Monadic refinements for relational cost analysis (Appendix)

Ivan Radiček Gilles Barthe Marco Gaboardi Deepak Garg
Florian Zuleger

Structure of the Appendix

In the appendix we give material that was omitted from the paper for the lack of space (see the table of contents below).

Note: In the appendix we distinguish different (small-step) reductions, such as β, ζ, ι and μ, and hence write $\to_{\beta,\zeta,\iota,\mu}$ (for their union), or any single appropriate reduction; for example $\to_\zeta$ is the impure reduction. In the paper, for simplicity, we conflate all the reductions to simply $\to$.

Contents

1 Generalized data types

In this section we generalize data types to also allow monadic types $\mathbb{C}(\cdot)$ in definitions, thus permitting reasoning with lazy data structures[1] in the style of Danielsson (2008).

Elementary types First we define types (named elementary) that we allow in data type definitions:

$$\sigma ::= \mathsf{b} \mid \theta \mid \mathbb{C}(\sigma)$$

where b ranges over *base types*, θ ranges of data types, and $\mathbb{C}(\sigma)$ is a monadic computation that returns a result of type σ.

Data types A data type θ is defined by an equation:

$$\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n})$$

where $K_1, \ldots, K_n$ are constructors.

All data type definitions are collected in a context Θ. A context Θ is well defined if for every $\theta \in \Theta$, an equation for θ satisfies:

1. For each $1 \leq i, j \leq n$ such that $i \neq j$, also $K_i \neq K_j$.
2. For each $1 \leq i \leq n$, $K_i \notin \Theta \setminus \{\theta\}$.
3. For each $1 \leq i \leq n$ and $1 \leq j \leq a_i$, if $\theta' \in \sigma_{i,j}$, then $\theta' \in \Theta$.

The first two conditions ensure that all constructors (across all data type definitions) are unique. The third condition ensures that any data-type θ' mentioned in an equation is defined in Θ (note that this definition allows mutually recursive data-types).

We assume some data-type environment Θ, and leave it implicit in all judgments.

Examples Some standard data types, also used in the examples through the paper, are:

- Unit: $\mathrm{unit} = \star()$.
- Boolean: $\mathrm{bool} = \mathsf{tt}() + \mathsf{ff}()$.
- List: $\mathrm{list}_\sigma = \mathsf{nil}() + \mathsf{cons}(\sigma \times \mathrm{list}_\sigma)$.
- Lazy list: $\mathrm{list}_{\mathbf{L}} = \mathsf{nil}() + \mathsf{cons}(\sigma \times \mathbb{C}(\mathrm{list}_{\mathbf{L}}))$ (note $\mathbb{C}(\mathrm{list}_{\mathbf{L}})$ in cons constructor).

Note that we use duplicate constructor names in examples, where it does not cause confusion. Also, in examples we often omit parentheses around constructors with no arguments (e.g., $\star$, tt, nil).

[1] See Section 6.5 and Section 6.6 for examples.

Depth of elementary types Depth is defined as an axiom scheme over σ; We write $|e, n|_\sigma$ to mean that e of type σ has depth n (of type $\mathbb{N}$). Let

$$\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta$$

$$\forall x : \mathsf{b}, n : \mathbb{N}.\ |x, n|_{\mathsf{b}} \Leftrightarrow n \doteq 0 \qquad \forall x : \theta.\ |x, 0|_\theta \Leftrightarrow \bigvee_{1 \le i \le n | a_i = 0} x \doteq K_i()$$

$$\forall x : \theta, n : \mathbb{N}.\ |x, n+1|_\theta \Leftrightarrow \bigvee_{1 \le i \le n | a_i > 0} (\exists x_1 : \sigma_{i,1}, n_1 : \mathbb{N}, \ldots, x_{a_i} : \sigma_{i,a_i}, n_{a_i} : \mathbb{N}.\ x \doteq$$

$$K_i(x_1, \ldots, x_{a_i}) \wedge \bigwedge_{1 \le j \le a_i} (|x, n_j|_{\sigma_{i,j}}) \wedge n \doteq \max\{n_1, \ldots, n_{a_i}\})$$

$$\forall x : \mathbb{C}(\sigma), n : \mathbb{N}.\ |x, n|_{\mathbb{C}(\sigma)} \Leftrightarrow \exists n' : \mathbb{R}^\infty, y : \sigma.\ x \doteq \{\mathsf{cstep}_{n'}(\mathsf{cret}(y))\} \wedge |y, n|_\sigma$$

The axioms formalize the following:

- Any value of a base type has depth zero.
- Any data type value, s.t. its constructor has not arguments (i.e. $K()$) has depth zero.
- Any data types value, s.t. its constructor has arguments has depth one larger than any of its arguments.
- Any monadic elementary type has depth equal to the depth of its underlying pure value.

It is not difficult to prove by induction on σ and $\mathrm{L^C}$'s other axioms that (a) $\forall x : \sigma. \exists n : \mathbb{N}. |x, n|_\sigma$ and (b) $\forall x : \sigma, n_1, n_2 : \mathbb{N}. (|x, n_1|_\sigma \wedge |x, n_2|_\sigma) \Rightarrow n_1 = n_2$. Hence, for every $x : \sigma$, there is a unique $n : \mathbb{N}$ such that $|x, n|_\sigma$. We abuse notation slightly and write $|x|_\sigma$ for this unique n. When σ is obvious from the context, we simplify this further to $|x|$.

Elementary type interpretation Next we define interpretation of elementary types in set theory. We assume $q \in \mathbb{N}$.

$$[\![\mathsf{b}]\!]_q \triangleq \mathsf{b} \qquad [\![\mathbb{C}(\sigma)]\!]_q \triangleq [\![\sigma]\!]_q \times \mathbb{R}^\infty$$

If $\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta$ then:

$$\begin{array}{lcl} [\![\theta]\!]_0 & \triangleq & \{K_i() \mid \text{ for all } 1 \le i \le n \text{ where } a_i = 0\} \\ [\![\theta]\!]_{q+1} & \triangleq & \bigcup_{1 \le i \le n} \{K_i(x_1, \ldots, x_{a_i}) \mid a_i > 0 \wedge x_j \in [\![\sigma_{i,j}]\!]_{q'} \text{ for any } 0 \le q' \le q \text{ and all } 1 \le j \le a_i\} \end{array}$$

These are well defined by induction on the index q.

Type interpretation Based on the definitions above we define the interpretation of data types as:

$$[\![\theta]\!] \triangleq \bigcup_{q \in \mathbb{N}} [\![\theta]\!]_q$$

Interpretation of the size predicate Next, we define the interpretation of depth predicate $|\cdot|_\sigma$. We define $(\!|v, n|_\sigma|\!) \triangleq v \in [\![\sigma]\!] \wedge n = \min\{q \mid v \in [\![\sigma]\!]_q\}$.

This yields the following interpretation for the depth *function*: $(\!|\cdot|_\sigma|\!)\ v = \min\{q \mid v \in [\![\sigma]\!]_q\}$.

Lemma. The following hold:

- For any $v \in [\![\mathsf{b}]\!]$, $\min\{q \mid v \in [\![\mathsf{b}]\!]_q\} = 0$
- For any $K(v_1, \ldots, v_n) \in [\![\theta]\!]$, such that $K(\sigma_1, \ldots, \sigma_n) \in \Theta$: if $n = 0$, then $\min\{q \mid v \in [\![\mathsf{b}]\!]_q\} = 0$, otherwise $\min\{q \mid v \in [\![\theta]\!]_q\} = \max\{q_1, \ldots, q_n\} + 1$, where $q_i = \min\{q \mid v \in [\![\sigma_i]\!]_q\}$, for all $1 \leq i \leq n$.
- For any $v = (v', n) \in [\![\sigma]\!] \times \mathbb{R}^\infty$, $\min\{q \mid v \in [\![\mathbb{C}(\sigma)]\!]_q\} = \min\{q \mid v' \in [\![\sigma]\!]_q\}$

Proof. Immediate from the definition of $[\![\sigma]\!]_q$. □

Lemma. All L^C axioms relating to $|\cdot|_\sigma$ are sound for the semantic definition of $|\cdot|_\sigma$.

Proof. By case analysis of the axioms that define $|\cdot|_\sigma$. See Proof 5. □

2 Rules

In this section we list most of the typing rules and most of the rules of L^C, U^C, and R^C; the exception are the rules from Aguirre et al. (2017), and standard typing rules.

2.1 Typing rules

We elide the standard typing rules for pairs, projection, abstraction, application.

$$\frac{c \in \mathsf{b}}{\Gamma \vdash c : \mathsf{b}} \qquad \frac{i \in \{1,2\} \qquad \Gamma \vdash e : \tau_i}{\Gamma \vdash \mathsf{inj}_i\ e : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 : \tau}$$

$$\frac{K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma \vdash e_i : \sigma_i \text{ for all } 1 \leq i \leq n}{\Gamma \vdash K(e_1, \ldots, e_n) : \theta}$$

$$\frac{\Gamma \vdash e : \theta \qquad \theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \qquad \Gamma \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau \text{ for all } 1 \leq i \leq n}{\Gamma \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n : \tau}$$

$$\frac{\Gamma, f : \theta \to \tau, x : \theta \vdash e : \tau \qquad \mathcal{D}ef(f, x, e)}{\Gamma \vdash \mathsf{rec}\, f(x).e : \theta \to \tau} \qquad \frac{\Gamma \vdash m \div \tau}{\Gamma \vdash \{m\} : \mathbb{C}(\tau)} \qquad \frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathsf{cret}(e) \div \tau}$$

$$\frac{\Gamma \vdash n : \mathbb{R}^\infty \qquad \Gamma \vdash m \div \tau}{\Gamma \vdash \mathsf{cstep}_n(m) \div \tau} \qquad \frac{\Gamma \vdash e : \mathbb{C}(\tau') \qquad \Gamma, x : \tau' \vdash m \div \tau}{\Gamma \vdash \mathsf{cbind}(e, \{x\}.\, m) \div \tau}$$

2.2 L^C rules

We elide the standard logic connectives rules; they can be found in Aguirre et al. (2017).

General rules for equality. Here, $u ::= e \mid m$.

$$\frac{u_1 \to_{\beta,\iota,\mu,\zeta} u_2}{\Gamma; \Psi \vdash_{\text{LC}} u_1 \doteq u_2}\text{BETA} \qquad \frac{}{\Gamma; \Psi \vdash_{\text{LC}} u \doteq u}\text{REFL} \qquad \frac{\Gamma; \Psi \vdash_{\text{LC}} u_1 \doteq u_2 \qquad \Gamma; \Psi \vdash_{\text{LC}} \phi[u_1/x]}{\Gamma; \Psi \vdash_{\text{LC}} \phi[u_2/x]}\text{SUBST}$$

The following rules are derivable:

$$\frac{\Gamma; \Psi \vdash_{\text{LC}} u_2 \doteq u_1}{\Gamma; \Psi \vdash_{\text{LC}} u_1 \doteq u_2}\text{SYM} \qquad \frac{\Gamma; \Psi \vdash_{\text{LC}} u_1 \doteq u' \qquad \Gamma; \Psi \vdash_{\text{LC}} u' \doteq u_2}{\Gamma; \Psi \vdash_{\text{LC}} u_1 \doteq u_2}\text{TRANS}$$

Axioms specific to monadic expression equality

$$\frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x, y \div \tau.\ \{x\} \doteq \{y\} \Rightarrow x \doteq y} \qquad \frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x : \mathbb{C}(\tau).\ \exists y \div \tau.\ x \doteq \{y\}}$$

$$\frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x \div \tau.\ \exists y : \mathbb{R}^\infty, z : \tau.\ x \doteq \mathsf{cstep}_y(\mathsf{cret}(z))}$$

$$\frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x_1, x_2 : \mathbb{R}^\infty,\ y_1, y_2 : \tau.\ \mathsf{cstep}_{x_1}(\mathsf{cret}(y_1)) \doteq \mathsf{cstep}_{x_2}(\mathsf{cret}(y_2)) \Rightarrow x_1 \doteq x_2 \wedge y_1 \doteq y_2}$$

The following axioms are derivable:

$$\frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x_1, x_2 : \mathbb{R}^\infty, y_1 \div \tau, y_2 : \tau.\ \mathsf{cstep}_{x_1}(y_1) \doteq \mathsf{cstep}_{x_2}(\mathsf{cret}(y_2)) \Rightarrow \exists x_3 : \mathbb{R}^\infty.\ x_2 \doteq x_1 + x_3 \wedge y_1 \doteq \mathsf{cstep}_{x_3}(\mathsf{cret}(y_2))}$$

$$\frac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x_1, x_2 : \tau.\ \mathsf{cret}(x_1) \doteq \mathsf{cret}(x_2) \Rightarrow x_1 \doteq x_2}$$

Rules about data types

$$\frac{\Gamma; \Psi \vdash_{\mathrm{LC}} K(e_1, \ldots, e_n) \doteq K'(e'_1, \ldots, e'_n)}{\Gamma; \Psi \vdash_{\mathrm{LC}} e_i \doteq e'_i}\text{CONS}$$

$$\frac{K \neq K'}{\Gamma; \Psi \vdash_{\mathrm{LC}} K(e_1, \ldots, e_n) \not\doteq K'(e'_1, \ldots, e'_m)}\text{NC}$$

$$\frac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma \vdash e : \theta \qquad \Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, e \doteq K_i(x_1, \ldots, x_{a_i}) \vdash_{\mathrm{LC}} \phi \\ \text{for all } 1 \leq i \leq n \qquad \text{where } x_1, \ldots, x_{a_i} \notin \phi\end{array}}{\Gamma; \Psi \vdash_{\mathrm{LC}} \phi}\text{ELIM}$$

$$\frac{\theta \in \Theta \qquad \Gamma, x : \theta; \Psi, \forall y : \theta.\ |y| < |x| \Rightarrow \phi[y/x] \vdash_{\mathrm{LC}} \phi}{\Gamma; \Psi \vdash_{\mathrm{LC}} \forall x : \theta.\ \phi}\text{IND}$$

$$\frac{\begin{array}{c}\theta_1, \theta_2 \in \Theta \\ \Gamma, x_1 : \theta_1, x_2 : \theta_2; \Psi, \forall y_1 : \theta_1, y_2 : \theta_2.\ (|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2] \vdash_{\mathrm{LC}} \phi\end{array}}{\Gamma; \Psi \vdash_{\mathrm{LC}} \forall x_1 : \theta_1, x_2 : \theta.\ \phi}\text{DBLIND}$$

2.3 U^{C} rules

Here we show the most interesting U^{C} rules, primarily all the rules related to cost and datatypes. Other rules can be found in Aguirre et al. (2017).

Rules for the pure judgment $\Gamma; \Psi \vdash e : \tau \mid \phi$

$$\frac{\begin{array}{c}\mathcal{D}ef(f, x, e) \\ \Gamma, x : \theta, f : \theta \to \tau; \Psi, \phi, \forall y.\ |y| < |x| \Rightarrow \phi[y/x] \Rightarrow \phi'[y/x][f\ y/\mathbf{r}] \vdash e : \tau \mid \phi'\end{array}}{\Gamma; \Psi \vdash \mathsf{rec}\, f(x).e : \theta \to \tau \mid \forall x.\ \phi \Rightarrow \phi'[\mathbf{r}\ x/\mathbf{r}]}\text{U-LETREC}$$

$$\frac{\Gamma; \Psi \vdash e_1 : \tau \to \tau' \mid \forall x.\phi \Rightarrow \phi'[\mathbf{r}\ x/\mathbf{r}] \qquad \Gamma; \Psi \vdash e_2 : \tau \mid \phi[\mathbf{r}/x]}{\Gamma; \Psi \vdash e_1\ e_2 : \tau' \mid \phi'}\text{U-APP}$$

$$\frac{\begin{array}{c}K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma; \Psi \vdash e_i : \sigma_i \mid \phi_i \quad \text{for all } 1 \leq i \leq n \\ \Gamma; \Psi \vdash_{\mathrm{LC}} \forall x_1 : \sigma_1, \ldots, x_n : \sigma_n.\ \phi_1[x_1/\mathbf{r}] \Rightarrow \cdots \Rightarrow \phi_n[x_n/\mathbf{r}] \Rightarrow \phi[K(x_1, \ldots, x_n)/\mathbf{r}]\end{array}}{\Gamma; \Psi \vdash K(e_1, \ldots, e_n) : \theta \mid \phi}\text{U-CONS}$$

$$\frac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma; \Psi \vdash e : \theta \mid \phi' \qquad \text{For all } 1 \leq i \leq n : \quad \Gamma; \Psi \vdash e_i : \sigma_{i,1} \to \cdots \sigma_{i,a_i} \to \tau \mid \phi'_i \quad \text{where} \\ \phi'_i \equiv \forall x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}.\ \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}] \Rightarrow \phi[(\mathbf{r}\ x_1\ \cdots\ x_{a_i})/\mathbf{r}]\end{array}}{\Gamma; \Psi \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n : \tau \mid \phi}\text{U-MATCH}$$

$$\frac{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi[\mathbf{r}/x]}{\Gamma; \Psi \vdash \{m\} : \mathbb{C}(\tau) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, x.\phi)}\text{U-MONAD}$$

Rules for the monadic judgment $\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi$

$$\frac{\Gamma; \Psi \vdash e : \tau \mid \phi}{\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid 0 \mid 0 \mid \phi}\text{U-RET} \qquad \frac{\Gamma \vdash n : \mathbb{R}^{\infty} \qquad \Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}{\Gamma; \Psi \vdash \mathsf{cstep}_n(m) \div \tau \mid k + n \mid \ell + n \mid \phi}\text{U-STEP}$$

$$\frac{\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \mathbb{C}_{\mathrm{U}}(\mathbf{r}, k', \ell', x.\phi_1) \qquad \Gamma, x : \tau_1; \Psi, \phi_1 \vdash m_2 \div \tau_2 \mid k \mid \ell \mid \phi_2 \qquad x \notin k, \ell, \phi_2}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau_2 \mid k' + k \mid \ell' + \ell \mid \phi_2}\text{U-BIND}$$

Structural rules

$$\frac{\Gamma; \Psi \vdash e : \tau \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \phi'[e/\mathbf{r}] \Rightarrow \phi[e/\mathbf{r}]}{\Gamma; \Psi \vdash e : \tau \mid \phi}\text{U-SUB} \qquad \frac{\Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq k' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \ell' \leq \ell}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}\text{U-SUBC}$$

$$\frac{\Gamma; \Psi \vdash_{\mathrm{LC}} \exists x : \tau.\phi \quad \Gamma, x : \tau; \Psi, \phi \vdash m \div \tau' \mid k \mid \ell \mid \phi' \quad x \notin m, n, \phi'}{\Gamma; \Psi \vdash m \div \tau' \mid k \mid \ell \mid \phi'}\exists\text{EM}$$

Admissible rules

$$\frac{\Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} m \doteq \mathsf{cstep}_n(\mathsf{cret}(e)) \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq n \leq \ell \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \phi'[e/\mathbf{r}] \Rightarrow \phi[e/\mathbf{r}]}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}\text{U-SUBM1}$$

$$\frac{\Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq k' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \ell' \leq \ell \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \forall \mathbf{r}.\ \phi' \Rightarrow \phi}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}\text{U-SUBM2}$$

$$\frac{\Gamma; \Psi \vdash e : \tau \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} e \doteq e' \qquad \Gamma \vdash e' : \tau}{\Gamma; \Psi \vdash e' : \tau \mid \phi}\text{U-EQ-PURE} \qquad \frac{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} m \doteq m' \qquad \Gamma \vdash m' \div \tau}{\Gamma; \Psi \vdash m' \div \tau \mid k \mid \ell \mid \phi}\text{U-EQ-MONADIC}$$

2.4 R^{C} rules

Here we show the most interesting R^{C} rules, primarily all the rules related to cost and datatypes. Other rules can be found in Aguirre et al. (2017).

Two-sided rules for the pure judgment $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$

$$\frac{\begin{array}{c}\mathcal{D}ef(f_1, x_1, e_1) \qquad \mathcal{D}ef(f_2, x_2, e_2) \\ \Gamma, x_1 : \theta_1, x_2 : \theta_2, f_1 : \theta_1 \to \tau_1, f_2 : \theta_2 \to \tau_2; \\ \Psi, \phi, \forall y_1 y_2.\ (|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2] \Rightarrow \phi'[y_1/x_1][y_2/x_2][f_1\ y_1/\mathbf{r}_1][f_2\ y_2/\mathbf{r}_2] \\ \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi'\end{array}}{\Gamma; \Psi \vdash \mathsf{rec}\, f_1(x_1).e_1 : \theta_1 \to \tau_1 \sim \mathsf{rec}\, f_2(x_2).e_2 : \theta_2 \to \tau_2 \mid \forall x_1 x_2.\ \phi \Rightarrow \phi'[\mathbf{r}_1\ x_1/\mathbf{r}_1][\mathbf{r}_2\ x_2/\mathbf{r}_2]}\text{R-LETREC}$$

$$\frac{\begin{array}{c}\Gamma; \Psi \vdash e_1 : \tau \to \sigma_1 \sim e_2 : \tau_1 \to \sigma_2 \mid \forall x_1 x_2. \phi \Rightarrow \phi'[\mathbf{r}_1\ x_1/\mathbf{r}_1][\mathbf{r}_2\ x_2/\mathbf{r}_2] \\ \Gamma; \Psi \vdash e_1' : \tau_1 \sim e_2' : \tau_2 \mid \phi[\mathbf{r}_1/x_1][\mathbf{r}_2/x_2]\end{array}}{\Gamma; \Psi \vdash e_1\ e_1' : \sigma_1 \sim e_2\ e_2' : \sigma_2 \mid \phi'}\text{R-APP}$$

$$\frac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma; \Psi \vdash e : \theta \sim e' : \theta \mid \phi' \\ \text{For all } 1 \le i, j \le n:\ \Gamma; \Psi \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau_1 \sim e_j' : \sigma_{j,1} \to \cdots \to \sigma_{j,a_j} \to \tau_2 \mid \phi_{i,j}' \text{ where} \\ \phi_{i,j}' \equiv \forall x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}, y_1 : \sigma_{j,1}, \ldots, y_{a_j} : \sigma_{j,a_j}.\ \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}_1][K_j(y_1, \ldots, y_{a_j})/\mathbf{r}_2] \\ \Rightarrow \phi[(\mathbf{r}_1\ x_1\ \cdots\ x_{a_i})/\mathbf{r}_1][(\mathbf{r}_2\ y_1\ \cdots\ y_{a_j})/\mathbf{r}_2]\end{array}}{\Gamma; \Psi \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n : \tau_1 \sim \mathsf{match}\ e'\ \mathsf{with}\ K_1 \mapsto e_1'; \ldots; K_n \mapsto e_n' : \tau_2 \mid \phi}\text{R-MATCH}$$

$$\frac{\begin{array}{c}K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma; \Psi \vdash e_i : \sigma_i \sim e_i' : \sigma_i \mid \phi_i \quad \text{ for all } 1 \le i \le n \\ \Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x_1, y_1 : \sigma_1, \ldots, x_n, y_n : \sigma_n.\ \phi_1[x_1/\mathbf{r}_1][y_1/\mathbf{r}_2] \Rightarrow \cdots \Rightarrow \phi_n[x_n/\mathbf{r}_1][y_n/\mathbf{r}_2] \\ \Rightarrow \phi[K(x_1, \ldots, x_n)/\mathbf{r}_1][K(y_1, \ldots, y_n)/\mathbf{r}_2]\end{array}}{\Gamma; \Psi \vdash K(e_1, \ldots, e_n) : \theta \sim K(e_1', \ldots, e_n') : \theta \mid \phi}\text{R-CONS}$$

$$\frac{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \{m_1\} : \mathbb{C}(\tau_1) \sim \{m_2\} : \mathbb{C}(\tau_2) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.\phi)}\text{R-MONAD}$$

Two-sided rules for the monadic judgment $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$

$$\frac{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi}{\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim \mathsf{cret}(e_2) \div \tau_2 \mid 0 \mid \phi}\text{R-RET}$$

$$\frac{\Gamma \vdash n_1 : \mathbb{R}^\infty \qquad \Gamma \vdash n_2 : \mathbb{R}^\infty \qquad \Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1) \div \tau_1 \sim \mathsf{cstep}_{n_2}(m_2) \div \tau_2 \mid n + n_1 - n_2 \mid \phi}\text{R-STEP}$$

$$\frac{\begin{array}{c}\Gamma; \Psi \vdash e_1 : \tau_1' \sim e_2 : \tau_2' \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n', x_1.x_2.\phi') \\ \Gamma, x_1 : \tau_1', x_2 : \tau_2'; \Psi, \phi' \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi \qquad x_1, x_2 \notin n, \phi\end{array}}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x_1\}.\, m_1) \div \tau_1 \sim \mathsf{cbind}(e_2, \{x_2\}.\, m_2) \div \tau_2 \mid n' + n \mid \phi}\text{R-BIND}$$

One-sided rules for the pure judgment $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$ (selected)

$$\frac{\mathcal{D}ef(f,x,e) \qquad \Gamma, x:\theta, f:\theta \to \tau_1; \Psi, \phi, \forall y.\ |y| < |x| \Rightarrow \phi[y/x] \Rightarrow \phi'[y/x][f\ y/\mathbf{r}_1][e_2/\mathbf{r}_2] \vdash e : \tau_1 \sim e_2 : \tau_2 \mid \phi'}{\Gamma; \Psi \vdash \mathsf{rec}\, f(x).e : \theta \to \tau_1 \sim e_2 : \tau_2 \mid \forall x.\ \phi \Rightarrow \phi'[\mathbf{r}_1\ x_1/\mathbf{r}_1]}\text{R-LETREC-L}$$

$$\frac{\Gamma; \Psi \vdash e : \tau \to \sigma_1 \sim e_2 : \sigma_2 \mid \forall x.\phi \Rightarrow \phi'[\mathbf{r}\ x/\mathbf{r}] \qquad \Gamma; \Psi \vdash e' : \tau \mid \phi[\mathbf{r}/x]}{\Gamma; \Psi \vdash e\ e' : \sigma_1 \sim e_2 : \sigma_2 \mid \phi'}\text{R-APP-L}$$

One-sided rules for the monadic judgment $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$

$$\frac{\Gamma \vdash e_1 \div \tau_1 \qquad \Gamma; \Psi \vdash m_2 \div \tau_2 \mid k \mid \ell \mid \phi[e_1/\mathbf{r}_1][\mathbf{r}/\mathbf{r}_2]}{\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim m_2 \div \tau_2 \mid -k \mid \phi}\text{R-RET-L}$$

$$\frac{\Gamma \vdash n_1 : \mathbb{R}^\infty \qquad \Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1) \div \tau_1 \sim m_2 \div \tau_2 \mid n + n_1 \mid \phi}\text{R-STEP-L}$$

$$\frac{\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1') \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, x.\phi') \qquad \Gamma, x:\tau_1'; \Psi, \phi' \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.\, m_1) \div \tau_1 \sim m_2 \div \tau_2 \mid \ell + n \mid \phi}\text{R-BIND-L}$$

Structural rules

$$\frac{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \Rightarrow \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]}{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi}\text{R-SUB}$$

$$\frac{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} n' \leq n}{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}\text{R-SUBC}$$

Admissible rules

$$\frac{\begin{array}{c}\Psi;\Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi' \\ \Gamma;\Psi \vdash_{\mathrm{L^C}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \qquad \Gamma;\Psi \vdash_{\mathrm{L^C}} m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \\ \Gamma;\Psi \vdash_{\mathrm{L^C}} n_1 - n_2 \leq n \qquad \Gamma;\Psi \vdash_{\mathrm{L^C}} \phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \Rightarrow \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]\end{array}}{\Psi;\Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}\text{R-SUBM1}$$

$$\frac{\Psi;\Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi' \qquad \Psi;\Gamma \vdash_{\mathrm{L^C}} n' \leq n \qquad \Psi;\Gamma \vdash_{\mathrm{L^C}} \forall \mathbf{r}_1, \mathbf{r}_2.\ \phi' \Rightarrow \phi}{\Psi;\Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}\text{R-SUBM2}$$

$$\frac{\Gamma;\Psi \vdash m_1 \div \tau_1 \mid k_1 \mid \ell_1 \mid \phi_1 \qquad \Gamma;\Psi \vdash m_2 \div \tau_2 \mid k_2 \mid \ell_2 \mid \phi_2 \qquad \Gamma;\Psi \vdash_{\mathrm{L^C}} \ell_1 - k_2 \leq n}{\Gamma;\Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}]}\text{R-SPLIT}$$

$$\frac{\Gamma;\Psi \vdash e_1 : \tau_1 \mid \phi_1 \qquad \Gamma;\Psi \vdash e_2 : \tau_2 \mid \phi_2}{\Gamma;\Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}]}\text{R-R}^{\mathrm{C}}\text{-U}^{\mathrm{C}}$$

$$\frac{\begin{array}{c}\Gamma;\Psi \vdash e_1 : \tau_1 \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k', \ell, \mathbf{r}.\phi_1) \qquad \Gamma;\Psi \vdash e_2 : \tau_2 \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell', \mathbf{r}.\phi_2) \\ \Gamma;\Psi \vdash_{\mathrm{L^C}} \ell - k \leq n\end{array}}{\Gamma;\Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.\phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}])}\text{R-SPLIT-PURE}$$

3 Metatheory proofs

In this section we give proofs of lemmas and theorems of our framework's metatheory stated in the paper, as well as some supporting lemmas.

3.1 $\mathbf{L^C}$ metatheory

Lemma 1. If m is closed and $m \Downarrow^n e$, then $m \doteq \mathsf{cstep}_n(\mathsf{cret}(e))$.

Proof. By induction on the forcing derivation.

Case $\dfrac{}{\mathsf{cret}(e) \Downarrow^0 e}$

TS: $\mathsf{cret}(e) \doteq \mathsf{cstep}_0(\mathsf{cret}(e))$.

Immediately from $\mathsf{cstep}_0(\mathsf{cret}(e)) \to_\zeta \mathsf{cret}(e)$.

Case $\dfrac{e_1 \to_\beta^* \{m_1\} \qquad m_1 \Downarrow^{n_1} e_1' \qquad m_2[e_1'/x] \Downarrow^{n_2} e_2}{\mathsf{cbind}(e_1, \{x\}.m_2) \Downarrow^{n+n_2} e_2}$

TS: $\mathsf{cbind}(e_1, \{x\}.\, m_2) \doteq \mathsf{cstep}_{n_1+n_2}(\mathsf{cret}(e_2))$.

From IH on the second premise we have $m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'))$, and then from the first premise we have $e_1 \doteq \{m_1\} \doteq \{\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'))\}$.

Then we have

$$\begin{array}{lcl}\mathsf{cbind}(e_1, \{x\}\, m_2) & \doteq & \mathsf{cbind}(\{\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'))\}, \{x\}.\, m_2) \\ & \to_\zeta & \{\!|(\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1')))/x|\!\}m_2 \\ & = & \mathsf{cstep}_{n_1}(\{\!|\mathsf{cret}(e_1')/x|\!\}m_2) \\ & = & \mathsf{cstep}_{n_1}(m_2[e_1'/x])\end{array}$$

By IH on the second premise we have $m_2[e_1'/x] \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$. Hence, $\mathsf{cbind}(e_1, \{x\}\, m_2) \doteq \mathsf{cstep}_{n_1}(\mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))) \to_\zeta \mathsf{cstep}_{n_1+n_2}(\mathsf{cret}(e_2))$, as required.

Case $\dfrac{m \Downarrow^{n'} e}{\mathsf{cstep}_n(m) \Downarrow^{n+n'} e}$

TS: $\mathsf{cstep}_n(m) \doteq \mathsf{cstep}_{n+n'}(\mathsf{cret}(e))$.

By IH on the premise we have $m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e))$. Hence, $\mathsf{cstep}_n(m) \doteq \mathsf{cstep}_n(\mathsf{cstep}_{n'}(\mathsf{cret}(e)) \to_\zeta \mathsf{cstep}_{n+n'}(\mathsf{cret}(e))$, as required.

□

Theorem 2 (Soundness (typing))**.** Let $\rho \vDash \Gamma$ mean that for each $x \in dom(\Gamma)$, $\rho(x) \in [\![\Gamma(x)]\!]$. Then,

1. If $\Gamma \vdash e : \tau$ and $\rho \vDash \Gamma$, then $(\!|e|\!)_\rho \in [\![\tau]\!]$.
2. If $\Gamma \vdash m \div \tau$ and $\rho \vDash \Gamma$, then $(\!|m|\!)_\rho \in [\![\tau]\!] \times \mathbb{R}^\infty$.

Proof. By simultaneous induction on the given typing derivations. We show some representative cases.

Proof of (1)

Case $\dfrac{c \in \mathsf{b}}{\Gamma \vdash c : \mathsf{b}}$

TS: $(\!|c|\!)_\rho \in [\![\mathsf{b}]\!]$.

Immediately by expanding the definitions.

Case $\dfrac{i \in \{1,2\} \qquad \Gamma \vdash e : \tau_i}{\Gamma \vdash \mathsf{inj}_i\ e : \tau_1 + \tau_2}$

TS: $(\!|\mathsf{inj}_i\ e|\!)_\rho \in [\![\tau_1 + \tau_2]\!]$.

Fix i. The result follows immediately by IH on the typing premise.

Case $\dfrac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 : \tau}$

TS: $(\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho \in [\![\tau]\!]$.

From the IH on the first premise, $(\!|e|\!)_\rho \in [\![\tau_1 + \tau_2]\!]$, hence $(\!|e|\!)_\rho = \mathsf{inj}_i\ v$ and $v \in [\![\tau_i]\!]$ for either $i = 1$ or $i = 2$. We consider the case where $i = 1$; the other case is similar. By IH on the second premise, we have $(\!|e_1|\!)_\rho \in [\![\tau_1 \to \tau]\!]$, hence $(\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho \triangleq (\!|e_1|\!)_\rho\ v \in [\![\tau]\!]$, as required.

