \rrbracket_{g', k'-1}$

s.t we know

$(H_1(l)[n]) \in \llbracket A \rrbracket_{g', k'-1}$

Because $k' - k'' \leq k' - 1$,

By Lemma 1, We get: $v \in \llbracket A \rrbracket_{g', k'-k''}$

STS3: $0 \leq c \leq \infty$

by the fact that $0 \leq c_{read} \leq \infty$, it is proved by (e)(f).

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

This completes the proof of case read.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Delta; \Phi \models I' \leq I \quad \Delta; \Phi \models I' \in \beta}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{read } t_1 \ t_2 \ominus \text{read } t'_1 \ t'_2 \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ . \square \tau \{\gamma \rightarrow \beta\}} \text{R-RB}$$

The proof is same as case read above.

$$\text{CASE } \frac{\Sigma; \Delta; \Phi; \Gamma \vdash t_1 \ominus t'_1 \lesssim D_1 : \text{Array}_\gamma[I] \tau \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_2 \ominus t'_2 \lesssim D_2 : \text{int}[I'] \quad \Sigma; \Delta; \Phi; \Gamma \vdash t_3 \ominus t'_3 \lesssim D_3 : \tau \quad \Delta; \Phi \models I' \leq I}{\Sigma; \Delta; \Phi; \Gamma \vdash \text{updt } t_1 \ t_2 \ t_3 \ominus \text{updt } t'_1 \ t'_2 \ t'_3 \lesssim 0 : \{P \star \gamma \rightarrow \beta\} \exists _ . \text{unit}_r \{P \star \gamma \rightarrow \beta \cup \{I'\}\}} \text{R-U}$$

Assume that $\models \Phi \delta$ and there exists Γ' s.t. $\text{FV}(\text{updt } t_1 \ t_2 \ t_3) \subseteq \text{dom}(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in \llbracket \delta \Gamma' \rrbracket_{G_{1,k}}$

TS: $\delta \sigma \text{ updt } t_1 \ t_2 \ t_3 \in \llbracket \delta(\{|P|_1 \star \gamma \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r | \{|P|_1 \star \gamma \rightarrow \mathbb{N}\}) \rrbracket_{G_{1,k}}^{E,(0,\infty)}$

STS: $\delta \sigma \text{ updt } t_1 \ t_2 \ t_3 \in \llbracket \delta(\{|P|_1 \star \gamma \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r | \{|P|_1 \star \gamma \rightarrow \mathbb{N}\}) \rrbracket_{G_{1,k}}^{\text{exec}(0,\infty)}$

Unfold the definition.

Pick $g' \supseteq G_{1,k}, k' \leq k, k'' < k'$ and assume $H \models_{g',k'} |P|_i \star \gamma \rightarrow \mathbb{N}$ and $H; \text{updt } t_1 \ t_2 \ t_3 \downarrow_f^{k''}$

STS: $H; \text{updt } t_1 \ t_2 \ t_3 \downarrow_f^{k'',c} v', H'$ and $0 \leq c \leq \infty$ and $H' \models_{g'',k'-k''} \gamma$ and $v' \in \llbracket \text{unit}_r \rrbracket_{g'',k'-k''}$

From $H; \text{updt } t_1 \ t_2 \ t_3 \downarrow_f^{k''}$, because t_1, t_2 are sub terms of $\delta \sigma \text{ updt } t_1 \ t_2 \ t_3$, we get

$$\delta \sigma t_1 \downarrow_f^{k_1} \wedge \delta \sigma t_2 \downarrow_f^{k_2} \wedge \delta \sigma t_3 \downarrow_f^{k_3} \quad (1)$$

From (1), we get:

$$\exists l, n, v, k_1, k_2, k_3. \delta \sigma t_1 \downarrow_f^{k_1, c_1} l \wedge \delta \sigma t_2 \downarrow_f^{k_2, c_1'} n \wedge \delta \sigma t_3 \downarrow_f^{k_3, c_1''} v \wedge k'' = k_1 + k_2 + k_3 + 1 \wedge c = c_1 + c_1' + c_1'' + c_{\text{update}} \quad (a)$$

From (a) and evaluation rules we get:

$$\delta \sigma(\text{updt } t_1 \ t_2 \ t_3); H \downarrow_f^{k'',c} 0; H[l][n] \leftarrow v \quad (b)$$

By IH 3 on the first premise instantiated with $\sigma \in \llbracket \delta \Gamma' \rrbracket_{g,k''}$ using lemma 1 and $\text{FV}(t_2) \subseteq \text{dom}(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta \sigma t_1) \in \llbracket \text{Array}_\gamma[I] \tau \rrbracket_{g',k'}^{E,(0,\infty)} \quad (2)$$

Similarly, By IH 3 on the second premise and third premise, we get

$$(\delta \sigma t_2) \in \llbracket \text{int}[I] \rrbracket_{g',k'}^{E,(0,\infty)} \quad (3)$$

$$(\delta \sigma t_3) \in \llbracket \tau \rrbracket_{g',k'}^{E,(0,\infty)} \quad (4)$$

From (2),(3),(4), we know:

$$l \in \llbracket \text{Array}_\gamma[I] A \rrbracket_{g',k'-k_1} \Rightarrow g'(\gamma) = (l, A, I) \wedge 0 \leq c_1 \leq \infty \quad (c)$$

$$n \in \llbracket \text{int}[I] \rrbracket_{g',k'-k_2} \Rightarrow n = I \wedge 0 \leq c_1' \leq \infty \quad (d)$$

$$v \in \llbracket A \rrbracket_{g',k'-k_3} \wedge 0 \leq c_1'' \leq \infty \quad (e)$$

Let us assume:

$$g'' = g' \quad (5)$$

$$H'_1 = H(l)[n] \leftarrow v \quad (6)$$

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

STS2: $(0) \in \llbracket \text{unit} \rrbracket_{g',k'-k''}$
It is proved by unfolding the definition.

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

STS4: $\exists n. |P|_i \star \gamma \rightarrow \mathbb{N} = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\} \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
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.

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

Assume that $\models \Phi \delta$ and there exists Γ' s.t. $\text{FV}(\text{return } t_1) \subseteq \text{dom}(\Gamma'), \Gamma' \subseteq \Gamma$ and $\sigma \in \llbracket \delta \Gamma'_1 \rrbracket_{G_1, k}$

TS: $\delta \sigma \text{return } t_1 \in \llbracket \delta(\{|P|_1\} \exists_{-} \{\tau\}_1 \{|P|_1\})_1 \rrbracket_{G_1, k}^{E, (0, \infty)}$

STS: $\delta \sigma \text{return } t_1 \in \llbracket \delta(\{|P|_1\} \exists_{-} \{\tau\}_1 \{|P|_1\})_1 \rrbracket_{G_1, k}^{\text{exec}(0, \infty)}$

Unfold the definition.

Pick $g' \supseteq G_1, k' \leq k, k'' < k'$ and assume $H \models_{g', k'} |P|_1$ and $H; \text{return } t_1 \Downarrow_f^{k''}$

STS: $\text{return } t_1; H \Downarrow_f^{k'', c} v', H'$ and $0 \leq c \leq \infty$ and $H' \models_{g'', k'-k''} \gamma$ and $v' \in \llbracket \tau \rrbracket_{g'', k'-k''}$

Because t_1 is the subterm of $\text{return } t_1$, we know:

$$\delta \sigma_1 t_1 \Downarrow^{k''} \tag{3}$$

$$\tag{2.6}$$

From (3), we know:

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

From (a), (b) and the evaluation rule R-ret:

$$\text{return } t_1; H_1 \Downarrow_f^{k'', c} v_1; H_1 \tag{b}$$

By IH 3 on the first premise instantiated with $\sigma \in \llbracket \delta \Gamma'_1 \rrbracket_{g, k''}$ using lemma 1 and $\text{FV}(t_2) \subseteq \text{dom}(\Gamma')$ and $\Gamma' \subseteq \Gamma$, we get:

$$(\delta \sigma t_1) \in \llbracket \delta \tau \rrbracket_{G', k'}^{E, (0, \infty)} \tag{5}$$

Unfold (5), we get:

$$v_1 \in (\delta A)_{g', k'-k_1} \wedge 0 \leq c_1 \leq \infty \tag{6}$$

Let us assume:

$$g'' = g' \tag{7}$$

$$H'_1 = H_1 \tag{8}$$

STS1: $H_1 \models_{g', k'-k''} |P|_1$

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

STS4: $\exists n. |P|_1 = \{\gamma_1 \rightarrow T_1, \dots, \gamma_n \rightarrow T_n\} \wedge \forall i \in [1, n]. g(\gamma_i) = (l_i, A, m) \Rightarrow \forall j. (H[l_i][j]) \neq H'[l_i][j] \Rightarrow j \in T_i$
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

```
InPlaceMap = fix map (f).λa.λk.λn.
  if k ≤ n then (let {x} = (read a k) in
    let {_} = (updt a k (f x)) in ((celim (map f)) a (k + 1) n)
  else return()
```

As a first example we want to prove

$$\vdash \text{InPlaceMap} : \forall L. \forall U : \mathbb{N}. (A \xrightarrow{\text{exec}(L,U)} A') \longrightarrow \forall n : \mathbb{N}. \forall i : \mathbb{N}. \forall \gamma : \mathbb{L}. \\ (i \leq n) \supset (\text{Array}_\gamma[n] A \longrightarrow \text{int}[i] \longrightarrow \xrightarrow{\text{exec}((L+c_r+c_u)*(n-i), (U+c_r+c_u)*(n-i))} \{\gamma \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow \mathbb{N}\})$$

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

$$\vdash \text{InPlaceMap} \ominus \text{InPlaceMap} \lesssim 0 : \forall r. \square(\tau \xrightarrow{\text{diff}(r)} \tau) \longrightarrow \forall n, i : \mathbb{N}. \forall \gamma : \text{ref}. \forall \beta : \mathbb{P}. \\ (i \leq n) \supset (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i] \longrightarrow \xrightarrow{\text{diff}((\beta \cap [i, n]) * r)} \{\gamma \rightarrow \beta\} \exists _ . \text{unit} \{\gamma \rightarrow \beta\})$$

$$\vdash \text{InPlaceMap} \ominus \text{InPlaceMap} \lesssim 0 : \forall r. (\tau \xrightarrow{\text{diff}(r)} \tau) \longrightarrow \forall n, i : \mathbb{N}. \forall \gamma : \text{ref}. \forall \beta : \mathbb{P}. (i \leq n) \supset \\ (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i] \longrightarrow \xrightarrow{\text{diff}((n-i) * r)} \{\gamma \rightarrow \beta\} \exists _ . \text{unit} \{\gamma \rightarrow \beta \cup [i, n]\})$$

Let us call M the following term:

$$(\text{celim}(\text{map } f) a (k + 1) n)$$

and call T the following term:

$$\left(\text{let } \{x\} = (\text{read } a k) \text{ in let } \{_ \} = (\text{updt } a i (f x)) \text{ in } (\text{celim}(\text{map } f) a (k + 1) n) \right)$$

and call Z the following term:

$$\text{let } \{_ \} = (\text{updt } a i (f x)) \text{ in } (\text{celim}(\text{map } f) a (k + 1) n)$$

and set V as:

$$\text{if } k < n, \text{ then } T \text{ else } (\text{return } ())$$

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 $i \leq n$ 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, $\Phi \wedge i \in \beta$

$$\frac{\Delta; \Phi; \Gamma \vdash a \ominus a \lesssim 0 : \text{Array}_\gamma[n] \tau \quad \Delta; \Phi; \Gamma \vdash k \ominus k \lesssim 0 : \text{int}[i] \quad \Delta; \Phi \models i \leq n \quad \Delta; \vdash \text{empty} \quad wf}{\Delta; \Phi; \Gamma \vdash \text{read } a \ k \ominus \text{read } a \ k \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ . \tau \{\gamma \rightarrow \beta\}} \text{R-R}$$

and

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

where $\Delta = n : \mathbb{N}, i : \mathbb{N}, r : \mathbb{R}$

$\Sigma = \gamma : \mathbb{L}$

$\Phi = i \leq n$

$\Gamma = a : \text{Array}_\gamma[n] \tau, k : \text{int}[i], f : \square(\tau \xrightarrow{\text{diff}(r)} \tau), \text{map} : \square(\tau \xrightarrow{\text{diff}(r)} \tau) \xrightarrow{\text{diff}(0)} \forall n : \mathbb{N}. \forall i : \mathbb{N}. \forall \gamma : \mathbb{L}. (i \leq n) \supset (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i] \longrightarrow \text{int}[n] \longrightarrow \{\gamma \rightarrow \beta\} \text{unit} \{\gamma \rightarrow \beta\})$.