Case $\dfrac{K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma \vdash e_i : \sigma_i \text{ for all } 1 \le i \le n}{\Gamma \vdash K(e_1, \ldots, e_n) : \theta}$

TS: $K((\!|e_1|\!)_\rho, \ldots, (\!|e_n|\!)_\rho) \in [\![\theta]\!]$.

By IH on e_i premises we have $(\!|e_i|\!)_\rho \in [\![\sigma_i]\!]$, hence by expanding the definition, there exists q_i, s.t. $(\!|e_i|\!)_\rho \in [\![\sigma_i]\!]_{q_i}$, for all $1 \le i \le n$. Therefore, $K((\!|e_1|\!)_\rho, \ldots, (\!|e_n|\!)_\rho) \in [\![\theta]\!]_{\max\{q_i | 1 \le i \le n\}+1}$, and then $K((\!|e_1|\!)_\rho, \ldots, (\!|e_n|\!)_\rho) \in [\![\theta]\!]$, as required.

Case $\dfrac{\Gamma \vdash e : \theta \qquad \theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \quad \Gamma \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau \text{ for all } 1 \le i \le n}{\Gamma \vdash \mathsf{match}\ x\ \mathsf{with}\ K_1 \mapsto e_1; \cdots ; K_n \mapsto e_n : \tau}$

TS: $(\!|\mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \cdots ; K_n \mapsto e_n|\!)_\rho \in [\![\tau]\!]$.

From the IH on the first premise, $(\!|e|\!)_\rho \in [\![\theta]\!]$. Therefore, $(\!|e|\!)_\rho = K_i(v_1, \ldots, v_{a_i})$, for some $1 \le i \le n$, and $v_j \in [\![\sigma_{i,j}]\!]$ for all $1 \le j \le a_i$.

By IH on e_i's premise, we have that $(\!|e_i|\!)_\rho \in [\![\sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau]\!]$, hence $(\!|e_i|\!)_\rho\ v_1\ \cdots\ v_{a_i} \in [\![\tau]\!]$. Since, $(\!|\mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \cdots ; K_n \mapsto e_n|\!)_\rho \triangleq (\!|e_i|\!)_\rho\ v_1\ \cdots\ v_{a_i}$, the required result follows immediately.

Case $\dfrac{\Gamma \vdash m \div \tau}{\Gamma \vdash \{m\} : \mathbb{C}(\tau)}$

TS: $(\!|\{m\}|\!) \in [\![\mathbb{C}(\tau)]\!]$.

From IH on the premise we have $(\!|m|\!)_\rho \in [\![\tau]\!] \times \mathbb{R}^\infty$, hence the result follows immediately by expanding the definitions.

Proof of (2)

Case $\dfrac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathsf{cret}(e) \div \tau}$

TS: $(\!|\mathsf{cret}(e)|\!)_\rho \in [\![\tau]\!] \times \mathbb{R}^\infty$.

From IH on the premise we have $(\!|e|\!)_\rho \in [\![\tau]\!]$, hence $(\!|\mathsf{cret}(e)|\!)_\rho \triangleq ((\!|e|\!)_\rho, 0) \in [\![\tau]\!] \times \mathbb{R}^\infty$.

Case $\dfrac{\Gamma \vdash n : \mathbb{R}^\infty \qquad \Gamma \vdash m \div \tau}{\Gamma \vdash \mathsf{cstep}_n(m) \div \tau}$

TS: $(\!|\mathsf{cstep}_n(m)|\!) \in [\![\tau]\!] \times \mathbb{R}^\infty$.

By IH on the second premise we have $(\!|m|\!)_\rho \in [\![\tau]\!] \times \mathbb{R}^\infty$, and then $(\!|m|\!)_\rho = (v, n')$ for some $v \in [\![\tau]\!]$ and $n' \in \mathbb{R}^\infty$. By IH on the first premise, $(\!|n|\!)_\rho \in \mathbb{R}^\infty$. Therefore $(\!|\mathsf{cstep}_n(m)|\!)_\rho \triangleq (v, (\!|n|\!)_\rho + n') \in [\![\tau]\!] \times \mathbb{R}^\infty$.

Case $\dfrac{\Gamma \vdash e : \mathbb{C}(\tau') \qquad \Gamma, x : \tau' \vdash m \div \tau}{\Gamma \vdash \mathsf{cbind}(e, \{x\}.\, m) \div \tau}$

TS: $(\!|\mathsf{cbind}(e, \{x\}.\, m)|\!)_\rho \in [\![\tau]\!] \times \mathbb{R}^\infty$.

By IH on the first premise we have $(\!|e|\!)_\rho \in [\![\mathbb{C}(\tau')]\!] \triangleq [\![\tau]\!] \times \mathbb{R}^\infty$, hence $(\!|e|\!)_\rho = (v', n_1)$ and $v' \in [\![\tau']\!]$ and $n_1 \in \mathbb{R}^\infty$.

Let $\rho' = \rho[v'/x]$. By IH on the second premise we have $(\!|m|\!)_{\rho'} \in [\![\tau]\!] \times \mathbb{R}^\infty$, hence $(\!|m|\!)_{\rho'} = (v, n_2)$, $v \in [\![\tau]\!]$ and $n_2 \in \mathbb{R}^\infty$.

Finally, then $(\!|\mathsf{cbind}(e, \{x\}.\, m)|\!)_\rho = (v, n_1 + n_2) \in [\![\tau]\!] \times \mathbb{R}^\infty$.

□

Lemma 3. The following hold:

1. If $\Gamma \vdash e_1 : \tau$ and $e_1 \to_{\beta,\iota,\mu,\zeta} e_2$ and $\rho \vDash \Gamma$, then $(\!|e_1|\!)_\rho = (\!|e_2|\!)_\rho$.

2. If $\Gamma \vdash m_1 \div \tau$ and $m_1 \to_{\beta,\iota,\mu,\zeta} m_2$ and $\rho \vDash \Gamma$, then $(\!|m_1|\!)_\rho = (\!|m_2|\!)_\rho$.

Proof. By simultaneous induction on the typing derivations and case analysis of $\to_{\beta,\iota,\mu,\zeta}$.

Proof of (1)

Case $\dfrac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 : \tau}$

Assume: $\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 \to_{\beta,\iota,\mu,\zeta} e'$.

We case-analyze $\to_{\beta,\iota,\mu,\zeta}$:

Subcase $\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 \to_{\beta,\iota,\mu,\zeta} \mathsf{case}\ e'\ \mathsf{of}\ e_1; e_2 \quad$ and $\quad e \to_{\beta,\iota,\mu,\zeta} e'$

TS: $(\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho = (\!|\mathsf{case}\ e'\ \mathsf{of}\ e_1; e_2|\!)_\rho$.
By IH we have $(\!|e|\!)_\rho = (\!|e'|\!)_\rho$, hence the result holds.

Subcase $\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 \to_{\beta,\iota,\mu,\zeta} \mathsf{case}\ e\ \mathsf{of}\ e_1'; e_2 \quad$ and $\quad e_1 \to_{\beta,\iota,\mu,\zeta} e_1'$

TS: $(\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho = (\!|\mathsf{case}\ e\ \mathsf{of}\ e_1'; e_2|\!)_\rho$.
By IH we have $(\!|e_1|\!)_\rho = (\!|e_1'|\!)_\rho$, hence the result holds.

Subcase $\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2 \to_{\beta,\iota,\mu,\zeta} \mathsf{case}\ e\ \mathsf{of}\ e_1; e_2' \quad$ and $\quad e_2 \to_{\beta,\iota,\mu,\zeta} e_2'$

TS: $(\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho = (\!|\mathsf{case}\ e\ \mathsf{of}\ e_1; e_2'|\!)_\rho$.
By IH we have $(\!|e_2|\!)_\rho = (\!|e_2'|\!)_\rho$, hence the result holds.

Subcase $\mathsf{case}\ \mathsf{inj}_i\ e\ \mathsf{of}\ e_1; e_2 \to_{\beta,\iota,\mu,\zeta} e_i\ e$

TS: $(\!|\mathsf{case}\ \mathsf{inj}_i\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho = (\!|e_i\ e|\!)_\rho$.
Since $(\!|\mathsf{case}\ \mathsf{inj}_i\ e\ \mathsf{of}\ e_1; e_2|\!)_\rho \triangleq (\!|e_i|\!)_\rho\ e$, the result holds.

Case $\dfrac{\begin{array}{c}\Gamma \vdash e : \theta \qquad \theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau \text{ for all } 1 \le i \le n\end{array}}{\Gamma \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n : \tau}$

Assume (for an arbitrary $1 \le i \le n$):

$$\mathsf{match}\ K_i(e_1', \ldots, e_{a_i}')\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n \to_\iota e_i\ e_1'\ \cdots\ e_{a_i}'$$

TS: $(\!|\mathsf{match}\ K_i(e_1', \ldots, e_{a_i}')\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n|\!)_\rho = (\!|e_i\ e_1'\ \cdots\ e_{a_i}'|\!)_\rho$.

Since $(\!|K_i(e_1', \ldots, e_{a_i}')|\!)_\rho \triangleq K_i((\!|e_1'|\!)_\rho, \ldots, (\!|e_{a_i}'|\!)_\rho)$, we have:

$$(\!|\mathsf{match}\ K_i(e_1', \ldots, e_{a_i}')\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n|\!)_\rho \triangleq (\!|e_i|\!)_\rho\ (\!|e_1'|\!)_\rho\ \cdots\ (\!|e_{a_i}'|\!)_\rho$$

as required.

The congruence rules are the same as in the previous case.

Case $\dfrac{\Gamma \vdash m \div \tau}{\Gamma \vdash \{m\} : \mathbb{C}(\tau)}$

Assume: $\{m\} \to_{\beta,\iota,\mu,\zeta} \{m'\}$ and $m \to_{\beta,\iota,\mu,\zeta} m'$.

TS: $(\!|\{m\}|\!))_\rho = (\!|\{m'\}|\!)_\rho$.

By the definition this reduces to $(\!|m|\!)_\rho = (\!|m'|\!)_\rho$. This follows by IH on m.

Proof of (2)

Case $\dfrac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathsf{cret}(e) \div \tau}$

Assume: $\mathsf{cret}(e) \to_{\beta,\iota,mu,\zeta} \mathsf{cret}(e')$ and $e \to_{\beta,\iota,\mu,\zeta} e'$.

TS: $(\!|\mathsf{cret}(e)_\rho|\!) \doteq (\!|\mathsf{cret}(e')|\!)_\rho$.

By the definition this reduces to $(\!|e|\!)_\rho \doteq (\!|e'|\!)_\rho$. This follows by IH on the premise.

Case $\dfrac{\Gamma \vdash n : \mathbb{R}^\infty \qquad \Gamma \vdash m \div \tau}{\Gamma \vdash \mathsf{cstep}_n(m) \div \tau}$

Subcase $\mathsf{cstep}_n(\mathsf{cstep}_{n'}(m')) \to_\zeta \mathsf{cstep}_{n+n'}(m')$

TS: $(\!|\mathsf{cstep}_n(\mathsf{cstep}_{n'}(m'))|\!)_\rho = (\!|\mathsf{cstep}_{n+n'}(m')|\!)_\rho$.

We have:

$$\begin{array}{ll} & (\!|\mathsf{cstep}_n(\mathsf{cstep}_{n'}(m'))|\!)_\rho \\ \triangleq & (\pi_1(\!|\mathsf{cstep}_{n'}(m')|\!)_\rho, \pi_2(\!|\mathsf{cstep}_{n'}(m')|\!)_\rho + n) \\ \triangleq & (\pi_1(\!|m'|\!)_\rho, \pi_2(\!|m'|\!)_\rho + n + n') \\ \triangleq & (\!|\mathsf{cstep}_{n+n'}(m')|\!)_\rho \end{array}$$

Subcase $\mathsf{cstep}_0(m) \to_\zeta m$

TS: $(\!|\mathsf{cstep}_0(m)|\!)_\rho = (\!|m|\!)_\rho$.

We have $(\!|\mathsf{cstep}_0(m)|\!)_\rho \triangleq (\pi_1(\!|m|\!)_\rho, \pi_2(\!|m|\!)_\rho + 0) \triangleq (\!|m|\!)_\rho$, as required.

Case $\dfrac{\Gamma \vdash e : \mathbb{C}(\tau') \qquad \Gamma, x : \tau' \vdash m \div \tau}{\Gamma \vdash \mathsf{cbind}(e, \{x\}.\, m) \div \tau}$

Assume: $\mathsf{cbind}(\{m_1\}, \{x\}.\, m_2) \to_\iota \{\!|m_1/x|\!\}m_2$.

TS: $(\!|\mathsf{cbind}(\{m_1\}, \{x\}.\, m_2)|\!)_\rho = (\!|\{\!|m_1/x|\!\}m_2|\!)_\rho$.

Let $(\!|m_1|\!)_\rho = (v', n_1) \in [\![\tau]\!] \times \mathbb{R}^\infty$, $\rho' = \rho[v'/x]$, and $(\!|m_2|\!)_{\rho'} = (v, n_2)$ (this well-defined, since $\{m_1\}$ and m_2 are well-typed). Then, we have $(\!|\mathsf{cbind}(\{m_1\}, \{x\}.\, m_2)|\!)_\rho \triangleq (v, n_1 + n_2)$.

We show that $(\!|\{\!|m_1/x|\!\}m_2|\!)_\rho = (v, n_1 + n_2)$ by sub-induction on m_1.

Subcase $m_1 \equiv \mathsf{cret}(e')$

TS: $(\!|\{\!|\mathsf{cret}(e')/x|\!\}m_2|\!)_\rho = (v, n_1 + n_2)$.

From the assumption we have $n_1 = 0$ and $(\!|e'|\!)_\rho = v'$. We have $(\!|\{\!|\mathsf{cret}(e')/x|\!\}m_2|\!)_\rho \triangleq (\!|m_2[e'/x]|\!)_\rho = (\!|m_2|\!)_{\rho'} = (v, n_2 + 0) = (v, n_1 + n_2)$, as required.

Subcase $m_1 \equiv \mathsf{cstep}_n(m')$

TS: $(\!|\{\!|\mathsf{cstep}_n(m')/x|\!\}m_2|\!)_\rho = (v, n_1 + n_2)$.

From the assumption we have $(\!|m'|\!)_\rho = (v', n_1')$ and $n_1 = n + n_1'$. By sub-IH on m' we then have $(\!|\{\!|m'/x|\!\}m_2|\!)_\rho = (v, n_1' + n_2)$. Hence, $(\!|\{\!|\mathsf{cstep}_n(m')/x|\!\}m_2|\!)_\rho = (v, n_1' + n + n_2) = (v, n_1 + n_2)$.

Subcase $m_1 \equiv \mathsf{cbind}(e', \{y\}.\, m')$

TS: $(\!|\{\!|\mathsf{cbind}(e', \{y\}.\, m')/x|\!\}m_2|\!)_\rho = (v, n_1 + n_2)$.
Let $(\!|e|\!)_\rho = (v'', n_1')$ and $(\!|m'|\!)_{\rho[v''/y]} = (v', n_1'')$ such that $n_1 = n_1' + n_1''$ (note that this is well-defined since by the assumption $e \equiv \{m_1\} \equiv \{\mathsf{cbind}(e', \{y\}.\, m_2')\}$ is well-typed).
Hence, by sub-IH on m' we have $(\!|\{\!|m'/x|\!\}m_2|\!)_{\rho[v''/y]} = (v, n_1'' + n_2)$. The we have $(\!|\{\!|\mathsf{cbind}(e', \{y\}.\, m')/x|\!\}m_2|\!)_\rho \triangleq \mathsf{cbind}(e, \{y\}.\, \{\!|m'/x|\!\}m_2)_\rho = (v, n_1' + n_1'' + n_2) = (v, n_1 + n_2)$, as required.

□

Lemma 4. The following hold:

- For any $v \in [\![\mathsf{b}]\!]$, $\min\{q \mid v \in [\![\mathsf{b}]\!]_q\} = 0$
- For any $K(v_1, \ldots, v_n) \in [\![\theta]\!]$, such that $K(\sigma_1, \ldots, \sigma_n) \in \Theta$: if $n = 0$, then $\min\{q \mid v \in [\![\mathsf{b}]\!]_q\} = 0$, otherwise $\min\{q \mid v \in [\![\theta]\!]_q\} = \max\{q_1, \ldots, q_n\} + 1$, where $q_i = \min\{q \mid v \in [\![\sigma_i]\!]_q\}$, for all $1 \le i \le n$.
- For any $v = (v', n) \in [\![\sigma]\!] \times \mathbb{R}^\infty$, $\min\{q \mid v \in [\![\mathbb{C}(\sigma)]\!]_q\} = \min\{q \mid v' \in [\![\sigma]\!]_q\}$

Proof. Immediate from the definition of $[\![\sigma]\!]_q$. □

Lemma 5. All L^{C} axioms relating to $|\cdot|_\sigma$ are sound for the semantic definition of $|\cdot|_\sigma$.

Proof. We case analyze the axioms that define $|\cdot|_\sigma$.

Case $(\!|\forall x : \mathsf{b}, s : \mathbb{N}.\ |x, s|_\mathsf{b} \Leftrightarrow s \doteq 0|\!)_\emptyset$

We need to show $(\!||x, s|_\mathsf{b} \Leftrightarrow s \doteq 0|\!)_\rho$ for all $v \in [\![\mathsf{b}]\!]$, $n \in \mathbb{N}$ and $\rho = \{x \mapsto v, s \mapsto n\}$.

$\Rightarrow$ direction: Assume: $v \in [\![\mathsf{b}]\!]$ and $n = \min\{q \mid v \in [\![\mathsf{b}]\!]_q\}$.

We need to show $n = 0$. This follows by Lemma 4.

$\Leftarrow$ direction: Assume $n = 0$. We need to show $v \in [\![\mathsf{b}]\!]$ and $0 = \min\{q \mid v \in [\![\mathsf{b}]\!]_q\}$. This follows by the assumptions and Lemma 4.

Case $(\!|\forall x : \theta.\ |x, 0|_\theta \Leftrightarrow \bigvee_{1 \le i \le n | a_i = 0} x \doteq K_i()|\!)_\emptyset$

We need to show $(\!||x, 0|_\theta \Leftrightarrow \bigvee_{1 \le i \le n | a_i = 0} x \doteq K_i()|\!)_\rho$ for all $v \in [\![\theta]\!]$ and $\rho = \{x \mapsto v\}$.
Let $\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta$ be the equation of θ.
$\Rightarrow$ direction: Assume: $(\!||x, 0|_\theta|\!)_{\rho'} \triangleq v \in [\![\theta]\!] \wedge 0 = \min\{q \mid v \in [\![\theta]\!]\}$. TS: $\bigvee_{1 \le i \le n | a_i = 0} v = K_i()$.
From the assumption we have $v \in [\![\theta]\!]_0$, and by the definition $v = K_i()$ for some i where $a_i = 0$, hence the result holds.
$\Leftarrow$ direction: Assume: $\bigvee_{1 \le i \le n | a_i = 0} x = K_i()$. TS: $v \in [\![\theta]\!]$ and $0 = \min\{q \mid v \in [\![\theta]\!]\}$.
Suffices to show: $v \in [\![\theta]\!]_0$; this follows from the assumption, since that means that $x = K_i()$ for some i.

Case $(\!|\forall x : \theta, s : \mathbb{N}.\ |x, s+1|_\theta \Leftrightarrow \bigvee_{1\le i\le n|a_i>0}(\exists x_1 : \sigma_{i,1}, s_1 : \mathbb{N}, \ldots, x_{a_i} : \sigma_{i,a_i}, s_{a_i} : \mathbb{N}.\ x \doteq K_i(x_1,\ldots,x_{a_i}) \wedge \bigwedge_{1\le j\le a_i}(|x, s_j|_{\sigma_{i,j}}) \wedge s \doteq \max\{s_1,\ldots,s_{a_i}\})|\!)_\emptyset$

We need to show $(\!|x, s+1|_\theta \Leftrightarrow \bigvee_{1\le i\le n|a_i>0}(\exists x_1 : \sigma_{i,1}, s_1 : \mathbb{N}, \ldots, x_{a_i} : \sigma_{i,a_i}, s_{a_i} : \mathbb{N}.\ x \doteq K_i(x_1,\ldots,x_{a_i}) \wedge \bigwedge_{1\le j\le a_i}(|x, s_j|_{\sigma_{i,j}}) \wedge s \doteq \max\{s_1,\ldots,s_{a_i}\})|\!)_\rho$ for all $v \in [\![\theta]\!]$, $k \in \mathbb{N}$ and $\rho = \{x \mapsto v, s \mapsto k\}$.

Let $\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta$ be the equation of θ.

$\Rightarrow$ direction: Assume: $v \in [\![\theta]\!]$ and $k+1 = \min\{q \mid v \in [\![\theta]\!]_q\}$.

Hence, $v = K_i(v_1,\ldots,v_{a_i})$, such that $a_i > 0$ and $v_j \in [\![\sigma_{i,j}]\!]_{k_j}$ for some $k_j \le k$, for all $1 \le j \le a_j$.

We instantiate $x_j := v_{i,j}$ and $s_j := k_j \triangleq \min\{q \mid v_j \in [\![\sigma_{i,j}]\!]\}$ for all $1 \le j \le a_j$. Hence, it remains to show: $k = \max\{k_1,\ldots,k_{a_i}\}$. This follows by Lemma 4.

$\Leftarrow$ direction: Assume $v = K_i(v_1,\ldots,v_{a_i})$ for some $a_i > 0$, $v_j \in [\![\sigma_{i,j}]\!]$, $k_j = \min\{q \mid v_j \in [\![\sigma_{i,j}]\!]\}$ for all $1 \le j \le a_i$, and $k = \max\{k_1,\ldots,k_{a_i}\}$.

We need to show: $v \in [\![\theta]\!]$ and $k+1 = \min\{q \mid [\![\theta]\!]_q\}$.

The first goal follows directly from the assumption. The second goal follows by Lemma 4.

Case $(\!|\forall x : \mathbb{C}(\sigma), s : \mathbb{N}.\ |x, s|_{\mathbb{C}(\sigma)} \Leftrightarrow \exists n : \mathbb{R}^\infty, y : \sigma.\ x \doteq \{\mathsf{cstep}_n(\mathsf{cret}(y))\} \wedge |y, s|_\sigma|\!)_\emptyset$

We need to show $(\!|x, s|_{\mathbb{C}(\sigma)} \Leftrightarrow \exists n : \mathbb{R}^\infty, y : \sigma.\ x \doteq \{\mathsf{cstep}_n(\mathsf{cret}(y))\} \wedge |y, s|_\sigma|\!)_\rho$ for all $(v, n) \in [\![\sigma]\!] \times \mathbb{R}^\infty$, $k \in \mathbb{N}$, and $\rho = \{x \mapsto (v, n), s \mapsto k\}$.

$\Rightarrow$ direction: Assume: $(v, n) \in [\![\mathbb{C}(\sigma)]\!] \triangleq [\![\sigma]\!] \times \mathbb{R}^\infty$ and $k = \min\{q \mid (v, n) \in [\![\mathbb{C}(\sigma)]\!]_q\}$.

We instantiate $n := n$, and $y := v$. We need to show (after expanding the definitions) $(v, n) = (v, n)$, $v \in [\![\sigma]\!]$ and $k = \min\{q \mid v' \in [\![\sigma]\!]\}$.

The first two goals follow directly from the assumptions, the third goal follows by Lemma 4.

$\Leftarrow$ direction: Assume: $v = (n', v')$, $v \in [\![\sigma]\!]$ and $k = \min\{q \mid v \in [\![\sigma]\!]\}$. From this follows that $n' = n$ and $v' = v$.

We need to show $(v, n) \in [\![\mathbb{C}(\sigma)]\!]$, and $k = \min\{q \mid (v, n) \in [\![\mathbb{C}(\sigma)]\!]\}$.

The first goal follows from the definitions, and the second goal follows by Lemma 4.

□

Theorem 6 (Soundness (L^{C}))**.** If $\Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \phi$, $\rho \vDash \Gamma$ and $\bigwedge_{\phi' \in \Psi}(\!|\phi'|\!)_\rho$, then $(\!|\phi|\!)_\rho$.

Proof. By induction on the L^{C} derivation. We consider L^{C} axioms and the additional data-type rules.

Case
$$\frac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma \vdash e : \theta \qquad \Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, e \doteq K_i(x_1,\ldots,x_{a_i}) \vdash_{\mathrm{L}^{\mathrm{C}}} \phi \\ \text{for all } 1 \le i \le n \qquad \text{where } x_1,\ldots,x_{a_i} \notin \phi\end{array}}{\Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \phi}\ \text{ELIM}$$

TS: $(\!|\phi|\!)_\rho$.

By Theorem 2 applied to the premise $\Gamma \vdash e : \theta$, we have $(\!|e|\!)_\rho \in [\![\theta]\!]$, hence by the first premise and the definition of $[\![\theta]\!]$, we have for some $1 \le i \le n$: $(\!|e|\!)_\rho = K_i(v_1,\ldots,v_{a_i})$ and $v_j \in [\![\sigma_{i,j}]\!]$ for all $1 \le j \le a_i$. Let $\rho' = \rho[v_1/x_1]\cdots[v_{a_i}/x_{a_i}]$. Since, $(\!|e|\!)_\rho = K_i(v_1,\ldots,v_{a_i})$,

we can apply IH, and we have $(\!|\phi|\!)_{\rho'}$. However, since $x_1 \ldots, x_{a_i} \notin \phi$, we also have $(\!|\phi|\!)_{\rho}$, as required.

Case $\dfrac{\Gamma, x : \theta; \Psi, \forall y : \theta. |y| < |x| \Rightarrow \phi[y/x] \vdash_{\text{L}^\text{C}} \phi}{\Gamma; \Psi \vdash_{\text{L}^\text{C}} \forall x : \theta.\ \phi}\text{IND}$

TS: $(\!|\forall x : \theta.\ \phi|\!)_\rho$.

This is the same as showing that $(\!|\phi|\!)_{\rho[v/x]}$ for all $v \in [\![\theta]\!]$. For all $v \in [\![\theta]\!]$, we show by induction on $|v|_\theta$ that $(\!|\phi|\!)_{\rho[v/x]}$.

When $|v|_\theta = 0$, we apply the IH to the premise with substitution $\rho' := \rho[v/x]$. This yields: $(\forall y \in [\![\theta]\!].(\!||y| < |x| \Rightarrow \phi[y/x]|\!)_{\rho[v/x]}) \Rightarrow (\!|\phi|\!)_{\rho[v/x]}$. Since $|v|_\theta = 0$, $(\!||y| < |x||\!)_{\rho[v/x]}$ is the same as $(\!||y| < 0|\!))_\emptyset$, which is $\bot$, so $(\forall y \in [\![\theta]\!].(\!||y| < |x| \Rightarrow \phi[y/x]|\!)_{\rho[v/x]})$ is the same as $\top$. Hence, we get $(\!|\phi|\!)_{\rho[v/x]}$, as needed.

When $|v|_\theta > 0$, we again apply the IH on the premise to get

$$(\forall y \in [\![\theta]\!].(\!||y| < |x| \Rightarrow \phi[y/x]|\!)_{\rho[v/x]}) \Rightarrow (\!|\phi|\!)_{\rho[v/x]}$$

which is the same as $(\forall y \in [\![\theta]\!].(\!||y| < |v| \Rightarrow \phi[y/x]|\!)_{\rho[v/x]}) \Rightarrow (\!|\phi|\!)_{\rho[v/x]}$. Now,

$$\forall y \in [\![\theta]\!].(\!||y| < |v| \Rightarrow \phi[y/x]|\!)_{\rho[v/x]}$$

holds by the subinduction hypothesis, so we get $(\!|\phi|\!)_{\rho[v/x]}$, as needed.

Case $\dfrac{\theta_1, \theta_2 \in \Theta \qquad \Gamma, x_1 : \theta_1, x_2 : \theta_2; \Psi, \forall y_1 : \theta_1, y_2 : \theta_2.\ (|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2] \vdash_{\text{L}^\text{C}} \phi}{\Gamma; \Psi \vdash_{\text{L}^\text{C}} \forall x_1 : \theta_1, x_2 : \theta_2.\ \phi}\text{DBLIND}$

This is the same as showing that $(\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$ for all $v_1 \in [\![\theta_1]\!]$ and $v_2 \in [\![\theta_2]\!]$. For all $v_1 \in [\![\theta_1]\!]$ and $v_2 \in [\![\theta_2]\!]$, we show by induction on $(|v_1|, |v_2|)$ that $(\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$.

When $(|v_1|, |v_2|) = (0, 0)$, we apply the IH to the premise with substitution $\rho' := \rho[v_1/x_1][v_2/x_2]$. This yields: $(\forall y_1 \in [\![\theta_1]\!], y_2 \in [\![\theta_2]\!].(\!|(|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2]|\!)_{\rho[v_1/x_1][v_2/x_2]}) \Rightarrow (\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$. Since $(|v_1|, |v_2|) = (0, 0)$, $(\!|(|y_1|, |y_2|) < (|x_1|, |x_2|\!)_{\rho[v_1/x_1][v_2/x_2]}$ is the same as $(\!|(|y_1|, |y_2|) < (0, 0)|\!))_\emptyset$, which is $\bot$; therefore

$$\forall y_1 \in [\![\theta_1]\!], y_2 \in [\![\theta_2]\!].\ (\!|(|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2]|\!)_{\rho[v_1/x_1][v_2/x_2]}$$

is the same as $\top$. Hence, we get $(\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$, as needed.

When $(|v_1|, |v_2|) > (0, 0)$, we again apply the IH on the premise to get $(\forall y_1 \in [\![\theta_1]\!], y_2 \in [\![\theta_2]\!].(\!|(|y_1|, |y_2|) < (|x_1|, |x_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2]|\!)_{\rho[v_1/x_1][v_2/x_2]}) \Rightarrow (\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$, which is the same as $(\forall y_1 \in [\![\theta_1]\!], y_2 \in [\![\theta_2]\!].(\!|(|y_1|, |y_2|) < (|v_1|, |v_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2]|\!)_{\rho[v_1/x_1][v_2/x_2]}) \Rightarrow (\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$. Now, $\forall y_1 \in [\![\theta]\!].y_2 \in [\![\theta_2]\!](\!|(|y_1|, |y_2|) < (|v_1|, |v_2|) \Rightarrow \phi[y_1/x_1][y_2/x_2]|\!)_{\rho[v_1/x_1][v_2/x_2]}$ holds by the subinduction hypothesis, so we get $(\!|\phi|\!)_{\rho[v_1/x_1][v_2/x_2]}$, as needed.

Case $\dfrac{}{\Gamma; \Psi \vdash_{\text{L}^\text{C}} e \doteq e}\text{REFL}$

TS: $(\!|e \doteq e|\!)_\rho$.

Immediately by reflexivity of $=$.

Case $\dfrac{\Gamma; \Psi \vdash_{\text{LC}} e_1 \doteq e_2 \qquad \Gamma; \Psi \vdash_{\text{LC}} \phi[e_1/x]}{\Gamma; \Psi \vdash_{\text{LC}} \phi[e_2/x]}\text{SUBST}$

TS: $(\!|\phi[e_2/x]|\!)_\rho$.

This is equivalent to $(\!|\phi|\!)_{\rho[v/x]}$ where $v = (\!|e_2|\!)_\rho$. By IH on the first premise $(\!|e_1 \doteq e_2|\!)_\rho$, hence $(\!|e_1|\!)_\rho = (\!|e_2|\!)_\rho = v$. By IH on the second premise $(\!|\phi[e_1/x]|\!)_\rho = (\!|\phi|\!)_{\rho[v/x]}$, as required.

Case $\dfrac{\Gamma \vdash e_1 : \tau \qquad e_1 \to_{\beta,\iota,\mu,\zeta} e_2}{\Gamma; \Psi \vdash_{\text{LC}} e_1 \doteq e_2}\text{BETA}$

TS: $(\!|e_1 \doteq e_2|\!)_\rho$.

Immediately by Lemma 3 applied to the premises.

Case $\dfrac{K \neq K'}{\Gamma; \Psi \vdash_{\text{LC}} K(e_1, \ldots, e_n) \not\doteq K'(e'_1, \ldots, e'_m)}\text{NC}$

TS: $(\!|K(e_1, \ldots, e_n) \not\doteq K'(e'_1, \ldots, e'_m)|\!)_\rho$.

This reduces to $(\!|K(e_1, \ldots, e_n)|\!)_\rho \neq (\!|K(e'_1, \ldots, e'_m|\!)_\rho$, and further to $K((\!|e_1|\!)_\rho, \ldots, (\!|e_n|\!)_\rho) \neq K'((\!|e'_1|\!)_\rho, \ldots, (\!|e'_m|\!)_\rho)$, which follows immediately by the assumption $K \neq K'$.

Case $\dfrac{\Gamma; \Psi \vdash_{\text{LC}} K(e_1, \ldots, e_n) \doteq K'(e'_1, \ldots, e'_n)}{\Gamma; \Psi \vdash_{\text{LC}} e_i \doteq e'_i}\text{CONS}$

TS: $(\!|e_i \doteq e'_i|\!)_\rho$ for any $1 \leq i \leq n$.

This reduces to $(\!|e_i|\!)_\rho = (\!|e'_i|\!)_\rho$.

By IH we have $(\!|K(e_1, \ldots, e_n) \doteq K'(e'_1, \ldots, e'_n)|\!)_\rho$, and then $(\!|K(e_1, \ldots, e_n)|\!)_\rho = (\!|K'(e'_1, \ldots, e'_n)|\!)_\rho$ and further $K((\!|e_1|\!)_\rho, \ldots, (\!|e_n|\!)_\rho) = K((\!|e'_1|\!)_\rho, (\!|e'_n|\!)_\rho)$, and hence $(\!|e_i|\!)_\rho = (\!|e'_i|\!)_\rho$, for all $1 \leq i \leq n$, as required.

Case $\dfrac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall x : \mathbb{C}(\tau).\ \exists m \div \tau.\ x \doteq \{m\}}$

TS: $(\!|\forall x : \mathbb{C}(\tau).\ \exists m \div \tau.\ x \doteq \{m\}|\!)_\rho$.

Let $v \in [\![\mathbb{C}(\tau)]\!] \triangleq [\![\tau]\!] \times \mathbb{R}^\infty$. Then, RTS: $(\!|\exists m \div \tau.\ x \doteq \{m\}|\!)_{\rho[v/x]}$.

We pick $m := v$, and then RTS: $(\!|x \doteq \{m\}|\!)_{\rho[v/x][v/m]} \triangleq v = v$, which holds trivially.

Case $\dfrac{}{\Gamma; \Psi \vdash_{\text{LC}} \forall m \div \tau.\ \exists n : \mathbb{R}^\infty, e : \tau.\ m \doteq \mathsf{cstep}_n(\mathsf{cret}(e))}$

TS: $(\!|\forall m \div \tau.\ \exists n : \mathbb{R}^\infty, e : \tau.\ m \doteq \mathsf{cstep}_n(\mathsf{cret}(e))|\!)_\rho$.

Let $v \in [\![\tau]\!] \times \mathbb{R}^\infty$, hence $v = (v', q)$, where $v' \in [\![\tau]\!]$ and $q \in \mathbb{R}^\infty$. Further, we pick $n := q$ and $e := v'$. Then RTS: $(\!|m \doteq \mathsf{cstep}_n(\mathsf{cret}(e))|\!)_{\rho[(v'/q)/m][q/n][v'/w]} \triangleq (v', q) = (v', q+0)$, which holds immediately.

Case $\dfrac{}{\Gamma; \Psi \vdash_{\text{LC}} \{m_1\} \doteq \{m_2\} \Rightarrow m_1 \doteq m_2}$

TS: $(\!|\{m_1\} \doteq \{m_2\} \Rightarrow m_1 \doteq m_2|\!)_\rho$.

Assume $(\!|\{m_1\} \doteq \{m_2\}|\!)_\rho$, which is equivalent to $(\!|m_1|\!)_\rho = (\!|m_2|\!)_\rho$. RTS: $(\!|m_1 \doteq m_2|\!)_\rho$, which is equivalent to $(\!|m_1|\!)_\rho = (\!|m_2|\!)_\rho$, which we have assumed.

Case $\dfrac{}{\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \Rightarrow e_1 \doteq e_2 \wedge n_1 \doteq n_2}$

TS: $(\!|\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \Rightarrow e_1 \doteq e_2 \wedge n_1 \doteq n_2|\!)_\rho$.

Assume $(\!|\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))|\!)_\rho$, which is equivalent to $((\!|e_1|\!)_\rho, n_1 + 0) = ((\!|e_2|\!)_\rho, n_2 + 0)$. RTS: $(\!|e_1|\!)_\rho = (\!|e_2|\!)_\rho$ and $n_1 = n_2$, which follows trivially from the assumption.

□

3.2 $\mathrm{U^C}$ metatheory

Theorem 7 ($\mathrm{U^C} \Rightarrow \mathrm{L^C}$)**.** The following hold:

1. If $\Gamma; \Psi \vdash e : \tau \mid \phi$ then $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[e/\mathbf{r}]$.
2. If $\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi$ then $\Gamma; \Psi \vdash_{\mathrm{L^C}} \exists e', n.\ m \doteq \mathsf{cstep}_n(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \leq n \leq \ell$.

Proof. By simultaneous induction on the given $\mathrm{U^C}$ derivations.

Proof of (1)

Case. $\dfrac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma; \Psi \vdash e : \theta \mid \phi' \qquad \text{For all } 1 \leq i \leq n: \quad \Gamma; \Psi \vdash e_i : \sigma_{i,1} \to \cdots \sigma_{i,a_i} \to \tau \mid \phi_i' \quad \text{where} \\ \phi_i' \equiv \forall x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}.\ \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}] \Rightarrow \phi[(\mathbf{r}\ x_1\ \cdots\ x_{a_i})/\mathbf{r}]\end{array}}{\Gamma; \Psi \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n : \tau \mid \phi}$ U-MATCH

To show: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[(\mathsf{match}\ x\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n)/\mathbf{r}]$.
By the $\mathrm{L^C}$ ELIM rule for θ, we need to show:

$$\Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, e \doteq K_i(x_1, \ldots, x_{a_i}) \vdash_{\mathrm{L^C}} \phi[(\mathsf{match}\ e\ \mathsf{with}\ \ \cdots)/\mathbf{r}]$$

for all $1 \leq i \leq n$. Fix some i. This is reduced to:

$$\Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, e = K_i(x_1, \ldots, x_{a_i}) \vdash_{\mathrm{L^C}} \phi[(e_i\ x_1\ \cdots\ x_{a_i})/\mathbf{r}]$$

By IH on the first premise we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi'[e/\mathbf{r}]$, and then combined with the above (and weakening):

$$\Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}] \vdash_{\mathrm{L^C}} \phi[(e_i\ x_1\ \cdots\ x_{a_i})/\mathbf{r}]$$

By IH on the given i we obtain (after eliminating $\forall$ and $\Rightarrow$):

$$\Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}] \vdash_{\mathrm{L^C}} \phi[(\mathbf{r}\ x_1\ \cdots\ x_{a_i})/\mathbf{r}][e_i/\mathbf{r}]$$

which is equivalent to the goal that we need to show.

Case. $\dfrac{\begin{array}{c}K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma; \Psi \vdash e_i : \sigma_i \mid \phi_i \quad \text{for all } 1 \leq i \leq n \\ \Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x_1 : \sigma_1, \ldots, x_n : \sigma_n.\ \phi_1[x_1/\mathbf{r}] \Rightarrow \cdots \Rightarrow \phi_n[x_n/\mathbf{r}] \Rightarrow \phi[K(x_1, \ldots, x_n)/\mathbf{r}]\end{array}}{\Gamma; \Psi \vdash K(e_1, \ldots, e_n) : \theta \mid \phi}$ U-CONS

To show: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[K(e_1, \ldots, e_n)/\mathbf{r}]$.
By IH on e_i premises we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi_i[e_i/\mathbf{r}]$ (*), for all $1 \leq i \leq n$. By instantiating the last premise with $x_i = e_i$, for all $1 \leq i \leq n$, and after eliminating all implications with (*), we obtain $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[K(e_1, \ldots, e_n)/\mathbf{r}]$.

Case. $\dfrac{\begin{array}{c}\mathcal{D}ef(f,x,e)\\ \theta \in \Theta \qquad \Gamma, x:\theta, f:\theta\to\tau; \Psi, \phi', \forall y.\ |y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][f\ y/\mathbf{r}] \vdash e : \tau \mid \phi\end{array}}{\Gamma; \Psi \vdash \mathsf{rec}\, f(x).e : \theta \to \tau \mid \forall x.\ \phi' \Rightarrow \phi[\mathbf{r}\ x/\mathbf{r}]}$ U-LETREC

TS: $\Gamma; \Psi \vdash_{\mathrm{LC}} \forall x.\phi' \Rightarrow \phi[((\mathsf{rec}\, f(x).e)\ x)/\mathbf{r}]$.
Let $\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n})$ be an equation for θ.
By induction principle IND for θ, it suffices to prove that:
$\Gamma, x:\theta; \Psi, \forall y.|y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][((\mathsf{rec}\, f(x).e)\ y)/\mathbf{r}] \vdash_{\mathrm{LC}} \phi' \Rightarrow \phi[((\mathsf{rec}\, f(x).e)\ x)/\mathbf{r}]$
Let $F \triangleq (\mathsf{rec}\, f(x).e)$. Then, the goal above is:
$\Gamma, x:\theta; \Psi, \forall y.|y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][(F\ y)/\mathbf{r}] \vdash_{\mathrm{LC}} \phi' \Rightarrow \phi[(F\ x)/\mathbf{r}]$
which further reduces (by the introduction rule for $\Rightarrow$) to:
$\Gamma, x:\theta; \Psi, \phi', \forall y.|y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][(F\ y)/\mathbf{r}] \vdash_{\mathrm{LC}} \phi[(F\ x)/\mathbf{r}]$
Now note that $F\ x \doteq e[F/f][x/x] \doteq e[F/f]$. Hence, we reduce further to:
$\Gamma, x:\theta; \Psi, \phi', \forall y.|y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][(F\ y)/\mathbf{r}] \vdash_{\mathrm{LC}} \phi[e[F/f]/\mathbf{r}]$

Applying the IH to the third premise of U-LETREC, we get:
$\Gamma, x:\theta, f:\theta\to\tau; \Psi, \phi', \forall y.|y|<|x| \Rightarrow \phi'[y/x] \Rightarrow \phi[y/x][(f\ y)/\mathbf{r}] \vdash_{\mathrm{LC}} \phi[e/\mathbf{r}]$
Instantiating with $f := F$, we get the required result.

Case. $\dfrac{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}{\Gamma; \Psi \vdash \{m\} : \mathbb{C}(\tau) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, \mathbf{r}.\phi)}$ U-MONAD

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \mathbb{C}_{\mathrm{u}}(\{m\}, k, \ell, \mathbf{r}.\phi) \triangleq \exists e', n.\ \{m\} \doteq \{\mathsf{cstep}_n(\mathsf{cret}(e'))\} \wedge \phi[e'/\mathbf{r}] \wedge k \le n \le \ell$.

By IH on the premise, we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n.\ m \doteq \mathsf{cstep}_n(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \le n \le \ell$. The required result trivially follows by picking the same e' and n. (Note that $m \doteq \mathsf{cstep}_n(\mathsf{cret}(e'))$ implies $\{m\} \doteq \{\mathsf{cstep}_n(\mathsf{cret}(e'))\}$.)

Proof of (2)

Case. $\dfrac{\Gamma; \Psi \vdash e : \tau \mid \phi}{\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid 0 \mid 0 \mid \phi}$ U-RET

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ \mathsf{cret}(e) \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge 0 \le n' \le 0$.
We pick $e' := e$ and $n' := 0$. Then, $\mathsf{cret}(e) \doteq \mathsf{cstep}_0(\mathsf{cret}(e))$ by the rule $\mathsf{cstep}_0(m) \to_\zeta m$, $\phi[e/\mathbf{r}]$ by the IH on the premise and $0 \le 0 \le 0$ trivially.

Case. $\dfrac{\Gamma \vdash n : \mathbb{R}^\infty \qquad \Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}{\Gamma; \Psi \vdash \mathsf{cstep}_n(m) \div \tau \mid k+n \mid \ell+n \mid \phi}$ U-STEP

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n''.\ \mathsf{cstep}_n(m) \doteq \mathsf{cstep}_{n''}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge (k+n) \le n'' \le (\ell+n)$.
By IH on the second premise we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e'', n'''.\ m \doteq \mathsf{cstep}_{n'''}(\mathsf{cret}(e'')) \wedge \phi[e''/\mathbf{r}] \wedge k \le n''' \le \ell$. We pick $e' := e''$ and $n'' := n + n'''$. Then, we have:

- $\mathsf{cstep}_n(m) \doteq \mathsf{cstep}_n(\mathsf{cstep}_{n'''}(\mathsf{cret}(e')) \doteq \mathsf{cstep}_{n+n'''}(\mathsf{cret}(e'))$ by rule $\mathsf{cstep}_n(\mathsf{cstep}_{n'}(m)) \to_\zeta \mathsf{cstep}_{n+n'}(m)$.
- $\phi[e'/\mathbf{r}]$ from the IH
- $(k+n) \le n'' \le (\ell+n)$, since $n'' \doteq n + n'''$ and $k \le n''' \le \ell$.

Case. $\dfrac{\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k', \ell', x.\phi_1) \quad \Gamma, x:\tau_1; \Psi, \phi_1 \vdash m_2 \div \tau_2 \mid k \mid \ell \mid \phi_2 \quad x \notin n_2, \phi_2}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau_2 \mid k'+k \mid \ell'+\ell \mid \phi_2}$ U-BIND

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ \mathsf{cbind}(e_1, \{x\}.m_2) \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi_2[e'/\mathbf{r}] \wedge (k' + k) \leq n' \leq (\ell' + \ell)$.

From i.h.(1) on the first premise, $\Gamma; \Psi \vdash_{\mathrm{LC}} \mathbb{C}\mathrm{u}(e_1, k', \ell', x.\phi_1) \triangleq \exists e_1', n_1'.\ e_1 \doteq \{\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))\} \wedge \phi_1[e_1'/x] \wedge k' \leq n_1' \leq \ell'$.

From i.h.(2) on the second premise, $\Gamma, x : \tau_1; \Psi, \phi_1 \vdash_{\mathrm{LC}} \exists e_2', n_2'.\ m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2')) \wedge \phi_2[e_2'/\mathbf{r}] \wedge k \leq n_2' \leq \ell$. Hence, $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_2', n_2'.\ m_2[e_1'/x] \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2')) \wedge \phi_2[e_2'/\mathbf{r}] \wedge k \leq n_2' \leq \ell$ (since $x \notin n_2, \phi_2$).

We choose $e' := e_2'$ and $n' := n_1' + n_2'$. Then, we have:

- $$\begin{array}{lll} \mathsf{cbind}(e_1, \{x\}.m_2) & \doteq & \mathsf{cbind}(\{\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))\}, \{x\}.m_2) \\ & \to_\zeta & \{\!|\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))/x|\!\}m_2 \\ & = & \mathsf{cstep}_{n_1'}(m_2[e_1'/x]) \\ & = & \mathsf{cstep}_{n_1'}(\mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))) \\ & \to_\zeta & \mathsf{cstep}_{n_1'+n_2'}(\mathsf{cret}(e_2')) \\ & = & \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \end{array}$$

- We already know that $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi_2[e_2'/\mathbf{r}] \triangleq \phi_2[e'/\mathbf{r}]$.

- $(k' + k) \leq n' \leq (\ell' + \ell)$, since $n' \doteq n_1' + n_2'$, $k' \leq n_1' \leq \ell'$, and $k \leq n_2' \leq \ell$.

Case. $\dfrac{\Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq k' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \ell' \leq \ell}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}$ U-SUBC

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n''.\ m \doteq \mathsf{cstep}_{n''}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \leq n'' \leq \ell$.

From the i.h., $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n''.\ m \doteq \mathsf{cstep}_{n''}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k' \leq n'' \leq \ell'$. Since $\Gamma; \Psi \vdash_{\mathrm{LC}} k \leq k'$ and $\Gamma; \Psi \vdash_{\mathrm{LC}} \ell' \leq \ell$, the required result follows immediately.

Case. $\dfrac{\Gamma; \Psi \vdash_{\mathrm{LC}} \exists x : \tau.\phi \quad \Gamma, x : \tau; \Psi, \phi \vdash m \div \tau' \mid k \mid \ell \mid \phi' \quad x \notin m, n, \phi'}{\Gamma; \Psi \vdash m \div \tau' \mid k \mid \ell \mid \phi'}$ ∃EM

To show: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi'[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$.

By i.h. on the second premise, $\Gamma, x : \tau; \Psi, \phi \vdash_{\mathrm{LC}} \exists e', n'.\ m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi'[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$. By rule ∃E in L^{C}, $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi'[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$, as required. □

Lemma 8 (Trivial refinements)**.** The following hold:

1. If $\Gamma \vdash e : \tau$ then $\Gamma; \Psi \vdash e : \tau \mid \top$.
2. If $\Gamma \vdash m \div \tau$ then $\Gamma; \Psi \vdash m \div \tau \mid -\infty \mid \infty \mid \top$.

Proof. By simultaneous induction on the given typing derivations.

Proof of (1)

Case $\dfrac{K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma \vdash e_i : \sigma_i \text{ for all } 1 \leq i \leq n}{\Gamma \vdash K(e_1, \ldots, e_n) : \theta}$

TS: $\Gamma; \Psi \vdash K(e_1, \ldots, e_n) : \theta \mid \top$.

By IH on the e_i premises we have $\Gamma; \Psi \vdash e_i : \sigma_i \mid \top$ for all $1 \leq i \leq n$, and the result follows by the rule U-CONS.

Case $\dfrac{\begin{array}{c}\theta = K_1(\sigma_{1,1} \times \cdots \times \sigma_{1,a_1}) + \cdots + K_n(\sigma_{n,1} \times \cdots \times \sigma_{n,a_n}) \in \Theta \\ \Gamma \vdash e : \theta \qquad \Gamma \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau \text{ for all } 1 \leq i \leq n\end{array}}{\Gamma \vdash \mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n : \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{match}\ x\ \mathsf{with}\ K_1 \mapsto e_1; \cdots; K_n \mapsto e_n : \tau \mid \top$.

By IH on the e premise we have $\Gamma; \Psi \vdash e : \theta \mid \top$. By IH on e_i premises we have $\Gamma; \Psi \vdash e_i : \sigma_{i,1} \to \cdots \to \sigma_{i,a_i} \to \tau \mid \top$, for all $1 \le i \le n$. The result follows from the rule U-MATCH.

Case $\dfrac{\mathcal{D}ef(f, x, e) \qquad \Gamma, f : \theta \to \tau, x : \theta \vdash e : \tau}{\Gamma \vdash \mathsf{rec}\, f(x).e : \theta \to \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{rec}\, f(x).e : \theta \to \tau \mid \top$.

By IH we have $\Gamma, f : \theta, x : \theta; \Psi \vdash e : \tau \mid \top$, from which we have $\Gamma, f : \theta, x : \theta; \Psi, \forall y.\ |y| < |x| \Rightarrow \top \Rightarrow \top \vdash e : \tau \mid \top$, and the result follows by the rule U-LETREC.

Case $\dfrac{\Gamma \vdash m \div \tau}{\Gamma \vdash \{m\} : \mathbb{C}(\tau)}$

TS: $\Gamma; \Psi \vdash \{m\} : \mathbb{C}(\tau) \mid \top$.

By IH on the premise we have $\Gamma; \Psi \vdash e \div \tau \mid -\infty \mid \infty \mid \top$, and hence $\Gamma; \Psi \vdash \{m\} : \mathbb{C}(\tau) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, -\infty, \infty, \mathbf{r}.\top)$ by the rule U-MONAD. Since $\Gamma; \Psi \vdash_{\mathrm{LC}} \mathbb{C}_{\mathrm{u}}(\{m\}, -\infty, \infty, \mathbf{r}.\top) \Rightarrow \top$, by U-SUB rule we obtain $\Gamma; \psi \vdash \{m\} : \mathbb{C}(\tau) \mid \top$, as required.

Proof of (2)

Case $\dfrac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathsf{cret}(e) \div \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid -\infty \mid \infty \mid \top$.

By IH on the premise we have $\Gamma; \Psi \vdash e : \tau \mid \top$, and by the U-RET rule we have $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid 0 \mid 0 \mid \top$. By rule U-SUBC, $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid -\infty \mid \infty \mid \top$.

Case $\dfrac{\Gamma \vdash n : \mathbb{R}^{\infty} \qquad \Gamma \vdash m \div \tau}{\Gamma \vdash \mathsf{cstep}_n(m) \div \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{cstep}_n(m) \div \tau \mid -\infty \mid \infty \mid \top$.

By IH on the second premise we have $\Gamma; \Psi \vdash m \div \tau \mid -\infty \mid \infty \mid \top$, and then by the U-STEP rule $\Gamma; \Psi \vdash \mathsf{cstep}_n(m) \div \tau \mid -\infty + n \mid \infty + n \mid \top$. Since $\Gamma; \Psi \vdash_{\mathrm{LC}} \infty + n \le \infty$, and $\Gamma; \Psi \vdash_{\mathrm{LC}} -\infty \le -\infty + n$, the result follows by rule U-SUBC.

Case $\dfrac{\Gamma \vdash e_1 : \mathbb{C}(\tau_1) \qquad \Gamma, x : \tau_1 \vdash m_2 \div \tau_2}{\Gamma \vdash \mathsf{cbind}(e_1, \{x\}.\, m_2) \div \tau_2}$

TS: $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.\, m_2) \div \tau_2 \mid -\infty \mid \infty \mid \tau_2$.

By IH on the premises we have $\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \top$ and $\Gamma, x : \tau_1; \Psi \vdash m_2 \div \tau_2 \mid -\infty \mid \infty \mid \top$.

By the (first two) L^{C} axioms we have $\Gamma; \Psi \vdash_{\mathrm{LC}} e_1 \doteq \{\mathsf{cstep}_n(\mathsf{cret}(e_1'))\}$, for some n and e_1'. Further we have $\Gamma; \Psi \vdash_{\mathrm{LC}} -\infty \le n \le \infty$ and hence $\Gamma; \Psi \vdash_{\mathrm{LC}} \mathbb{C}_{\mathrm{u}}(e_1, -\infty, \infty, r.\top)$ and from that $\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, -\infty, \infty, r.\top)$ (by rule U-SUB on $\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \top$), and then by the rule U-BIND $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.\, m_2) \div \tau_2 \mid (-\infty) + (-\infty) \mid \infty + \infty \mid \top$. The result follows by rule U-SUBC. □

Theorem 9 (L^{C}⇒U^{C}, pure)**.** If $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e/\mathbf{r}]$ and $\Gamma \vdash e : \tau$ then $\Gamma; \Psi \vdash e : \tau \mid \phi$.

Proof. Immediate from Lemma 8(1) and the subsumption rule U-SUB. □

Theorem 10 (L^{C}⇒U^{C}, monadic)**.** If $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \le n' \le \ell$ and $\Gamma \vdash m \div \tau$ then $\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi$.

Proof. By induction on the derivation of $\Gamma \vdash m \div \tau$.

Case. $\dfrac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathsf{cret}(e) \div \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid k \mid \ell \mid \phi$.

From assumption, $\mathsf{cret}(e) = m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e'))$ and, hence, by an L^{C} axiom, $e \doteq e'$ and $n' \doteq 0$. Therefore, from $\phi[e'/\mathbf{r}]$, we get $\phi[e/\mathbf{r}]$. By Theorem 9 on the premise, $\Gamma; \Psi \vdash e : \tau \mid \phi$. By the rule U-RET, we get $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid 0 \mid 0 \mid \phi$ and by U-SUBC, since $k \leq n' \doteq 0 \leq \ell$, we get $\Gamma; \Psi \vdash \mathsf{cret}(e) \div \tau \mid k \mid \ell \mid \phi$, as required.

Case. $\dfrac{\Gamma \vdash n_1 : \mathbb{R}^{\infty} \qquad \Gamma \vdash m' \div \tau}{\Gamma \vdash \mathsf{cstep}_{n_1}(m') \div \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m') \div \tau \mid k \mid \ell \mid \phi$.

From assumption, $\mathsf{cstep}_{n_1}(m') = m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e'))$. By an L^{C} axiom, there is a n_2 such that $n' \doteq n_1 + n_2$ and $m' \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e'))$. From the latter, and the remaining assumptions it follows that $m' \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge (k - n_1) \leq n_2 \leq (\ell - n_1)$. Hence, by i.h. on the second premise, $\Gamma; \Psi \vdash m' \div \tau \mid k - n_1 \mid \ell - n_1 \mid \phi$. By the rule U-STEP if follows that $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m') \div \tau \mid k - n_1 + n_1 \mid \ell - n_1 + n_1 \mid \phi$. By rule U-SUBC, $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m') \div \tau \mid k \mid \ell \mid \phi$, as needed.

Case. $\dfrac{\Gamma \vdash e_1 : \mathbb{C}(\tau_1) \qquad \Gamma, x : \tau_1 \vdash m_2 \div \tau}{\Gamma \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau}$

TS: $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau \mid k \mid \ell \mid \phi$.

From the premise $\Gamma \vdash e_1 : \mathbb{C}(\tau_1)$ and L^{C}'s axioms, it follows that there exist e_1', n_1' such that $\Gamma; \bullet \vdash_{\mathrm{L^C}} e_1 \doteq \{\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))\}$. By definition, it follows that $\Gamma; \bullet \vdash_{\mathrm{L^C}} \mathbb{C}_{\mathrm{u}}(e_1, n_1', n_1', x.x \doteq e_1')$. Applying Theorem 9 on the first premise, we get

$$\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, n_1', n_1', x.x \doteq e_1')$$

Similarly, from the premise $\Gamma, x : \tau_1 \vdash m_2 \div \tau$ and L^{C}'s axioms, it follows that $\Gamma, x : \tau_1; \bullet \vdash_{\mathrm{L^C}} \exists n_2', e_2'.\ m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))$. Consequently, $\Gamma; \bullet \vdash_{\mathrm{L^C}} \exists n_2', e_2'.\ m_2[e_1'/x] \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))$. Therefore, there exist e_2', n_2' such that $\Gamma; \bullet \vdash_{\mathrm{L^C}} m_2[e_1'/x] \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))$. It follows that $\Gamma, x : \tau_1; \Psi, x \doteq e_1' \vdash_{\mathrm{L^C}} m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))$. Call this statement (A).

Further, $\Gamma; \bullet \vdash_{\mathrm{L^C}} \mathsf{cbind}(e_1, \{x\}.m_2) \doteq \mathsf{cbind}(\{\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))\}, \{x\}.m_2) \doteq \mathsf{cstep}_{n_1'}(m_2[e_1'/x]) \doteq \mathsf{cstep}_{n_1'}(\mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))) \doteq \mathsf{cstep}_{n_1'+n_2'}(\mathsf{cret}(e_2'))$. However, from assumption, $\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathsf{cbind}(e_1, \{x\}.m_2) = m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e'))$. Hence, $\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \doteq \mathsf{cstep}_{n_1'+n_2'}(\mathsf{cret}(e_2'))$. By an L^{C} axiom, $\Gamma; \Psi \vdash_{\mathrm{L^C}} n' \doteq n_1' + n_2' \wedge e' \doteq e_2'$.

Next, $\Gamma; \Psi \vdash_{\mathrm{L^C}} e' \doteq e_2'$ and the assumption $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[e'/\mathbf{r}]$ imply (by weakening) that $\Gamma, x : \tau_1; \Psi, x \doteq e_1' \vdash_{\mathrm{L^C}} \phi[e_2'/\mathbf{r}]$. Call this statement (B).

Applying the i.h. to the second premise, using statements (A) and (B), we get

$$\Gamma, x : \tau_1; \Psi, x \doteq e_1' \vdash m_2 \div \tau \mid n_2' \mid n_2' \mid \phi$$

Using the rule U-BIND, we derive $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau \mid n_1' + n_2' \mid n_1' + n_2' \mid \phi$. Since $\Gamma; \Psi \vdash_{\mathrm{L^C}} k \leq n' \doteq n_1' + n_2'$, and $\Gamma; \Psi \vdash_{\mathrm{L^C}} n_1' + n_2' \doteq n' \leq \ell$, by rule U-SUBC, we get $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.m_2) \div \tau \mid k \mid \ell \mid \phi$, as required. □

Corollary 11 (Pure equivalence closure)**.** If $\Gamma; \Psi \vdash e : \tau \mid \phi$ and $\Gamma; \Psi \vdash_{\mathrm{L^C}} e \doteq e'$ and $\Gamma \vdash e' : \tau$, then $\Gamma; \Psi \vdash e' : \tau \mid \phi$.

Proof. From Theorem 7(1) applied to $\Gamma; \Psi \vdash e : \tau \mid \phi$, we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e/\mathbf{r}]$. From $\Gamma; \Psi \vdash_{\mathrm{LC}} e \doteq e'$ we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e'/\mathbf{r}]$. The required result follows from Theorem 9. □

Corollary 12 (Monadic equivalence closure)**.** If $\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi$ and $\Gamma; \Psi \vdash_{\mathrm{LC}} m \doteq m'$ and $\Gamma \vdash m' \div \tau$, then $\Gamma; \Psi \vdash m' \div \tau \mid k \mid \ell \mid \phi$.

Proof. From Theorem 7(2) applied to $\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi$, we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$. Combining with $\Gamma; \Psi \vdash_{\mathrm{LC}} m \doteq m'$, we get $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e', n'.\ m' \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$. The required result follows from Theorem 10. □

Theorem 13 (Subsumption, monadic)**.** The following rules are admissible:

$$\frac{\Gamma; \Psi \vdash_{\mathrm{LC}} m \doteq \mathsf{cstep}_n(\mathsf{cret}(e)) \qquad \Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq n \leq \ell \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \phi'[e/\mathbf{r}] \Rightarrow \phi[e/\mathbf{r}]}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}\ \text{U-SUBM1}$$

$$\frac{\Gamma; \Psi \vdash m \div \tau \mid k' \mid \ell' \mid \phi' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} k \leq k' \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \ell' \leq \ell \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \forall \mathbf{r}.\ \phi' \Rightarrow \phi}{\Gamma; \Psi \vdash m \div \tau \mid k \mid \ell \mid \phi}\ \text{U-SUBM2}$$

Proof. (U-SUBM1) From Theorem 7(2) applied to the first premise, $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, n_1.\ m \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge \phi'[e_1/\mathbf{r}] \wedge k' \leq n_1 \leq \ell'$. Using the second premise, $\Gamma; \Psi \vdash_{\mathrm{LC}} e_1 \doteq e$. From the last premise, $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e/\mathbf{r}]$. The conclusion follows from the third premise and Theorem 10.

(U-SUBM2) From Theorem 7(2) applied to the first premise, $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, n_1.\ m \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge \phi'[e_1/\mathbf{r}] \wedge k' \leq n_1 \leq \ell'$. Using the second, third and fourth premises, $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, n_1.\ m \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge \phi[e_1/\mathbf{r}] \wedge k \leq n_1 \leq \ell$. The conclusion follows from Theorem 10. □

Theorem 14 (Forcing subject reduction)**.** If $\bullet; \bullet \vdash m \div \tau \mid k \mid \ell \mid \phi$ and $m \Downarrow^n e$, then $k \leq n \leq \ell$ and $\bullet; \bullet \vdash e : \tau \mid \phi$.

Proof. Suppose $\bullet; \bullet \vdash m \div \tau \mid k \mid \ell \mid \phi$ and $m \Downarrow^n e$. By Theorem 7(2), there exist e', n' such that $m \doteq \mathsf{cstep}_{n'}(\mathsf{cret}(e')) \wedge \phi[e'/\mathbf{r}] \wedge k \leq n' \leq \ell$. From Lemma 1, it follows that $m \doteq \mathsf{cstep}_n(\mathsf{cret}(e))$, hence $n' \doteq n$, $e' \doteq e$, and $k \leq n \leq \ell$. Further, since $\phi[e'/\mathbf{r}]$ and $e' \doteq e$, we also have $\phi[e/\mathbf{r}]$ and, hence, by Theorem 9, that $\bullet; \bullet \vdash e : \tau \mid \phi$. (We also need to prove that $\bullet \vdash e : \tau$, but this follows from type preservation since $m \Downarrow^n e$ and $\bullet \vdash m \div \tau$ from $\bullet; \bullet \vdash m \div \tau \mid k \mid \ell \mid \phi$.) □

3.3 R^{C} metatheory

Theorem 15 ($\mathrm{R}^{\mathrm{C}} \Rightarrow \mathrm{L}^{\mathrm{C}}$)**.** The following hold:

1. If $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$ then $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$
2. If $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$ then $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$.

Proof. By simultaneous induction on the given R^{C} derivations. We show some representative cases.