Because at position i , $a_1[i] \neq a_2[i]$, s.t. $\beta \cup \{i\} = \beta$.

We will show:

$$\begin{array}{c} \text{S-RM} \\ \hline \Delta; \Phi \models \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \sqsubseteq \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \Delta; \Phi; \Gamma \vdash \text{return } 0 \ominus \text{return } 0 \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \hline \Delta; \Phi; \Gamma \vdash \text{return } 0 \ominus \text{return } 0 \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \text{R-EXEC} \\ \Delta; \Phi; \Gamma \vdash k < n \ominus k < n \lesssim 0 : \text{unit} + \text{unit} \quad \Delta; \Phi; \Gamma \vdash T \ominus T \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \hline \Delta; \Phi; \Gamma, \{x : \tau\} \vdash V \ominus V \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \text{R-CASE} \end{array}$$

Using R-Let rules, we get

$$\frac{P_2 = \text{empty} \quad \Delta; \Phi; \Gamma \vdash \text{read } a \ k \ominus \text{read } a \ k \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{int} \{\gamma \rightarrow \beta\} \quad \Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\}}{\Delta; \Phi; \Gamma, \{x : \text{int}\} \vdash T \ominus T \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\}} \text{R-LET}$$

$$\frac{P_2 = \text{empty} \quad \Delta; \Phi; \Gamma \vdash \text{updt } a \ k \ (f \ x) \ominus \text{updt } a \ k \ (f \ x) \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \quad \Delta; \Phi; \Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\}}{\Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\}} \text{R-LET}$$

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

$$\begin{array}{c} \text{R-CELIM} \\ \Delta; \Phi \models i+1 \leq n \quad \Delta; \Phi \Gamma \vdash ((\text{map } f)) \ominus ((\text{map } f)) \lesssim 0: \\ \frac{(i+1 \leq n) \supset (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i+1] \longrightarrow \text{int}[n] \longrightarrow \{\gamma \rightarrow \beta\} \exists_- : \text{unit} \{\gamma \rightarrow \beta\})}{\Delta; \Phi; \Gamma \vdash (\text{celim}(\text{map } f)) \ominus (\text{celim}(\text{map } f)) \lesssim 0:} \\ \text{Array}_\gamma[n] \tau \rightarrow \text{int}[i+1] \rightarrow \text{int}[n] \rightarrow \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\} \end{array}$$

$$\Delta; \Phi; \Gamma \vdash (\text{celim}(\text{map } f) a(k+1) n) \ominus (\text{celim}(\text{map } f) a(k+1) n) \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\}$$

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

$$\begin{array}{c} \Delta; \Phi; \Gamma \vdash a \ominus a \lesssim 0: \text{Array}_\gamma[n] \tau \\ \Delta; \Phi; \Gamma \vdash k \ominus k \lesssim 0: \text{int}[i] \quad \Delta; \Phi \models i \leq n \quad \Delta; \Phi \models i \notin \beta \\ \frac{}{\Delta; \Phi; \Gamma \vdash \text{read } a k \ominus \text{read } a k \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- . \square \tau \{\gamma \rightarrow \beta\}} \text{R-RB} \end{array}$$

and

$$\begin{array}{c} \Delta; \Phi; \Gamma \vdash a \ominus a \lesssim 0: \text{Array}_\gamma[n] \tau \quad \Delta; \Phi; \Gamma \vdash k \ominus k \lesssim 0: \text{int}[i] \\ \Delta; \Phi \models \square(\tau \xrightarrow{\text{diff}(r)} \tau) \sqsubseteq \square \tau \xrightarrow{\text{diff}(0)} \square \tau \text{ S-R-BOX-DIFF} \\ \frac{}{\Delta; \Phi \vdash f \ominus f \lesssim 0: \square \tau \xrightarrow{\text{diff}(0)} \square \tau} \\ \frac{\Delta; \Phi; \Gamma x: \square \tau \vdash f x \ominus f x \lesssim 0: \square \tau \quad \Delta; \Phi \models i \leq n}{\Delta; \Phi; \Gamma, \vdash \text{updt } a k (f x) \ominus \text{updt } a k (f x) \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta / \{i\}\}} \text{R-UB} \end{array}$$

where $\Delta = n : \mathbb{N}, i : \mathbb{N}, r : \mathbb{R}$

$\Sigma = \gamma : \mathbb{L}$

$\Phi = i \leq n$

$\Gamma = a : \text{Array}_\gamma[n] \tau, k : \text{int}[i], f : \square(\tau \xrightarrow{\text{diff}(r)} \tau), \text{map} : \square(\tau \xrightarrow{\text{diff}(r)} \tau) \xrightarrow{\text{diff}(0)} \forall n : \mathbb{N}. \forall i : \mathbb{N}. \forall \gamma, \beta : . (i \leq n) \supset (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i] \longrightarrow \text{int}[n] \longrightarrow \exists \beta. \{\gamma \rightarrow \beta\} \text{unit} \{\gamma \rightarrow \beta\})$.

Because at position i , $a_1[i] \neq a_2[i]$, s.t. $\beta \cup \{i\} = \beta$.

Using R-Let rules, we get

$$\begin{array}{c} P_2 = \text{empty} \quad \Delta; \Phi; \Gamma \vdash \text{read } a k \ominus \text{read } a k \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{int} \{\gamma \rightarrow \beta\} \\ \Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \frac{}{\Delta; \Phi; \Gamma, \{x : \text{int}\} \vdash T \ominus T \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\}} \text{R-LET} \end{array}$$

$$\begin{array}{c} P_2 = \text{empty} \quad \Delta; \Phi; \Gamma \vdash \text{updt } a k (f x) \ominus \text{updt } a k (f x) \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \Delta; \Phi; \Gamma \vdash M \ominus M \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\} \\ \frac{}{\Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0: \{\gamma \rightarrow \beta\} \exists_- : \text{unit}_r \{\gamma \rightarrow \beta\}} \text{R-LET} \end{array}$$

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 :

$$\vdash \text{InPlaceMap} \ominus \text{InPlaceMap} \lesssim 0 : \forall r : \mathbb{N}. (\tau \xrightarrow{\text{diff}(r)} \tau) \longrightarrow \forall n : \mathbb{N}. \forall i : \mathbb{N}. \forall \gamma : \mathbb{L}. \forall \beta : \mathbb{P}. \\ (i \leq n) \supset (\text{Array}_\gamma[n] \tau \longrightarrow \text{int}[i] \longrightarrow \{\gamma \rightarrow \beta\} \text{unit} \{\gamma \rightarrow \beta \cup [i, n]\})$$

Let start with read and update from index k.

$$\frac{\Delta; \Phi; \Gamma \vdash a \ominus a \lesssim 0 : \text{Array}_\gamma[n] \tau \quad \Delta; \Phi; \Gamma \vdash k \ominus k \lesssim 0 : \text{int}[i] \quad \Delta; \Phi \models i \leq n \quad \Delta; \vdash \text{empty} \quad wf}{\Delta; \Phi; \Gamma \vdash \text{read } a \ominus \text{read } a \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ . \tau \{\gamma \rightarrow \beta\}} \text{R-R}$$

and

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

where $\Delta = n : \mathbb{N}, i : \mathbb{N}, r : \mathbb{R}$

$\Sigma = \gamma : \mathbb{L}$

$\Phi = i \leq n$

$\Gamma = a : \text{Array}_\gamma[n] \tau, k : \text{int}[i], f : (\tau \xrightarrow{\text{diff}(r)} \tau),$

$\text{map} : (\tau \xrightarrow{\text{diff}(r)} \tau) \rightarrow \forall n, i, \gamma, \beta. (i \leq n) \supset (\text{Array}_\gamma[n] \tau \rightarrow \text{int}[i] \rightarrow \text{int}[n] \{\gamma \rightarrow \beta\} \text{unit} \{\gamma \rightarrow \beta \cup [i, n]\}).$

We will show:

$$\frac{\text{S-RM} \quad \frac{\frac{\text{diff}(0)}{\vdash \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\} \sqsubseteq \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}}{\Delta; \Phi; \Gamma \vdash \text{return } () \ominus \text{return } () \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta\}} \text{R-EXEC}}{\Delta; \Phi; \Gamma \vdash \text{return } () \ominus \text{return } () \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}} \text{R-EXEC}}{\Delta; \Phi; \Gamma \vdash k < n \ominus k < n \lesssim 0 : \text{unit} + \text{unit} \quad \Delta; \Phi; \Gamma \vdash T \ominus T \lesssim 0 : \{\gamma \rightarrow S\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}} \text{R-CASE}}{\Delta; \Phi; \Gamma, \{x : \tau\} \vdash V \ominus V \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}} \text{R-CASE}}$$

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

$$\frac{\Delta; \Phi; \Gamma \vdash \text{read } a \ k \ominus \text{read } a \ k \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{int} \{\gamma \rightarrow \beta\} \quad \Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}}{\Delta; \Phi; \Gamma, \{x : \text{int}\} \vdash T \ominus T \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}} \text{R-LET}$$

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 $\{\gamma \rightarrow \beta\}$ and $P_2 = \text{empty}, Q_1 = \{\gamma \rightarrow \beta \cup \{i\}\}, Q_2 = \text{empty}, Q = \{\gamma \rightarrow \beta \cup \{i\} \cup [i+1, n]\}$, we get

$$\frac{\Delta; \Phi; \Gamma \vdash \text{updt } a \ k \ (f \ x) \ominus \text{updt } a \ k \ (f \ x) \lesssim 0 : \{\gamma \rightarrow \beta\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup \{i\}\} \quad \Delta; \Phi; \Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \rightarrow \beta \cup \{i\}\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup \{i\} \cup [i+1, n]\}}{\Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow S\} \exists _ : \text{unit}_r \{\gamma \rightarrow \beta \cup [i, n]\}} \text{R-LET}$$

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

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

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

3.2 MergeSort

```

copy = fix cp ().Λ.Λ.Λ.Λ.Λ.λx.λy.λλ.λu.
  if l ≤ u then
    let {a} = ( read x l ) in
    let {_} = ( updt y l a ) in
    (celim(cp() [] [] [] [] [] x y (l + 1) u))
  else return()

```

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

$\vdash \text{copy} \ominus \text{copy} \lesssim 0$:

$$\forall n, L, U, \gamma_1, \gamma_2, S, S'. L \leq U \leq n \wedge S' \cap [L, U] = \emptyset \supset$$

$$\text{Array}_{\gamma_1}[n] U(\text{int}, \text{int}) \rightarrow \text{Array}_{\gamma_2}[n] U(\text{int}, \text{int})$$

$$\rightarrow \text{int}[L] \rightarrow \text{int}[U] \rightarrow \{\gamma_1 \rightarrow S, \gamma_2 \rightarrow S'\} \exists_{\text{diff}(0)} \text{unit} \{\gamma_1 \rightarrow S, \gamma_2 \rightarrow \mathbb{N}\}$$

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

$\vdash \text{copy} \ominus \text{copy} \lesssim 0$:

$$\forall n, L, U, \gamma_1, \gamma_2, S. L \leq U \leq n \supset \text{Array}_{\gamma_1}[n] U(\text{int}, \text{int}) \rightarrow \text{Array}_{\gamma_2}[n] U(\text{int}, \text{int})$$

$$\rightarrow \text{int}[L] \rightarrow \text{int}[U] \rightarrow \{\gamma_1 \rightarrow S, \gamma_2 \rightarrow \mathbb{N}\} \exists_{\text{diff}(0)} \text{unit} \{\gamma_1 \rightarrow S \cup [L, U], \gamma_2 \rightarrow \mathbb{N}\}$$

```

mergeip = fix mg (λ. (k : int[K]). λ a : Arrayγ1 [n] int .
    λ ls : int[i], λ le : int[j], λ rs : int[m], λ re : int[u]. λ b : Arrayγ2 [n] int .
    if ls < le then
        if rs < re then
            let {x} = (read a ls) in
            let {y} = (read a rs) in
            if x < y then
                (let {} = (updt b k x) in
                 ((celim mg () (k+1)) a (ls+1) le rs re b))
            else
                (let {} = (updt b k y) in
                 ((celim mg () (k+1)) a ls le (rs+1) re, b))
        else
            (let {x} = (read a ls) in
             let {} = (updt b k x) in
             ((celim mg () (k+1)) a (ls+1) le rs re b))
    else
        if rs < re
            (let {x} = (read a rs) in
             let {} = (updt b k x) in
             ((celim mg () (k+1)) a ls le (rs+1) re b))
        else (return())

```

$$\vdash \text{merge}_{ip} : \forall n, l_s, l_e, r_s, r_e, K, \gamma, \gamma'. l_s \leq l_e \leq r_s \leq r_e \leq n \supset \text{unit}_r \rightarrow \text{int}[K] \rightarrow \text{Array}_\gamma [n] \text{int} \rightarrow \text{int}[l_s] \rightarrow \text{int}[l_e] \rightarrow \text{int}[r_s] \rightarrow \text{int}[r_e] \rightarrow \text{Array}_{\gamma'} [n] \text{int} \rightarrow \{\gamma \rightarrow \emptyset, \gamma' \rightarrow \overset{\text{exec}(U,L)}{[K, r_e]}\} \exists _ . \text{unit} \{\gamma, \gamma'\}$$

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

$$U = (c_r + c_r + c_u) * ((l_e - l_s) + (r_e - r_s))$$

In merge_{ip} , 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_{ip} , 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 $S \cap [K, r_e] = \emptyset$

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

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 .