Proof of (1)

Case $\dfrac{\begin{array}{c}\mathcal{D}ef(f_1,x_1,e_1)\qquad \mathcal{D}ef(f_2,x_2,e_2)\\ \Gamma, x_1:\theta_1, x_2:\theta_2, f_1:\theta_1\to\tau_1, f_2:\theta_2\to\tau_2;\\ \Psi,\phi,\forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow \phi[y_1/x_1][y_2/x_2]\Rightarrow \phi'[y_1/x_1][y_2/x_2][f_1\ y_1/\mathbf{r}_1][f_2\ y_2/\mathbf{r}_2]\\ \vdash e_1:\tau_1\sim e_2:\tau_2\mid\phi'\end{array}}{\Gamma;\Psi\vdash \mathsf{rec}\, f_1(x_1).e_1:\theta_1\to\tau_1\sim \mathsf{rec}\, f_2(x_2).e_2:\theta_2\to\tau_2\mid \forall x_1x_2.\ \phi\Rightarrow\phi'[\mathbf{r}_1\ x_1/\mathbf{r}_1][\mathbf{r}_2\ x_2/\mathbf{r}_2]}$ R-LETREC

TS: $\Gamma;\Psi\vdash_{\mathrm{L^C}} \forall x_1x_2.\ \phi\Rightarrow\phi'[((\mathsf{rec}\, f_1(x_1).e_1)\ x_1)/\mathbf{r}_1][((\mathsf{rec}\, f_2(x_2).e_2)\ x_2)/\mathbf{r}_2]$.
By double-induction principle DBLIND for θ_1,θ_2, it suffices to show that:

$$\Gamma, x_1:\theta_1, x_2:\theta_2; \Psi, \forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow \phi[y_1/x_1][y_2/x_2]\Rightarrow$$
$$\phi'[y_1/x_1][y_2/x_2][((\mathsf{rec}\, f_1(x_1).e_1)\ y_1)/\mathbf{r}_1][((\mathsf{rec}\, f_2(x_2).e_2)\ y_2)/\mathbf{r}_2]\vdash_{\mathrm{L^C}}\phi\Rightarrow$$
$$\phi'[((\mathsf{rec}\, f_1(x_1).e_1)\ x_1)/\mathbf{r}_1][((\mathsf{rec}\, f_2(x_2).e_2)\ x_2)/\mathbf{r}_2]$$

Let $F_1\triangleq \mathsf{rec}\, f_1(x_1).e_1$ and $F_2\triangleq \mathsf{rec}\, f_2(x_2).e_2$. Then the above goal is:

$$\Gamma, x_1:\theta_1, x_2:\theta_2; \Psi, \forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow \phi[y_1/x_1][y_2/x_2]\Rightarrow$$
$$\phi'[y_1/x_1][y_2/x_2][(F_1\ y_1)/\mathbf{r}_1][(F_2\ y_2)/\mathbf{r}_2]\vdash_{\mathrm{L^C}}\phi\Rightarrow\phi'[(F_1\ x_1)/\mathbf{r}_1][(F_2\ x_2)/\mathbf{r}_2]$$

which is (by the introduction rule for $\Rightarrow$) reduced to:

$$\Gamma, x_1:\theta_1, x_2:\theta_2; \Psi,\phi, \forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow \phi[y_1/x_1][y_2/x_2]\Rightarrow$$
$$\phi'[y_1/x_1][y_2/x_2][(F_1\ y_1)/\mathbf{r}_1][(F_2\ y_2)/\mathbf{r}_2]\vdash_{\mathrm{L^C}}\Rightarrow\phi'[(F_1\ x_1)/\mathbf{r}_1][(F_2\ x_2)/\mathbf{r}_2]$$

Note that $F_i\ x_1\doteq e_i[F_i/f_i][x_i/x_i]\doteq e_i[F_i/f_i]$ (for $i\in\{1,2\}$), hence this further reduces to:

$$\Gamma, x_1:\theta_1, x_2:\theta_2; \Psi,\phi, \forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow \phi[y_1/x_1][y_2/x_2]\Rightarrow$$
$$\phi'[y_1/x_1][y_2/x_2][(F_1\ y_1)/\mathbf{r}_1][(F_2\ y_2)/\mathbf{r}_2]\vdash_{\mathrm{L^C}}\Rightarrow\phi'[e_1[F_1/f_1]/\mathbf{r}_1][e_2[F_2/f_2]/\mathbf{r}_2]$$

Applying the IH to the third premise we get:
$\Gamma, x_1:\theta_1, x_2:\theta_2, f_1:\theta_1\to\tau_1, f_2:\theta_2\to\tau_2;\Psi,\forall y_1y_2.\ (|y_1|,|y_2|)<(|x_1|,|x_2|)\Rightarrow\phi[y_1/x_1][y_2/x_2]\Rightarrow$
$\phi'[y_1/x_1][y_2/x_2][(f_1\ y_1)/\mathbf{r}_1][(f_2\ y_2)/\mathbf{r}_2]\vdash_{\mathrm{L^C}}\phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$
By instantiating it with $f_1:=F_1$ and $f_2:=F_2$, we get the required result.

Case $\dfrac{\begin{array}{c}\theta=K_1(\sigma_{1,1}\times\cdots\times\sigma_{1,a_1})+\cdots+K_n(\sigma_{n,1}\times\cdots\times\sigma_{n,a_n})\in\Theta\\ \Gamma;\Psi\vdash e:\theta\sim e':\theta\mid\phi'\\ \text{For all } 1\le i,j\le n:\\ \Gamma;\Psi\vdash e_i:\sigma_{i,1}\to\cdots\to\sigma_{i,a_i}\to\tau_1\sim e'_j:\sigma_{j,1}\to\cdots\to\sigma_{j,a_j}\to\tau_2\mid\phi'_{i,j}\text{ where}\\ \phi'_{i,j}\equiv\forall x_1:\sigma_{i,1},\ldots,x_{a_i}:\sigma_{i,a_i},y_1:\sigma_{j,1},\ldots,y_{a_j}:\sigma_{j,a_j}.\ \phi'[K_i(x_1,\ldots,x_{a_i})/\mathbf{r}_1][K_j(y_1,\ldots,y_{a_j})/\mathbf{r}_2]\\ \Rightarrow\phi[(\mathbf{r}_1\ x_1\ \cdots\ x_{a_i})/\mathbf{r}_1][(\mathbf{r}_2\ y_1\ \cdots\ y_{a_j})/\mathbf{r}_2]\end{array}}{\Gamma;\Psi\vdash \mathsf{match}\ e\ \mathsf{with}\ K_1\mapsto e_1;\ldots;K_n\mapsto e_n:\tau_1\sim\mathsf{match}\ e'\ \mathsf{with}\ K_1\mapsto e'_1;\ldots;K_n\mapsto e'_n:\tau_2\mid\phi}$ R-MATCH

TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[(\mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n)/\mathbf{r}_1][(\mathsf{match}\ e'\ \mathsf{with}\ K_1 \mapsto e'_1; \ldots; K_n \mapsto e'_n)/\mathbf{r}_2]$.

By the $\mathrm{L^C}$ ELIM rule (on e) for θ we need to show:

$\Gamma, x_1 : \sigma_{1,i}, \ldots, x_{a_i} : \sigma_{i,a_i}; \Psi, e \doteq K_i(x_1, \ldots, x_{a_i}) \vdash_{\mathrm{L^C}} \phi[(\mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n)/\mathbf{r}_1][(\mathsf{match}\ e'\ \mathsf{with}\ K_1 \mapsto e'_1; \ldots; K_n \mapsto e'_n)/\mathbf{r}_2]$

for all $1 \leq i \leq n$. Fix an i.

By the $\mathrm{L^C}$ ELIM rule (on e') for θ we need to show:

$\Gamma, x_1 : \sigma_{1,i}, \ldots, x_{a_i} : \sigma_{i,a_i}, y_1 : \sigma_{j,1}, \ldots, y_{a_j} : \sigma_{j,a_j}; \Psi, e \doteq K_i(x_1, \ldots, x_{a_i}), e' \doteq K_j(y_1, \ldots, y_{a,j}) \vdash_{\mathrm{L^C}} \phi[(\mathsf{match}\ e\ \mathsf{with}\ K_1 \mapsto e_1; \ldots; K_n \mapsto e_n)/\mathbf{r}_1][(\mathsf{match}\ e'\ \mathsf{with}\ K_1 \mapsto e'_1; \ldots; K_n \mapsto e'_n)/\mathbf{r}_2]$

for all $1 \leq j \leq n$. Fix an j.

This reduces to:

$\Gamma, x_1 : \sigma_{1,i}, \ldots, x_{a_i} : \sigma_{i,a_i}, y_1 : \sigma_{j,1}, \ldots, y_{a_j} : \sigma_{j,a_j}; \Psi, e \doteq K_i(x_1, \ldots, x_{a_i}), e' \doteq K_j(y_1, \ldots, y_{a,j}) \vdash_{\mathrm{L^C}} \phi[(e_i\ x_1\ \cdots\ x_{a_i})/\mathbf{r}_1][(e_j\ y_1\ \cdots\ y_{a_j})/\mathbf{r}_2]$

By IH on the first premise we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi'[e/\mathbf{r}_1][e'/\mathbf{r}_2]$, and then obtained with the above (and weakening):

$\Gamma, x_1 : \sigma_{1,i}, \ldots, x_{a_i} : \sigma_{i,a_i}, y_1 : \sigma_{j,1}, \ldots, y_{a_j} : \sigma_{j,a_j}; \Psi, \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}_1][K_j(y_1, \ldots, y_{a,j})/\mathbf{r}_2] \vdash_{\mathrm{L^C}} \phi[(e_i\ x_1\ \cdots\ x_{a_i})/\mathbf{r}_1][(e_j\ y_1\ \cdots\ y_{a_j})/\mathbf{r}_2]$

By IH on the given i, j pair we have (after eliminating $\forall$ and $\Rightarrow$):

$\Gamma, x_1 : \sigma_{i,1}, \ldots, x_{a_i} : \sigma_{i,a_i}, y_1 : \sigma_{j,1}, \ldots, y_{a_j} : \sigma_{j,a_j}; \Psi, \phi'[K_i(x_1, \ldots, x_{a_i})/\mathbf{r}_1][K_j(y_j, \ldots, y_{a_j})/\mathbf{r}_2] \vdash_{\mathrm{L^C}} \phi[(e\ x_1\ \cdots\ x_{a_i})/\mathbf{r}_1][(e'\ y_i\ \cdots\ y_{a_j})/\mathbf{r}_2]$.

This is exactly the goal that we need to show.

Case $\dfrac{\begin{array}{c} K(\sigma_1 \times \cdots \times \sigma_n) \in \Theta(\theta) \qquad \Gamma; \Psi \vdash e_i : \sigma_i \sim e'_i : \sigma_i \mid \phi_i \quad \text{for all } 1 \leq i \leq n \\ \Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x_1, y_1 : \sigma_1, \ldots, x_n, y_n : \sigma_n.\ \phi_1[x_1/\mathbf{r}_1][y_1/\mathbf{r}_2] \Rightarrow \cdots \Rightarrow \phi_n[x_n/\mathbf{r}_1][y_n/\mathbf{r}_2] \\ \Rightarrow \phi[K(x_1, \ldots, x_n)/\mathbf{r}_1][K(y_1, \ldots, y_n)/\mathbf{r}_2] \end{array}}{\Gamma; \Psi \vdash K(e_1, \ldots, e_n) : \theta \sim K(e'_1, \ldots, e'_n) : \theta \mid \phi}\ \text{R-CONS}$

TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[K(e_1, \ldots, e_n)/\mathbf{r}_1][K(e'_1, \ldots, e'_n)/\mathbf{r}_2]$.

By IH on e_i premises we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi_i[e_i/\mathbf{r}_1][e'_i/\mathbf{r}_2]$ (*), for all $1 \leq i \leq n$.

By instantiating the last premise with $x_i = e_i$ and $y_i = e'_i$, for all $1 \leq i \leq n$, and after eliminating all implications with (*), we obtain $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[K(e_1, \ldots, e_n)/\mathbf{r}_1][K(e'_1, \ldots, e'_n)/\mathbf{r}_2]$, as required.

Case $\dfrac{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \{m_1\} : \mathbb{C}(\tau_1) \sim \{m_2\} : \mathbb{C}(\tau_2) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.\phi)}\ \text{R-MONAD}$

TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathbb{C}_{\mathrm{r}}(\{m_1\}, \{m_2\}, n, \mathbf{r}_1.\mathbf{r}_2.\phi) \triangleq \exists e'_1, n_1, e'_2, n_2.\ \{m_1\} \doteq \{\mathsf{cstep}_{n_1}(\mathsf{cret}(e'_1))\} \wedge \{m_2\} \doteq \{\mathsf{cstep}_{n_2}(\mathsf{cret}(e'_2))\} \wedge \phi[e'_1/\mathbf{r}_1][e'_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$.

By IH on the premise we have: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$. We pick $e'_1 := e_1$, $e'_2 := e_2$, $n_1 := n_1$, $n_2 := n_2$, and then the required result follows, since $m_i \doteq \mathsf{cstep}_{n_i}(\mathsf{cret}(e'_i))$ implies $\{m_i\} \doteq \{\mathsf{cstep}_{n_i}(\mathsf{cret}(e'_i))\}$ (for $i \in \{1, 2\}$).

Proof of (2)

Case $\dfrac{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi}{\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim \mathsf{cret}(e_2) \div \tau_2 \mid 0 \mid \phi}\text{R-RET}$

TS: $\Gamma; \Psi \vdash_{\text{LC}} \exists e_1', e_2', n_1, n_2.\ \mathsf{cret}(e_1) \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1')) \wedge \mathsf{cret}(e_2) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2')) \wedge \phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2] \wedge n_1 - n_2 \leq 0$.

We pick $e_i' := e_i$, $n_1, n_2 := 0$. Then $\mathsf{cret}(e_i) \doteq \mathsf{cstep}_0(\mathsf{cret}(e_i))$ by the $\mathsf{cstep}_0 \to_\zeta$ rule, $\phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$ from the IH on the premise, and we have that $0 - 0 \leq 0$.

Case $\dfrac{\Gamma \vdash n_1 : \mathbb{R}^\infty \qquad \Gamma \vdash n_2 : \mathbb{R}^\infty \qquad \Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1) \div \tau_1 \sim \mathsf{cstep}_{n_2}(m_2) \div \tau_2 \mid n + n_1 - n_2 \mid \phi}\text{R-STEP}$

TS: $\Gamma; \Psi \vdash_{\text{LC}} \exists e_1, e_2, n_1', n_2'.\ \mathsf{cstep}_{n_1}(m_1) \doteq \mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1)) \wedge \mathsf{cstep}_{n_2} m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1' - n_2' \leq (n + n_1 - n_2)$.

By IH on the third premise we have: $\Gamma; \Psi \vdash_{\text{LC}} \exists e_1', e_2', n_1'', n_2''.\ m_1 \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1')) \wedge m_2 \doteq \mathsf{cstep}_{n_2''}(\mathsf{cret}(e_2')) \wedge \phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2] \wedge n_1'' - n_2'' \leq n$. We pick $e_i := e_i'$, $n_i' := (n_i'' + n_i)$. Then we have:

(a) $\mathsf{cstep}_{n_i}(m_i) \doteq \mathsf{cstep}_{n_i}(\mathsf{cstep}_{n_i''}(\mathsf{cret}(e_i'))) \doteq \mathsf{cstep}_{n_i + n_i''}(\mathsf{cret}(e_i'))$ (by $\mathsf{cstep} \to_\zeta$ rule);

(b) $\phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2]$ from the IH; and

(c) $n_1' - n_2' \leq n + n_1 - n_2$, since $n_1' = n_1'' + n_1$, $n_2' = n_2'' + n_2$, and $n_1'' - n_2'' \leq n$.

Case $\dfrac{\Gamma; \Psi \vdash e_1 : \tau_1' \sim e_2 : \tau_2' \mid \mathbb{C}_{\text{R}}(\mathbf{r}_1, \mathbf{r}_2, n', x_1.x_2.\phi') \qquad \Gamma, x_1 : \tau_1', x_2 : \tau_2'; \Psi, \phi' \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi \qquad x_1, x_2 \notin n, \phi}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x_1\}.\, m_1) \div \tau_1 \sim \mathsf{cbind}(e_2, \{x_2\}.\, m_2) \div \tau_2 \mid n' + n \mid \phi}\text{R-BIND}$

TS: $\Gamma; \Psi \vdash_{\text{LC}} \exists e_1'', e_2'', n_1, n_2.\ \mathsf{cbind}(e_1, \{x_1\}.\, m_1) \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'')) \wedge \mathsf{cbind}(e_2, \{x_2\}.\, m_2) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2'')) \wedge \phi[e_1''/\mathbf{r}_1][e_2''/\mathbf{r}_2] \wedge n_1 - n_2 \leq (n' + n)$.

From IH on the first premise we have:

$$\Gamma; \Psi \vdash_{\text{LC}} \mathbb{C}_{\text{R}}(e_1, e_2, n', x_1.x_2.\phi')$$

$\triangleq \exists e_1', e_2', n_1', n_2'.\ e_1 \doteq \{\mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1'))\} \wedge e_2 \doteq \{\mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2'))\} \wedge \phi'[e_1'/x_1][e_2'/x_2] \wedge n_1' - n_2' \leq n'$

Using ϕ', from IH on the second premise we have (noting that $x_1, x_2 \notin n, \phi$):

$$\Gamma; \Psi \vdash_{\text{LC}} \exists e_1''', e_2''', n_1'', n_2''.\ m_1[e_1'/x_1] \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1''')) \wedge m_2[e_2'/x_2] \doteq \mathsf{cstep}_{n_2''}(\mathsf{cret}(e_2'''))$$

$$\wedge\ \phi[e_1'''/\mathbf{r}_1][e_2'''/\mathbf{r}_2] \wedge n_1'' - n_2'' \leq n$$

We pick $e_i'' := e_i'''$, and $n_i := n_i' + n_i''$, and then we have:

(a)

$$\begin{array}{rcl}
\mathsf{cbind}(e_i, \{x_i\}.\, m_i) & \doteq & \mathsf{cbind}(\{\mathsf{cstep}_{n_i'}(\mathsf{cret}(e_i'))\}, \{x_i\}.\, m_i) \\
& \to_\zeta & \{\!|\mathsf{cstep}_{n_i'}(\mathsf{cret}(e_i'))/x_i|\!\}m_i \\
& = & \mathsf{cstep}_{n_i'}(m_i[e_i'/x_i]) \\
& \doteq & \mathsf{cstep}_{n_i'}(\mathsf{cstep}_{n_i''}(\mathsf{cret}(e_i'''))) \\
& \to_\zeta & \mathsf{cstep}_{n_i' + n_i''}(\mathsf{cret}(e_i''')) \\
& = & \mathsf{cstep}_{n_i}(\mathsf{cret}(e_i'')).
\end{array}$$

(b) We already established $\phi[e_1'''/\mathbf{r}_1][e_2'''/\mathbf{r}_2] \triangleq \phi[e_1''/\mathbf{r}_1][e_2''/\mathbf{r}_2]$.

(c) We have $n_1 - n_2 \leq (n' + n)$, since $n_1 = n_1' + n_1''$, $n_2 = n_2' + n_2''$, $n_1' - n_2' \leq n'$, and $n_1'' - n_2'' \leq n$.

Case $\dfrac{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} n' \leq n}{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}$ R-SUBC

TS: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$.

From IH on the first premise we have: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n'$ From the second premise we have $n_1 - n_2 \leq n' \leq n$, hence the result holds.

Case $\dfrac{\Gamma \vdash e_1 \div \tau_1 \qquad \Gamma; \Psi \vdash m_2 \div \tau_2 \mid k \mid \ell \mid \phi[e_1/\mathbf{r}_1][\mathbf{r}/\mathbf{r}_2]}{\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim m_2 \div \tau_2 \mid -k \mid \phi}$ R-RET-L

TS: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1', e_2', n_1, n_2.\ \mathsf{cret}(e_1) \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'))$
$\wedge\ m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2')) \wedge \phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2] \wedge n_1 - n_2 \leq -k$.

By Theorem 7(2) applied to the second premise,

$$\exists e_2', n_2.\ m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2')) \wedge \phi[e_1/\mathbf{r}_1][e_2'/\mathbf{r}_2] \wedge k \leq n_2 \leq \ell$$

(and by noting that $(\phi[e_1/\mathbf{r}_1][\mathbf{r}/\mathbf{r}_2])[e_2'/\mathbf{r}] = \phi[e_1/\mathbf{r}_1][e_2'/\mathbf{r}_2]$).

We pick $e_1' := e_1$, $e_2' := e_2'$, $n_1 := 0$, $n_2 := n_2$. Then we have:

(a) $\mathsf{cret}(e_1) \doteq \mathsf{cstep}_0(\mathsf{cret}(e_1))$
(b) $m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2'))$
(c) $\phi[e_1/\mathbf{r}_1][e_2'/\mathbf{r}_2]$
(d) From $k \leq n_2$, we have $0 - n_2 \leq -k$.

Case $\dfrac{\Gamma \vdash n_1 : \mathbb{R}^{\infty} \qquad \Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \psi}{\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1) \div \tau_1 \sim m_2 \div \tau_2 \mid n + n_1 \mid \psi}$ R-STEP-L

TS: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1', n_2'.\ \mathsf{cstep}_{n_1}(m_1) \doteq \mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1' - n_2' \leq (n + n_1)$.

By IH on the second premise we have: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1'', n_2'.\ m_1 \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1'' - n_2' \leq n$

We pick $e_i := e_i$, $n_2' := n_2'$, and $n_1' := n_1'' + n_1$. Since $m_1 \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1))$, we have $\mathsf{cstep}_{n_1}(m_1) \doteq \mathsf{cstep}_{n_1}(\mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1))) \to_{\zeta} \mathsf{cstep}_{n_1+n_1''}(\mathsf{cret}(e_1))$.

Hence, it remains to show that $n_1 + n_1'' - n_2' \leq (n + n_1)$, which follows from $n_1'' - n_2' \leq n$.

Case $\dfrac{\Gamma; \Psi \vdash e_1' : \mathbb{C}(\tau_1') \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, k, \ell, x.\phi') \qquad \Gamma, x : \tau_1'; \Psi, \phi' \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}{\Gamma; \Psi \vdash \mathsf{cbind}(e_1', \{x\}.\, m_1) \div \tau_1 \sim m_2 \div \tau_2 \mid \ell + n \mid \phi}$ R-BIND-L

TS: $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1', n_2'.\ \mathsf{cbind}(e_1', \{x\}.m_1) \doteq \mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1' - n_2' \leq (\ell + n)$.

By Theorem 7(1) applied to the first premise we have $\Gamma; \Psi \vdash_{\mathrm{LC}} \mathbb{C}_{\mathbf{u}}(e_1', k, \ell, x.\phi') \triangleq \exists e_1'', n_1''.\ e_1' \doteq \{\mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1''))\} \wedge \phi'[e_1''/x] \wedge k \leq n_1'' \leq \ell$.

Using ϕ', by IH on the second premise we have: $\Gamma, x : \tau_1'; \Psi, \phi' \vdash_{\mathrm{LC}} \exists e_1''', e_2, n_1''', n_2'.\ m_1 \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1''')) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1'''/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1''' - n_2' \leq n$. Hence,

$\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1''', e_2, n_1''', n_2'.\ m_1[e_1''/x] \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1''')) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2)) \wedge \phi[e_1'''/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1''' - n_2' \leq n$ (since, $x \notin n, \phi$).

We pick $e_1 := e_1'''$, $e_2 := e_2$, $n_1' := n_1'' + n_1'''$, and $n_2 := n_2$, and then we have:

(a)

$$\begin{array}{rcl} \mathsf{cbind}(e_1', \{x\}.\, m_1) & \doteq & \mathsf{cbind}(\{\mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1''))\}, \{x\}.\, m_1) \\ & \to_\zeta & \{\!|\mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1''))/x|\!\}m_1 \\ & = & \mathsf{cstep}_{n_1''}(m_1[e_1''/x]) \\ & \doteq & \mathsf{cstep}_{n_1''}(\mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))) \\ & \doteq & \mathsf{cstep}_{n_1''+n_1'''}(\mathsf{cret}(e_1''')) \end{array}$$

(b) $m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2))$

(c) $\phi[e_1'''/\mathbf{r}_1][e_2/\mathbf{r}_2]$

(d) From, $n_1'' \leq \ell$ and $n_1''' - n_2' \leq n$, we have $n_1'' + n_1''' - n_2' \leq \ell + n$, and from $n_1' = n_1'' + n_1'''$, we then have $n_1' - n_2' \leq \ell + n$.

□

Lemma 16 (Trivial refinements)**.** The following hold:

1. If $\Gamma \vdash e_1 : \tau_1$ and $\Gamma \vdash e_2 : \tau_2$, then $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \top$.
2. If $\Gamma \vdash m_1 \div \tau_1$ and $\Gamma \vdash m_2 \div \tau_2$, then $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$.

Proof. **Proof of (1)** Assume $\Gamma \vdash e_1 : \tau_1$ and $\Gamma \vdash e_2 : \tau_2$. By Lemma 8(1) on the first premise, we have $\Gamma; \Psi \vdash e_1 : \tau_1 \mid \top$. Using the rule UHOL-L of the original RHOL paper, we now derive $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \top$, as required.

Proof of (2) By induction on the first typing derivation ($\Gamma \vdash m_1 \div \tau_1$).

Case $\dfrac{\Gamma \vdash e_1 : \tau_1}{\Gamma \vdash \mathsf{cret}(e_1) \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$.

By Lemma 8(2) applied to the second assumption we have:

$$\Gamma; \Psi \vdash m_2 \div \tau_2 \mid -\infty \mid \infty \mid \top$$

Then, by rule R-RET-L, with the first premise we have:

$$\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim m_2 \div \tau_2 \mid -(-\infty) \mid \top$$

and the required result follows by rule R-SUBC.

Case $\dfrac{\Gamma \vdash n_1 : \mathbb{R}^\infty \qquad \Gamma \vdash m_1' \div \tau_1}{\Gamma \vdash \mathsf{cstep}_{n_1}(m_1') \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$.

By IH on the second premise we have:

$$\Gamma; \Psi \vdash m_1' \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$$

and then by rule R-STEP-L:

$$\Gamma; \Psi \vdash \mathsf{cstep}_{n_1}(m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid \infty + n_1 \mid \top$$

The required result follows by rule R-SUBC (since $\Gamma; \Psi \vdash_{\mathrm{LC}} \infty + n_1 \leq \infty$.

Case $\dfrac{\Gamma \vdash e_1 : \mathbb{C}(\tau_1') \qquad \Gamma, x : \tau_1' \vdash m_1' \div \tau_1}{\Gamma \vdash \mathsf{cbind}(e_1, \{x\}.\, m_1') \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.\, m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$.

By Lemma 8(2) on the first premise we have:

$$\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1') \mid \top$$

and by rule U-SUB:

$$\Gamma; \Psi \vdash e_1 : \mathbb{C}(\tau_1') \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, -\infty, \infty, x.\top)$$

From the second assumption we also have (by weakening):

$$\Gamma, x : \tau_1' \vdash m_2 \div \tau_2$$

Hence, by IH on the second premise we have:

$$\Gamma, x : \tau_1'; \Psi \vdash m_1' \div \tau_1 \sim m_2 \div \tau_2 \mid \infty \mid \top$$

Then by rule R-BIND-L we have:

$$\Gamma; \Psi \vdash \mathsf{cbind}(e_1, \{x\}.\, m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid \infty + (-\infty) \mid \top$$

and the result follows by rule R-SUBC (since $\Gamma; \Psi \vdash_{\mathrm{LC}} \infty + (-\infty) \leq \infty$).

□

Theorem 17 ($\mathrm{L}^{\mathrm{C}}{\Rightarrow}\mathrm{R}^{\mathrm{C}}$, pure)**.** If $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$ and $\Gamma \vdash e_1 : \tau_1$ and $\Gamma \vdash e_2 : \tau_2$, then $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$.

Proof. By Lemma 16(1) and the subsumption R-SUB rule. □

Theorem 18 ($\mathrm{L}^{\mathrm{C}}{\Rightarrow}\mathrm{R}^{\mathrm{C}}$, monadic)**.** If $\Gamma; \Psi \vdash_{\mathrm{LC}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$ and $\Gamma \vdash m_1 \div \tau_1$ and $\Gamma \vdash m_2 \div \tau_2$, then $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$.

Proof. By induction of the first typing derivation ($\Gamma \vdash m_1 \div \tau_1$).

Case $\dfrac{\Gamma \vdash e_1 : \tau_1}{\Gamma \vdash \mathsf{cret}(e_1') \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cret}(e_1') \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$.

By assumption $m = \mathsf{cret}(e_1') \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1))$, hence by $\rightarrow_\zeta$ and an L^{C} axiom, $n_1 \doteq 0$ and $e_1' \doteq e_1$. Further, from the assumption we then have:

$$\exists e_2, n_2.\ m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge (\phi[e_1/\mathbf{r}_1][\mathbf{r}/\mathbf{r}_2])[e_2/\mathbf{r}] \wedge -n \leq n_2 \leq \infty$$

Then, by Theorem 10 (with the second typing assumption), $\Gamma; \Psi \vdash m_2 \div \tau_2 \mid -n \mid \infty \mid \phi[e_1/\mathbf{r}_1][\mathbf{r}/\mathbf{r}_2]$.

Then, by rule R-RET-L (using the above and the first premise), $\Gamma; \Psi \vdash \mathsf{cret}(e_1) \div \tau_1 \sim m_2 \div \tau_2 \mid -(-n) \mid \phi$, and finally the required result follows by rule R-SUBC, since $-(-n) \leq n$.

Case $\dfrac{\Gamma \vdash n_1' : \mathbb{R}^\infty \qquad \Gamma \vdash m_1' \div \tau_1}{\Gamma \vdash \mathsf{cstep}_{n_1'}(m_1') \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1'}(m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$.

From assumption $m = \mathsf{cstep}_{n_1'}(m_1') \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1))$. By an $\mathrm{L^C}$ axiom, there is a n_1'' such that $n_1 \doteq n_1' + n_1''$ and $m_1' \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1))$. From the latter, and the remaining assumptions it follows that

$$m_1' \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1'' - n_2 \leq n - n_1'$$

Hence, by IH on the second premise, $\Gamma; \Psi \vdash m_1' \div \tau_1 \sim m_2 \div \tau_2 \mid n - n_1' \mid \phi$. By rule R-STEP-L, it follows that $\Gamma; \Psi \vdash \mathsf{cstep}_{n_1'}(m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid n - n_1' + n_1' \mid \phi$. The required result follows from R-SUBC rule, since $n - n_1' + n_1' \leq n$.

Case $\dfrac{\Gamma \vdash e_1' : \mathbb{C}(\tau_1') \qquad \Gamma, x : \tau_1' \vdash m_1' \div \tau_1}{\Gamma \vdash \mathsf{cbind}(e_1', \{x\}.\, m_1') \div \tau_1}$

TS: $\Gamma; \Psi \vdash \mathsf{cbind}(e_1', \{x\}.\, m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$.

From the first premise and $\mathrm{L^C}$'s axioms, it follows that there exists e_1'', n_1'' such that $\Gamma; \bullet \vdash_{\mathrm{L^C}} e_1' \doteq \mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1''))$, hence, by definition, $\Gamma; \bullet \vdash_{\mathrm{L^C}} \mathbb{C}_{\mathrm{U}}(e_1', n_1'', n_1'', x.x \doteq e_1'')$. Applying Theorem 9 on the first premise, we then get:

$$\Gamma; \Psi \vdash e_1' : \mathbb{C}(\tau_1') \mid \mathbb{C}_{\mathrm{U}}(\mathbf{r}, n_1'', n_1'', x.x \doteq e_1'')$$

Similarly, from the second premise and $\mathrm{L^C}$'s axioms it follows that $\Gamma, x : \tau_1'; \bullet \vdash_{\mathrm{L^C}} \exists e_1''', n_1'''.\, m_1' \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))$, and consequently $\Gamma; \bullet \vdash_{\mathrm{L^C}} \exists e_1''', n_1'''.\, m_1'[e_1''/x] \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))$. Therefore, there exists e_1''', n_1''' such that $\Gamma; \bullet \vdash_{\mathrm{L^C}} m_1'[e_1''/x] \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))$. It follows that $\Gamma, x : \tau_1'; \Psi, x \doteq e_1'' \vdash_{\mathrm{L^C}} m_1' \doteq \mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))$. Call this statement (A).

From assumption $\Gamma; \Psi \vdash_{\mathrm{L^C}} m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$, by weakening, we have: $\Gamma, x : \tau_1'; \Psi, x \doteq e_1'' \vdash_{\mathrm{L^C}} m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$. Call this statement (B).

Further, $\Gamma; \bullet \vdash_{\mathrm{L^C}} \mathsf{cbind}(e_1', \{x\}.\, m_1') \doteq \mathsf{cbind}(\mathsf{cstep}_{n_1''}(\mathsf{cret}(e_1'')), \{x\}.\, m_1') \doteq \mathsf{cstep}_{n_1''}(m_1'[e_1''/x]) \doteq \mathsf{cstep}_{n_1''}(\mathsf{cstep}_{n_1'''}(\mathsf{cret}(e_1'''))) \doteq \mathsf{cstep}_{n_1''+n_1'''}(\mathsf{cret}(e_1'''))$. However, from assumption $\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathsf{cbind}(e_1', \{x\}.\, m_1') = m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1))$. Hence, we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \doteq \mathsf{cstep}_{n_1''+n_1'''}(\mathsf{cret}(e_1'''))$. By an $\mathrm{L^C}$ axiom: $\Gamma; \Psi \vdash_{\mathrm{L^C}} n_1 \doteq n_1'' + n_1''' \wedge e_1 \doteq e_1'''$.

Next, $\Gamma; \Psi \vdash_{\mathrm{L^C}} e \doteq e_1'''$ and the assumption $\Gamma; \Psi \vdash_{\mathrm{L^C}} \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$ imply (by weakening) that $\Gamma, x : \tau_1'; \Psi, x \doteq e_1'' \vdash_{\mathrm{L^C}} \phi[e_1'''/\mathbf{r}_1][e_2/\mathbf{r}_2]$. Call this statement (C).

Also, $\Gamma; \Psi \vdash_{\mathrm{L^C}} n \doteq n_1'' + n_1'''$ and the assumption $\Gamma; \Psi \vdash_{\mathrm{L^C}} n_1 - n_2 \leq n$ imply (by weakening), $\Gamma, x : \tau_1'; \Psi, x \doteq e_1'' \vdash_{\mathrm{L^C}} n_1''' - n_2 \leq n - n_1''$. Call this statement (D).

Applying the IH to the second premise, using statements (A)-(D), we get

$$\Gamma, x : \tau_1'; \Psi, x \doteq e_1'' \vdash m_1' \div \tau_1 \sim m_2 \div \tau_2 \mid n - n_1' \mid \phi$$

Using the rule R-BIND-L, we derive $\Gamma; \Psi \vdash \mathsf{cbind}(e_1', \{x\}.\, m_1') \div \tau_1 \sim m_2 \div \tau_2 \mid n - n_1'' + n_1'' \mid \phi$, and the result follows by rule R-SUBC, since $n - n_1'' + n_1'' \leq n$.

□

Corollary 19 (Pure equivalence closure ($\mathrm{R^C}$))**.** If $\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$ and $\Gamma; \Psi \vdash_{\mathrm{L^C}} e_1 \doteq e_1'$ and $\Gamma; \Psi \vdash_{\mathrm{L^C}} e_2 \doteq e_2'$ and $\Gamma \vdash e_1' : \tau_1$ and $\Gamma \vdash e_2' : \tau_2$, then $\Gamma; \Psi \vdash e_1' : \tau_1 \sim e_2' : \tau_2 \mid \phi$.

Proof. From Theorem 15(1), $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$. Hence, $\Gamma; \Psi \vdash_{\mathrm{LC}} \phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2]$. The required result follows from Theorem 18. □

Corollary 20 (Monadic equivalence closure ($\mathrm{R^C}$))**.** If $\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$ and $\Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq m_1'$ and $\Gamma; \Psi \vdash_{\mathrm{LC}} m_2 \doteq m_2'$ and $\Gamma \vdash m_1' \div \tau_1$ and $\Gamma \vdash m_2' \div \tau_2$, then $\Gamma; \Psi \vdash m_1' \div \tau_1 \sim m_2' \div \tau_2 \mid n \mid \phi$.

Proof. From the first assumption, by Theorem 15(2), we have

$$\exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$$

From the second and third assumptions we then have:

$$\exists e_1, e_2, n_1, n_2.\ m_1' \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2' \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$$

The conclusion follows from Theorem 18. □

Theorem 21 (Subsumption, monadic)**.** The following rules are admissible:

$$\frac{\begin{array}{c}\Psi; \Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi' \\ \Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \\ \Gamma; \Psi \vdash_{\mathrm{LC}} n_1 - n_2 \leq n \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \Rightarrow \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]\end{array}}{\Psi; \Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}\ \text{R-SUBM1}$$

$$\frac{\Psi; \Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n' \mid \phi' \qquad \Psi; \Gamma \vdash_{\mathrm{LC}} n' \leq n \qquad \Psi; \Gamma \vdash_{\mathrm{LC}} \forall \mathbf{r}_1, \mathbf{r}_2.\ \phi' \Rightarrow \phi}{\Psi; \Gamma \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi}\ \text{R-SUBM2}$$

Proof. **Proof of (R-SUBM1)** From Theorem 15(2) applied to the first premise we have:

$$\Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n'$$

From the last four premises we have:

$$\Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$$

The result follows by Theorem 18 (note that the typing follows from the first premise).

Proof of (R-SUBM2) From Theorem 15(2) applied to the first premise we have:

$$\Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi'[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n'$$

Then from the last two premises we have:

$$\Gamma; \Psi \vdash_{\mathrm{LC}} m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$$

The result follows by Theorem 18.

□

Theorem 22 (SPLIT, monadic)**.** The following rule is admissible:

$$\frac{\Gamma; \Psi \vdash m_1 \div \tau_1 \mid k_1 \mid \ell_1 \mid \phi_1 \qquad \Gamma; \Psi \vdash m_2 \div \tau_2 \mid k_2 \mid \ell_2 \mid \phi_2 \qquad \Gamma; \Psi \vdash_{\mathrm{LC}} \ell_1 - k_2 \leq n}{\Gamma; \Psi \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}]}\ \text{R-SPLIT}$$

Proof. From Theorem 7(2) applied to the first two premises we have (for $i \in \{1,2\}$):

$$\Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \exists e_i, n_i.\ m_i \doteq \mathsf{cstep}_{n_i}(\mathsf{cret}(e_i)) \wedge \phi_i[e_i/\mathbf{r}] \wedge k_i \leq n_i \leq \ell_i$$

(for $i \in \{1,2\}$). Hence, we have (from the third premise):

$$\Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \exists e_1, e_2, n_1, n_2.\ m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$$

$$\wedge\ (\phi_1[\mathbf{r}_1/\mathbf{r}])[e_1/\mathbf{r}_1] \wedge (\phi_2[\mathbf{r}_2/\mathbf{r}])[e_2/\mathbf{r}_2] \wedge n_1 - n_2 \leq n$$

The result follows from Theorem 18. □

Lemma 23 (R^{C}-U^{C})**.** The following rule is admissible:

$$\frac{\Gamma; \Psi \vdash e_1 : \tau_1 \mid \phi_1 \qquad \Gamma; \Psi \vdash e_2 : \tau_2 \mid \phi_2}{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}]}\ \mathrm{R}^{\mathrm{C}}\text{-}\mathrm{U}^{\mathrm{C}}$$

Proof. By Theorem 7(1) applied to the premises (with the L^{C} rule for $\wedge$) we have:

$$\Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} (\phi_1[\mathbf{r}_1/\mathbf{r}])[e_1/\mathbf{r}_1] \wedge (\phi_2[\mathbf{r}_2/\mathbf{r}])[e_2/\mathbf{r}_2]$$

The result follows by Theorem 17. □

Lemma 24 (SPLIT, pure)**.** The following rule is admissible:

$$\frac{\Gamma; \Psi \vdash e_1 : \tau_1 \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k', \ell, \mathbf{r}.\phi_1) \qquad \Gamma; \Psi \vdash e_2 : \tau_2 \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell', \mathbf{r}.\phi_2) \qquad \Gamma; \Psi \vdash_{\mathrm{L}^{\mathrm{C}}} \ell - k \leq n}{\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.\phi_1[\mathbf{r}_1/\mathbf{r}] \wedge \phi_2[\mathbf{r}_2/\mathbf{r}])}\ \text{R-SPLIT-PURE}$$

Proof. By rule R^{C}-U^{C} we have:

$$\Gamma; \Psi \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}_1, k', \ell, \mathbf{r}_1.\phi_1[\mathbf{r}_1/\mathbf{r}]) \wedge \mathbb{C}_{\mathrm{u}}(\mathbf{r}_2, k, \ell', \mathbf{r}_2.\phi_2[\mathbf{r}_2/\mathbf{r}])$$

$$\triangleq \exists e_1', e_2', n_1, n_2.\ e_1 \doteq \{\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1'))\} \wedge e_2 \doteq \{\mathsf{cstep}_{n_2}(\mathsf{cret}(e_2'))\}$$

$$\wedge\ (\phi_1[\mathbf{r}_1/\mathbf{r}])[e_1'/\mathbf{r}_1] \wedge (\phi_2[\mathbf{r}_2/\mathbf{r}])[e_2'/\mathbf{r}_2] \wedge k' \leq n_1 \leq \ell \wedge k \leq n_2 \leq \ell'$$

The result follows by rule R-SUB, using the last premise, and folding the definition the monadic proposition. □

Theorem 25 (Forcing subject reduction (R^{C}))**.** If $\bullet; \bullet \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$ and $m_1 \Downarrow^{n_1} e_1$ and $m_2 \Downarrow^{n_2} e_2$, then $n_1 - n_2 \leq n$ and $\bullet; \bullet \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$.

Proof. Suppose $\bullet; \bullet \vdash m_1 \div \tau_1 \sim m_2 \div \tau_2 \mid n \mid \phi$, and $m_1 \Downarrow^{n_1} e_1$ and $m_2 \Downarrow^{n_2} e_2$. By Theorem 15(2), there exists e_1', e_2', n_1', n_2', such that $m_1 \doteq \mathsf{cstep}_{n_1'}(\mathsf{cret}(e_1')) \wedge m_2 \doteq \mathsf{cstep}_{n_2'}(\mathsf{cret}(e_2')) \wedge \phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2] \wedge n_1' - n_2' \leq n$. By Lemma 1 it follows that $m_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1))$ and $m_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$, hence $n_1 \doteq n_1'$, $n_2 \doteq n_2'$, $e_1 \doteq e_1'$, and $e_2 \doteq e_2'$. Therefore, from $n_1' - n_2' \leq n$, we have that $n_1 - n_2 \leq n$. Also, from $\phi[e_1'/\mathbf{r}_1][e_2'/\mathbf{r}_2]$ we have $\phi[e_1/\mathbf{r}_1][e_2/\mathbf{r}_2]$, and then, by Theorem 17, we have $\bullet; \bullet \vdash e_1 : \tau_1 \sim e_2 : \tau_2 \mid \phi$. □

4 Embedding of RelCost

In this section we give some RelCost rules and translation details that were omitted from the paper, as well as proofs of the stated theorems.

4.1 Types and rules

$$
\begin{array}{lcll}
A & ::= & \text{int} \mid \text{list}[n]\ A \mid A_1 \xrightarrow{\text{exec}(k,\ell)} A_2 \mid \forall i \overset{\text{exec}(k,\ell)}{::} S.\ A & \text{Unary types} \\
\tau & ::= & \text{int}_r \mid \text{list}[n]^\alpha\ \tau \mid \tau_1 \xrightarrow{\text{diff}(n)} \tau_2 \mid \forall i \overset{\text{diff}(n)}{::} S.\ \tau \mid UA \mid \Box\tau & \text{Relational types}
\end{array}
$$

$$\boxed{\Delta; \Phi; \Omega \vdash_k^\ell e : A \qquad \text{Unary typing judgment}}$$

$$\frac{\Omega(x) = A}{\Delta; \Phi; \Omega \vdash_0^0 x : A} \qquad \frac{\Delta; \Phi; \Omega, x : A_1 \vdash_k^\ell e : A_2}{\Delta; \Phi; \Omega \vdash_0^0 \lambda x.\ e : A_1 \xrightarrow{\text{exec}(k,\ell)} A_2}$$

$$\frac{\Delta; \Phi; \Omega \vdash_{k_1}^{\ell_1} e_1 : A_1 \xrightarrow{\text{exec}(k,\ell)} A_2 \qquad \Delta; \Phi; \Omega \vdash_{k_2}^{\ell_2} e_2 : A_1}{\Delta; \Phi; \Omega \vdash_{k_1+k_2+k+c_{app}}^{\ell_1+\ell_2+\ell+c_{app}} e_1\ e_2 : A_2}$$

$$\boxed{\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim n : \tau \qquad \text{Relational typing judgment}}$$

$$\frac{\Gamma(x) = \tau}{\Delta; \Phi; \Gamma \vdash x \ominus x \lesssim 0 : \tau} \qquad \frac{\Delta; \Phi; x : \tau_1, \Gamma \vdash e_1 \ominus e_2 \lesssim n : \tau_2}{\Delta; \Phi; \Gamma \vdash \lambda x.\ e_1 \ominus \lambda x.\ e_2 \lesssim 0 : \tau_1 \xrightarrow{\text{diff}(n)} \tau_2}$$

$$\frac{\Delta; \Phi; \overline{\Gamma} \vdash_{\ell_1}^{k_1} e_1 : A \qquad \Delta; \Phi; \overline{\Gamma} \vdash_{\ell_2}^{k_2} e_2 : A}{\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim \ell_1 - k_2 : UA}$$

$$\frac{\Delta; \Phi; \Gamma \vdash e \ominus e \lesssim n : \tau \qquad \forall x \in dom(\Gamma).\ \Delta; \Phi \vDash \Gamma(x) \sqsubseteq \Box\Gamma(x)}{\Delta; \Phi; \Gamma, \Gamma' \vdash e \ominus e \lesssim 0 : \Box\tau}$$

$$\frac{i, \Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim n : \tau \qquad i \notin FIV(\Phi, \Gamma)}{\Delta; \Phi; \Gamma \vdash \Lambda e_1 \ominus \Lambda e_2 \lesssim 0 : \forall i \overset{\text{diff}(n)}{::} S.\ \tau}$$

$$\frac{\begin{array}{c}\Delta; \Phi; \Gamma \vdash e \ominus e' \lesssim m : \text{list}_{\tau'}[n]^\alpha \qquad \Delta; \Phi \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \lesssim m' : \tau \\ i, \Delta; \Phi \wedge n = i + 1; \Gamma, h : \Box\tau', t : \text{list}_{\tau'}[i]^\alpha \vdash e_2 \ominus e_2' \lesssim m' : \tau \\ i, \beta, \Delta; \Phi \wedge n = i + 1 \wedge \alpha = \beta + 1; \Gamma, h : \Box\tau', t : \text{list}_{\tau'}[i]^\beta \vdash e_2 \ominus e_2' \lesssim m' : \tau\end{array}}{\Delta; \Phi; \Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ \mathsf{nil} \to e_1 \mid h :: t \to e_2 \ominus \mathsf{case}\ e'\ \mathsf{of}\ \mathsf{nil} \to e_1 \mid h :: t \to e_2 \lesssim m + m' : \tau}$$

4.2 Translation rules

Erasing translation from RelCost types to simple types

$$|\text{int}| \triangleq \|\text{int}_r\| \triangleq \mathbb{Z} \qquad |\text{list}[n]\ A| \triangleq \text{list}_{|A|} \qquad \|\text{list}[n]^{\alpha}\ \tau\| \triangleq \text{list}_{\|\tau\|}$$

$$|A_1 \xrightarrow{\text{exec}(k,\ell)} A_2| \triangleq |A_1| \to \mathbb{C}(|A_2|) \qquad \|\tau_1 \xrightarrow{\text{diff}(n)} \tau_2\| \triangleq \|\tau_1\| \to \mathbb{C}(\|\tau_2\|)$$

$$|\forall i \overset{\text{exec}(k,\ell)}{::} S.\ A| \triangleq \text{unit} \to \mathbb{C}(|A|) \qquad \|\forall i \overset{\text{diff}(n)}{::} S.\ \tau\| \triangleq \text{unit} \to \mathbb{C}(\|\tau\|) \qquad \|UA\| \triangleq |A|$$

$$\|\Box\tau\| \triangleq \|\tau\|$$

$$|x_1 : A_1, \ldots, x_n : A_n| \triangleq x_1 : |A_1|, \ldots, x_n : |A_n|$$

$$\|x_1 : \tau_1, \ldots, x_n : \tau_n\| \triangleq x_1^1 : \|\tau_1\|, x_1^2 : \|\tau_1\| \ldots, x_n^2 : \|\tau_n\|, x_n^2 : \|\tau_n\|$$

Expression translation

$$(\!|x|\!) \triangleq \{\mathsf{cret}(x)\} \qquad (\!|\mathsf{nil}|\!) \triangleq \{\mathsf{cret}(\mathsf{nil})\}$$

$$(\!|\mathsf{cons}(e_1, e_2)|\!) \triangleq \{\mathsf{cbind}((\!|e_1|\!), \{h\}.\, \mathsf{cbind}((\!|e_2|\!), \{t\}.\, \mathsf{cret}(\mathsf{cons}(h,t))))\}$$

$$(\!|\mathsf{case}\ e\ \mathsf{of}\ \mathsf{nil} \to e_1 \mid h :: t \to e_2|\!) \triangleq \{\mathsf{cbind}((\!|e|\!), \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto (\!|e_1|\!); \mathsf{cons} \mapsto$$

$$\lambda h, t.\ (\!|e_2|\!), \{y\}.\, \mathsf{cstep}_{c_{case}}(\mathsf{cret}(y))))\} \qquad (\!|\lambda x.\ e|\!) \triangleq \{\mathsf{cret}(\lambda x.\ (\!|e|\!))\}$$

$$(\!|e_1\ e_2|\!) \triangleq \{\mathsf{cbind}((\!|e_1|\!), \{x\}.\, \mathsf{cbind}((\!|e_2|\!), \{y\}.\, \mathsf{cbind}(x\ y, \{z\}.\, \mathsf{cstep}_{c_{app}}(\mathsf{cret}(z)))))\}$$

$$(\!|\Lambda e|\!) \triangleq \{\mathsf{cret}(\lambda\_.\ (\!|e|\!))\} \qquad (\!|e[]|\!) \triangleq \{\mathsf{cbind}((\!|e|\!), \{x\}.\, \mathsf{cbind}(x\ \star, \{z\}.\, \mathsf{cret}(z)))\}$$

Translation of unary refinements

$$\forall \ell.\ \text{listU}_A(\ell, 0) \Leftrightarrow \ell \doteq \mathsf{nil}$$

$$\forall \ell, n.\ \text{listU}_A(\ell, n+1) \Leftrightarrow \exists h, t.\ \ell \doteq \mathsf{cons}(h,t) \wedge \lfloor A \rfloor(h) \wedge \text{listU}_A(t, n)$$

$$\lfloor \text{int} \rfloor(x) \triangleq \top \qquad \lfloor \text{list}[n]\ A \rfloor(x) \triangleq \text{listU}_A(x, n)$$

$$\lfloor A_1 \xrightarrow{\text{exec}(k,\ell)} A_2 \rfloor(x) \triangleq \forall y.\ \lfloor A_1 \rfloor(y) \Rightarrow \mathbb{C}_{\text{u}}(x\ y, k, \ell, \mathbf{r}.\lfloor A_2 \rfloor(\mathbf{r}))$$

$$\lfloor \forall i \overset{\text{exec}(k,\ell)}{::} S.\ A \rfloor(x) \triangleq \forall y.\ \top \Rightarrow \forall i.\ \mathbb{C}_{\text{u}}(x\ y, k, \ell, \mathbf{r}.\lfloor A \rfloor(\mathbf{r}))$$

$$\lfloor x_1 : A_1, \ldots, x_n : A_n \rfloor \triangleq \lfloor A_1 \rfloor(x_1), \ldots, \lfloor A_n \rfloor(x_n)$$

Translation of relational refinements

$$\forall \ell_1, \ell_2, a.\ \mathrm{listR}_\tau(\ell_1, \ell_2, 0, a) \Leftrightarrow \ell_1 \doteq \ell_2 \doteq \mathsf{nil}$$

$$\begin{aligned}\forall \ell_1, \ell_2, n, a.\ \mathrm{listR}_\tau(\ell_1, \ell_2, n+1, a) \Leftrightarrow \exists h_1, h_2, t_1, t_2.\ \ell_1 \doteq \mathsf{cons}(h_1, t_1) \wedge \ell_2 \doteq \mathsf{cons}(h_2, t_2)\\ \wedge \llbracket\tau\rrbracket(h_1, h_2) \wedge ((h_1 \doteq h_2 \wedge \mathrm{listR}_\tau(t_1, t_2, n, a))\\ \vee (a > 0 \wedge \exists b.\ a = b + 1 \wedge \mathrm{listR}_\tau(t_1, t_2, n, b)))\end{aligned}$$

$$\llbracket\mathrm{int}_r\rrbracket(x, y) \triangleq x \doteq y \qquad\qquad \llbracket\mathrm{list}[n]^\alpha\ \tau\rrbracket(x, y) \triangleq \mathrm{listR}_\tau(x, y, n, \alpha)$$

$$\begin{aligned}\llbracket\tau_1 \xrightarrow{\mathrm{diff}(n)} \tau_2\rrbracket(x, y) \triangleq \lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(x) \wedge \lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(y) \wedge\\ (\forall x_1, x_2.\ \llbracket\tau_1\rrbracket(x_1, x_2) \Rightarrow \mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, n, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau_2\rrbracket(\mathbf{r}_1, \mathbf{r}_2)))\end{aligned}$$

$$\begin{aligned}\llbracket\forall i \overset{\mathrm{diff}(n)}{::} S.\ \tau\rrbracket(x, y) \triangleq \lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau}\rfloor(x) \wedge \lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau}\rfloor(y) \wedge\\ (\forall x_1, x_2.\ \top \Rightarrow \forall i.\ \mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, n, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2)))\end{aligned}$$

$$\llbracket UA\rrbracket(x, y) \triangleq \lfloor A\rfloor(x) \wedge \lfloor A\rfloor(y) \qquad\qquad \llbracket\Box\tau\rrbracket(x, y) \triangleq x \doteq y \wedge \llbracket\tau\rrbracket(x, y)$$

$$\llbracket x_1 : \tau_1, \ldots, x_n : \tau_n\rrbracket \triangleq \llbracket\tau_1\rrbracket(x_1^1, x_1^2), \ldots, \llbracket\tau_n\rrbracket(x_n^1, x_n^2)$$

4.3 Proofs

We translate RelCost's constraints in the obvious way: Equality $=$ maps to L^{C}'s equality $\doteq$, and comparisons like $\leq$ map to themselves.

Theorem 26 (RelCost, unary)**.** If $\Delta; \Phi; \Omega \vdash_k^\ell e : A$ in RelCost, then $|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash (\!|e|\!) : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, \mathbf{r}.\lfloor A\rfloor(\mathbf{r}))$ in U^{C}.

Proof. By induction on the RelCost derivation. We show some representative cases.

Case $\dfrac{\Omega(x) = A}{\Delta; \Phi; \Omega \vdash_0^0 x : A}$

TS: $|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash \{\mathsf{cret}(x)\} : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \mathbf{r}.\lfloor A\rfloor(\mathbf{r}))$.

By the rules U-MONAD and U-RET, RTS:

$$|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash x : |A| \mid \lfloor A\rfloor(\mathbf{r})$$

From the premise $\Omega(x) = A$ we have $(x : |A|) \in |\Omega|.$ and $\lfloor A\rfloor(x) \in \lfloor\Omega\rfloor$, hence the result follows by the rule U-VAR.

Case $\dfrac{\Delta; \Phi; \Omega, x : A_1 \vdash_k^\ell e : A_2}{\Delta; \Phi; \Omega \vdash_0^0 \lambda x.\ e : A_1 \xrightarrow{\mathrm{exec}(k,\ell)} A_2}$

TS: $|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash \{\mathsf{cret}(\lambda x.\ (\!|e|\!))\} : \mathbb{C}(|A_1| \to \mathbb{C}(|A_2|)) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, r.\forall x.\ \lfloor A_1\rfloor(x) \Rightarrow \mathbb{C}_{\mathrm{u}}(r\ x, k, \ell, \mathbf{r}.\lfloor A_2\rfloor(\mathbf{r})))$

By the rules U-MONAD, U-RET, and U-ABS, RTS:

$$|\Omega|, \Delta, x : |A_1|; \Phi, \lfloor\Omega\rfloor, \lfloor A_1\rfloor(x) \vdash (\!|e|\!) : \mathbb{C}(|A_2|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, \mathbf{r}.\lfloor A_2\rfloor(\mathbf{r}))$$

This follows directly by IH on the premise.

Case $\dfrac{\Delta;\Phi;\Omega \vdash^{\ell_1}_{k_1} e_1 : A_1 \xrightarrow{\mathrm{exec}(k,\ell)} A_2 \qquad \Delta;\Phi;\Omega \vdash^{\ell_2}_{k_2} e_2 : A_1}{\Delta;\Phi;\Omega \vdash^{\ell_1+\ell_2+\ell+c_{app}}_{k_1+k_2+k+c_{app}} e_1\ e_2 : A_2}$

TS: $|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash \{\mathsf{cbind}(\Downarrow e_1\Downarrow, \{x\}.\, \mathsf{cbind}(\Downarrow e_2\Downarrow, \{y\}.\, \mathsf{cbind}(x\ y, \{z\}.\, \mathsf{cstep}_{c_{app}}(\mathsf{cret}(z)))))\} : \mathbb{C}(|A_2|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k_1+k_2+k+c_{app}, \ell_1+\ell_2+\ell+c_{app}, \mathbf{r}.\lfloor A_2\rfloor(\mathbf{r}))$.

By IH on the first premise:

$$|\Omega|, \Delta; \Phi, \lfloor\Omega\rfloor \vdash \Downarrow e_1\Downarrow : \mathbb{C}(|A_1| \to \mathbb{C}(|A_2|)) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k_1, \ell_1, x.\overbrace{\forall w.\ \lfloor A_1\rfloor(w) \Rightarrow \mathbb{C}_{\mathrm{u}}(x\ w, k, \ell, x.\lfloor A_2\rfloor(x))}^{\phi'})$$

Hence, by the rule U-BIND, RTS:

$$|\Omega|, \Delta, x : |A_1| \to \mathbb{C}(|A_2|); \Phi, \lfloor\Omega\rfloor, \phi' \vdash \mathsf{cbind}(\Downarrow e_2\Downarrow, \{y\}.\ \cdots) \div |A_2| \mid k_2{+}k{+}c_{app} \mid \ell_2{+}\ell{+}c_{app} \mid \lfloor A_2\rfloor(\mathbf{r})$$

By IH on the second premise (with weakening):

$$|\Omega|, \Delta, x; \Phi, \lfloor\Omega\rfloor, \phi' \vdash \Downarrow e_2\Downarrow : |A_1| \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k_2, \ell_2, y.\overbrace{\lfloor A_1\rfloor(y)}^{\phi''})$$

Hence, by the rule U-BIND, RTS:

$$|\Omega|, \Delta, x, y; \Phi, \lfloor\Omega\rfloor, \phi', \phi'' \vdash \mathsf{cbind}(x\ y, \{z\}.\, \mathsf{cstep}_{c_{app}}(\mathsf{cret}(w))) \div |A_2| \mid k{+}c_{app} \mid \ell{+}c_{app} \mid \lfloor A_2\rfloor(\mathbf{r})$$

By U^{C} rule U-APP we have:

$$|\Omega|, \Delta, \ldots; \Phi, \lfloor\Omega\rfloor, \ldots \vdash y\ z : \mathbb{C}(|A_2|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, k, \ell, z.\lfloor A_2\rfloor(z))$$

Finally, by the rule U-BIND it remains to show:

$$|\Omega|, \Delta, x, y, z : |A_2|; \Phi, \lfloor\Omega\rfloor, \ldots, \lfloor A_2\rfloor(z) \vdash \mathsf{cstep}_{c_{app}}(\mathsf{cret}(z)) \div |A_2| \mid c_{app} \mid c_{app} \mid \lfloor A_2\rfloor(\mathbf{r})$$

This follows by the rules U-STEP, U-RET, and U-VAR.

□

Notation remark. $\overline{\Gamma_i}$ (for $i \in \{1,2\}$) is obtained from Γ by replacing each variable x in $\overline{\Gamma}$ with x^1 and x^2, respectively.

We assume that RelCost types are well-formed, formally $\Gamma; \Phi \vdash \tau\ \mathsf{wf}$ as defined in the original paper.

Before showing the main theorem for relational translation we show couple of supporting lemmas needed in the proof.

Lemma 27. The following hold:

1. $\|\tau\| = |\overline{\tau}|$.
2. $\|\Gamma\| = |\overline{\Gamma_1}|, |\overline{\Gamma_2}|$.
3. $\Delta; \Phi \vdash_{\mathrm{LC}} \forall x, y.\ [\![\tau]\!](x,y) \Rightarrow \lfloor\overline{\tau}\rfloor(x) \wedge \lfloor\overline{\tau}\rfloor(y)$.
4. $[\![\Gamma]\!] \Rightarrow \lfloor\overline{\Gamma_1}\rfloor \wedge \lfloor\overline{\Gamma_2}\rfloor$.

Proof. **Proof of (1)** By induction on τ.

Case int_r
TS: $\|\mathrm{int}_r\| \triangleq \mathbb{Z} = |\overline{\mathrm{int}_r}| \triangleq |\mathrm{int}| \triangleq \mathbb{Z}$. Trivially holds.

Case $\mathrm{list}[n]^\alpha\ \tau'$
TS: $\|\mathrm{list}[n]^\alpha\ \tau'\| \triangleq \mathrm{list}_{\|\tau'\|} = |\overline{\mathrm{list}[n]^\alpha\ \tau'}| \triangleq |\mathrm{list}[n]_{\overline{\tau'}}| \triangleq \mathrm{list}_{|\overline{\tau'}|}$. This follows by IH on τ'.

Case $\tau_1 \xrightarrow{\mathrm{diff}(n)} \tau_2$
TS: $\|\tau_1 \xrightarrow{\mathrm{diff}(n)} \tau_2\| \triangleq \|\tau_1\| \to \mathbb{C}(\|\tau_2\|) = |\overline{\tau_1 \xrightarrow{\mathrm{diff}(n)} \tau_2}| \triangleq |\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}| \triangleq |\overline{\tau_1}| \to \mathbb{C}(|\overline{\tau_2}|)$. This follows by IH on τ_1 and τ_2.

Case $\forall i \overset{\mathrm{diff}(n)}{::} S.\ \tau'$
TS: $\|\forall i \overset{\mathrm{diff}(n)}{::} S.\ \tau'\| \triangleq \mathrm{unit} \to \mathbb{C}(\|\tau'\|) = |\overline{\forall i \overset{\mathrm{diff}(n)}{::} S.\ \tau'}| \triangleq |\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau'}| \triangleq \mathrm{unit} \to \mathbb{C}(\overline{\tau'})$. This follows by IH on τ'.

Case UA
TS: $\|UA\| \triangleq |A| = |\overline{A}| \triangleq |A|$. Trivially holds.

Case $\Box\tau$
TS: $\|\Box\tau'\| \triangleq \|\tau'\| = |\overline{\Box\tau'}| \triangleq |\overline{\tau'}|$. This follows by IH on τ'.

Proof of (2) Let $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$. Then, using (1), $\|x_1 : \tau_1, \ldots, x_n : \tau_n\| \triangleq x_1^1 : \|\tau_1\|, x_2^1 : \|\tau_1\|, \ldots x_1^n : \|\tau_n\|, x_2^n : \|\tau_n\| = x_1^1 : |\overline{\tau_1}|, x_2^1 : |\overline{\tau_1}|, \ldots, x_1^n : |\overline{\tau_n}|, x_2^n : |\overline{\tau_n}| \triangleq |\overline{\Gamma_1}|, |\overline{\Gamma_2}|$.

Proof of (3) By induction on τ.

Case int_r
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ x \doteq y \Rightarrow \top \wedge \top$. Holds trivially.

Case $\mathrm{list}[n]^\alpha\ \tau'$
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ \mathrm{listR}_{\tau'}(x, y, n, \alpha) \Rightarrow \mathrm{listU}_{\overline{\tau'}}(x, n) \wedge \mathrm{listU}_{\overline{\tau'}}(y, n)$.
By ($\mathrm{L^C}$) sub-induction on x and y, by noting that by (main) IH on τ' we have $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ [\![\tau']\!](x, y) \Rightarrow \lfloor\overline{\tau'}\rfloor(x) \wedge \lfloor\overline{\tau'}\rfloor(y)$.

Case $\tau_1 \xrightarrow{\mathrm{diff}(n)} \tau_2$
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ (\lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(x) \wedge \lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(y) \wedge \cdots) \Rightarrow \lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(x) \wedge \lfloor\overline{\tau_1} \xrightarrow{\mathrm{exec}(0,\infty)} \overline{\tau_2}\rfloor(y)$. This holds trivially.

Case $\forall i \overset{\mathrm{diff}(n)}{::} S.\ \tau'$
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ (\lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau'}\rfloor(x) \wedge \lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau'}\rfloor(y) \wedge \cdots) \Rightarrow \lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau'}\rfloor(x) \wedge \lfloor\forall i \overset{\mathrm{exec}(0,\infty)}{::} S.\ \overline{\tau'}\rfloor(y)$. This holds trivially.

Case UA
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ \lfloor A\rfloor(x) \wedge \lfloor A\rfloor(y) \Rightarrow \lfloor A\rfloor(x) \wedge \lfloor A\rfloor(y)$. This holds trivially.

Case $\Box\tau'$
TS: $\Gamma; \Psi \vdash_{\mathrm{L^C}} \forall x, y.\ x \doteq y \wedge [\![\tau']\!](x, y) \Rightarrow \lfloor\overline{\tau'}\rfloor(x) \wedge \lfloor\overline{\tau'}\rfloor(y)$. This follows by IH on τ'.

Proof of (4) Let $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$. Assume: $[\![\tau_1]\!](x_1^1, x_2^1), \ldots, [\![\tau_n]\!](x_1^n, x_2^n)$. TS: $\lfloor\overline{\tau_1}\rfloor(x_1^1) \wedge \lfloor\overline{\tau_1}\rfloor(x_2^1), \ldots, \lfloor\overline{\tau_n}\rfloor(x_1^n) \wedge \lfloor\overline{\tau_n}\rfloor(x_2^n)$. This follows directly from (3). □

Lemma 28. If $\Delta;\Phi;\Gamma \vdash e_1 \ominus e_2 \lesssim \tau : n$ in RelCost, then $\Delta;\Phi;\overline{\Gamma} \vdash_0^\infty e_i : \overline{\tau}$ for $i \in \{1,2\}$.

Proof. By induction on the given derivation. □

Lemma 29. If $\Delta;\Phi \vDash \tau_1 \sqsubseteq \tau_2$, then $\Delta;\Phi \vdash_{\text{LC}} \forall x, y.\ [\![\tau_1]\!](x,y) \Rightarrow [\![\tau_2]\!](x,y)$.

Proof. By induction on the given derivation. We show couple of representative cases.

Case $\dfrac{}{\Delta;\Phi \vDash \tau \sqsubseteq U\overline{\tau}}$

TS: $\Delta;\Phi \vdash_{\text{LC}} \forall x, y.\ [\![\tau]\!](x,y) \Rightarrow \lfloor\overline{\tau}\rfloor(x) \wedge \lfloor\overline{\tau}\rfloor(y)$.

By Lemma 27(3).

Case $\dfrac{\Delta;\Phi \vDash \tau_1' \sqsubseteq \tau_1 \qquad \Delta;\Phi \vDash \tau_2 \sqsubseteq \tau_2' \qquad \Delta;\Phi \vDash n \leq n'}{\Delta;\Phi \vDash \tau_1 \xrightarrow{\text{diff}(n)} \tau_2 \sqsubseteq \tau_1' \xrightarrow{\text{diff}(n')} \tau_2'}$

TS: $\Delta;\Phi \vdash_{\text{LC}} \forall x, y.\ (\forall x_1, x_2.\ [\![\tau_1]\!](x_1,x_2) \Rightarrow \mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1,r_2)))$
$\Rightarrow (\forall x_1, x_2.\ [\![\tau_1']\!](x_1,x_2) \Rightarrow \mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n', r_1.r_2.[\![\tau_2']\!](r_1,r_2)))$

Assume: $\forall x, y.\ \forall x_1, x_2.\ [\![\tau_1]\!](x_1,x_2) \Rightarrow \mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1,r_2))$ and $[\![\tau_1']\!](x_1,x_2)$ for some x_1, x_2. Then RTS:

$$\mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n', r_1.r_2.[\![\tau_2']\!](r_1,r_2))$$

From IH on the first premise, applied to the second assumption from above, we have:

$$[\![\tau_1]\!](x_1,x_2)$$

This applied to the first assumption above, gives:

$$\mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1,r_2))$$

then by IH on the second premise applied to $[\![\tau_2]\!](r_1,r_2)$, and from $n \leq n'$ we have:

$$\mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n', r_1.r_2.[\![\tau_2']\!](r_1,r_2))$$

as required.