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

and

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

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

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

Its type is $\{\gamma \rightarrow S, \gamma' \rightarrow N\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma' \rightarrow N\}$. 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.

$$\frac{\begin{array}{c} \Delta; \Phi; |\Gamma|_1 \vdash_0^0 \text{merge}_{lp} () \ k \ a \ l_s \ l_e \ r_s \ r_e \ b : \{\gamma \rightarrow \emptyset, \gamma' \rightarrow [K, r_e]\} \exists _ : \text{unit} \{\gamma, \gamma'\} \\ \Delta; \Phi; |\Gamma|_2 \vdash_0^0 \text{merge}_{lp} () \ k \ a \ l_s \ l_e \ r_s \ r_e \ b : \{\gamma \rightarrow \emptyset, \gamma' \rightarrow [K, r_e]\} \exists _ : \text{unit} \{\gamma, \gamma'\} \end{array}}{\Delta; \Phi; \Gamma \vdash \text{merge}_{lp} () \ k \ a \ l_s \ l_e \ r_s \ r_e \ b \ominus \text{merge}_{lp} () \ k \ a \ l_s \ l_e \ r_s \ r_e \ b \lesssim 0 :} \text{R-S} \xrightarrow{\text{diff}(0)}$$

$$U(\{\gamma \rightarrow \emptyset, \gamma' \rightarrow [K, r_e]\} \exists _ : \text{unit} \{\gamma, \gamma'\}, \{\gamma \rightarrow \emptyset, \gamma' \rightarrow [K, r_e]\} \exists _ : \text{unit} \{\gamma, \gamma'\})$$

and we know from S-RUM.

$$\begin{aligned} & \models U(\{\gamma, \gamma' \rightarrow [K, r_e]\} \exists _ : \text{unit} \{\gamma, \gamma'\}, \{\gamma, \gamma' \rightarrow [L, U]\} \exists _ : \text{unit} \{\gamma, \gamma'\}) \sqsubseteq \\ & \{\gamma \rightarrow S, \gamma' \rightarrow S'\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma' \rightarrow S' \cup [K, r_e]\} \end{aligned}$$

$$\begin{aligned} & \forall n, l_s, l_e, r_s, r_e, K, \gamma, \gamma'. l_s \leq l_e \leq r_s \leq r_e \leq n \supset \text{unit}_r \rightarrow \text{int}[K] \rightarrow \text{Array}_\gamma[n] U(\text{int}, \text{int}) \\ \vdash \text{merge}_{lp} \ominus \text{merge}_{lp} \lesssim 0 : & \rightarrow \text{int}[l_s] \rightarrow \text{int}[l_e] \rightarrow \text{int}[r_s] \rightarrow \text{int}[r_e] \rightarrow \text{Array}_{\gamma'}[n] U(\text{int}, \text{int}) \\ & \xrightarrow{\text{diff}((U-L))} \rightarrow \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N} \cup [K, r_e]\} \end{aligned}$$

Function merge

$$\begin{aligned} \text{merge} &= \lambda a. \lambda b. \lambda l. \lambda u. \lambda \text{mid}. \\ &\quad \text{let } \{_ \} = (\text{celim } (\text{merge}_{lp} \ 0 \ l) \ a \ l \ \text{mid} \ \text{mid} + 1 \ u \ b) \ \text{in} \\ &\quad (\text{celim } \text{copy } l) \ a \ b \ u). \end{aligned}$$

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

$$\begin{aligned} \vdash \text{merge} \ominus \text{merge} \lesssim 0 : & \quad \forall n, l, m, u, \gamma, \gamma', S. l \leq m \leq u \leq n \wedge S \cap [l, u] = \emptyset \supset \text{Array}_{\gamma}[n] \ U(\text{int}, \text{int}) \xrightarrow{\text{diff}(0)} \\ & \quad \text{Array}_{\gamma'}[n] \ U(\text{int}, \text{int}) \rightarrow \text{int}[l] \rightarrow \text{int}[m] \rightarrow \text{int}[u] \rightarrow \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \\ \text{R-LET} & \quad \vdash \text{celim } \text{merge}_{lp} \ l \ a \ l \ m \ (m+1) \ u \ b \ominus \text{celim } \text{merge}_{lp} \ l \ a \ l \ m \ (m+1) \ u \ b \lesssim 0 : \\ & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N} \setminus [l, u]\} \\ \Delta; \Phi; \Gamma \vdash \text{celim } \text{copy } l \ a \ b \ u \ominus \text{celim } \text{copy } l \ a \ b \ u \lesssim 0 : & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N} \setminus [l, u]\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N} \setminus [l, u]\} \\ \Delta; \Phi; \Gamma \vdash \text{merge } a \ b \ l \ m \ u \ominus \text{merge } a \ b \ l \ m \ u \lesssim 0 : & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \\ \text{we know: } \models \mathbb{N} / [l, u] \sqsubseteq \mathbb{N}, \text{ By } S\text{-RM, we can get the post condition of } \text{merge} \text{ to be} & \end{aligned}$$

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

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

$$\begin{aligned} \vdash \text{merge} \ominus \text{merge} \lesssim 0 : & \quad \forall n, l, m, u, \gamma, \gamma', S. l \leq m \leq u \leq n \supset \text{Array}_{\gamma}[n] \ U(\text{int}, \text{int}) \rightarrow \text{Array}_{\gamma'}[n] \ U(\text{int}, \text{int}) \rightarrow \text{int}[l] \rightarrow \\ & \quad \text{int}[m] \rightarrow \text{int}[u] \rightarrow \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \text{R-LET} & \quad \Delta; \Phi; \Gamma \vdash \text{celim } \text{merge}_{lp} \ l \ a \ l \ m \ (m+1) \ u \ b \ominus \text{celim } \text{merge}_{lp} \ l \ a \ l \ m \ (m+1) \ u \ b \lesssim 0 : \\ & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N} \cup [l, u]\} \\ \Delta; \Phi; \Gamma \vdash \text{celim } \text{copy } l \ a \ b \ u \ominus \text{celim } \text{copy } l \ a \ b \ u \lesssim 0 : & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash \text{merge } a \ b \ l \ m \ u \ominus \text{merge } a \ b \ l \ m \ u \lesssim 0 : & \quad \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \end{aligned}$$

msort

$$\begin{aligned} \text{msort} &= \text{fix } \text{sort} \ (a : \text{Array}_{\gamma_1}[n] \ \tau) \ b : \text{Array}_{\gamma_2}[n] \ \tau \ . \lambda l. \lambda u. \\ &\quad \text{if } (l \geq u) \ \text{then} \\ &\quad \quad (\text{return } 0) \\ &\quad \text{else } \left(\right. \\ &\quad \quad \text{let } _ = (\text{celim } (\text{sort}) \ a \ b \ l \ (l + \lfloor (u-l)/2 \rfloor)) \ \text{in} \\ &\quad \quad \text{let } _ = (\text{celim } (\text{sort}) \ a \ b \ (l + \lfloor (u-l)/2 \rfloor + 1) \ u) \ \text{in} \\ &\quad \quad \left. (\text{celim } \text{merge} \ a \ b \ l \ (l + \lfloor (u-l)/2 \rfloor) \ u) \right) \\ \vdash \quad \text{msort} \quad : & \quad \forall n, l, u, \gamma, \gamma'. l \leq u \leq n \supset \text{Array}_{\gamma}[n] \ \text{int} \\ & \quad \rightarrow \text{int}[l] \rightarrow \text{int}[u] \rightarrow \text{Array}_{\gamma'}[n] \ \text{int} \rightarrow \{\gamma \rightarrow [l, u], \gamma' \rightarrow [l, u]\} \exists _ . \text{unit} \{\gamma, \gamma'\} \end{aligned}$$

Let us call M the following term:

$$\text{celim } \text{merge } a b l (l + \lfloor (u-l)/2 \rfloor) u$$

and call $S1$ the following term:

$$\text{celim } (\text{sort}) a b l l + \lfloor (u-l)/2 \rfloor$$

and call $S2$ the following term:

$$\text{celim } (\text{sort}) a b l + \lfloor (u-l)/2 \rfloor + 1 u$$

and call Z the following term:

$$\text{let } _ = S1 \text{ in let } _ = S2 \text{ in } M$$

and set V as:

$$\text{if } l = u, \text{ then (return ()) else } Z$$

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

$$\begin{array}{c} \forall n, l, u, \gamma, \gamma' : \mathbb{N}, S.l \leq u \leq n \wedge S \cap [l, u] = \emptyset \supset \text{Array}_{\gamma}[n] U(\text{int}, \text{int}) \rightarrow \text{int}[l] \rightarrow \text{int}[u] \\ \vdash \text{msort} \ominus \text{msort} \lesssim 0 : \text{Array}_{\gamma'}[n] U(\text{int}, \text{int}) \rightarrow \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash S2 \ominus S2 \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \quad P_2 = \text{empty} \\ \hline \Delta; \Phi; \Gamma \vdash \text{let } _ = S2 \text{ in } M \ominus \text{let } _ = S2 \text{ in } M \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash S1 \ominus S1 \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \quad P_2 = \text{empty} \\ \hline \Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \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 \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \\ \hline \Delta; \Phi; \Gamma \vdash V \ominus V \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \end{array}$$

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

$$\begin{array}{c} \forall n, l, u, \gamma, \gamma' : \mathbb{N}, S.l \leq u \leq n \supset \text{Array}_{\gamma}[n] U(\text{int}, \text{int}) \rightarrow \text{int}[l] \rightarrow \text{int}[u] \\ \vdash \text{msort} \ominus \text{msort} \lesssim 0 : \text{Array}_{\gamma'}[n] U(\text{int}, \text{int}) \rightarrow \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \end{array}$$

where

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

S-RM

$$\begin{array}{c} \vdash \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \sqsubseteq \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash \text{return } () \ominus \text{return } () \lesssim 0 : \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \hline \Delta; \Phi; \Gamma \vdash \text{return } () \ominus \text{return } () \lesssim 0 : \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \Delta; \Phi; \Gamma \vdash l = u \ominus l = u \lesssim 0 : \text{unit} + \text{unit} \\ \Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \\ \hline \Delta; \Phi; \Gamma \vdash V \ominus V \lesssim 0 : \{\gamma \rightarrow S, \gamma' \rightarrow \mathbb{N}\} \exists _ : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma' \rightarrow \mathbb{N}\} \end{array}$$

$$\begin{array}{c}
P_2 = \text{empty} \\
\frac{\Delta; \Phi; \Gamma \vdash S2 \ominus S2 \lesssim 0 : \{\gamma \rightarrow S \cup [l, \lfloor (u-l)/2 \rfloor], \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma \rightarrow \mathbb{N}\}}{\Delta; \Phi; \Gamma \vdash M \ominus M \lesssim 0 : \{\gamma \rightarrow S \cup [l, u], \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma \rightarrow \mathbb{N}\}} \text{R-LET} \\
\frac{\Delta; \Phi; \Gamma \vdash \text{let}_- = S2 \text{ in } M \ominus \text{let}_- = S2 \text{ in } M \lesssim 0 : \{\gamma \rightarrow S \cup [l, \lfloor (u-l)/2 \rfloor], \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma \rightarrow \mathbb{N}\}}{\Delta; \Phi; \Gamma \vdash S1 \ominus S1 \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, \lfloor (u-l)/2 \rfloor], \gamma \rightarrow \mathbb{N}\}} \text{R-LET} \\
\frac{P_2 = \text{empty} \quad \Delta; \Phi; \Gamma \vdash S1 \ominus S1 \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, \lfloor (u-l)/2 \rfloor], \gamma \rightarrow \mathbb{N}\}}{\Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma \rightarrow S, \gamma \rightarrow \mathbb{N}\} \exists_- : \text{unit}_r \{\gamma \rightarrow S \cup [l, u], \gamma \rightarrow \mathbb{N}\}} \text{R-LET}
\end{array}$$

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 :

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

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 $n=u-l$, $\alpha = |S \cap [l, l + \lfloor \frac{u-l}{2} \rfloor]|$ 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 $S \cap [l, u] \neq \emptyset$

Set $\Gamma = _ : \text{unit}, f : \text{unit} \xrightarrow{\text{diff}(0)} \sigma, f U(A_1, A_2)$

$$\frac{\Gamma \vdash \Lambda \dots \lambda s \dots \ominus \Lambda \dots \lambda s \dots \lesssim 0 : \sigma \quad |\Gamma|_1 \vdash_0^0 \text{Fix } f(_). \Lambda \dots \lambda s \dots : A_1 \quad |\Gamma|_2 \vdash_0^0 \text{Fix } f(_). \Lambda \dots \lambda s \dots : A_2}{\Gamma \vdash \text{Fix } f(_). \Lambda \dots \ominus \text{Fix } f(_). \Lambda \dots \lesssim 0 : \text{unit} \xrightarrow{\text{diff}(0)} \sigma} \text{R-FIX-EXT}$$

We first show how do we get the unary type A_1 .

Set $A_1 = \text{unit} \rightarrow B_1$. So $B_1 = \forall \gamma_1 \gamma_2, \gamma_3 : \mathbb{L} \dots$

Set $\Omega = |\Gamma|_1 = _ : \text{unit}, f : A_1$.

unary type

First, we apply U-fix rule.

$$\frac{\Omega \vdash_0^0 \Lambda \dots \lambda S \dots : B_1}{\Omega \vdash_0^0 \text{Fix } f(_). \Lambda \dots : A_1} \text{U-FIX}$$

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, $\Omega = s : \text{Array}_{\gamma_1} [N] \text{int}, w : \text{Array}_{\gamma_2} [Q] \text{int}, m : \text{int}[M], i : \text{int}[I], l_w : \text{int}[Q], p : \text{Array}_{\gamma_3} [N] \text{int}$. we will show:

$$\frac{\{\gamma_3 \rightarrow \{M\}\} = \{\gamma_3 \rightarrow \{M\}\} \star \text{empty} \quad \vdash_0^0 \text{read } s(m+i) : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{int} \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_r, c_r)}{\Delta; \Phi; a : \text{int}, \Omega \vdash_0^0 \text{let } \{b\} = \text{read } b(i) \dots : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{int} \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_r + c_u, c_r + (Q-I-1)*r + c_u)}{\Delta; \Phi; \Omega \vdash \text{let } \{a\} = \text{read } s(m+i) \dots : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{unit} \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u + r, (Q-I)*r + c_u)} \text{U-LET}$$

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.

$$\frac{\Delta; \Phi; \Omega' \vdash_0^0 \text{if } (a == b)(1) : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{int} \{\gamma_3 \rightarrow \mathbb{N}\} \quad \text{exec}(c_u, (Q-I-1)*r + c_u)}{\Delta; \Phi; \Omega' \vdash_0^0 \text{if } (a == b)(2) : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{int} \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, (Q-I-1)*r + c_u)}{\Delta; \Phi; \Omega' \vdash_0^0 \text{if } (i+1) == l_w \text{ then if } (a == b)(1) \text{ else if } (a == b)(2) : \{\gamma_3 \rightarrow \{M\}\} \exists _ \text{int} \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, (Q-I-1)*r + c_u)} \text{U-CASE}$$

For if (2):

$$\frac{\Delta; \Phi; \Omega' \vdash_0^0 \text{celim}(f \ 0 \ \square \ \square \ \square \ \square \ \square \ \square \ \square \ \square) \ w \ m \ (i+1) \ l_w \ p : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, (Q-I-1)*r+c_u)}{\Delta; \Phi; \Omega' \vdash_0^0 \text{updt} \ p \ m \ 0 : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \vdash_0^0 \ a == b : \text{bool}} \text{U-CASE}$$

$$\Delta; \Phi; \Omega' \vdash_0^0 \text{if} \ (a == b) \ \text{then} \ \text{celim}(f \ 0 \ \square \ \square \ \square \ \square \ \square \ \square \ \square \ \square) \ w \ m \ (i+1) \ l_w \ p$$

$$\text{else} \ \text{updt} \ p \ m \ 0 : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, (Q-I-1)*r+c_u)$$

By applying U-app, U-lapp, U-celim, we get the type for $\text{celim}(f \ 0 \ \square \ \square \ \square \ \square \ \square \ \square \ \square \ \square) \ w \ m \ (i+1) \ l_w \ p$ is $\{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\}$. Along with the constraint $I+1 < Q < N \wedge M+I+1 < N$.

For the other branch, the term $\text{updt} \ p \ m \ 0$. We use U-X and subtyping rule S-A-MONAD.

$$\frac{\Delta; \Phi; \Omega' \vdash_0^0 \text{updt} \ p \ m \ 0 : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, c_u)}{\Delta; \Phi \models \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \sqsubseteq \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \Delta \models 0 \leq 0 \quad \Delta \models 0 \leq 0} \text{U-EXEC}$$

$$\Delta; \Phi; \Omega' \vdash_0^0 \text{updt} \ p \ m \ 0 : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{int} \ \{\gamma_3 \rightarrow \{M\}\} \quad \text{exec}(c_u, (Q-I-1)*r+c_u)$$

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 $\tau = \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}$.

To show the type of part Z , we will use the rule R-SPLIT first.

$$\frac{\Sigma; \Delta; \Phi \wedge C \wedge I \notin \beta_2; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau \dots (\text{Case 1}) \quad \Sigma; \Delta; \Phi \wedge C \wedge I \in \beta_2; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau \dots (\text{Case 2})}{\Sigma; \Delta; \Phi \wedge C; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau} \text{R-SPLIT}$$

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

$$\frac{\Delta; \Phi'; \Gamma \vdash a \ominus a \lesssim 0 : \text{Array}_\gamma[n] \ \tau \quad \Delta; \Phi'; \Gamma \vdash k \ominus k \lesssim 0 : \text{int}[i] \quad \Delta; \Phi' \models i \leq n \quad \Delta; \vdash \text{empty} \ w \ f}{\Delta; \Phi'; \Gamma \vdash \text{read} \ S \ (m+i) \ \ominus \ \text{read} \ S \ (m+i) \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{int} \ \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\}} \text{diff}(0)$$

Set R as the outer if if $(i+1 == l_w)$ then (1) else (2).

Set V as let $b = \text{read} \ W \ i$ in R .

$$\begin{array}{c}
\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 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}}{\Delta; \Phi; \Gamma \vdash (2) \ominus (2) \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}} \text{R-CASE} \\
\frac{\Delta; \Phi; \Gamma \vdash R \ominus R \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}}{\Delta; \Phi; \Gamma \vdash \text{read } W \ i \ominus \text{read } W \ i \lesssim 0 : \{\gamma_2 \rightarrow \beta_2\} \exists _ : \text{int} \{\gamma_2 \rightarrow \beta_2\} \quad P_2 = \text{empty}} \text{R-LET} \\
\frac{\Delta; \Phi; \Gamma \vdash V \ominus V \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}}{\Delta; \Phi; \Gamma \vdash \text{read } S \ (m + i) \ominus \text{read } S \ (m + i) \lesssim 0 : \{\gamma_1 \rightarrow \emptyset\} \exists _ : \text{int} \{\gamma_1 \rightarrow \emptyset\} \quad P_2 = \text{empty}} \text{R-LET} \\
\Delta; \Phi; \Gamma \vdash Z \ominus Z \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}
\end{array}$$

(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 $(Q - 1 - \min(\text{MIN}(\beta_2 \cap [I, \infty)), Q - 1)) * r$ by using relational typing rules R-APP and R-IAPP. Because we already know $i \notin \beta$, it is easy to infer that $\text{MIN}(\beta_2 \cap [I, \infty)) = \text{MIN}(\beta_2 \cap [I + 1, \infty))$.

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 $\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}$. Because $i \in \beta_2$, $(Q - 1 - \min(\text{MIN}(\beta_2 \cap [I, \infty)), Q - 1)) = (Q - I - 1)$.

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:

$$\text{exec}(c_u, (Q - I - 1) * r + c_u) \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\}$$

By applying unary subtyping rule S-UM, we know the following unary typing used in R-S.

$$\frac{\Delta; \Phi; |\Gamma|_1 \vdash_0^0 (1) : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\} \quad \Delta; \Phi; |\Gamma|_2 \vdash_0^0 (1) : \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\}}{\Delta; \Phi; \Gamma \vdash (1) \ominus (1) \lesssim 0 : \quad U(\{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\}, \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\})} \text{R-S}$$

$$\begin{aligned}
& \models U(\{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\}, \{\gamma_3 \rightarrow \{M\}\} \exists _ . \text{unit} \{\gamma_3 \rightarrow \{M\}\}) \sqsubseteq \\
& \quad \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . U(\text{unit}, \text{unit}) \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}
\end{aligned}$$

By subtyping rule S-R-UNIT, we get the relational type.

NSS function

```

NSS = fix F (S). λW.λm.λls.λlw.λP.
  if (m + lw) ≤ ls then
    let {_} = (celim search SW m 0 ls lw P) in
      F(S) W (m + 1) ls lw P
  else
    return ()

```

$$\vdash \text{NSS} \ominus \text{NSS} \lesssim 0 : \forall M. \forall \gamma_1, \gamma_2, \gamma_3 : \mathbb{L}. \forall Q, N : \mathbb{N}. (Q \leq N \wedge M + Q \leq N) \subset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow$$

$$\text{Array}_{\gamma_2} [Q] \text{int} \rightarrow \text{int} [M] \rightarrow \text{int} [N] \rightarrow \text{int} [Q] \rightarrow \text{Array}_{\gamma_3} [N] \text{int} \rightarrow$$

$$\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup [M, N]\}$$

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 $(Q \leq N \wedge M + Q \leq N)$ into the constraint environment Φ , the cost comes from the first branch.

Set K as (celim helper $SW\ m\ 0\ l_s\ l_w\ P$)

Set Q as $F(S)\ W\ (m + 1)\ l_s\ l_w\ P$.

Set X as let $_ =$ (celim helper $SW\ m\ 0\ l_s\ l_w\ P$) in
 $F(S)\ W\ (m + 1)\ l_s\ l_w\ P$.

R-LET

$$\{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} = \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} * \text{empty}$$

$$\Delta; \Phi; \Gamma \vdash K \ominus K \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\}$$

$$\Delta; \Phi; \Gamma \vdash Q \ominus Q \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup \{M\} \cup [M + 1, N]\}$$

$$\Delta; \Phi; \Gamma \vdash X \ominus X \lesssim 0 : \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \emptyset, \gamma_2 \rightarrow \beta_2, \gamma_3 \rightarrow \beta_3 \cup [M, N]\}$$

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 $\{\gamma_1 \rightarrow \beta_1\} \exists_{\perp}.U(\text{bool}, \text{bool}) \{\gamma_1 \rightarrow \beta_1\}$ for the reason that under the assumption of $M \in \beta_1$, $(N - 1 - \min(\text{MIN}(\beta_1 \cap [M, \infty)), N)) = (N - 1 - M)$.