Case $\dfrac{}{\Delta;\Phi \vDash \Box(\tau_1 \xrightarrow{\text{diff}(n)} \tau_2) \sqsubseteq \Box\tau_1 \xrightarrow{\text{diff}(0)} \Box\tau_2}$

TS: $\Delta;\Phi \vdash_{\text{LC}} \forall x, y.\ (x \doteq y \wedge \lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(x) \wedge \lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(y) \wedge \forall x_1, x_2.\ [\![\tau_1]\!](x_1,x_2) \Rightarrow$
$\mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1,r_2))) \Rightarrow (\lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(x) \wedge \lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(y) \wedge \forall x_1, x_2.\ x_1 \doteq$
$x_2 \wedge [\![\tau_1]\!](x_1,x_2) \Rightarrow \mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, 0, r_1.r_2.r_1 \doteq r_2 \wedge [\![\tau_2]\!](r_1,r_2)))$.

Assume:

(a) $x \doteq y$

(b) $\lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(x)$ and $\lfloor\overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}\rfloor(y)$

(c) $\forall x_1, x_2.\ [\![\tau_1]\!](x_1,x_2) \Rightarrow \mathbb{C}_{\text{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1,r_2))$

RTS: $\lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(x) \wedge \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(y) \wedge \forall x_1, x_2.\ x_1 \doteq x_2 \wedge [\![\tau_1]\!](x_1, x_2) \Rightarrow \mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, 0, r_1.r_2.r_1 \doteq r_2 \wedge [\![\tau_2]\!](r_1, r_2))$.

By (a), assume (d) $x_1 \doteq x_2$, (e) $[\![\tau_1]\!](x_1, x_2)$, and then RTS:

$$\mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, 0, r_1.r_2.r_1 \doteq r_2 \wedge [\![\tau_2]\!](r_1, r_2))$$

By (e) applied to (c) we have:

$$\mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, n, r_1.r_2.[\![\tau_2]\!](r_1, r_2))$$

which expands (by the definition) to (for some e_1, e_2, n_1, n_2):

$$x\ x_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge y\ x_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge [\![\tau_2]\!](e_1, e_2) \wedge n_1 - n_2 \leq n$$

From (a) and (d) we have $x\ x_1 \doteq y\ x_2$, and hence $\mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2))$, and further $n_1 \doteq n_2$ and $e_1 \doteq e_2$. Hence we can conclude:

$$x\ x_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge y\ x_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge e_1 \doteq e_2 \wedge [\![\tau_2]\!](e_1, e_2) \wedge n_1 - n_2 \leq 0$$

and by folding the definition:

$$\mathbb{C}_{\mathrm{r}}(x\ x_1, y\ x_2, 0, r_1.r_2.r_1 \doteq r_2 \wedge \lfloor \tau_2 \rfloor(r_1, r_2))$$

as required.

□

Theorem 30 (RelCost, relational)**.** If $\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim n : \tau$ in RelCost, then $\|\Gamma\|, \Delta; \Phi, [\![\Gamma]\!] \vdash (\!|e_1|\!)_1 : \|\tau\| \sim (\!|e_2|\!)_2 : \|\tau\| \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.[\![\tau]\!](\mathbf{r}_1, \mathbf{r}_2))$ in R^{C}, where $(\!|e_i|\!)_i$ is a copy of $(\!|e_i|\!)$ where each variable x if replace by a variable x^i, for $i \in \{1, 2\}$.

Proof. By induction on the RelCost typing derivation. We show some representative cases.

Case $\dfrac{\Gamma(x) = \tau}{\Delta; \Phi; \Gamma \vdash x \ominus x \lesssim 0 : \tau}$

TS: $\|\Gamma\|, \Delta; \Phi, [\![\Gamma]\!] \vdash \{\mathsf{cret}(x_1)\} : \|\tau\| \sim \{\mathsf{cret}(x_2)\} : \|\tau\| \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.[\![\tau]\!](\mathbf{r}_1, \mathbf{r}_2))$.

By the rules R-MONAD and R-RET, RTS:

$$\|\Gamma\|, \Delta; \Phi, [\![\Gamma]\!] \vdash x_1 : \|\tau\| \sim x_2 : \|\tau\| \mid [\![\tau]\!](\mathbf{r}_1, \mathbf{r}_2)$$

From the premise $\Gamma(x) = \tau$, we have $x_1, x_2 : \|\tau\| \in \|\Gamma\|$, and $[\![\tau]\!](x_1, x_2) \in [\![\Gamma]\!]$, hence the result follows by the rule R-VAR.

Case $\dfrac{\Delta; \Phi; x : \tau_1, \Gamma \vdash e_1 \ominus e_2 \lesssim \tau_2 : n}{\Delta; \Phi; \Gamma \vdash \lambda x.\ e_1 \ominus \lambda x.\ e_2 \lesssim 0 : \tau_1 \xrightarrow{\text{diff}(n)} \tau_2}$

TS: $\|\Gamma\|, \Delta; \Phi, [\![\Gamma]\!] \vdash \{\mathsf{cret}(\lambda x_1.\ (\!|e_1|\!)_1)\} : \mathbb{C}(\|\tau_1\| \to \mathbb{C}(\|\tau_2\|)) \sim \{\mathsf{cret}(\lambda x_2.\ (\!|e_2|\!)_2)\} : \mathbb{C}(\|\tau_1\| \to \mathbb{C}(\|\tau_2\|)) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\phi)$, where $\phi \triangleq \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(\mathbf{r}_1) \wedge \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(\mathbf{r}_2) \wedge \forall x_1, x_2.\ [\![\tau_1]\!](x_1, x_2) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ x_1, \mathbf{r}_2\ x_2, n, \mathbf{r}_1'.\mathbf{r}_2'.[\![\tau_2]\!](\mathbf{r}_1', \mathbf{r}_2'))$.

By the rules R-MONAD and R-RET, RTS:

$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash \lambda x_1.\ (\!|e_1|\!)_1 : \|\tau_1\| \to \mathbb{C}(\|\tau_2\|) \sim \lambda x_2.\ (\!|e_2|\!)_2 : \|\tau_1\| \to \mathbb{C}(\|\tau_2\|) \mid \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(\mathbf{r}_1) \wedge \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(\mathbf{r}_2) \wedge \forall x_1, x_2.\ \llbracket\tau_1\rrbracket(x_1, x_2) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ x_1, \mathbf{r}_2\ x_2, n, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau_2\rrbracket(\mathbf{r}_1, \mathbf{r}_2))$.

By Lemma 28, we have: $\Delta; \Phi; \overline{\Gamma} \vdash_0^\infty \lambda x.\ e_i : \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2}$ for $i \in \{1, 2\}$. Hence, by Theorem 26, variable renaming and weakening, we have:

$$\Delta, |\overline{\Gamma}_1|, |\overline{\Gamma}_2|; \Phi, \lfloor\overline{\Gamma_1}\rfloor, \lfloor\overline{\Gamma_2}\rfloor \vdash \lambda x.\ (\!|e_i|\!)_i : |\overline{\tau_1}| \to \mathbb{C}(\overline{\tau_2}) \mid \lfloor \overline{\tau_1} \xrightarrow{\text{exec}(0,\infty)} \overline{\tau_2} \rfloor(\mathbf{r})$$

Hence, by Lemma 27, translation to L^{C} and applying to the goal, RTS:

$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash \lambda x_1.\ (\!|e_1|\!)_1 : \|\tau_1\| \to \mathbb{C}(\|\tau_2\|) \sim \lambda x_2.\ (\!|e_2|\!)_2 : \|\tau_1\| \to \mathbb{C}(\|\tau_2\|) \mid \forall x_1, x_2.\ \llbracket\tau_1\rrbracket(x_1, x_2) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ x_1, \mathbf{r}_2\ x_2, n, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau_2\rrbracket(\mathbf{r}_1, \mathbf{r}_2))$.

This follows directly by the rule R-ABS, and by IH.

Case $\dfrac{\Delta; \Phi; \overline{\Gamma} \vdash_{\ell_1}^{k_1} e_1 : A \qquad \Delta; \Phi; \overline{\Gamma} \vdash_{\ell_2}^{k_2} e_2 : A}{\Delta; \Phi; \Gamma \vdash e_1 \ominus e_2 \lesssim \ell_1 - k_2 : UA}$

TS: $\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash (\!|e_1|\!)_1 : \mathbb{C}(|A|) \sim (\!|e_2|\!)_2 : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, \ell_1 - k_2, \mathbf{r}_1.\mathbf{r}_2.\lfloor A\rfloor(\mathbf{r}_1) \wedge \lfloor A\rfloor(\mathbf{r}_2))$.

By Lemma 27(2) and (4), STS:

$|\overline{\Gamma}_1|, |\overline{\Gamma}_2|, \Delta; \Phi, \lfloor\overline{\Gamma}_1\rfloor, \lfloor\overline{\Gamma}_2\rfloor \vdash (\!|e_1|\!)_1 : \mathbb{C}(|A|) \sim (\!|e_2|\!)_2 : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, \ell_1 - k_2, \mathbf{r}_1.\mathbf{r}_2.\lfloor A\rfloor(\mathbf{r}_1) \wedge \lfloor A\rfloor(\mathbf{r}_2))$

By applying Theorem 26 to the premises, we have:

$$|\overline{\Gamma}|, \Delta; \Phi, \lfloor\overline{\Gamma}\rfloor \vdash (\!|e_i|\!) : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, \ell_i, k_i, \mathbf{r}.\lfloor A\rfloor(\mathbf{r}))$$

for $i \in \{1, 2\}$.

By weakening, and renaming of the variables in both expression, we have:

$$|\overline{\Gamma_1}|, |\overline{\Gamma_2}|, \Delta; \Phi, \lfloor\overline{\Gamma_1}\rfloor, \lfloor\overline{\Gamma_2}\rfloor \vdash (\!|e_i|\!)_i : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, \ell_i, k_i, \mathbf{r}.\lfloor A\rfloor(\mathbf{r}))$$

for $i \in \{1, 2\}$

Hence, by the rule R-SPLIT-PURE:

$|\overline{\Gamma_1}|, |\overline{\Gamma_2}|, \Delta; \Phi, \lfloor\overline{\Gamma_1}\rfloor, \lfloor\overline{\Gamma_2}\rfloor \vdash (\!|e_1|\!)_1 : \mathbb{C}(|A|) \sim (\!|e_2|\!)_2 : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, \ell_1 - k_2, \mathbf{r}_1.\mathbf{r}_2.\lfloor A\rfloor(\mathbf{r}_1) \wedge \lfloor A\rfloor(\mathbf{r}_2))$

as required.

Case $\dfrac{\Delta; \Phi; \Gamma \vdash e \ominus e \lesssim n : \tau \qquad \forall x \in dom(\Gamma).\ \Delta; \Phi \vDash \Gamma(x) \sqsubseteq \Box\Gamma(x)}{\Delta; \Phi; \Gamma, \Gamma' \vdash e \ominus e \lesssim 0 : \Box\tau}$

TS: $\|\Gamma, \Gamma'\|, \Delta; \Phi, \llbracket\Gamma, \Gamma'\rrbracket \vdash (\!|e|\!)_1 : \mathbb{C}(\|\tau\|) \sim (\!|e|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\mathbf{r}_1 \doteq \mathbf{r}_2 \wedge \llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2))$.

By Lemma 29 applied to the second premise, we have for all $x \in dom(\Gamma)$ that $\Delta; \Phi \vdash_{\mathrm{LC}} \llbracket\Gamma(x)\rrbracket(x_1, x_2) \Rightarrow \llbracket\Box\Gamma(x)\rrbracket(x_1, x_2)$. Since $\llbracket\Box\Gamma(x)\rrbracket(x, y) \Rightarrow x \doteq y$, and from $\llbracket\Gamma\rrbracket$ we know $\llbracket\Gamma(x)\rrbracket(x_1, x_2)$, it follows that $\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash_{\mathrm{LC}} x_1 \doteq x_2$. Since this holds for every $x \in dom(\Gamma)$, it follows immediately that

$$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash_{\mathrm{LC}} (\!|e|\!)_1 \doteq (\!|e|\!)_2$$

Call this statement (A).

By IH on the first premise we have:

$$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash (\!|e|\!)_1 : \mathbb{C}(\|\tau\|) \sim (\!|e|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2))$$

By L^{C} equivalence (Theorem 15(1)) and unfolding of monadic constraint we have (for some e_1, e_2, n_1, n_2):

$$(\!|e|\!)_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge (\!|e|\!)_2 \doteq \mathsf{cstep}_{n_2}(\mathsf{cret}(e_2)) \wedge \llbracket\tau\rrbracket(e_1, e_2) \wedge n_1 - n_2 \leq n$$

However, from (A) and an L^{C} axiom we have $n_1 \doteq n_2 \wedge e_1 \doteq e_2$. Hence:

$$(\!|e|\!)_1 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge (\!|e|\!)_2 \doteq \mathsf{cstep}_{n_1}(\mathsf{cret}(e_1)) \wedge \llbracket\tau\rrbracket(e_1, e_2) \wedge e_1 \doteq e_2 \wedge n_1 - n_1 \leq 0$$

By L^{C} equivalence (Theorem 17) it follows:

$$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash (\!|e|\!)_1 : \mathbb{C}(\|\tau\|) \sim (\!|e|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2) \wedge \mathbf{r}_1 \doteq \mathbf{r}_2)$$

The required result follows by weakening.

Case
$$\frac{\begin{array}{c}\Delta; \Phi; \Gamma \vdash e \ominus e' \lesssim m : \mathrm{list}[n]^{\alpha}\ \tau' \qquad \Delta; \Phi \wedge n = 0; \Gamma \vdash e_1 \ominus e_1' \lesssim m' : \tau \\ i, \Delta; \Phi \wedge n = i + 1; \Gamma, h : \square\tau', t : \mathrm{list}[i]^{\alpha}\ \tau' \vdash e_2 \ominus e_2' \lesssim m' : \tau \\ i, \beta, \Delta; \Phi \wedge n = i + 1 \wedge \alpha = \beta + 1; \Gamma, h : \square\tau', t : \mathrm{list}[i]^{\beta}\ \tau' \vdash e_2 \ominus e_2' \lesssim m' : \tau\end{array}}{\Delta; \Phi; \Gamma \vdash \mathsf{case}\ e\ \mathsf{of}\ \mathsf{nil} \to e_1 \mid h :: t \to e_2 \ominus \mathsf{case}\ e'\ \mathsf{of}\ \mathsf{nil} \to e_1' \mid h :: t \to e_2' \lesssim m + m' : \tau}$$

TS: $\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash \{\mathsf{cbind}((\!|e|\!)_1, \{x_1\}.\ \cdots)\} : \mathbb{C}(\|\tau\|) \sim \{\mathsf{cbind}((\!|e'|\!)_2, \{x_2\}.\ \cdots)\} : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, m + m', \mathbf{r}_1.\mathbf{r}_2.\llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2))$.

By IH on the first premise we have:

$$\|\Gamma\|, \Delta; \Phi, \llbracket\Gamma\rrbracket \vdash (\!|e|\!)_1 : \mathbb{C}(\mathrm{list}_{\|\tau'\|}) \sim (\!|e'|\!)_2 : \mathbb{C}(\mathrm{list}_{\|\tau'\|}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, m, x_1.x_2.\overbrace{\mathrm{listR}_{\tau'}(x_1, x_2, n, \alpha)}^{\phi'})$$

Hence, by the rules R-MONAD and R-BIND it remains to show:

$\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash \mathsf{cbind}(\mathsf{match}\ x_1\ \mathsf{with}\ \mathsf{nil} \mapsto (\!|e_1|\!)_1; \mathsf{cons} \mapsto \lambda h_1, t_1.\ (\!|e_2|\!)_1, \{y_1\}.\ \cdots) \div \|\tau\| \sim \mathsf{cbind}(\mathsf{match}\ x_2\ \mathsf{with}\ \mathsf{nil} \mapsto (\!|e_1'|\!)_2; \mathsf{cons} \mapsto \lambda h_1, t_1.\ (\!|e_2'|\!)_2, \{y_2\}.\ \cdots) \div \|\tau\| \mid m' \mid \llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2)$

Note that we have $\mathrm{listR}_{\tau'}(x_1, x_2, n, \alpha)$. We consider 4 cases for the inner match expression.

Case nil, nil: We want to show: (A) $\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash (\!|e_1|\!)_1 : \mathbb{C}(\|\tau\|) \sim (\!|e_1'|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathrm{listR}_{\tau'}(\mathsf{nil}, \mathsf{nil}, n, \alpha) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, m', y_1.y_2.\llbracket\tau\rrbracket(y_1, y_2))$

From $\mathrm{listR}(\mathsf{nil}, \mathsf{nil}, n, \alpha)$ we have $n \doteq 0$. Hence, STS:

$$\|\Gamma\|, \Delta, x_1, x_2; \Phi \wedge n \doteq 0, \llbracket\Gamma\rrbracket, \phi' \vdash (\!|e_1|\!)_1 : \mathbb{C}(\|\tau\|) \sim (\!|e_1'|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, m', y_1.y_2.\llbracket\tau\rrbracket(y_1, y_2))$$

This follows by IH on second premise with weakening.

Case nil, cons: We want to show: (B) $\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash (\!|e_1|\!)_1 : \mathbb{C}(\|\tau\|) \sim \lambda h_2, t_2.\ (\!|e_2'|\!)_2 : \|\tau'\| \to \mathrm{list}_{\|\tau'\|} \to \mathbb{C}(\|\tau\|) \mid \forall h, t.\ \mathrm{listR}_{\tau'}(\mathsf{nil}, \mathsf{cons}(h, t), n, \alpha) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2\ h\ t, m', y_1.y_2.\llbracket\tau\rrbracket(y_1, y_2))$.
From $\mathrm{listR}_{\tau'}(\mathsf{nil}, \mathsf{cons}(h, t), n, \alpha)$ we have a contradiction, hence this holds trivially.

Case cons, nil: We want to show: (C) $\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash \lambda h, t.\ (\!|e_2|\!)_1 : \|\tau'\| \to \mathrm{list}_{\|\tau'\|} \to \mathbb{C}(\|\tau\|) \sim (\!|e_1'|\!)_2 : \mathbb{C}(\|\tau\|) \mid \forall h, t.\ \mathrm{listR}_{\tau'}(\mathsf{cons}(h, t), \mathsf{nil}, n, \alpha) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ h\ t, \mathbf{r}_2, m', y_1.y_2.\llbracket\tau\rrbracket(y_1, y_2))$.

We obtain a contradiction in the same way as in the last sub-case.

Case $\mathsf{cons}, \mathsf{cons}$: We want to show: (D) $\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash \lambda h_1, t_1.\ (\!|e_1|\!)_1 : \|\tau'\| \to \mathrm{list}_{\|\tau'\|} \to \mathbb{C}(\|\tau\|) \sim \lambda h_2, t_2.\ (\!|e_1'|\!)_2 : \|\tau'\| \to \mathrm{list}_{\|\tau'\|} \to \mathbb{C}(\|\tau\|) \mid \forall h_1, h_2, t_1, t_2.\ \mathrm{listR}_{\tau'}(\mathsf{cons}(h_1, t_1), \mathsf{cons}(h_2, t_2), n, \alpha) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ h_1\ t_1, \mathbf{r}_2\ h_2\ t_2, m', y_1.y_2.\llbracket\tau\rrbracket(y_1, y_2))$

From $\mathrm{listR}_{\tau'}(\mathsf{cons}(h_1, t_1), \mathsf{cons}(h_2, t_2), n, \alpha)$ we distinguish two sub-cases.

Sub-case $h_1 \doteq h_2 \wedge \mathrm{listR}(t_1, t_2, n-1, \alpha)$, STS:

$\|\Gamma\|, i, \Delta, x_1, x_2, h_1, h_2, t_1, t_2; \Phi \ \wedge\ n \doteq i+1, \llbracket\Gamma\rrbracket, \llbracket\Box\|\tau'\|\rrbracket(h_1, h_1), \llbracket\mathrm{list}_{\|\tau'\|}\rrbracket(t_1, t_2, i, \alpha) \vdash (\!|e_1|\!)_1 \div \|\tau\| \sim (\!|e_1'|\!)_2 \div \|\tau\| \mid m' \mid \llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2)$

This follows by IH on the third premise.

Sub-case $\alpha > 0 \wedge \exists\beta.\ \alpha \doteq \beta + 1 \wedge \mathrm{listR}(t_1, t_2, n-1, \beta)$, STS:

$\|\Gamma\|, i, \beta, \Delta, x_1, x_2, h_1, h_2, t_1, t_2; \Phi \wedge n \doteq i{+}1, \alpha \doteq \beta{+}1, \llbracket\Gamma\rrbracket, \llbracket\|\tau'\|\rrbracket(h_1, h_1), \llbracket\mathrm{list}_{\|\tau'\|}\rrbracket(t_1, t_2, i, \beta) \vdash (\!|e_1|\!)_1 \div \|\tau\| \sim (\!|e_1'|\!)_2 \div \|\tau\| \mid m' \mid \llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2)$

This follows by IH on the fourth premise.

From (A), (B), (C) and (D) by the rule R-MATCH we have:

$\|\Gamma\|, \Delta, x_1, x_2; \Phi, \llbracket\Gamma\rrbracket, \phi' \vdash \mathsf{match}\ x_1\ \mathsf{with}\ \mathsf{nil} \mapsto (\!|e_1|\!)_1; \mathsf{cons} \mapsto \lambda h_1, t_1.\ (\!|e_2|\!)_1 : \mathbb{C}(\|\tau\|) \sim \mathsf{match}\ x_2\ \mathsf{with}\ \mathsf{nil} \mapsto (\!|e_1'|\!)_2; \mathsf{cons} \mapsto \lambda h_1, t_1.\ (\!|e_2'|\!)_2 : \mathbb{C}(\|\tau\|) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, m', y_1.y_2.\overbrace{\llbracket\tau\rrbracket(y_1, y_2)}^{\phi''})$.

Hence, by the rule R-BIND, RTS:

$\|\Gamma\|, \Delta, x_1, x_2, y_1, y_2; \Phi, \llbracket\Gamma\rrbracket, \ldots, \phi'' \vdash \mathsf{cstep}_{c_{case}}(\mathsf{cret}(y_1)) \div \|\tau\| \sim \mathsf{cstep}_{c_{case}}(\mathsf{cret}(y_2)) \div \|\tau\| \mid 0 \mid \llbracket\tau\rrbracket(\mathbf{r}_1, \mathbf{r}_2)$

This follows by the rules R-STEP, R-STEP, R-RET, and R-VAR.

□

5 Embedding of amortized cost analysis

In this section we give some RAML rules and translation details that were omitted from the paper, as well as proofs of the stated theorems.

5.1 Syntax

RAML expressions and types

Expressions	e	::=	$x \mid n \mid f(x) \mid \mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2 \mid \mathsf{nil} \mid$ $\mathsf{cons}(x_h, x_t) \mid \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 \mid \mathsf{cons}(x_h, x_t) \mapsto e_2$
Programs	P	::=	$(e_{f_1}, y_{f_1}), \ldots, (e_{f_n}, y_{f_n})$
Types	A	::=	$\mathrm{int} \mid L^q(A)$
Function types	F	::=	$A_1 \xrightarrow{q/q'} A_2$

RAML typing rules

$$\frac{n \in \mathbb{Z}}{\Sigma; \emptyset \vdash_q^{q+K^{int}} n : \text{int}}\text{L:ConstI} \qquad \frac{}{\Sigma; x : B \vdash_q^{q+K^{var}} x : B}\text{L:Var}$$

$$\frac{\Sigma; \Gamma_1 \vdash_p^{q-K_1^{let}} e_1 : A \qquad \Sigma; \Gamma_2 \vdash_{q'+K_3^{let}}^{p-K_2^{let}} e_2 : B}{\Sigma; \Gamma_1, \Gamma_2 \vdash_{q'}^{q} \mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2 : B}\text{L:Let} \qquad \frac{A \xrightarrow{q/q'} B \in \Sigma(f)}{\Sigma; x : A \vdash_{q'-K_2^{app}}^{q+K_1^{app}} f(x) : B}\text{L:App}$$

$$\frac{}{\Sigma; \emptyset \vdash_q^{q+K^{nil}} \mathsf{nil} : L^p(A)}\text{L:Nil} \qquad \frac{}{\Sigma; x_h : A, h_t : L^p(A) \vdash_q^{q+p+K^{cons}} \mathsf{cons}(x_h, x_t) : L^p(A)}\text{L:Cons}$$

$$\frac{\Sigma; \Gamma \vdash_{q'+K_2^{matN}}^{q-K_1^{matN}} e_1 : B \qquad \Sigma; \Gamma, x_h : A, x_t : L^p(A) \vdash_{q'+K_2^{matC}}^{q+p-K_1^{matC}} e_2 : B}{\Sigma; \Gamma, x : L^p(A) \vdash_{q'}^{q} \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2 : B}\text{L:MatL}$$

$$\frac{\Sigma; \Gamma, x : A_1, y : A_2 \vdash_{q'}^{q} e : B \qquad A \curlyvee (A_1, A_2)}{\Sigma; \Gamma, z : A \vdash_{q'}^{q} e[z/x][z/y] : B}\text{L:Share}$$

$$\frac{\Sigma; \Gamma, x : A \vdash_{q'}^{q} e : B \qquad A' <: A}{\Sigma; \Gamma, x : A' \vdash_{q'}^{q} e : B}\text{L:Supertype} \qquad \frac{\Sigma; \Gamma \vdash_{q'}^{q} e : B \qquad B <: B'}{\Sigma; \Gamma \vdash_{q'}^{q} e : B'}\text{L:Subtype}$$

$$\frac{\Sigma; \Gamma \vdash_{p'}^{p} e : B \qquad q \geq p \qquad q - p \geq q' - p'}{\Sigma; \Gamma \vdash_{q'}^{q} e : B}\text{L:Relax} \qquad \frac{\Sigma; \Gamma \vdash_{q'}^{q} e : B}{\Sigma; \Gamma, x : A \vdash_{q'}^{q} e : B}\text{L:Augment}$$

Type sharing & subtyping

$$\begin{array}{l} \text{int} \curlyvee (\text{int}, \text{int}) \\ L^p(A) \curlyvee (L^q(A_1), L^r(A_2)) \quad \text{if } A \curlyvee (A_1, A_2) \text{ and } p = q + r \end{array}$$

$$\begin{array}{ll} \text{int} <: \text{int} & \\ L^p(A) <: L^q(B) & \text{if } A <: B \text{ and } p \geq q \end{array}$$

5.2 Semantics

Set of values *Val* is defined as:

$$v ::= \ell \mid n \mid \textsc{Null}$$

Where $\ell \in Loc$ is a location, and *Loc* is an infinite set of locations.

Let $H : Loc \to Val$ be a *heap*, and $V : VID \to Val$ (where *VID* is set of variables) a *stack*. The evaluation judgment is then:

$$V, H \vdash e \rightsquigarrow v, H' \mid q$$

where $q \in \mathbb{R}^+$.

Note: As it was discussed in the paper, we limit ourselves to the analysis of additive costs only, hence the judgment above has only one component q, instead of (q, q').

$$\frac{x \in dom(V)}{V, H \vdash x \rightsquigarrow V(x), H \mid K^{var}}\text{E:Var} \qquad \frac{n \in \mathbb{Z}}{V, H \vdash n \rightsquigarrow n, H \mid K^{int}}\text{E:ConstI}$$

$$\frac{V(x) = v' \qquad [y_f \mapsto v'], H \vdash e_f \rightsquigarrow v, H' \mid q}{V, H \vdash f(x) \rightsquigarrow v, H' \mid K_1^{app} + q + K_2^{app}}\text{E:App}$$

$$\frac{V, H \vdash e_1 \rightsquigarrow v_1, H_1 \mid q \qquad V[x \mapsto v_1], H_1 \vdash e_2 \rightsquigarrow v_2, H_2 \mid p}{V, H \vdash \mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2 \rightsquigarrow v_2, H_2 \mid K_1^{let} + q + K_2^{let} + p + K_3^{let}}\text{E:Let}$$

$$\frac{}{V, H \vdash \mathsf{nil} \rightsquigarrow \textsc{Null}, H \mid K^{nil}}\text{E:Nil}$$

$$\frac{x_h, x_t \in dom(V) \qquad v = (V(x_h), V(x_t)) \qquad \ell \notin dom(H)}{V, H \vdash \mathsf{cons}(x_h, x_t) \rightsquigarrow \ell, H[\ell \mapsto v] \mid K^{cons}}\text{E:Cons}$$

$$\frac{V(x) = \textsc{Null} \qquad V, H \vdash e_1 \rightsquigarrow v, H' \mid q}{V, H \vdash \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2 \rightsquigarrow v, H' \mid K_1^{matN} + q + K_2^{matN}}\text{E:MatNil}$$

$$\frac{V(x) = \ell \qquad H(\ell) = (v_h, v_t) \qquad V[x_h \mapsto v_h, x_t \mapsto v_t], H \vdash e_2 \rightsquigarrow v, H' \mid q}{V, H \vdash \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2 \rightsquigarrow v, H' \mid K_1^{matC} + q + K_2^{matC}}\text{E:MatCons}$$

For each type A, $[\![A]\!]$ denotes the set of semantic values of type A, defined in a usual way. $H \vDash v \mapsto a : A$ denotes that v defines the semantic value $a \in [\![A]\!]$ when pointers are followed in H.

$\boxed{H \vDash v \mapsto a : A}$ v defines the semantic value $a \in [\![A]\!]$

$$\frac{v = \textsc{Null}}{H \vDash v \mapsto [] : L^q(A)}\text{V:Nil}$$

$$\frac{H(v) = (v_1, v_2) \qquad H' = H \setminus v \qquad H' \vDash v_1 \mapsto a_1 : A \qquad H' \vDash v_2 \mapsto [a_2, \ldots, a_n] : L^q(A) \qquad v \in Loc}{H \vDash v \mapsto [a_1, \ldots, a_n] : L^q(A)}\text{V:Cons}$$

$\boxed{\Phi(a : A)}$ the potential of a semantic value a of type A

$$\Phi(n : \text{int}) \triangleq 0 \qquad \Phi([a_1; \cdots; a_n] : L^q(A)) \triangleq q \cdot n + \sum_{1 \le i \le n} \Phi(a_i : A)$$

Let Γ be a standard typing context. A stack V and a heap H are *well-formed* with respect to a context Γ, written $H \vDash V : \Gamma$, if $H \vDash V(x) : \Gamma(x)$ holds for every $x \in dom(\Gamma)$.

Let A be a type, let H be a heap, and let $v \in Val$ be a value such that $H \vDash v \mapsto a : A$. The potential of v under type A in H is defined as $\Phi_H(v : A) = \Phi(a : A)$.