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:

$$\{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\}^{\text{exec}(3, (N-M)*3+1)}$$

By applying unary subtyping rule S-UM, we know the following unary typing used in R-S.

$$\frac{\Delta; \Phi; |\Gamma|_1 \vdash_0^0 (1) : \{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\}^{\text{exec}(3, (N-M)*3+1)} \quad \Delta; \Phi; |\Gamma|_2 \vdash_0^0 (1) : \{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\}^{\text{exec}(3, (N-M)*3+1)}}{\Delta; \Phi; \Gamma \vdash (1) \ominus (1) \lesssim 0 : \quad U(\{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\}, \{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\})^{\text{exec}(3, (N-M)*3+1)}} \text{R-S}$$

$$\begin{aligned} & \models U(\{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\}, \{\gamma_1 \rightarrow \emptyset\} \exists_{\perp}. \text{bool} \{\gamma_1 \rightarrow \emptyset\})^{\text{exec}(3, (N-M)*3+1)} \sqsubseteq \\ & \quad \{\gamma_1 \rightarrow \beta_1\} \exists_{\perp}. U(\text{bool}, \text{bool}) \{\gamma_1 \rightarrow \beta_1\}^{\text{diff}((N-1-M)*3+1)} \end{aligned}$$

3.5 Boolean Or, two implementations

We have two implementations for the above Boolean or functions.//

```

fix BoolOr1 (□).Λ.Λ.Λ.Λ.λs. λm.λls.
  if m < ls then
    let {a} = read s m in
      if a then
        return true
      else
        (celim BoolOr1() [] [] [] [] s (m + 1) ls)
  else
    return false

```

The unary type of BoolOr1 as follows.

$$\vdash \text{BoolOr1} : \text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall M, N : \mathbb{N}. (M \leq N) \supset \text{Array}_{\gamma_1} [N] \text{bool} \rightarrow \text{int}[M] \rightarrow \text{int}[N] \rightarrow \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\}$$

The following BoolOr2 is another implementation of BoolOr.

```

fix BoolOr2 (□).Λ.Λ.Λ.Λ.λs. λm.λls.
  if m < ls then
    let {a} = read s m in
      if (celim BoolOr2() [] [] [] [] s (m + 1) ls) then
        return true
      else
        return a
  else
    return false

```

With its unary type presented below.

$$\vdash \text{BoolOr2} : \text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall M, N : \mathbb{N}. (M \leq N) \supset \text{Array}_{\gamma_1} [N] \text{bool} \rightarrow \text{int}[M] \rightarrow \text{int}[N] \rightarrow \text{exec}(3 * (N - M) + 1, 3 * (N - M) + 1) \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\}$$

We can obtain the relational type with relative cost 0.

$$\vdash \text{BoolOr1} \ominus \text{BoolOr2} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall M, N : \mathbb{N}. \forall \beta_1. (M \leq N) \supset U(\text{Array}_{\gamma_1} [N] \text{bool}, \text{Array}_{\gamma_1} [N] \text{bool}) \xrightarrow{\text{diff}(0)} U(\text{int}[M], \text{int}[M]) \rightarrow U(\text{int}[N], \text{int}[N]) \rightarrow \{\gamma_1 \rightarrow \beta_1\} \exists _ . U(\text{bool}, \text{bool}) \{\gamma_1 \rightarrow \beta_1\}$$

The derivation mainly relies on the switch rules.

$$\frac{\begin{array}{l} \Delta; \Phi; |\Gamma|_1 \vdash_0^0 \text{BoolOr1} : A \rightarrow \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\} \\ \Delta; \Phi; |\Gamma|_2 \vdash_0^0 \text{BoolOr2} : A \rightarrow \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\} \end{array}}{\Delta; \Phi; \Gamma \vdash \text{BoolOr1} \ominus \text{BoolOr2} \lesssim 0 : U(A \rightarrow \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\}, A \rightarrow \{\gamma_1 \rightarrow \emptyset\} \exists _ . \text{bool} \{\gamma_1 \rightarrow \emptyset\})} \text{R-S}$$

We set A to be $\text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall M, N : \mathbb{N}. (M \leq N) \supset \text{Array}_{\gamma_1}[N] \text{ bool} \rightarrow \text{int}[M] \rightarrow \text{int}[N] \rightarrow$.

We set A_1 to be $\{\gamma_1 \rightarrow \emptyset\} \exists_{\text{diff}(0)} \text{bool} \{\gamma_1 \rightarrow \emptyset\}$. set A_2 to be $\{\gamma_1 \rightarrow \emptyset\} \exists_{\text{diff}(0)} \text{bool} \{\gamma_1 \rightarrow \emptyset\}$.

We set τ to be $\text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall M, N : \mathbb{N}. \forall \beta_1. (M \leq N) \supset U(\text{Array}_{\gamma_1}[N] \text{ bool}, \text{Array}_{\gamma_1}[N] \text{ bool}) \rightarrow U(\text{int}[M], \text{int}[M]) \rightarrow U(\text{int}[N], \text{int}[N])$.

We set τ' to be $\{\gamma_1 \rightarrow \beta_1\} \exists_{\text{diff}(0)} U(\text{bool}, \text{bool}) \{\gamma_1 \rightarrow \beta_1\}$.

Use the subtyping rules S-R-FORALL-U, S-R-CIMPL-U and S-R-EXECDIFF, we can show that

$$\models U(A \rightarrow A_1, A \rightarrow A_2) \sqsubseteq \tau \rightarrow U(A_1, A_2)$$

Next, using the monadic sutyping S-RUM, we can show that

$$\begin{aligned} \models U(\{\gamma_1 \rightarrow \emptyset\} \exists_{\text{diff}(0)} \text{bool} \{\gamma_1 \rightarrow \emptyset\}, \{\gamma_1 \rightarrow \emptyset\} \exists_{\text{diff}(0)} \text{bool} \{\gamma_1 \rightarrow \emptyset\}) \sqsubseteq \\ \{\gamma_1 \rightarrow \beta_1\} \exists_{\text{diff}(0)} U(\text{bool}, \text{bool}) \{\gamma_1 \rightarrow \beta_1\} \end{aligned}$$

Finally, using the rule R-EXEC, we can derive the relational type shown above.

3.6 Insertion sort

```

fix ISort ().Λ.Λ.Λ.Λ.λs.λi.λls.
  if i < ls then
    let {a} = read s i in
    let {b} = celim(insert() [] [] [] []) s a 0 i in
      celim(ISort() [] [] [] [])s(i + 1)ls
  else
    return ()

```

$$\vdash \text{ISort} : \text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, I : \mathbb{N}. (I \leq N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \text{int}[I] \rightarrow \text{int}[N] \rightarrow \text{exec} \left(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1) \right) \{ \gamma_1 \rightarrow \mathbb{N} \} \exists _ . \text{unit} \{ \gamma_1 \rightarrow \mathbb{N} \}$$

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.

$$\vdash \text{ISort} \ominus \text{ISort} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, I : \mathbb{N}. \forall \beta_1. (I \leq N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \text{int}[I] \rightarrow \text{int}[N] \rightarrow \text{diff} \left(\frac{N*(N+1)-k*(k+1)}{2} \right) \{ \gamma_1 \rightarrow \beta_1 \} \exists _ . \text{unit}_r \{ \gamma_1 \rightarrow \mathbb{N} \}$$

where $k = \max(I, \min(\text{MIN}(\beta_1), N))$.

```

fix insert () .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\lambda s$ .  $\lambda a$ .  $\lambda idx$ .  $\lambda i$ .
  let {b} = read s idx in
  if  $a \geq b$  then
    celim(insert () [] [] [] [] s a (idx + 1) i
  else
    let _ = celim(shift () [] [] [] [] s idx (i - 1) in
    updt s idx a

```

$$\vdash \text{insert} : \begin{array}{l} \text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, A, IDX, I : \mathbb{N}. (IDX \leq N \wedge I \leq N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \\ \text{int} \rightarrow \text{int}[IDX] \rightarrow \text{int}[I] \rightarrow \\ \text{exec}(I - IDX + 1, 2 * (I - IDX) + 2) \\ \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\} \end{array}$$

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.

$$\vdash \text{insert} \ominus \text{insert} \lesssim 0 : \begin{array}{l} \text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, A, IDX, I : \mathbb{N}. \forall \beta_1. (IDX \leq N \wedge I \leq N \wedge \beta_1 \cap [IDX, I] = \emptyset) \\ \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \text{int}[M] \rightarrow \text{int}[N] \rightarrow \\ \text{diff}(0) \\ \{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \beta_1\} \end{array}$$

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.

```

fix shift ().Λ.Λ.Λ.Λ.λs.λidx.λi.
  if idx ≤ i then
    let {c} = read s i in
    let {} = updt s i + 1 c in
    celim (shift () [] [] [] []) s idx (i - 1)
  else
    return ()

```

\vdash shift : $\text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, IDX, I : \mathbb{N}. (IDX \leq I \wedge I < N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow$
 $\text{int}[IDX] \rightarrow \text{int}[I] \rightarrow$
 $\text{exec}(2*(I-IDX+1), 2*(IDX-I+1))$
 $\{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}$

where cost 2 comes from one read and one update operation.

\vdash shift \ominus shift $\lesssim 0$: $\text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, IDX, I : \mathbb{N}. (IDX \leq I \wedge I < N) \supset \text{Array}_{\gamma_1} [N] \text{int}$
 $\rightarrow \text{int}[IDX] \rightarrow \text{int}[I] \rightarrow$
 $\text{diff}(0)$
 $\{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \beta_1 \cup [ID]\}$

Let us look at the derivation of the relational type of ISort.

```

fix ISort().Λ.Λ.Λ.Λ.λs.λi.λls.
  if i < ls then
    let {a} = read s i in
    let {b} = celim(insert() [] [] [] [] s a 0 i in
      celim(ISort() [] [] [] [] s(i+1) ls
    else
      return ()

```

$$\begin{aligned}
\vdash \text{ISort} & : \text{unit} \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, I : \mathbb{N}. (I \leq N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \\
& \text{int}[I] \rightarrow \text{int}[N] \rightarrow \text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1)) \\
& \quad \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\} \\
\vdash \text{ISort} \ominus \text{ISort} \lesssim 0 & : \text{unit}_r \rightarrow \forall \gamma_1 : \mathbb{L}. \forall N, I : \mathbb{N}. \forall \beta_1. (I \leq N) \supset \text{Array}_{\gamma_1} [N] \text{int} \\
& \rightarrow \text{int}[I] \rightarrow \text{int}[N] \rightarrow \{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}\} \\
& \quad \text{diff}(\frac{N*(N+1)-k*(k+1)}{2})
\end{aligned}$$

where $k = \max(I, \min(\text{MIN}(\beta_1), N))$.

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 $C = (I \leq N)$ and

$$\tau = \{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}\}.$$

To show the type of part Z , we will use the rule R-SPLIT first.

$$\frac{\begin{array}{l} \Sigma; \Delta; \Phi \wedge C \wedge \beta_1 \cap [0, I] = \emptyset; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau \dots (\text{Case1}) \\ \Sigma; \Delta; \Phi \wedge C \wedge \beta_1 \cap [0, I] \neq \emptyset; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau \dots (\text{Case2}) \end{array}}{\Sigma; \Delta; \Phi \wedge C; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau} \text{R-SPLIT}$$

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.

$$\frac{\Delta; \Phi' \vdash i < l_s \ominus i < l_s \lesssim 0 : \text{unit} + \text{unit} \quad \Delta; \Phi'; \Gamma \vdash (1) \ominus (1) \lesssim 0 : \tau \quad \Delta; \Phi'; \Gamma \vdash (2) \ominus (2) \lesssim 0 : \tau}{\Delta; \Phi'; \Gamma \vdash Z \ominus Z \lesssim 0 : \tau} \text{R-CASE}$$

(a) The relative cost for (2) is 0, return () as a unit operation causes no relative cost. We use sub-typing 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.

Set (3) as let $b = \text{celim}(\text{insert}() \dots) \text{ s a } 0 \text{ i in}$
 $\text{celim}(\text{ISort}() \dots) \text{ s}(i+1) \text{ I}_s$

$$\frac{\Delta; \Phi'; \Gamma \vdash \text{insert} \dots \ominus \text{insert} \dots \lesssim 0 : \{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \beta_1\} \quad \text{diff}(0)}{\Delta; \Phi'; \Gamma \vdash \text{ISort} \dots \ominus \text{ISort} \dots \lesssim 0 : \tau[k/k']} \text{R-LET}$$

$$\Delta; \Phi'; \Gamma \vdash (3) \ominus (3) \lesssim 0 : \tau$$

where $k' = \max(I+1, \min(\text{MIN}(\beta_1), N))$.

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:

$$\frac{\Delta; \Phi; \Omega \vdash_{c_1}^{c_1} \text{insert} \dots : \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\} \quad \text{exec}(I+1, 2I+2)}{\Delta; \Phi; \Omega \vdash_{c_2}^{c_2} \text{ISort} \dots : \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}} \text{U-LET}$$

$$\Delta; \Phi; \Omega \vdash_{c_1+c_2+1}^{c_1+c_2+1} (3) : \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}$$

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 $\{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}\}$.

By applying unary subtyping rule S-UM, we know the following unary typing used in R-S.

$$\frac{\Delta; \Phi; |\Gamma|_1 \vdash_{c_1+c_2+1}^{c_1+c_2+1} (3) : \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\} \quad \text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1))}{\Delta; \Phi; |\Gamma|_2 \vdash_{c_1+c_2+1}^{c_1+c_2+1} (3) : \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}} \text{R-S}$$

$$\Delta; \Phi; \Gamma \vdash (3) \ominus (3) \lesssim 0 :$$

$$U(\frac{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}}, \frac{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}})$$

$$\models U(\frac{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}}, \frac{\text{exec}(\frac{(N+1)*(N+2)-(I+1)*(I+2)}{2}, (2N+1)*(N+1)-(2I+3)*(I+1))}{\{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}}) \sqsubseteq$$

$$\frac{\text{diff}(N*(N+1)-I*(I+1))}{\{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}\}}$$

3.7 Cooley Tukey FFT algorithm

```

fix sp () .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\Lambda$ .  $\lambda x$ .  $\lambda n$ .  $\lambda i$ .  $\lambda y$ .  $\lambda pr$ .
  if  $i < (n/2)$  then
    let { $a$ } = read  $x (pr + 2 * i)$  in
    let { $b$ } = read  $x (pr + 2 * i + 1)$  in
    let { $\_$ } = updt  $y (pr + i)$   $a$  in
    let { $\_$ } = updt  $y (pr + i + n/2)$   $b$  in
      celim(sp () [] [] [] [] [] []  $x n (i + 1) y pr$ )
  else
    return ()

```

\vdash sp : $\text{unit} \rightarrow \forall \gamma_1, \gamma_2 : \mathbb{L}. \forall M, N, I, PR : \mathbb{N}. (I \leq (M/2) \wedge (M + PR) < N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow$
 $\text{int}[M] \rightarrow \text{int}[I] \rightarrow \text{Array}_{\gamma_2} [N] \text{int} \rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$
exec($4 * (M/2 - I), 4 * (M/2 - I)$)

$\text{unit}_r \rightarrow \forall \gamma_1 \gamma_2 : \mathbb{L}. \forall M, N, I, PR : \mathbb{N}. \forall \beta_1. (I \leq (M/2) \wedge (M + PR) < N)$
 \vdash sp \ominus sp \lesssim 0 : $\supset \text{Array}_{\gamma_1} [N] U(\text{int}, \text{int}) \rightarrow \text{int}[M] \rightarrow \text{int}[I] \rightarrow \text{Array}_{\gamma_2} [N] U(\text{int}, \text{int})$
 $\rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$
diff(0)

fix cp($_$). $\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda y.\lambda l.\lambda u.$

if $l \leq u$ then

let $\{a\} = (\text{read } x \ l)$ in

let $\{_ \} = (\text{updt } y \ l \ a)$ in

(celim(cp() [] [] [] [])) $x \ y \ (l + 1) \ u$

else return()

$\vdash \text{cp} : \text{unit} \rightarrow \forall \gamma_1, \gamma_2. \forall L, U, N. (L \leq U \leq N) \supset \text{Array}_{\gamma_1}[N] \text{int} \rightarrow \text{Array}_{\gamma_2}[N] \text{int} \rightarrow$
 $\text{int}[L] \rightarrow \text{int}[U] \rightarrow$
 $\text{exec}(2*(U-L+1), 2*(U-L+1))$
 $\{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$

$\vdash \text{cp} \ominus \text{cp} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1, \gamma_2, \beta_1. \forall L, U, N. (L \leq U \leq N) \supset \text{Array}_{\gamma_1}[N] U(\text{int}, \text{int}) \rightarrow \text{Array}_{\gamma_2}[N] U(\text{int}, \text{int})$
 $\rightarrow \text{int}[L] \rightarrow \text{int}[U] \rightarrow \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$
 $\text{diff}(0)$

separate

fix separate($_$). $\Lambda.\Lambda.\Lambda.\Lambda.\Lambda.\lambda x.\lambda n.\lambda y.\lambda pr.$

let $_ = \text{celim}(\text{sp}() [] [] [] [])$ $x \ n \ 0 \ y \ pr$ in

celim(cp() [] [] [] []) $y \ x \ pr \ (n + pr)$

$\vdash \text{separate} : \text{unit} \rightarrow \forall \gamma_1, \gamma_2. \forall M, N, PR. (M + PR < N) \supset \text{Array}_{\gamma_1}[N] \text{int} \rightarrow$
 $\text{int}[M] \rightarrow \text{Array}_{\gamma_2}[N] \text{int} \rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$
 $\text{exec}(4*M, 4*M)$

$\vdash \text{separate} \ominus \text{separate} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1, \gamma_2, \beta_1. \forall M, N, PR. (M + PR < N) \supset \text{Array}_{\gamma_1}[N] U(\text{int}, \text{int}) \rightarrow$
 $\text{int}[M] \rightarrow \text{Array}_{\gamma_2}[N] U(\text{int}, \text{int}) \rightarrow \text{int}[PR] \rightarrow$
 $\{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$
 $\text{diff}(0)$

```

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 ()

```

$\vdash \text{loop} : \text{unit} \rightarrow \forall \gamma_1. \forall K, M, N, PR. (PR + M < N) \supset \text{int}[K] \rightarrow \text{int}[M] \rightarrow \text{Array}_{\gamma_1} [N] \text{int} \rightarrow$
 $\text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}\}$

$\vdash \text{loop} \ominus \text{loop} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1, \beta_1. \forall K, M, N, PR. (PR + M < N) \supset \text{int}[K] \rightarrow \text{int}[M] \rightarrow$
 $\text{Array}_{\gamma_1} [N] U(\text{int}, \text{int}) \rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \beta_1\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}\}$

```

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 ()

```

$\vdash \text{FFT} : \text{unit} \rightarrow \forall \gamma_1, \gamma_2. \forall M, N, PR. (PR + M < N) \supset \text{Array}_{\gamma_1} [N] \text{int} \rightarrow \text{Array}_{\gamma_2} [N] \text{int} \rightarrow$
 $\text{int}[M] \rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit} \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$

$\vdash \text{FFT} \oplus \text{FFT} \lesssim 0 : \text{unit}_r \rightarrow \forall \gamma_1, \gamma_2, \beta_1. \forall M, N, PR. (PR + M < N) \supset \text{Array}_{\gamma_1} [N] U(\text{int}, \text{int}) \rightarrow$
 $\text{Array}_{\gamma_2} [N] U(\text{int}, \text{int}) \rightarrow \text{int}[M] \rightarrow \text{int}[PR] \rightarrow \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \mathbb{N}\} \exists _ . \text{unit}_r \{\gamma_1 \rightarrow \mathbb{N}, \gamma_2 \rightarrow \mathbb{N}\}$

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.

```

fix sam () . $\Lambda$ . $\Lambda$ . $\Lambda$ . $\Lambda$ . $\Lambda$ . $\Lambda$ . $\Lambda$ . $\lambda x$ . $\lambda a$ . $\lambda i$ . $\lambda n$ .
  if  $i \leq (n - 1)$  then
    let { $b$ } = read  $a$   $i$  in
    let { $r$ } = celim(sam () [] [] [] [] [] []  $x$   $a$  ( $i + 1$ )  $n$  in
      return(func  $x$   $b$   $r$ )
  else
    if  $x = 0$  return 1
    return  $x$ 

```

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 relational 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.

$$\vdash \text{sam} \ominus \text{sam} \lesssim 0: \begin{array}{l} \text{unit}_r \rightarrow \forall \gamma_1. \forall I, N, X, \beta. (I < N) \supset \text{int}[X] \rightarrow \text{Array}_{\gamma_1}[N] U(\text{bool}, \text{bool}) \rightarrow \\ \text{int}[I] \rightarrow \text{int}[N] \rightarrow \{\gamma_1 \rightarrow \beta\} \xrightarrow{\text{diff}(\beta \cap \{I, N\})} \exists _ . U(\text{int}) \{\gamma_1 \rightarrow \beta\} \end{array}$$

function func

```

fix func (x). $\lambda r$ . $\lambda b$ .
  if  $b$  then
    return  $x * r * r$ 
  else
    return  $r * r$ 

```

$$\vdash \text{func} \ominus \text{func} \lesssim 0: U(\text{int}) \rightarrow U(\text{int}) \rightarrow \text{int}_r \xrightarrow{\text{diff}(0)} U(\text{int})$$

$$\vdash \text{func} \ominus \text{func} \lesssim 0: U(\text{int}) \rightarrow U(\text{int}) \rightarrow U(\text{int}) \xrightarrow{\text{diff}(1)} U(\text{int})$$

3.9 Constant-time comparison

```

fix comp (). λ l1. λ l2. λ i. λ n.
  if i < n then
    let {a} = read l1 i in
    let {b} = read l2 i in
    return boolAnd (celim (comp () [] [] [] l1 l2 (i + 1) n, eq (a, b)))
  else
    return true

```

where function boolAnd has the type $(U(\text{bool}) \times U(\text{bool})) \xrightarrow{\text{diff}(0)} U(\text{bool})$, and eq has type $(U(\text{int}) \times U(\text{int})) \xrightarrow{\text{diff}(0)} U(\text{bool})$.

$$\vdash \text{comp} \ominus \text{comp} \lesssim 0 : \begin{array}{l} \text{unit}_r \rightarrow \forall \gamma_1, \gamma_2. \forall I, N, \beta_1, \beta_2. (I \geq 0) \Rightarrow \text{Array}_{\gamma_1} [N] U(\text{int}, \text{int}) \rightarrow \text{Array}_{\gamma_2} [N] U(\text{int}, \text{int}) \rightarrow \\ \text{int}[I] \rightarrow \text{int}[N] \rightarrow \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \beta_2\} \exists _ . \overset{\text{diff}(0)}{U(\text{bool})} \{\gamma_1 \rightarrow \beta_1, \gamma_2 \rightarrow \beta_2\} \end{array}$$

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.

4 Bidirectional type checking

Terms	$t ::= x \mid l \mid n \mid r \mid () \mid \Lambda.t \mid t[] \mid \lambda x.t \mid tu \mid \text{return } t \mid \text{let } \{x\} = t \text{ in } t_1 \mid$ $\text{alloc } t t \mid \text{updt } t t t \mid \text{read } t t \mid \text{inl } t \mid \text{inr } t \mid \text{case } (t, x, t_1, y, t_2) \mid$ $\text{pack } t \mid \text{fix } f(x).t \mid \text{unpack } t_1 \text{ as } x \text{ in } t_2 \mid \text{celim } t$ $\text{switch } t \mid \mathbb{N}C t \mid \text{split } t \text{ with } C \mid \text{contra } t \mid (t : \tau, D) \mid (t : A, L, U) \mid \text{FIXEXT } f(x).t \text{ with } A$
Values	$v ::= n \mid l \mid r \mid () \mid \lambda x.t \mid \Lambda.t \mid \text{return } t \mid \text{alloc } t t \mid \text{updt } t t t \mid \text{read } t t \mid$ $\text{inl } v \mid \text{inr } v \mid \text{let } \{x\} = t \text{ in } t_1 \mid \text{pack } v \mid \Lambda.t \mid \text{fix } f(x).t$

Figure 23: Syntax of values and expressions in the bidirectional type checking version

$\boxed{\Sigma; \Delta; \Psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t_2 \downarrow \tau, D \Rightarrow \Phi}$ 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 .