The potential of a typing context Γ with respect to a heap H and a stack V is:

$$\Phi_{V,H}(\Gamma) = \sum_{x \in dom(\Gamma)} \Phi_H(V(x) : \Gamma(x))$$

5.3 Translation rules

Erasing translation from RAML types to simple types

$$|\text{int}| \triangleq \text{int} \qquad |L^q(A)| \triangleq \text{list}_{|A|} \qquad |A_1 \xrightarrow{q/q'} A_2| \triangleq |A_1| \to \mathbb{C}(|A_2|)$$

Expression translation

$$(\!|n|\!) \triangleq \{\mathsf{cstep}_{K^{int}}(\mathsf{cret}(n))\} \text{ (for } n \in \mathbb{Z}) \qquad (\!|x|\!) \triangleq \{\mathsf{cstep}_{K^{var}}(\mathsf{cret}(x))\}$$

$$(\!|f(x)|\!) \triangleq \{\mathsf{cbind}(f\ x, \{r\}.\, \mathsf{cstep}_{K_1^{app}+K_2^{app}}(\mathsf{cret}(r)))\}$$

$$(\!|\mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2|\!) \triangleq \{\mathsf{cbind}((\!|e_1|\!), \{x\}.\, \mathsf{cbind}((\!|e_2|\!), \{r\}.\, \mathsf{cstep}_{K_1^{let}+K_2^{let}+K^{let_3}}(\mathsf{cret}(r))))\}$$

$$(\!|\mathsf{nil}|\!) \triangleq \{\mathsf{cstep}_{K^{nil}}(\mathsf{cret}(\mathsf{nil}))\} \qquad (\!|\mathsf{cons}(x_h, x_t)|\!) \triangleq \{\mathsf{cstep}_{K^{cons}}(\mathsf{cret}(\mathsf{cons}(x_h, x_t)))\}$$

$$(\!|\mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2|\!) \triangleq \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto$$

$$\{\mathsf{cbind}((\!|e_1|\!), \{r\}.\, \mathsf{cstep}_{K_1^{matN}+K_2^{matN}}(\mathsf{cret}(r)))\}; \mathsf{cons} \mapsto$$

$$\lambda x_h, x_t.\ \{\mathsf{cbind}((\!|e_2|\!), \{r\}.\, \mathsf{cstep}_{K_1^{matC}+K_2^{matC}}(\mathsf{cret}(r)))\}$$

Value translation

$$\|n\| \triangleq n \text{ (for } n \in \mathbb{Z}) \qquad \|[v_1; \cdots; v_n]\| \triangleq \mathsf{cons}(\|v_1\|, \mathsf{cons}(\|v_2\|, \cdots \mathsf{cons}(\|v_n\|, \mathsf{nil}) \cdots))$$

Predicate $\tilde{\Phi}_A(e, p)$ that encodes potentials

$$\forall x, p.\ \tilde{\Phi}_{\text{int}}(x, p) \Leftrightarrow p \doteq 0$$

$$\forall x, p.\ \tilde{\Phi}_{L^q(A)}(x, p) \Leftrightarrow \begin{array}{l} (p \doteq 0 \wedge x \doteq \mathsf{nil}) \vee \\ (\exists h, t, p_h, p_t.\ x \doteq \mathsf{cons}(h, t) \wedge \tilde{\Phi}_A(h, p_h) \\ \quad \wedge \tilde{\Phi}_{L^q(A)}(t, p_t) \wedge p \doteq q + p_h + p_t) \end{array}$$

Using the above axioms, it is not difficult to prove by induction on A that

$$\tilde{\Phi}_A(x, p_1) \wedge \tilde{\Phi}_A(x, p_2) \Rightarrow p_1 \doteq p_2$$

Translations for contexts

$$|x_1 : A_1, \ldots, x_n : A_n| \triangleq x_1 : |A_1|, x_1^p : \mathbb{R}^\infty, \ldots, x_n : |A_n|, x_n^p : \mathbb{R}^\infty$$

$$\lfloor x_1 : A_1, \ldots, x_n : A_n \rfloor \triangleq \tilde{\Phi}_{A_1}(x_1, x_1^p), \ldots, \tilde{\Phi}_{A_n}(x_n, x_n^p)$$

$$|\Sigma| \triangleq \{f : |F| \mid \text{for all } f \in dom(\Sigma) \text{ and some } F \in \Sigma(f)\}$$

$$\lfloor \Sigma \rfloor \triangleq \{\lfloor F \rfloor(f) \mid \text{for all } f \in dom(\Sigma) \text{ and all } F \in \Sigma(f)\}$$

Predicate over function types, that relates potentials to costs

$$\lfloor A_1 \xrightarrow{q/q'} A_2 \rfloor(f) \triangleq \forall y : |A_1|.\ \top \Rightarrow \forall y^p : \mathbb{R}^+.\ \tilde{\Phi}_{A_1}(y, y^p) \Rightarrow \exists p_r.\ \mathbb{C}_{\mathrm{u}}(f\ y, 0, y^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_{A_2}(\mathbf{r}, p_r))$$

5.4 Proofs

We need the following additional lemma for the next theorem below:

Lemma 31. The following hold:

1. If $A <: B$, then $|A| = |B|$.
2. If $A <: B$ then $\Gamma; \Psi \vdash_{\mathrm{LC}} \forall x, x^p.\ \tilde{\Phi}_A(x, x^p) \Rightarrow \exists x^{p'} \tilde{\Phi}_B(x, x^{p'}) \Rightarrow x^{p'} \leq x^p$.
3. If $A \curlyvee (A_1, A_2)$ then $|A| = |A_1| = |A_2|$.
4. If $A \curlyvee (A_1, A_2)$ then $\Gamma; \Psi \vdash_{\mathrm{LC}} \forall x, p_1, p_2.\ \tilde{\Phi}_{A_1}(x, p_1) \wedge \tilde{\Phi}_{A_2}(x, p_2) \Rightarrow \tilde{\Phi}_A(x, p_1 + p_2)$

Proof. **Proof of (1) and (2)** By induction on $A' <: A$.

Proof of (3) and (4) By induction on $A \curlyvee (A_1, A_2)$ □

Theorem 32 (Soundness of embedding)**.** If $\Sigma; \Gamma \vdash_{q'}^{q} e : A$ then $|\Sigma|, |\Gamma|; \lfloor \Sigma \rfloor, \lfloor \Gamma \rfloor \vdash (\!|e|\!) : \mathbb{C}(|A|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + q - q' - p_r, r.\tilde{\Phi}_A(r, p_r))$, where $\tilde{\Phi}(\Gamma) \triangleq \sum_{x \in dom(\Gamma)} x^p$ (the sum of all potential variables in the context).

Proof. By induction on the RAML derivation; we show some representative cases.

Case $\dfrac{n \in \mathbb{Z}}{\Sigma; \emptyset \vdash_q^{q+K^{int}} n : \mathrm{int}}$ L:ConstI

TS: $|\Sigma|; \lfloor \Sigma \rfloor \vdash \{\mathsf{cstep}_{K^{int}}(\mathsf{cret}(n))\} : \mathbb{C}(\mathrm{int}) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, q + K^{int} - q - p_r, \mathbf{r}.\tilde{\Phi}_{\mathrm{int}}(\mathbf{r}, p_r))$.

We pick $p_r := 0$. By the rule U-MONAD, RTS:

$$|\Sigma|; \lfloor \Sigma \rfloor \vdash \mathsf{cstep}_{K^{int}}(\mathsf{cret}(n)) \div \mathrm{int} \mid 0 \mid K^{int} \mid \tilde{\Phi}_{\mathrm{int}}(\mathbf{r}, 0)$$

This follows trivially by the rules U-STEP and U-RET since $\tilde{\Phi}_{\mathrm{int}}(n, 0)$.

Case $\dfrac{}{\Sigma; x : B \vdash_{q}^{q+K^{var}} x : B}\,\text{L:Var}$

TS: $|\Sigma|, x : |B|, x^p : \mathbb{R}^{\infty}; \lfloor\Sigma\rfloor, \tilde{\Phi}_B(x, x^p) \vdash \{\mathsf{cstep}_{K^{var}}(\mathsf{cret}(x))\} : \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, x^p + q + K^{var} - q - p_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p_r))$.

We pick $p_r := x^p$. Hence, after applying the rule U-MONAD RTS:

$$|\Sigma|, x : |B|, x^p; \lfloor\Sigma\rfloor, \tilde{\Phi}_B(x, x^p) \vdash \mathsf{cstep}_{K^{var}}(\mathsf{cret}(x)) \div |B| \mid 0 \mid K^{var} \mid \tilde{\Phi}_B(\mathbf{r}, x^p)$$

This follows by the rules U-STEP and U-RET, and $\tilde{\Phi}_B(x, x^p)$.

Case $\dfrac{\Sigma; \Gamma_1 \vdash_{p}^{q-K_1^{let}} e_1 : A \qquad \Sigma; \Gamma_2 \vdash_{q'+K_3^{let}}^{p-K_2^{let}} e_2 : B}{\Sigma; \Gamma_1, \Gamma_2 \vdash_{q'}^{q} \mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2 : B}\,\text{L:Let}$

TS: $|\Sigma|, |\Gamma_1|, |\Gamma_2|; \lfloor\Sigma\rfloor, \lfloor\Gamma_1\rfloor, \lfloor\Gamma_2\rfloor \vdash \{\mathsf{cbind}(\langle\!|e_1|\!\rangle, \{x\}.\, \mathsf{cbind}(\langle\!|e_2|\!\rangle, \{r\}.\, \mathsf{cstep}_{K_1^{let}+K_2^{let}+K_3^{let}}(\mathsf{cret}(r))))\} : \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma_1) + \tilde{\Phi}(\Gamma_2) + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p_r))$.

By IH on the first premise we have:

$$|\Sigma|, |\Gamma_1|; \lfloor\Sigma\rfloor, \lfloor\Gamma_1\rfloor \vdash \langle\!|e_1|\!\rangle : \mathbb{C}(|A|) \mid \exists p_{r1}.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma_1) + q - K_1^{let} - p - p_{r1}, x.\tilde{\Phi}_A(x, p_{r1}))$$

Call this statement (A).

By IH on the second premise we have:

$|\Sigma|, |\Gamma_2|, x : |A|, x^p; \lfloor\Sigma\rfloor, \lfloor\Gamma_2\rfloor, \tilde{\Phi}_A(x, x^p) \vdash \langle\!|e_2|\!\rangle : \mathbb{C}(|B|) \mid \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma_2) + x^p + p - K_2^{let} - q' - K_3^{let} - p'_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p'_r))$.

Instantiating this with $x^p := p_{r1}$, we get: $|\Sigma|, |\Gamma_2|, x : |A|; \lfloor\Sigma\rfloor, \lfloor\Gamma_2\rfloor, \tilde{\Phi}_A(x, p_{r1}) \vdash \langle\!|e_2|\!\rangle : \mathbb{C}(|B|) \mid \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma_2) + p_{r1} + p - K_2^{let} - q' - K_3^{let} - p'_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p'_r))$.

Call this statement (B).

We pick $p_r := p'_r$. By the rule U-MONAD statement (A) (with weakening) and the rule U-BIND, RTS:

$\ldots, x : |A|; \ldots, \tilde{\Phi}_A(x, p_{r1}) \vdash (\mathsf{cbind}(\langle\!|e_2|\!\rangle, \{r\}.\ \cdots) \div |B| \mid 0 \mid \tilde{\Phi}(\Gamma_1) + \tilde{\Phi}(\Gamma_2) + q - q' - p'_r - (\tilde{\Phi}(\Gamma_1) + q - K_1^{let} - p - p_{r1}) \mid \tilde{\Phi}_B(\mathbf{r}, p'_r)$.

Note that the upper cost bound is (by U-SUBC rule): $\tilde{\Phi}(\Gamma_2) - q' - p'_r + K_1^{let} + p + p_{r1}$. Using statement (B) (with weakening) and the U-BIND, RTS:

$\ldots, r : |B|; \ldots, \tilde{\Phi}_B(r, p'_r) \vdash \mathsf{cstep}_{K_1^{let}+K_2^{let}+K_3^{let}}(\mathsf{cret}(r)) \div |B| \mid 0 \mid \tilde{\Phi}(\Gamma_2) - q' - p'_r + K_1^{let} + p + p_{r1} - (\tilde{\Phi}(\Gamma_2) + p_{r1} + p - K_2^{let} - q' - K_3^{let} - p'_r) \mid \tilde{\Phi}_B(\mathbf{r}, p'_r)$

Simplified by U-SUBC, RTS:

$\ldots, r : |B|; \ldots, \tilde{\Phi}_B(r, p'_r) \vdash \mathsf{cstep}_{K_1^{let}+K_2^{let}+K_3^{let}}(\mathsf{cret}(r)) \div |B| \mid 0 \mid K_1^{let} + K_2^{let} + K_3^{let} \mid \tilde{\Phi}_B(\mathbf{r}, p'_r)$

This holds by the rules U-STEP and U-RET.

Case $\dfrac{A \xrightarrow{q/q'} B \in \Sigma(f)}{\Sigma; x : A \vdash_{q'-K_2^{app}}^{q+K_1^{app}} f(x) : B}\,\text{L:App}$

TS: $|\Sigma|, x : |A|, x^p; \lfloor\Sigma\rfloor, \tilde{\Phi}_A(x, x^p) \vdash \{\mathsf{cbind}(f\ x, \{r\}.\, \mathsf{cstep}_{K_1^{app}+K_2^{app}}(\mathsf{cret}(r)))\} : \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, x^p + q + K_1^{app} - (q' - K_2^{app}) - p_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p_r))$.

From $A \xrightarrow{q/q'} B \in \Sigma(f)$, we have and $f : |A| \to \mathbb{C}(|B|) \in |\Sigma|$ and

$$\forall y, y^p.\ \tilde{\Phi}_A(y, y^p) \Rightarrow \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(f\ y, 0, y^p + q - q' - p'_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p'_r)) \in \lfloor \Sigma \rfloor$$

Hence, by Theorem 9:

$$|\Sigma|, x : |A|, x^p; \lfloor \Sigma \rfloor, \tilde{\Phi}_A(x, x^p) \vdash f\ x : \mathbb{C}(|B|) \mid \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, x^p + q - q' - p'_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p'_r))$$

We pick $p_r := p'_r$. Then, by the rules U-MONAD and U-BIND, RTS:
$\ldots, r : |B|; \ldots, \tilde{\Phi}_B(r, p'_r) \vdash \mathsf{cstep}_{K_1^{app_1} + K^{app_2}}(\mathsf{cret}(r)) \doteqdot |B| \mid 0 \mid x^p + q + K_1^{app} - (q' - K_2^{app}) - p'_r - (x^p + q - q' - p'_r) \mid \tilde{\Phi}_B(\mathbf{r}, p'_r)$
Since $x^p + q + K_1^{app} - (q' - K_2^{app}) - p_r - (x^p + q - q' - p'_r) \doteq K_1^{app} + K_2^{app}$, the result follows by rule U-SUBC and the rules U-STEP and U-RET.

Case $\dfrac{}{\Sigma; x_h : A, h_t : L^p(A) \vdash^{q+p+K^{cons}}_{q} \mathsf{cons}(x_h, x_t) : L^p(A)}$ L:Cons

TS: $|\Sigma|, x_h : |A|, x_h^p, h_t : \mathrm{list}_{|A|}; \lfloor \Sigma \rfloor, \tilde{\Phi}_A(x_h, x_h^p), \tilde{\Phi}_{L^p(A)}(x_t, x_t^p) \vdash \{\mathsf{cstep}_{K^{cons}}(\mathsf{cret}(\mathsf{cons}(x_h, x_t)))\} : \mathbb{C}(\mathrm{list}_{|A|}) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, x_h^p + x_t^p + q + p + K^{cons} - q - p_r, \mathbf{r}.\tilde{\Phi}_{L^p(A)}(\mathbf{r}, p_r))$.

We pick $p_r := x_h^p + x_t^p + p$. After applying the rules U-MONAD and U-SUBC, RTS:

$|\Sigma|, x_h : |A|, x_h^p, h_t : \mathrm{list}_{|A|}; \lfloor \Sigma \rfloor, \tilde{\Phi}_A(x_h, x_h^p), \tilde{\Phi}_{L^p(A)}(x_t, x_t^p) \vdash \mathsf{cstep}_{K^{cons}}(\mathsf{cret}(\mathsf{cons}(x_h, x_t))) \doteqdot \mathrm{list}_{|A|} \mid 0 \mid K^{cons} \mid \tilde{\Phi}_{L^p(A)}(\mathbf{r}, p_r)$

Since, $\tilde{\Phi}_A(x_h, x_h^p)$, $\tilde{\Phi}_{L^p(A)}(x_t, x_t^p)$, we have $\tilde{\Phi}_{L^p(A)}(\mathsf{cons}(x_h, x_t), x_h^p + x_t^p + p)$, and since $p_r \doteq x_h^p + x_t^p + p$, the result follows by the rules U-STEP and U-RET.

Case $\dfrac{\Sigma; \Gamma \vdash^{q - K_1^{matN}}_{q' + K_2^{matN}} e_1 : B \qquad \Sigma; \Gamma, x_h : A, x_t : L^p(A) \vdash^{q+p-K_1^{matC}}_{q' + K_2^{matC}} e_2 : B}{\Sigma; \Gamma, x : L^p(A) \vdash^{q}_{q'} \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2 : B}$ L:MatL

TS: $|\Sigma|, |\Gamma|, x : \mathrm{list}_{|A|}, x^p; \lfloor \Sigma \rfloor, \lfloor \Gamma \rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p) \vdash \mathsf{match}\ x\ \mathsf{with}$
$\mathsf{nil} \mapsto \{\mathsf{cbind}(\llparenthesis e_1 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matN} + K_2^{matN}}(\mathsf{cret}(r)))\};$
$\mathsf{cons} \mapsto \lambda x_h, x_t.\ \{\mathsf{cbind}(\llparenthesis e_2 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matC} + K_2^{matC}}(\mathsf{cret}(r)))\}$
$: \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_B(\mathbf{r}, p_r))$

By IH on the first premise we have (with weakening; note especially $x \doteq \mathsf{nil}$):

$|\Sigma|; |\Gamma|, x, x^p; \lfloor \Sigma \rfloor, \lfloor \Gamma \rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p), x \doteq \mathsf{nil} \vdash \llparenthesis e_1 \rrparenthesis : \mathbb{C}(|B|) \mid \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + q - K_1^{matN} - (q' + K_2^{matN}) - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

Note that from $x \doteq \mathsf{nil}$ and $\tilde{\Phi}_{L^p(A)}(x, x^p)$ we have $x^p \doteq 0$. Hence by using the rules U-MONAD, U-STEP and U-RET and $\Rightarrow$, we have:

$|\Sigma|; |\Gamma|, x; \lfloor \Sigma \rfloor, \lfloor \Gamma \rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p) \vdash \{\mathsf{cbind}(\llparenthesis e_1 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matN} + K_2^{matN}}(\mathsf{cret}(r)))\} : \mathbb{C}(|B|) \mid x \doteq \mathsf{nil} \Rightarrow \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - q' - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

Call this statement (A).

By IH on the second premise we have (with weakening, note especially $x \doteq \mathsf{cons}(x_h, x_t)$):

$|\Sigma|; |\Gamma|, x, x_h, x_t, x^p, x_h^p, x_t^p; \lfloor \Sigma \rfloor, \lfloor \Gamma \rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p), \tilde{\Phi}_A(x_h, x_h^p), \tilde{\Phi}_{L^p(A)}(x_t, x_t^p), x \doteq \mathsf{cons}(x_h, x_t) \vdash \llparenthesis e_2 \rrparenthesis : \mathbb{C}(|B|) \mid \exists p'_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x_h^p + x_t^h + q + p - K_1^{matC} - (q' + K_2^{matC}) - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

From $\tilde{\Phi}_{L^p(A)}(x, x^p)$ and $x \doteq \mathsf{cons}(x_h, x_t)$, there must be some p_h and p_t such that $\tilde{\Phi}_A(x_h, p_h)$, $\tilde{\Phi}_{L^p(A)}(x_t, p_t)$ and $x^p \doteq p + p_h + p_t$. It follows immediately that $x_h^p \doteq p_h$ and $x_t^p \doteq p_t$. Hence, from the above, we also get:

$|\Sigma|; |\Gamma|, x, x_h, x_t; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p), x \doteq \mathsf{cons}(x_h, x_t) \vdash (\!|e_2|\!) : \mathbb{C}(|B|) \mid \exists p'_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + p_h + p_q + q + p - K_1^{matC} - (q' + K_2^{matC}) - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

which is the same as (via U-SUBC):

$|\Sigma|; |\Gamma|, x, x_h, x_t; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p), x \doteq \mathsf{cons}(x_h, x_t) \vdash (\!|e_2|\!) : \mathbb{C}(|B|) \mid \exists p'_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - K_1^{matC} - (q' + K_2^{matC}) - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

Hence by using the rules U-MONAD, U-STEP, U-RET, and $\Rightarrow$ and rule U-ABS, we have:

$|\Sigma|; |\Gamma|, x; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{L^p(A)}(x, x^p)) \vdash \lambda x_h, x_t.\ \{\mathsf{cbind}((\!|e_2|\!), \{r\}.\, \mathsf{cstep}_{K_1^{matC} + K_2^{matC}}(\mathsf{cret}(r)))\} : \mathbb{C}(|B|) \mid \forall x_h, x_t.\ x \doteq \mathsf{cons}(x_h, x_t) \Rightarrow \exists p'_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - q' - p'_r, r.\tilde{\Phi}_B(r, p'_r))$

Call this statement (B).

Now, from (A) and (B) the required goal follows by rule U-MATCH.

Case $\dfrac{\Sigma; \Gamma, x : A_1, y : A_2 \vdash^q_{q'} e : B \qquad A \curlyvee (A_1, A_2)}{\Sigma; \Gamma, z : A \vdash^q_{q'} e[z/x][z/y] : B}$ L:Share

TS: $|\Sigma|, |\Gamma|, z : |A|, z^p : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_A(z, z^p) \vdash (\!|e[z/x][z/y]|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + z^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

By IH on the first premise we have:

$|\Sigma|, |\Gamma|, x : |A_1|, y : |A_2|, x^p, y^p : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{A_1}(x, x^p), \tilde{\Phi}_{A_2}(y, y^p) \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + y^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

We can replace all occurrences of x and y by z and using Lemma 31:

$|\Sigma|, |\Gamma|, z : |A|, x^p, y^p : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_A(z, x^p + y^p) \vdash (\!|e[z/x][z/y]|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + y^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

Finally we replace $x^p + y^p$ by a variable z^p:

$|\Sigma|, |\Gamma|, z : |A|, z^p : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_A(z, z^p) \vdash (\!|e[z/x][z/y]|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + z^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

as required.

Case $\dfrac{\Sigma; \Gamma, x : A \vdash^q_{q'} e : B \qquad A' <: A}{\Sigma; \Gamma, x : A' \vdash^q_{q'} e : B}$ L:Supertype

TS: $|\Sigma|, |\Gamma|, x : |A'|, x^p : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{A'}(x, x^p) \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$.

By Lemma 31 using the premise $A' <: A$:

$|\Sigma|, |\Gamma|, x : |A'|, x^p, x^{p'} : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_{A'}(x, x^p), \tilde{\Phi}_A(x, x^{p'}), x^{p'} \leq x^p \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^p + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$.

Since $x^{p'} \leq x^p$, in L^{C} we can rewrite this to (with weakening):

$|\Sigma|, |\Gamma|, x : |A'|, x^{p'} : \mathbb{R}^\infty; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \tilde{\Phi}_A(x, x^{p'}) \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + x^{p'} + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$.

This follows by IH on the first premise.

Case $\dfrac{\Sigma;\Gamma \vdash^{p}_{p'} e : B \qquad q \geq p \qquad q - p \geq q' - p'}{\Sigma;\Gamma \vdash^{q}_{q'} e : B}\,\text{L:Relax}$

$|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + q - q' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

By IH on the first premise we have: $|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor \vdash (\!|e|\!) : \mathbb{C}(|B|) \mid \exists p_r.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + p - p' - p_r, \mathbf{r}.\tilde{\Phi}_A(\mathbf{r}, p_r))$

We pick $p_r := p_r$. The result follows since we have, from the third premise, that $\tilde{\Phi}(\Gamma) + p - p' - p_r \leq \tilde{\Phi}(\Gamma) + q - q' - p_r$.

□

We define translation of values in context Γ (under a stack V and a heap H):

$$\lfloor\Gamma\rfloor_{V,H} \triangleq \{x \doteq \|a\| \mid x \in dom(\Gamma) \text{ and } H \vDash V(x) \mapsto a : \Gamma(x)\}$$

In the next theorem we implicitly use the RAML's (simple) typing derivation, which is not stated here, but can be obtained from the resource annotated typing derivation by erasing all resources. We will also use (without explicitly mentioning) the following RAML meta-theoretical results:

Lemma 33. The following hold:

- If $V, H \vdash e \rightsquigarrow v, H' \mid p$ then $H'(\ell) = H(\ell)$ for all $\ell \in dom(H)$.
- If $H \vDash v \mapsto a : A$ and $H \vDash v \mapsto a' : A$ then $a = a'$.
- If $\Sigma;\Gamma \vdash e : A$, $H \vDash V : \Gamma$ and $V, H \vdash e \rightsquigarrow v, H' \mid p$ then $H' \vDash V : \Gamma$ and $H' \vDash v : B$.

We define $\|\Gamma\| \triangleq |\Gamma| \setminus \{x^p \mid x \in dom(\Gamma)\}$, i.e, a context translation without potential variables x^p.

Theorem 34 (RAML, U^{C} soundness)**.** Let $H \vDash V : \Gamma$ and $\Sigma, \Gamma \vdash e : A$. If $V, H \vdash e \rightsquigarrow v, H' \mid q$ and $H' \vDash v \mapsto a : A$ then $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \|e\| : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, q, q, \mathbf{r}.\mathbf{r} \doteq \|a\|)$.

Proof. By induction on the RAML evaluation relation; we show some representative cases.

Case $\dfrac{n \in \mathbb{Z}}{V, H \vdash n \rightsquigarrow n, H \mid K^{int}}\,\text{E:ConstI}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \{\mathsf{cstep}_{K^{int}}(\mathsf{cret}(n))\} : \mathbb{C}(\text{int}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K^{int}, K^{int}, \mathbf{r}.\mathbf{r} \doteq \|a\|)$ where $H \vDash n \mapsto a : \text{int}$.

We have $H \vDash n \mapsto n : \text{int}$ and $\|n\| \triangleq n$, and hence $\|a\| = n$.

Case $\dfrac{x \in dom(V)}{V, H \vdash x \rightsquigarrow V(x), H \mid K^{var}}\,\text{E:Var}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \{\mathsf{cstep}_{K^{var}}(\mathsf{cret}(x))\} : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K^{var}, K^{var}, \mathbf{r}.\mathbf{r} \doteq \|a\|)$ where $H \vDash V(x) \mapsto a : A$.

We have $x \doteq \|a\| \in \lfloor\Gamma\rfloor_{V,H}$ for $H \vDash V(x) \mapsto a : A$, hence the result holds immediately.

Case $\dfrac{V, H \vdash e_1 \rightsquigarrow v_1, H_1 \mid q \qquad V[x \mapsto v_1], H_1 \vdash e_2 \rightsquigarrow v_2, H_2 \mid p}{V, H \vdash \mathsf{let}\ x \leftarrow e_1\ \mathsf{in}\ e_2 \rightsquigarrow v_2, H_2 \mid K_1^{let} + q + K_2^{let} + p + K_3^{let}}\text{E:Let}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \{\mathsf{cbind}((\!|e_1|\!), \{x\}.\, \mathsf{cbind}((\!|e_2|\!), \{r\}.\, \mathsf{cstep}_{K_1^{let}+K_2^{let}+K_3^{let}}(\mathsf{cret}(r))))\} : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K_1^{let}+q+K_2^{let}+p+K_3^{let}, K_1^{let}+q+K_2^{let}+p+K_3^{let}, \mathbf{r}.\mathbf{r} \doteq \|a_2\|)$ where $H_2 \vDash v_2 \mapsto a_2 : A$.

By IH on the first premise we have (statement (A)):

$$|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash (\!|e_1|\!) : \mathbb{C}(|B|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, q, q, x.x \doteq \|a_1\|)$$

where $H_1 \vDash v_1 \mapsto a_1 : B$.

By IH on the second premise (and using Lemma 33) we have (statement (B)):

$$|\Sigma|, \|\Gamma\|, x : |B|; \lfloor\Gamma\rfloor_{V,H_1}, x \doteq \|a_1\| \vdash (\!|e_2|\!) : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, p, p, x.x \doteq \|a_2\|)$$

where $H_2 \vDash v_2 \mapsto a_2 : A$.

The required result is obtained by using the U-MONAD, then (A) with the rule U-BIND, and then (B) with the rule U-BIND, and finally by the rules U-STEP and U-RET.

Case $\dfrac{V(x) = v' \qquad [y_f \mapsto v'], H \vdash e_f \rightsquigarrow v \mid q}{V \vdash f(x) \rightsquigarrow v \mid K_1^{app} + q + K_2^{app}}\text{E:App}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \{\mathsf{cbind}(f\ x, \{r\}.\, \mathsf{cstep}_{K_1^{app}+K_2^{app}}(\mathsf{cret}(r)))\} : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K_1^{app} + q + K_2^{app}, K_1^{app} + q + K_2^{app}, \mathbf{r}.\mathbf{r} \doteq \|a\|)$ where $H \vDash v \mapsto a : A$.

Since $f(x) : A$, that means that $A' \xrightarrow{-/-} A \in \Sigma(f)$ and $\Sigma, y_f : A' \vdash e_f : A$ for some A'. From $V(x) = v'$ we have $x \doteq a' \in \lfloor\Gamma\rfloor_{V,H}$, where $H \vDash v' \mapsto a' : A'$. Hence, we can apply IH on the second premise and we obtain (by weakening):

$$|\Sigma|, \|\Gamma\|, y_f : |A'|, y_f^p; \lfloor\Gamma\rfloor_{V,H}, y_f \doteq \|a'\| \vdash (\!|e_f|\!) : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, q, q, \mathbf{r}.\mathbf{r} \doteq \|a\|)$$

Since $y_f \doteq x \doteq \|a'\|$, and $f \triangleq \mathsf{rec} f(y_f).(\!|e_f|\!)$, then $f\ x \doteq f\ y_f \doteq (\!|e_f|\!)[f/f][y_f/y_f] \doteq (\!|e_f|\!)$; and hence:

$$|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash f\ x : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, q, q, \mathbf{r}.\mathbf{r} \doteq \|a\|)$$

By the rule U-MONAD, RTS:

$|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H}, r \doteq \|a\| \vdash \mathsf{cstep}_{K_1^{app}+K_2^{app}}(\mathsf{cret}(r)) \div |A| \mid K_1^{app} + q + K^{app_2} - q \mid K_1^{app} + q + K^{app_2} - q \mid \mathbf{r} \doteq \|a\|$

This follows by rule U-SUBC, U-STEP and U-RET.

Case $\dfrac{x_h, x_t \in dom(V) \qquad v = (V(x_h), V(x_t)) \qquad \ell \notin dom(H)}{V, H \vdash \mathsf{cons}(x_h, x_t) \rightsquigarrow \ell, H[\ell \mapsto v] \mid K^{cons}}\text{E:Cons}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \{\mathsf{cstep}_{K^{cons}}(\mathsf{cret}(\mathsf{cons}(x_h, x_t)))\} : \mathbb{C}(\mathrm{list}_{|A|}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K^{cons}, K^{cons}, r.r \doteq \|a\|)$ where $H[\ell \mapsto v] \vDash \ell \mapsto a : A$.

We have $H \vDash V(x_h) \mapsto a_h : A$ and $H \vDash V(x_t) \mapsto a_t : L(A)$ for some a_h and a_t. Hence $a_t = [a_1, \ldots, a_n] \in [\![L(A)]\!]$, and $\|a_t\| \doteq \mathsf{cons}(a_1, \ldots, \mathsf{cons}(a_n, \mathsf{nil}) \cdots)$. Further, $a = [a_h, a_1, \ldots, a_n]$.

Note that we have $x_h \doteq \|a_h\|$, $x_t \doteq \|a_t\| \in \lfloor\Gamma\rfloor_{V,H}$. But then

$$\|a\| \doteq \mathsf{cons}(\|a_h\|, \mathsf{cons}(\|a_1\|, \ldots, \mathsf{cons}(\|a_n\|, \mathsf{nil}) \cdots) \doteq \mathsf{cons}(\|a_h\|, \|a_t\|) \doteq \mathsf{cons}(x_h, x_t)$$

The required result the follows by the rules U-MONAD, U-STEP, and U-RET.

Case $\dfrac{V(x) = \ell \qquad H(\ell) = (v_h, v_t) \qquad V[x_h \mapsto v_h, x_t \mapsto v_t], H \vdash e_2 \rightsquigarrow v, H' \mid q}{V, H \vdash \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto e_1 | \mathsf{cons}(x_h, x_t) \mapsto e_2 \rightsquigarrow v, H' \mid K_1^{matC} + q + K_2^{matC}}\,\text{E:MatCons}$

TS: $|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash \mathsf{match}\ x\ \mathsf{with}\ \mathsf{nil} \mapsto \{\mathsf{cbind}(\llparenthesis e_1 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matN}+K_2^{matN}}(\mathsf{cret}(r)))\}; \mathsf{cons} \mapsto \lambda x_h, x_t.\, \{\mathsf{cbind}(\llparenthesis e_2 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matC}+K_2^{matC}}(\mathsf{cret}(r)))\} : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K_1^{matC}+q+K_2^{matC}, K_1^{matC}+q+K_2^{matC}, r.r \doteq \|a\|)$ where $H' \vDash \ell \mapsto a : A$.

From $x : L(B)$, $V(x) = \ell$ and $H(v_h, v_t)$, we have $H \vDash V(x) \mapsto [a_h, a_1, \ldots, a_n]$, where $H \vDash v_h \mapsto a_h : B$ and $H \vDash v_t \mapsto \overbrace{[a_1, \ldots, a_n]}^{a_t} : L(B)$. Then from $x \doteq \|[a_h, a_1, \ldots, a_n]\|$, we have that $x \doteq \mathsf{cons}(\|a_h\|, \|a_t\|)$.

Hence, by applying Lemma 33 (note H', instead of H), STS:

$|\Sigma|, \|\Gamma\|; \lfloor\Gamma\rfloor_{V,H'} \vdash \lambda x_h, x_t.\, \{\mathsf{cbind}(\llparenthesis e_2 \rrparenthesis, \{r\}.\, \mathsf{cstep}_{K_1^{matC}+K_2^{matC}}(\mathsf{cret}(r)))\}\ \|a_h\|\ \|a_t\| : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, K_1^{matC} + q + K_2^{matC}, K_1^{matC} + q + K_2^{matC}, r.r \doteq \|a\|)$

By IH we have:

$|\Sigma|, \|\Gamma\|, x_h : |B|, x_t : \mathrm{list}_{|B|}; \lfloor\Gamma\rfloor_{V,H'}, x_h \doteq \|a_h\|, x_t \doteq \|a_t\| \vdash \llparenthesis e_2 \rrparenthesis : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, q, q, r.r \doteq \|a\|)$

The goal now follows by the U^{C} syntax-directed monadic rules, the rule U-ABS.

□

We will need the following lemmas to establish the RAML soundness result.

Lemma 35. The following hold:

- For all (semantic) values $a : A$ we have $\vdash_{\mathrm{L^C}} \forall p.\, \tilde{\Phi}_A(\|a\|, p) \Rightarrow \Phi(a : A) \doteq p$.
- Let $H \vDash V : \Gamma$. Then: $\|\Gamma\|; \lfloor\Gamma\rfloor_{V,H} \vdash_{\mathrm{L^C}} \Phi_{V,H}(\Gamma) \doteq \tilde{\Phi}(\Gamma)$.

Proof. **Proof of (1)** By induction on A.