$\boxed{\Sigma; \Delta; \Psi_a; \Phi_a \models^A A_1 \sqsubseteq A_2 \Rightarrow \Phi}$ 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 24: Bidirectional algorithmic typing judgment explanation

constant integer n

$$\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash n \uparrow \text{int} \Rightarrow [\cdot], 0, 0, \top \text{ alg-u-n-}\uparrow \quad \Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash n \ominus n \uparrow \text{int}_r \Rightarrow [\cdot], 0, \top \text{ alg-r-n-}\uparrow$$

fix

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; f : A_1 \xrightarrow{\text{exec}(L', U')} A_2, x : A_1, \Omega \vdash t \downarrow A_2, L', U' \Rightarrow \Phi}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash \text{fix } f(x).t \downarrow A_1 \xrightarrow{\text{exec}(L', U')} A_2, L, U \Rightarrow \Phi \wedge L \doteq 0 \wedge 0 \doteq U} \text{ alg-u-fix-}\downarrow$$

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; f : \tau_1 \xrightarrow{\text{diff}(D')} \tau_2, x : \tau_1, \Gamma \vdash t \ominus t' \downarrow \tau_2, D' \Rightarrow \Phi}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{fix } f(x).t \ominus \text{fix } f(x).t' \downarrow \tau_1 \xrightarrow{\text{diff}(D')} \tau_2, D \Rightarrow \Phi \wedge 0 \doteq D} \text{ alg-r-fix-}\downarrow$$

$$\frac{\Delta; \psi_a; \Phi_a; |\Gamma| \vdash \text{fix } f(x).t \downarrow A_1, 0, 0 \Rightarrow \Phi_1 \quad \Delta; \psi_a; \Phi_a; |\Gamma| \vdash \text{fix } f(x).t' \downarrow A_2, 0, 0 \Rightarrow \Phi_2 \quad \Sigma; \Delta; \psi_a; \Phi_a; f : \tau_1 \xrightarrow{\text{diff}(D')} \tau_2, f : U(A_1, A_2), x : \tau_1, \Gamma \vdash t \ominus t' \downarrow \tau_2, D' \Rightarrow \Phi \quad \Phi_r = \Phi \wedge \Phi_1 \wedge \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{FIXEXT } f(x).t \text{ with } A_1 \ominus \text{FIXEXT } f(x).t' \text{ with } A_2 \downarrow \tau_1 \xrightarrow{\text{diff}(D')} \tau_2, D \Rightarrow \Phi_r \wedge 0 \doteq D} \text{ alg-fixext}\downarrow$$

variable x

$$\frac{\Omega(x) = A}{\Delta; \psi_a; \Phi_a; \Omega \vdash x \uparrow A \Rightarrow [\cdot], 0, 0, \top} \text{ alg-u-var-}\uparrow \quad \frac{\Gamma(x) = \tau}{\Delta; \psi_a; \Phi_a; \Gamma \vdash x \ominus x \uparrow \tau \Rightarrow [\cdot], 0, \top} \text{ alg-r-var-}\uparrow$$

switch

$$\frac{\Delta; \psi_a; \Phi_a; |\Gamma| \vdash t_1 \uparrow A \Rightarrow [\psi_1], _, U_1, \Phi_1 \quad \Delta; \psi_a; \Phi_a; |\Gamma| \vdash t_2 \uparrow A \Rightarrow [\psi_2], L_2, _, \Phi_2}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{switch } t_1 \ominus \text{switch } t_2 \uparrow U A \Rightarrow [\psi_1, \psi_2], U_1 - L_2, \Phi_1 \wedge \Phi_2} \text{ alg-r-switch}\uparrow$$

$$\frac{L_1, U_1, L_2, U_2 \in \text{fresh}(\mathbb{R}) \quad \Delta; U_1, L_1, \psi_a; \Phi_a; |\Gamma| \vdash t_1 \downarrow A, L_1, U_1 \Rightarrow \Phi_1 \quad \Delta; U_2, L_2, \psi_a; \Phi_a; |\Gamma| \vdash t_2 \downarrow A, L_2, U_2 \Rightarrow \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{switch } t_1 \ominus \text{switch } t_2 \downarrow D, U A \Rightarrow \exists L_1, U_1 :: \mathbb{R}. (\Phi_1 \wedge \exists L_2, U_2 :: \mathbb{R}. \Phi_2 \wedge U_1 - L_2 \doteq D)} \text{ alg-r-switch}\downarrow$$

split

$$\frac{\Sigma; \Delta; \psi_a; C \wedge \Phi_a; \Omega \vdash t_1 \downarrow A, L, U \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \neg C \wedge \Phi_a; \Omega \vdash t_1 \downarrow A, L, U \Rightarrow \Phi_2 \quad \Delta \vdash C \text{ wf}}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash \text{split } (t_1) \text{ with } C \downarrow A, L, U \Rightarrow C \rightarrow \Phi_1 \wedge \neg C \rightarrow \Phi_2} \text{ alg-u-split}\downarrow$$

$$\frac{\Sigma; \Delta; \psi_a; C \wedge \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \downarrow \tau, D \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \neg C \wedge \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \downarrow \tau, D \Rightarrow \Phi_2 \quad \Delta \vdash C \text{ wf}}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{split } (t_1) \text{ with } C \ominus \text{split } (t'_1) \text{ with } C \downarrow \tau, D \Rightarrow C \rightarrow \Phi_1 \wedge \neg C \rightarrow \Phi_2} \text{ alg-r-split}\downarrow$$

contra

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a \models \perp}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash \text{contra } t \downarrow A, L, U \Rightarrow \top} \text{ alg-u-contra}\downarrow$$

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a \models \perp}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{contra } t \ominus \text{contra } t' \downarrow \tau, D \Rightarrow \top} \text{ alg-r-contra}\downarrow$$

Figure 25: Bidirectional algorithmic typing rules, part 1

↑↓

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash t \uparrow A' \Rightarrow [\psi], L', U', \Phi_1 \quad \Sigma; \Delta; \psi, \psi_a; \Phi_a \models^A A' \sqsubseteq A \Rightarrow \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash t \downarrow A, L, U \Rightarrow \exists(\psi). \Phi_1 \wedge \Phi_2 \wedge L' \leq L \wedge U \leq U'} \text{alg-}\uparrow\downarrow$$

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash t \ominus t' \uparrow \tau' \Rightarrow [\psi], D', \Phi_1 \quad \Delta; \psi, \psi_a; \Phi_a \models \tau' \equiv \tau \Rightarrow \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \exists(\psi). \Phi_1 \wedge \Phi_2 \wedge D' \leq D} \text{alg-r-}\uparrow\downarrow$$

annotation

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash t \downarrow A, L, U \Rightarrow \Phi \quad \Delta; \Phi_a \vdash^A A \text{ wf} \quad \text{FIV}(A, L, U) \in \Delta}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash (t : A, L, U) \uparrow A \Rightarrow [\cdot], L, U, \Phi} \text{alg-u-anno-}\uparrow$$

$$\frac{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \Phi \quad \Delta; \Phi_a \vdash \tau \text{ wf} \quad \text{FIV}(\tau, D) \in \Delta}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash (t : \tau, D) \ominus (t' : \tau, D) \uparrow \tau \Rightarrow [\cdot], D, \Phi} \text{alg-r-anno-}\uparrow$$

Figure 26: Bidirectional algorithmic typing rules, part 2

Alloc algo

$$\frac{\Delta \vdash L_1, U_1, L_2, U_2 \in \text{fresh}(\mathbb{R}) \quad \Sigma; \Delta; L_1, U_1, \psi_a; \Phi_a; \Omega \vdash t_1 \downarrow \text{int}[I], L_1, U_1 \Rightarrow \Phi_1 \quad \Sigma; \Delta; L_2, U_2, \psi_a; \Phi_a; \Omega \vdash t_2 \downarrow A, L_2, U_2 \Rightarrow \Phi_2 \quad \Phi = \Phi_2 \wedge L_1 + L_2 + L_a \doteq L \wedge U_1 + U_2 + U_a \doteq U \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \psi_a; \Phi_a; \Omega \vdash \text{alloc } t_1 t_2 \downarrow \{P\} \exists \gamma : \text{Array}_\gamma[I] A \{P \star \gamma \rightarrow \mathbb{N}\}, 0, 0 \Rightarrow \exists L_1, U_1 :: \mathbb{R}. (\Phi_1 \wedge \exists L_2, U_2 :: \mathbb{R}. \Phi)} \text{alg-u-alc-}\downarrow$$

$$\frac{D_1, D_2 \in \text{fresh}(\mathbb{R}) \quad \Sigma; \Delta; D_1, \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \downarrow \text{int}[I], D_1 \Rightarrow \Phi_1 \quad \Sigma; \Delta; D_2, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \downarrow \tau, D_2 \Rightarrow \Phi_2 \quad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \quad \Sigma \vdash \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{alloc } t_1 t_2 \ominus \text{alloc } t'_1 t'_2 \downarrow \{P\} \exists \gamma. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \mathbb{N}\}, 0 \Rightarrow \exists D_1 :: \mathbb{R}. \Phi_1 \wedge (\exists D_2 :: \mathbb{R}. \Phi)} \text{alg-r-alc-}\downarrow$$

$$\frac{D_1, D_2 \in \text{fresh}(\mathbb{R}) \quad \Sigma; \Delta; D_1, \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \downarrow \text{int}[I], D_1 \Rightarrow \Phi_1 \quad \Sigma; \Delta; D_2, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \downarrow \square \tau, D_2 \Rightarrow \Phi_2 \quad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \quad \gamma \text{ fresh} \quad \Sigma; \Delta \vdash P \text{ wf}}{\Sigma; \Delta; \psi_a; \Phi_a; \Gamma \vdash \text{alloc}_b t_1 t_2 \ominus \text{alloc}_b t'_1 t'_2 \downarrow \{P\} \exists \gamma. \text{Array}_\gamma[I] \tau \{P \star \gamma \rightarrow \emptyset\}, 0 \Rightarrow \exists D_1 :: \mathbb{R}. \Phi_1 \wedge (\exists D_2 :: \mathbb{R}. \Phi)} \text{alg-r-alcB-}\downarrow$$

Figure 27: Bidirectional algorithm-typing rules (alloc)

Read algo

$$\begin{array}{c}
\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \text{Array}_\gamma[I] A \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Omega \vdash t_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \\
\Delta; \psi_a; \Phi_a \models I' \leq I \quad \Phi = \Phi_2 \wedge L_1 + L_2 + c_r \doteq L \wedge U_1 + U_2 + c_r \doteq U \quad P = P' \star \gamma \rightarrow _ \quad \Sigma; \Delta \vdash P \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Omega \vdash \text{read } t_1 t_2 \downarrow \{P\} \exists _ : A \{P\}, 0, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).\Phi)) \\
\text{alg-u-read-}\downarrow \\
\text{exec}(L, U) \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \text{Array}_\gamma[I] \tau \Rightarrow [\psi_1], D_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\
P = P' \star \gamma \rightarrow _ \quad \Delta; \psi_a; \Phi_a \models I' \leq I \quad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \quad \Sigma; \Delta \vdash P \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{read } t_1 t_2 \ominus \text{read } t'_1 t'_2 \downarrow \{P\} \exists _ . \tau \{P\}, 0 \Rightarrow \exists(\psi_1).\Phi_1 \wedge (\exists(\psi_2).\Phi) \\
\text{diff}(D) \\
\text{alg-r-read-}\downarrow \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \text{Array}_\gamma[I] \tau \Rightarrow [\psi_1], D_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\
P = P' \star \gamma \rightarrow \beta \quad \Delta; \psi_a; \Phi_a \models I' \leq I \wedge \neg(I' \in \beta) \quad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D \quad \Sigma; \Delta \vdash P \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{read}_b t_1 t_2 \ominus \text{read}_b t'_1 t'_2 \downarrow \{P\} \exists _ . \tau \{P\}, 0 \Rightarrow \exists(\psi_1).\Phi_1 \wedge (\exists(\psi_2).\Phi) \\
\text{diff}(D) \\
\text{alg-r-readB-}\downarrow
\end{array}$$

Figure 28: Bidirectional algorithm-typing rules (read)

Update algo

$$\begin{array}{c}
\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \text{Array}_\gamma[I] A \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Omega \vdash t_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \\
\Delta; \psi_a; \Phi_a \models I' \leq I \wedge I' \in \beta \quad L_3, U_3 \in \text{fresh}(\mathbb{R}) \quad \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 \quad P = P' \star \gamma \rightarrow \beta \quad \Sigma; \Delta \vdash P' \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Omega \vdash \text{update } t_1 t_2 t_3 \downarrow \{P\} \exists _ : \text{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))) \\
\text{alg-u-updt-}\downarrow \\
\text{exec}(L, U) \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \text{Array}_\gamma[I] \tau \Rightarrow [\psi_1], D_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\
\Delta; \psi_a; \Phi_a \models I' \leq I \wedge \beta' = \beta \cup \{I'\} \quad D_3 \in \text{fresh}(\mathbb{R}) \quad \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 \quad Q = P' \star \gamma \rightarrow \beta' \quad \Phi = \Phi_2 \wedge D_1 + D_2 + D_3 \doteq D \quad \Sigma; \Delta \vdash P' \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{update } t_1 t_2 t_3 \ominus \text{update } t'_1 t'_2 t'_3 \downarrow \{P\} \exists _ . \text{unit } \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi))) \\
\text{diff}(D) \\
\text{alg-r-updt-}\downarrow \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \text{Array}_\gamma[I] \tau \Rightarrow [\psi_1], D_1, \Phi_1 \quad \Delta; \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \uparrow \text{int}[I'] \Rightarrow [\psi_2], D_2, \Phi_2 \\
\Delta; \psi_a; \Phi_a \models I' \leq I \wedge \beta' = \beta / \{I'\} \quad D_3 \in \text{fresh}(\mathbb{R}) \quad \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 \quad Q = P' \star \gamma \rightarrow \beta' \quad \Phi = \Phi_2 \wedge D_1 + D_2 + D_3 \doteq D \quad \Sigma; \Delta \vdash P' \quad wf \\
\hline
\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{update}_b t_1 t_2 t_3 \ominus \text{update}_b t'_1 t'_2 t'_3 \downarrow \{P\} \exists _ . \text{unit } \{Q\}, 0 \Rightarrow \exists(\psi_1).(\Phi_1 \wedge (\exists(\psi_2).(\Phi_2 \wedge \exists D_3 :: \mathbb{R}.\Phi))) \\
\text{diff}(D) \\
\text{alg-r-updtB-}\downarrow
\end{array}$$

Figure 29: Bidirectional algorithm-typing rules (update)

Let algo

$$\begin{array}{c}
\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \{P\} \exists \vec{\gamma}_1 : A_1 \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \\
\frac{\text{exec}(L, U)}{\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \{P\} \exists \vec{\gamma}_1 : A_1 \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1} \\
\frac{\text{exec}(L', U')}{L_2, U_2 \in \text{fresh}(\mathbb{R}) \quad \Delta; L_2, U_2, \psi_a; \Phi_a; \Omega \vdash t_2 \downarrow \{P'\} \exists \vec{\gamma}_2 : A_2 \{Q\}, L_2, U_2 \Rightarrow \Phi_2} \\
\frac{\Phi = \Phi_2 \wedge L_1 + L_2 + c_l \doteq L_3 \wedge U_1 + U_2 + c_l \doteq U_3 \wedge L + L' \doteq L'' \wedge U + U' \doteq U''}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{let}_m \{x\} = t_1 \text{ in } t_2 \downarrow \{P\} \exists_- : A_2 \{Q\}, L_3, U_3 \Rightarrow \exists(\psi_1). \Phi_1 \wedge (\exists L_2, U_2 :: \mathbb{R}. \Phi)} \text{alg-u-let-}\downarrow \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \{P\} \exists \vec{\gamma}_1 : \tau_1 \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1 \quad D_2 \in \text{fresh}(\mathbb{R}) \\
\frac{\text{diff}(D)}{\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \{P\} \exists \vec{\gamma}_1 : \tau_1 \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1} \\
\frac{\text{diff}(D')}{\Delta; D_2, \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \downarrow \{P'\} \exists \vec{\gamma}_2 : \tau_2 \{Q\}, D_2 \Rightarrow \Phi_2 \quad \Phi = \Phi_2 \wedge D_1 + D_2 \doteq D_3 \wedge D + D' \doteq D''} \text{alg-r-let-}\downarrow \\
\Delta; \psi_a; \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_- . \tau_2 \{Q\}, D_3 \Rightarrow \exists(\psi_1). (\Phi_1 \wedge (\exists D_2 :: \mathbb{R}. \Phi)) \\
\frac{\text{diff}(D'')}{\Delta; \psi_a; \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_- . \tau_2 \{Q\}, D_3 \Rightarrow \exists(\psi_1). (\Phi_1 \wedge (\exists D_2 :: \mathbb{R}. \Phi))} \\
\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \{P\} \exists \vec{\gamma}_1 : A_1 \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1 \\
\frac{\text{exec}(L, U)}{\Delta; \psi_a; \Phi_a; \Omega \vdash t_1 \uparrow \{P\} \exists \vec{\gamma}_1 : A_1 \{P'\} \Rightarrow [\psi_1], L_1, U_1, \Phi_1} \\
\frac{\text{exec}(L', U')}{\Delta; \psi_1, \psi_a; \Phi_a; \Omega \vdash t_2 \uparrow \{Q'\} \exists \vec{\gamma}_2 : A_2 \{Q\} \Rightarrow [\psi_2], L_2, U_2, \Phi_2 \quad P' = Q'} \\
\frac{\text{exec}(L+L', L+L')}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{let}_m \{x\} = t_1 \text{ in } t_2 \uparrow \{P\} \exists_- : A_2 \{Q\} \Rightarrow [\psi_1, \psi_2], L_1 + L_2 + c_l, U_1 + U_2 + c_l, \Phi_1 \wedge \Phi_2} \text{alg-u-let-}\uparrow \\
\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \{P\} \exists \vec{\gamma}_1 : \tau_1 \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1 \\
\frac{\text{diff}(D)}{\Delta; \psi_a; \Phi_a; \Gamma \vdash t_1 \ominus t'_1 \uparrow \{P\} \exists \vec{\gamma}_1 : \tau_1 \{P'\} \Rightarrow [\psi_1], D_1, \Phi_1} \\
\frac{\text{diff}(D')}{\Delta; \psi_1, \psi_a; \Phi_a; \Gamma \vdash t_2 \ominus t'_2 \uparrow \{Q'\} \exists \vec{\gamma}_2 : \tau_2 \{Q\} \Rightarrow [\psi_2], D_2, \Phi_2 \quad Q' = P'} \\
\frac{\text{diff}(D+D')}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{let}_m \{x\} = t_1 \text{ in } t_2 \ominus \text{let}_m \{x\} = t'_1 \text{ in } t'_2 \uparrow \{P\} \exists_- . \tau_2 \{Q\} \Rightarrow [\psi_1, \psi_2], D_1 + D_2, \Phi_1 \wedge \Phi_2} \text{alg-r-let-}\uparrow
\end{array}$$

Figure 30: Bidirectional algorithm-typing rules (let)

Return algo

$$\begin{array}{c}
L, U \in \text{fresh}(\mathbb{R}) \quad \Delta; L, U, \psi_a; \Phi_a; \Omega \vdash t \downarrow A, L, U \Rightarrow \Phi \\
\frac{\text{exec}(L, U)}{\Delta; \psi_a; \Phi_a; \Omega \vdash \text{return } t \downarrow \{P\} \exists_- : A_2 \{P\}, 0, 0 \Rightarrow \exists k, t :: \mathbb{R}. \Phi} \text{alg-u-ret-}\downarrow \\
D \in \text{fresh}(\mathbb{R}) \quad \Delta; D, \psi_a; \Phi_a; \Gamma \vdash t \ominus t' \downarrow \tau, D \Rightarrow \Phi \\
\frac{\text{diff}(D)}{\Delta; \psi_a; \Phi_a; \Gamma \vdash \text{return } t \ominus \text{return } t' \downarrow \{P\} \exists_- . \tau \{P\}, 0 \Rightarrow \exists D :: \mathbb{R}. \Phi} \text{alg-r-ret-}\downarrow
\end{array}$$

Figure 31: Bidirectional algorithmic typing rules (return)

Algorithmic Reltional subtyping

$$\begin{array}{c}
\frac{}{\Sigma; \Delta; \psi_a; \Phi_a \models \text{int}_r[I] \equiv \text{int}_r[I'] \Rightarrow I \doteq I'} \text{alg-r-int-I} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi \quad \Sigma \models \gamma = \gamma'}{\Sigma; \Delta; \psi_a; \Phi_a \models \text{Array}_\gamma[I] \tau \equiv \text{Array}_{\gamma'}[I'] \tau' \Rightarrow I \doteq I' \wedge \Phi} \text{alg-r-array} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models \tau \equiv \tau' \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \Phi_a \models P' \equiv P \Rightarrow \Phi_2 \quad \Sigma; \Delta; \psi_a; \Phi_a \models Q \equiv Q' \Rightarrow \Phi_3 \quad \Sigma \models \vec{\gamma}_1 = \vec{\gamma}_2}{\Sigma; \Delta; \psi_a; \Phi_a \models \{P\} \exists \vec{\gamma}_1 : \tau \{Q\} \equiv \{P'\} \exists \vec{\gamma}_2 : \tau' \{Q'\} \Rightarrow D \doteq D' \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_3} \text{alg-r-monad} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models \tau_1 \equiv \tau'_1 \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \Phi_a \models \tau_2 \equiv \tau'_2 \Rightarrow \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a \models \tau_1 \xrightarrow{\text{diff}(D)} \tau_2 \equiv \tau'_1 \xrightarrow{\text{diff}(D')} \tau'_2 \Rightarrow D \doteq D' \wedge \Phi_1 \wedge \Phi_2} \text{alg-r-fun} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models P \equiv P' \Rightarrow \Phi}{\Sigma; \Delta; \psi_a; \Phi_a \models P \star \gamma \rightarrow \beta_1 \equiv P' \star \gamma \rightarrow \beta_2 \Rightarrow \Phi \wedge \beta_1 \doteq \beta_2} \text{alg-r-predicate} \\
\frac{}{\Sigma; \Delta; \psi_a; \Phi_a \models \emptyset \equiv \emptyset \Rightarrow \top} \text{alg-r-predicate-empty}
\end{array}$$

Figure 32: Bidirectional relational algorithmic subtyping rules

Algo Unary subtyping

$$\begin{array}{c}
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models^A A'_1 \sqsubseteq A_1 \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \Phi_a \models^A A_2 \sqsubseteq A'_2 \Rightarrow \Phi_2}{\Sigma; \Delta; \psi_a; \Phi_a \models^A A_1 \xrightarrow{\text{exec}(L,U)} A_2 \sqsubseteq A'_1 \xrightarrow{\text{exec}(L',U')} A'_2 \Rightarrow \Phi_1 \wedge \Phi_2 \wedge L' < L \wedge U < U'} \text{alg-u-fun} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models^A A \sqsubseteq A' \Rightarrow \Phi_1 \quad \Sigma; \Delta; \psi_a; \Phi_a \models^A P' \sqsubseteq P \Rightarrow \Phi_2 \quad \Sigma; \Delta; \psi_a; \Phi_a \models^A Q \sqsubseteq Q' \Rightarrow \Phi_3 \quad \Sigma \models \vec{\gamma}_1 \sqsubseteq \vec{\gamma}_2}{\Sigma; \Delta; \psi_a; \Phi_a \models^A \{P\} \exists \vec{\gamma}_1 : A \{Q\} \equiv \{P'\} \exists \vec{\gamma}_2 : A' \{Q'\} \Rightarrow \Phi_1 \wedge k' \leq k \wedge t \leq t'} \text{alg-u-monad} \\
\frac{\Sigma; \Delta; \psi_a; \Phi_a \models^A P \sqsubseteq P' \Rightarrow \Phi}{\Sigma; \Delta; \psi_a; \Phi_a \models^A P \star \gamma \rightarrow \beta_1 \sqsubseteq P' \star \gamma \rightarrow \beta_2 \Rightarrow \Phi \wedge \beta_1 \sqsubseteq \beta_2} \text{alg-u-predicate}
\end{array}$$

Figure 33: Bidirectional unary algorithmic subtyping rules

5 Experimental Evaluation

Benchmark	LOC	# of Anno	typechecking time	SMT total	SMT phrase 2
map(1)	19	3	0.802s	1.061s	0.01s(11)
map(2)	12	2	1.247s	1.014s	0.02s(25)
boolOr	48	8	1.574s	3.511	2.38s(10000)
seperate	36	8	1.351s	2.158s	0.01s(19)
loop	23	5	1.167s	2.124s	0.01 (9)
FFT	66	17	2.591s	4.278s	0.01s (31)
Search	62	10	3.753s	10.996s	6.56s(25216)
NSS	94	12	4.158s	14.443s	10.03(3355)
shift	14	3	0.660s	1.404s	0.01s (11)
insert	22	6	1.001s	3.029s	0.01s(34)
iSort	134	12	2.897s	16.881s	10.70 (2262)
merge(1)	29	8	2.203s	2.242s	0.01s (24)
merge(2)	64	11	3.231s	0.369s	0.02s (39)
sam	19	4	0.946s	0.103s	0.02(20)
comp	20	3	1.138s	0.122s	0.01(10)

Table 1: Experimental results of the type checker

Benchmark	LOC	# of Annotation	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