Proof of (2) Follows from (1). □

Corollary 36 (RAML, soundness)**.** Let $H \vDash V : \Gamma$ and $\Sigma; \Gamma \vdash^{q}_{q'} e : A$. If $V \vdash e \rightsquigarrow v \mid p$ then $|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \lfloor\Gamma\rfloor_{V,E} \vdash_{\mathrm{L^C}} p \leq \Phi_{V,H}(\Gamma) + q - (\Phi_H(v : A) + q')$.

Proof. By applying Theorem 32 to the second assumption we have:

$$|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor \vdash \llparenthesis e \rrparenthesis : \mathbb{C}(|A|) \mid \exists p_r.\, \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \tilde{\Phi}(\Gamma) + q - q' - p_r, r.\tilde{\Phi}_A(r, p_r))$$

Hence, by translation to $\mathrm{L^C}$:

$$|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor \vdash_{\mathrm{L^C}} \exists n, v', p_r.\, \llparenthesis e \rrparenthesis \doteq \{\mathsf{cstep}_n(\mathsf{cret}(v'))\} \wedge \tilde{\Phi}_A(v, p_r) \wedge 0 \leq \tilde{\Phi}(\Gamma) + q - q' - p_r \leq n$$

By applying Theorem 34 to the first assumption and $V \vdash e \rightsquigarrow v \mid p$ (with weakening):

$$|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \lfloor\Gamma\rfloor_{V,H} \vdash \llparenthesis e \rrparenthesis : \mathbb{C}(|A|) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, p, p, r.r \doteq \|a\|)$$

where $H \vDash v \mapsto a : A$; and by translation to $\mathrm{L^C}$:

$$|\Sigma|, |\Gamma|; \lfloor\Sigma\rfloor, \lfloor\Gamma\rfloor, \lfloor\Gamma\rfloor_{V,H} \vdash_{\mathrm{L^C}} \exists n, v'.\, \llparenthesis e \rrparenthesis \doteq \mathsf{cstep}_n(\mathsf{cret}(v')) \wedge v' \doteq \|a\| \wedge p \leq n \leq p$$

Therefore, by an $\mathrm{L^C}$ axiom, $n = p \leq \tilde{\Phi}(\Gamma) + q - q' - p_r$ and $\Phi_H(v : A) = p_r$; hence by Lemma 35 $p \leq \Phi_{V,H}(\Gamma) + q - q' - \Phi_H(v : A)$, as required. □

6 Additional examples

In this section we consider some additional examples to those already discussed in the paper.

We point out that in the proofs for the examples we usually do not discuss how to apply the rules for basic connectives of L^{C} (e.g., introduction and elimination rules for $\Rightarrow$, $\wedge$, $\forall$), but instead focus on the application of U^{C} and R^{C} rules. Further, we also sometimes omit the applications of the U-SUBC and R-SUBC rules, when this is obvious.

6.1 List flattening (unary and relational)

In the following example we show that the cost of the *list flattening function* (i.e., the function that appends inner lists of a nested list) is the sum of the lengths of the inner lists. More precisely, we show that the *relational cost* of the flattening function is the difference between the sums of lengths of inner lists (for two nested lists of the same length).

We point out that this cost is more precise than the $n \cdot m$, where n is the length of the outer list, and m is the maximal length of any inner list.

Further, by using the relational reasoning we do not need to establish the precise lower and upper bound, hence the proof is simpler than a unary proof would be. However, as we discuss below, we establish a precise unary cost of the *list append* function (which the flattening function uses), since this proof is simpler in the unary setting.

We start by discussing the append function. First we define the list type, and the predicate Len that denotes the length of the list. Note that we omit the parentheses after the constructor nil that has no arguments:

$$\begin{array}{lcl} \mathrm{list}_\sigma & = & \mathsf{nil} + \mathsf{cons}(\sigma \times \mathrm{list}_\sigma) \\ \\ \forall \ell.\ \mathrm{Len}(\ell, 0) & \Leftrightarrow & \ell \doteq \mathsf{nil} \\ \forall \ell.\ \mathrm{Len}(\ell, n+1) & \Leftrightarrow & \exists h, t.\ \ell \doteq \mathsf{cons}(h, t) \wedge \mathrm{Len}(t, n) \end{array}$$

Here σ denotes any (elementary) type, which we omit when it is clear or not relevant.

Next we define the append function:

$$\begin{array}{l} \mathit{append} \triangleq \mathsf{rec}\, f(\ell_1).\lambda \ell_2. \\ \quad \mathsf{match}\ \ell_1\ \mathsf{with}\ \mathsf{nil} \mapsto \{\mathsf{cret}(\ell_2)\}; \mathsf{cons} \mapsto \lambda h, t.\ \{\mathsf{cbind}(f\ t\ \ell_2, \{z\}.\, \mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(h, z))))\} \end{array}$$

The function implementation is standard, except for the monadic constructs. Note that each recursive call incurs unit cost, denoted by cstep_1 after the recursive call $f\ t$.

For the append function, we show, using unary reasoning, that the cost is exactly the length of the first list. We establish this in unary setting because we need (below, for the flattening function) the cost for a pair of lists of different lengths, and therefore a relational proof would be more complicated. However, our framework allows us to use this unary result in the relational proof below (via the rule R-SPLIT), as we discuss below in the proof of the flattening function. Formally, we show:

$$\mathit{append} : \mathrm{list} \to \mathrm{list} \to \mathbb{C}(\mathrm{list}) \mid \forall \ell_1.\ \top \Rightarrow \forall \ell_2.\ \top \Rightarrow \forall n.\ \mathrm{Len}(\ell_1, n) \Rightarrow \mathbb{C}_{\mathrm{u}}(\mathbf{r}\ \ell_1\ \ell_2, n, n, \_.\top)$$

Note that we could easily show that the length of the resulting list is $n_1 + n_2$, however, this is information is not necessary for the cost of the flattening function.

Proof. First we apply the rules U-LETREC, U-ABS and U-MATCH. By LETREC we have the following inductive hypothesis (IH):

$$\forall m.\ |m| < |\ell_1| \Rightarrow \forall \ell_2.\ \top \Rightarrow \forall n.\ \mathrm{Len}(m, n) \Rightarrow \mathbb{C}_{\mathrm{u}}(f\ m\ \ell_2, n, n, \_.\top)$$

For the rule U-MATCH we need to consider two cases for ℓ_1.

Case $\ell_1 \doteq \mathsf{nil}$. From $\mathrm{Len}(\ell_1, n)$ we have $n \doteq 0$. We need to show:

$$\{\mathsf{cret}(\ell_2)\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \_.\top)$$

This follows by the rules U-MONAD and U-RET.

Case $\ell_1 \doteq \mathsf{cons}(h, t)$. From $\mathrm{Len}(\ell_1, n)$ we have $\mathrm{Len}(t, n')$, s.t. $n \doteq n' + 1$. We need to show:

$$\{\mathsf{cbind}(f\ t\ \ell_2, \{z\}.\, \mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(h, z))))\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, n, n, \_.\top)$$

By IH we have:

$$\mathbb{C}_{\mathrm{u}}(f\ t\ \ell_2, n', n', \_.\top)$$

Hence, by the rules U-MONAD and U-BIND, it remains to show:

$$\mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(h, z))) \div \mathrm{list} \mid 1 \mid 1 \mid \top$$

This follows by the rules U-STEP and U-RET.

Next we turn to the list flattening function. First we define the predicate LenDiff, where $\mathrm{LenDiff}(\ell_1, \ell_2, n)$ states that the difference of the sums of (inner) lengths of the lists ℓ_1 and ℓ_2 is n, and ℓ_1 and ℓ_2 have the same (outer) length:

$$\forall \ell_1, \ell_2, n.\ \mathrm{LenDiff}(\ell_1, \ell_2, n) \Leftrightarrow \begin{array}{l} (\ell_1 \doteq \ell_2 \doteq \mathsf{nil} \wedge n \doteq 0) \vee \\ (\exists h_1, h_2, t_1, t_2, n_1, n_2, n'.\ \ell_1 \doteq \mathsf{cons}(h_1, t_1) \wedge \\ \quad \ell_2 \doteq \mathsf{cons}(h_2, t_2) \wedge \mathrm{Len}(h_1, n_1) \wedge \mathrm{Len}(h_2, n_2) \wedge \\ \quad \mathrm{LenDiff}(t_1, t_2, n') \wedge n \doteq n_1 - n_2 + n') \end{array}$$

Then, we state the flattening function:

$$\begin{array}{rl} \mathit{flatten} \triangleq \mathsf{rec}\, f(\ell).\mathsf{match}\ \ell\ \mathsf{with} & \\ \mathsf{nil} \mapsto & \{\mathsf{cret}(\mathsf{nil})\}; \\ \mathsf{cons} \mapsto & \lambda h, t.\ \{\mathsf{cbind}(f\ t, \{y\}.\, \mathsf{cbind}(\mathit{append}\ h\ y, \{z\}.\, \mathsf{cstep}_1(\mathsf{cret}(z))))\} \end{array}$$

Same as above, this function is standard, except for the monadic constructs, and the cost is also incurred by each recursive call.

As discussed in the beginning we show that the relational cost is exactly the difference LenDiff; formally:

$$\mathit{flatten}_1 : \mathrm{list}_{\mathrm{list}_\sigma} \to \mathbb{C}(\mathrm{list}_\sigma) \sim \mathit{flatten}_2 : \mathrm{list}_{\mathrm{list}_\sigma} \to \mathbb{C}(\mathrm{list}_\sigma) \mid \forall \ell_1, \ell_2.\ \top \Rightarrow$$

$$\forall n.\ \mathrm{LenDiff}(\ell_1, \ell_2, n) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1\ \ell_1, \mathbf{r}_2\ \ell_2, n, \_.\_.\top)$$

(where $\mathit{flatten}_i$ is obtained from *flatten* by replacing each variable x with x_i, for $i \in \{1, 2\}$).

Proof. First we apply the rules R-LETREC and R-MATCH. By R-LETREC we have the following IH:

$$\forall m_1, m_2.\ (|m_1|, |m_2|) < (|\ell_1|, |\ell_2|) \Rightarrow \forall n.\ \mathrm{LenDiff}(m_1, m_2, n) \Rightarrow \mathbb{C}_{\mathrm{r}}(f_1\ m_1, f_2\ m_2, n, \_.\_.\top)$$

For the rule R-MATCH we need to consider four cases (for all pairs of the constructors of ℓ_1 and ℓ_2). However, when $\ell_1 \doteq \mathsf{nil}$ and $\ell_2 \doteq \mathsf{cons}(h_2, t_2)$, from $\mathrm{LenDiff}(\ell_1, \ell_2, n)$ we obtain the contradiction, and similarly for the opposite case of different constructors. Hence, we consider

the remaining two cases.

Case $\ell_1 \doteq \ell_2 \doteq \mathsf{nil}$. From $\mathrm{LenDiff}(\ell_1, \ell_2, n)$ we have $n \doteq 0$. We need to show:

$$\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}) \sim \{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \_.\_.\top)$$

This follows by the rules R-MONAD and R-RET.

Case $\ell_1 \doteq \mathsf{cons}(h_1, t_1)$ and $\ell_2 \doteq \mathsf{cons}(h_2, t_2)$. From $\mathrm{LenDiff}(\ell_1, \ell_2, n)$ we have $\mathrm{Len}(h_1, n_1)$, $\mathrm{Len}(h_2, n_2)$, and $\mathrm{LenDiff}(t_1, t_2, n')$ s.t. $n \doteq n_1 - n_2 + n'$. We need to show:

$$\{\mathsf{cbind}(f_1\ t_1, \{y_1\}.\, \mathsf{cbind}(\mathit{append}\ h_1\ y_1, \{z_1\}.\, \mathsf{cstep}_1(\mathsf{cret}(z_1))))\} : \mathbb{C}(\mathrm{list}) \sim$$

$$\{\mathsf{cbind}(f_2\ t_2, \{y_2\}.\, \mathsf{cbind}(\mathit{append}\ h_2\ y_2, \{z_2\}.\, \mathsf{cstep}_1(\mathsf{cret}(z_2))))\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n, \_.\_.\top)$$

By IH we have:

$$\mathbb{C}_{\mathrm{r}}(f_1\ t_1, f_2\ t_2, n', \_.\_.\top)$$

Hence, by the rules R-MONAD, R-BIND, and R-SUBC with $(n_1 - n_2) + n' \leq n$, it remains to show:

$$\mathsf{cbind}(\mathit{append}\ h_1\ y_1, \{z_1\}.\, \mathsf{cstep}_1(\mathsf{cret}(z_1))) \div \mathrm{list} \sim$$

$$\mathsf{cbind}(\mathit{append}\ h_2\ y_2, \{z_2\}.\, \mathsf{cstep}_1(\mathsf{cret}(z_2))) \div \mathrm{list} \mid n_1 + n_2 \mid \top$$

Now, using the result obtained above for *append* we have:

$$\mathit{append}\ h_i\ y_i : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, n_i, n_i, \_.\top)$$

for $i \in \{1, 2\}$. Then by the rule R-SPLIT-PURE we have:

$$\mathit{append}\ h_1\ y_1 : \mathbb{C}(\mathrm{list}) \sim \mathit{append}\ h_2\ y_2 : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, n_1 - n_2, \_.\_.\top)$$

The resulting goal then follows by the rules R-BIND and R-RET.

6.2 Red-black tree search (unary)

In this example we show the unary cost of a search function (defined below) on the *red-black* trees (also defined below). A red-black tree is a self-balancing binary tree that has two types of nodes (red and black) with the following invariants: (1) The root and all leaves are black; (2) If a node is red, then both of its children are black; (3) Every path from a given node to any of its descendant leafs contains the same number of black nodes.

These constraints enforce a key property of red-black trees: *the longest path from the root to any leaf is at most as twice long as the shortest path from the root to any leaf.* The shortest path consists of all black nodes, and this is called the *black-height*, while the longest path alternates black and red nodes. Hence, our goal is to show that the cost of the search is lower bounded by h, and upper bounded by $2 \cdot h$, where h is the black-height.

First we define the red-black tree data type rbtree and the black-height predicate bheight (where σ_D and σ_K are types of leaf data and node keys, respectively):

$$\begin{array}{rcl} \mathrm{rbtree} & = & \mathsf{leaf}(\sigma_D) + \mathsf{rnode}(\mathrm{rbtree} \times \sigma_K \times \mathrm{rbtree}) \\ & & \quad + \mathsf{bnode}(\mathrm{rbtree} \times \sigma_K \times \mathrm{rbtree}) \end{array}$$

$$\begin{array}{rcl} \forall t.\ \mathrm{bheight}(t, 0, \mathsf{tt}) & \Leftrightarrow & \exists d.\ t \doteq \mathsf{leaf}(d) \\ \forall t, h.\ \mathrm{bheight}(t, h, \mathsf{ff}) & \Leftrightarrow & \exists t_1, t_2, k.\ t \doteq \mathsf{rnode}(t_1, k, t_2) \wedge \mathrm{bheight}(t_1, h, \mathsf{tt}) \wedge \mathrm{bheight}(t_2, h, \mathsf{tt}) \\ \forall t, h.\ \mathrm{bheight}(t, h+1, \mathsf{tt}) & \Leftrightarrow & \exists t_1, t_2, k, b_1, b_2.\ t \doteq \mathsf{bnode}(t_1, k, t_2) \wedge \mathrm{bheight}(t_1, h, b_1) \\ & & \quad \wedge\ \mathrm{bheight}(t_2, h, b_2) \end{array}$$

The predicate $\mathrm{bheight}(t, h, b)$ defines that a tree t has black-height h, and a Boolean b is true if t is a black node, and false otherwise.

Next we define the search function:

$$
\begin{array}{l}
\mathit{search} \triangleq \lambda g.\ \mathsf{rec}\, f(t). \\
\quad \mathsf{match}\ t\ \mathsf{with} \\
\qquad \mathsf{leaf} \mapsto \quad \lambda d.\ \{\mathsf{cret}(d)\}; \\
\qquad \mathsf{rnode} \mapsto \\
\qquad \mathsf{bnode} \mapsto \quad \lambda t_1, k, t_2.\ \{\mathsf{cbind}(g\ k, \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x\ \mathsf{with} \quad \mathsf{le} \mapsto (f\ t_1); \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathsf{ge} \mapsto (f\ t_2), \{y\}.\, \mathsf{cret}(y)))\}
\end{array}
$$

Where g is a comparison function, such that $g\ k$ (where k is a node's key) returns le (less) or ge (greater) and incurs a unit of cost (this is the only place where cost is incurred). The *search* function uses g to decide which sub-tree to traverse, and returns the data of the leaf (when the search reaches leaf).

Now we show that for any black tree t, such that, $\mathrm{bheight}(t, h, \mathsf{tt})$ the cost of *search* $g\ t$ is bounded by h and $2 \cdot h$. Actually, for the proof we need the following stronger property:

$$\mathit{search} : (\sigma_K \to \mathrm{cmp}) \to \mathrm{rbtree} \to \mathbb{C}(\sigma_D) \mid \forall g.\ (\forall k.\ \top \Rightarrow \mathbb{C}_{\mathrm{u}}(g\ k, 1, 1, \_.\top)) \Rightarrow \forall t.\ \top \Rightarrow$$

$$\forall h, b.\ \mathrm{bheight}(t, h, b) \Rightarrow \mathbb{C}_{\mathrm{u}}(\mathbf{r}\ g\ t, h, 2 \cdot h + (b\ ?\ 0 : 1), \_.\top)$$

Where $b\ ?\ 0 : 1$ evaluates to 0 when b is tt, and to 1 otherwise, and cmp is defined with the equation $\mathrm{cmp} = \mathsf{le} + \mathsf{ge}$.

Proof. We use U^{C} rules U-ABS, U-LETREC and U-MATCH. Hence we need to consider the three cases for t.

Case $t \doteq \mathsf{leaf}(d)$. From $\mathrm{bheight}(t, n, b)$ we have $n \doteq 0$ and $b \doteq \mathsf{tt}$. We need to show:

$$\{\mathsf{cret}(d)\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \_.\top)$$

This follows trivially by the U^{C} monadic rules.

Case $t \doteq \mathsf{rnode}(t_1, k, t_2)$. From $\mathrm{bheight}(t, h, b)$ we have $b \doteq \mathsf{ff}$, $\mathrm{bheight}(t_1, h, \mathsf{tt})$ and $\mathrm{bheight}(t_2, h, \mathsf{tt})$. We need to show:

$$\{\mathsf{cbind}(g\ k, \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x\ \mathsf{with}\ \mathsf{le} \mapsto (f\ t_1); \mathsf{ge} \mapsto (f\ t_2), \{y\}.\, \mathsf{cret}(y)))\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, h, 2{\cdot}h{+}1, \_.\top)$$

From the assumption about g we have:

$$g\ k : \mathbb{C}(\mathrm{cmp}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 1, 1, \_.\top)$$

From IH we have:

$$f\ t_i : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, h, 2 \cdot h, \_.\top)$$

for $i \in \{1, 2\}$. The conclusion follows by the rules of U^{C}; note that we have to use the rule U-SUBC to show lower bound $h \leq h + 1$.

Case $t \doteq \mathsf{bnode}(t_1, k, t_2$. From $\mathrm{bheight}(t, h, b)$ we have $b \doteq \mathsf{tt}$, $\mathrm{bheight}(t_1, h', b_1)$ and $\mathrm{bheight}(t_2, h', b_2)$ for some h, b_1, b_2, s.t. $h \doteq h' + 1$. We need to show:

$$\{\mathsf{cbind}(g\ k, \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x\ \mathsf{with}\ \mathsf{le} \mapsto (f\ t_1); \mathsf{ge} \mapsto (f\ t_2), \{y\}.\, \mathsf{cret}(y)))\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, h, 2{\cdot}h, \_.\top)$$

From the assumption about g we have:

$$g\ k : \mathbb{C}(\text{cmp}) \mid \mathbb{C}_{\text{u}}(\mathbf{r}, 1, 1, \_.\top)$$

From IH we have:

$$f\ t_i : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\text{u}}(\mathbf{r}, h', 2 \cdot h' + (b_i\ ?\ 0 : 1), \_.\top)$$

for $i \in \{1, 2\}$. The conclusion follows by the rules of U^{C}, and additionally showing, using the rule U-SUBC, that $h \leq h' + 1$ and $2 \cdot h' + (b_i\ ?\ 0 : 1) \leq 2 \cdot h$. This holds since $h \doteq h' + 1$.

6.3 Balanced binary tree search (relational)

In this example we show that the cost of a search function (defined below) on the *balanced binary trees* (also defined below) depends only on the *height of the tree*. More precisely, we show that given two trees, if the left tree has smaller depth, the relational cost is bounded by zero.

The setting is very similar to the setting of the previous example. We define the balance binary tree tree type and the predicate bheight that defines *balanced* height (where σ_D and σ_K are, as in the previous examples, types of leaf data and node keys, respectively):

$$\text{tree} \quad = \quad \mathsf{leaf}(\sigma_D) + \mathsf{node}(\text{tree} \times \sigma_K \times \text{tree})$$

$$\begin{array}{rcl} \forall t.\ \text{bheight}(t, 0) & \Leftrightarrow & \exists d.\ t \doteq \mathsf{leaf}(d) \\ \forall t, h.\ \text{bheight}(t, h+1) & \Leftrightarrow & \exists t_1, t_2, k.\ t \doteq \mathsf{node}(t_1, k, t_2) \wedge \text{bheight}(t_1, h) \wedge \text{bheight}(t_2, h) \end{array}$$

The predicate $\text{bheight}(t, h)$ defines that a tree t has balanced height h.

Next we define the search function, in almost the identical way as in the previous example:

$$\begin{array}{l} \mathit{search} \triangleq \lambda g.\ \mathsf{rec}\, f(t). \\ \quad \mathsf{match}\ t\ \mathsf{with} \\ \qquad \mathsf{leaf} \mapsto \lambda d.\ \{\mathsf{cret}(d)\}; \\ \qquad \mathsf{node} \mapsto \lambda t', k, t''.\ \{\mathsf{cbind}(g\ k, \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x\ \mathsf{with} \quad \mathsf{le} \mapsto (f\ t'); \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathsf{ge} \mapsto (f\ t''), \{y\}.\, \mathsf{cret}(y)))\} \end{array}$$

Where g is a comparison function same as in the previous example.

Now we show that the relational cost of the function *search* applied to two lists, where the left one has smaller height, is 0. Formally:

$$\mathit{search}_1 : (\sigma_K \to \mathbb{C}(\text{cmp})) \to \text{tree} \to \mathbb{C}(\sigma_D) \sim \mathit{search}_2 : (\sigma_K \to \mathbb{C}(\text{cmp})) \to \text{tree} \to \mathbb{C}(\sigma_D) \mid$$

$$\forall g_1, g_2.\ g_1 \doteq g_2 \wedge (\forall k.\ \top \Rightarrow \mathbb{C}_{\text{u}}(g_1\ k, 1, 1, \_.\top)) \Rightarrow \forall t_1, t_2.\ \top \Rightarrow \forall h_1, h_2.\ \text{bheight}(t_1, h_1) \wedge$$

$$\text{bheight}(t_2, h_2) \wedge h_1 \leq h_2 \Rightarrow \mathbb{C}_{\text{r}}(\mathbf{r}_1\ g_1\ t_1, \mathbf{r}_2\ g_2\ t_2, 0, \_.\_.\top)$$

We use the following lemma without proof:

$$\forall g, t.\ \mathbb{C}_{\text{u}}(\mathit{search}\ g\ t, 0, \infty, \_.\top)$$

This lemma states that the unary cost of *search* is positive (which is trivially true, since there is nothing to incur negative cost).

Proof. We use the rules R-ABS, R-LETREC, R-MATCH. We need to consider four cases for t_1 and t_2.

Case $t_1 \doteq \mathsf{leaf}(d_1)$ and $t_2 \doteq \mathsf{leaf}(d_2)$. We need to show:

$$\{\mathsf{cret}(d_1)\} : \mathbb{C}(\sigma_D) \sim \{\mathsf{cret}(d_1)\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \_.\_.\top)$$

This follows trivially by the rules of R^{C}.

Case $t_1 \doteq \mathsf{leaf}(d_1)$ and $t_2 \doteq \mathsf{node}(t_2', k_2, t_2'')$. We need to show:

$$\{\mathsf{cret}(d_1)\} : \mathbb{C}(\sigma_D) \sim \{\mathsf{cbind}(g_2\ k_2, \{x\}.\, \mathsf{cbind}(\mathsf{match}\ x_2\ \mathsf{with}\ \mathsf{le} \mapsto (f_2\ t_2'); \mathsf{ge} \mapsto$$

$$(f_2\ t_2''), \{y_2\}.\, \mathsf{cret}(y_2)))\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \_.\_.\top)$$

We trivially have

$$\{\mathsf{cret}(d_1)\} : \mathbb{C}(\sigma_D) \sim \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \_.\top)$$

Call this statement (A). From the assumption about g_2 we have:

$$g_2\ k_2 : \mathbb{C}(\mathrm{cmp}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 1, 1, \_.\top)$$

and from the lemma state above:

$$f_2\ t_2' : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \infty, \_.\top)$$

and the same for $f_2\ t_2''$. Therefore we have:

$\{\mathsf{cbind}(g_2\ k_2, \{x_2\}.\, \mathsf{cbind}(\mathsf{match}\ x_2\ \mathsf{with}\ \mathsf{le} \mapsto (f_2\ t_2'); \mathsf{ge} \mapsto (f_2\ t_2''), \{y_2\}.\, \mathsf{cret}(y_2)))\} : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \infty + 1, \_.\top)$

Call this statement (B). The required goal now follows by the R-SPLIT-PURE.

Case $t_1 \doteq \mathsf{node}(t_1', k_1, t_1'')$ and $t_2 \doteq \mathsf{leaf}(d_2)$. From $\mathrm{bheight}(t_1, h_1)$ we have $h_1 \doteq h_1' + 1$ for some $h_1' \in \mathbb{N}$. From $\mathrm{bheight}(t_2, h_2)$ we have $h_2 \doteq 0$. Hence, from $h_1 \leq h_2$ we have a contradiction, therefore this case is trivially done.

Case $t_1 \doteq \mathsf{node}(t_1', k_1, t_1'')$ and $t_2 \doteq \mathsf{node}(t_2', k_2, t_2'')$. We need to show:

$$\{\mathsf{cbind}(g_1\ k_1, \{x_1\}.\, \mathsf{cbind}(\mathsf{match}\ x_1\ \mathsf{with}\ \mathsf{le} \mapsto (f_1\ t_1'); \mathsf{ge} \mapsto (f_1\ t_1''), \{y_1\}.\, \mathsf{cret}(y_1)))\} : \mathbb{C}(\sigma_D) \sim$$

$$\{\mathsf{cbind}(g_2\ k_2, \{x_2\}.\, \mathsf{cbind}(\mathsf{match}\ x_2\ \mathsf{with}\ \mathsf{le} \mapsto (f_2\ t_2'); \mathsf{ge} \mapsto (f_2\ t_2''), \{y_2\}.\, \mathsf{cret}(y_2)))\} : \mathbb{C}(\sigma_D) \mid$$

$$\mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, 0.\_.\top)$$

From the assumptions about g_1 and g_2 and the rule R-SPLIT-PURE we have:

$$g_1\ k_1 : \mathbb{C}(\mathrm{cmp}) \sim g_2\ k_2 : \mathbb{C}(\mathrm{cmp}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \_.\_.\top)$$

From $\mathrm{bheight}(t_i, h_i)$ we have $\mathrm{bheight}(t_i', h_i')$ and $\mathrm{bheight}(t_i'', h_i')$, s.t. $h_i \doteq h_i' + 1$, for $i \in \{1, 2\}$. Hence, from $h_1 \leq h_2$ we have $h_1' \leq h_2'$. Then by IH we have:

$$f_1\ t_1' : \mathbb{C}(\sigma_D) \sim f_2\ t_2' : \mathbb{C}(\sigma_D) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \_.\_.\top)$$

and similar for the other pairs: (t_1', t_2''), (t_1'', t_2'), and (t_1'', t_2''). Therefore, the required goal follows by the R^{C} rules.

6.4 Lookup in random-access list (unary)

The following example explains the cost analysis of a data structure with a nontrivial invariant. The data structure is random access lists, described in Xi (1999). These are lists represented as trees, with a size invariant that makes them highly balanced. This allows fast (log-time) lookup, hence, the name "random access".

Formally, the datatype and the invariant are defined with:

$$\text{rlist} = \mathsf{nil} + \mathsf{one}(\sigma) + \mathsf{even}(\text{rlist} \times \text{rlist}) + \mathsf{odd}(\text{rlist} \times \text{rlist}) \qquad \forall \ell, n.\ \text{size}(\ell, n) \Leftrightarrow (n \doteq 0 \wedge \ell \doteq \mathsf{nil}) \vee$$

$$(\exists x.\ n \doteq 1 \wedge \ell \doteq \mathsf{one}(x)) \vee (\exists \ell_1, \ell_2, n'.\ n > 0 \wedge \ell \doteq \mathsf{even}(\ell_1, \ell_2) \wedge \text{size}(\ell_1, n') \wedge \text{size}(\ell_2, n') \wedge$$

$$n \doteq 2 \cdot n') \vee (\exists \ell_1, \ell_2, n'.\ n > 0 \wedge \ell \doteq \mathsf{odd}(\ell_1, \ell_2) \wedge \text{size}(\ell_1, n' + 1) \wedge \text{size}(\ell_2, n') \wedge n \doteq 2 \cdot n' + 1)$$

The type rlist defines lists of σs. A list can be empty (nil), hold one item of data (one), or consist of two sub-lists (even or odd). The invariant, captured by the size predicate, is that in the even case both of the sub-lists must be of the same size, and in the odd case the sizes of the sub-lists must differ by exactly one. This makes the data structure very balanced.

Next, we define a *lookup* function on rlist that returns the i^{th} element from the list ℓ. The function exploits the invariant, and runs in $\mathcal{O}(\log_2 n)$ time, where n is the size of the list.

$$\begin{array}{l}
\mathit{lookup} \triangleq \mathsf{rec}\, f(\ell).\lambda i. \\
\quad \mathsf{match}\ \ell\ \mathsf{with} \\
\qquad \begin{array}{rl}
\mathsf{one} & \mapsto \lambda x.\ \{\mathsf{cret}(x)\} \\
\mathsf{even} & \mapsto \lambda \ell_1, \ell_2.\ \mathsf{match}\ i \bmod 2 = 0\ \mathsf{with} \\
 & \qquad \mathsf{tt} \mapsto \{\mathsf{cbind}(f\ \ell_1\ (i/2), \{x\}.\, \mathsf{cstep}_1(\mathsf{cret}(x)))\} \\
 & \qquad \mathsf{ff} \mapsto \{\mathsf{cbind}(f\ \ell_2\ (i/2), \{x\}.\, \mathsf{cstep}_1(\mathsf{cret}(x)))\} \\
\mathsf{odd} & \mapsto \lambda \ell_1, \ell_2.\ \mathsf{match}\ i \bmod 2 = 0\ \mathsf{with} \\
 & \qquad \mathsf{tt} \mapsto \{\mathsf{cbind}(f\ \ell_1\ (i/2), \{x\}.\, \mathsf{cstep}_1(\mathsf{cret}(x)))\} \\
 & \qquad \mathsf{ff} \mapsto \{\mathsf{cbind}(f\ \ell_2\ ((i-1)/2), \{x\}.\, \mathsf{cstep}_1(\mathsf{cret}(x)))\}
\end{array}
\end{array}$$

lookup recurs on the bit-representation of the index i. If a bit is 0, it goes left, else it goes right. A cost is incurred for every bit. Formally, we show the following:

$$\vdash \mathit{lookup} : \text{rlist} \to \mathbb{N} \to \mathbb{C}(\sigma) \mid \forall \ell.\ \top \Rightarrow \forall i.\ \top \Rightarrow \forall n.\ \text{size}(\ell, n) \wedge i < n \Rightarrow$$

$$\mathbb{C}_{\mathrm{u}}(\mathbf{r}\ \ell\ i, \lfloor \log_2 n \rfloor, \lceil \log_2 n \rceil, \_.\top)$$

In words, the property says that the cost of *lookup* lies between $\lfloor \log_2 n \rfloor$ and $\lceil \log_2 n \rceil$ where n is the size of ℓ. The proof follows the syntax of *lookup*.

Proof. We first apply the U-LETREC, U-ABS and U-MATCH rules. We need to consider four cases (for each rlist's constructor one).

Case $\ell \doteq \mathsf{nil}$. From $\text{size}(\mathsf{nil}, n)$ we have $n \doteq 0$, and from $i < n$, we have a contradiction, hence this case is done.

Case $\ell \doteq \mathsf{one}(x)$. From $\text{size}(\mathsf{one}(x), n)$ we have $n \doteq 1$. We need to show:

$$\{\mathsf{cret}(x)\} : \mathbb{C}(\sigma) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \_.\top)$$

This follows by the rules U-MONAD, U-RET, and U-VAR.

Case $\ell \doteq \mathsf{even}(\ell_1, \ell_2)$. From $\mathrm{size}(\mathsf{even}(\ell_1, \ell_2), n)$ we have $\mathrm{size}(\ell_1, n')$ and $\mathrm{size}(\ell_2, n')$, s.t. $n \doteq 2 \cdot n'$ and $n' > 0$. We use U-MATCH on $i \bmod 2 \doteq 0$, and we need to distinguish two cases, however, they are completely symmetric, so we show one (we expand $n \doteq 2 \cdot n'$):

$$\vdash \{\mathsf{cbind}(f\ \ell_1\ (i/2), \{x\}.\, \mathsf{cstep}_1(\mathsf{cret}(x)))\} : \mathbb{C}(\sigma) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, \lfloor \log_2(2 \cdot n') \rfloor, \lceil \log_2(2 \cdot n') \rceil, \_.\top)$$

Note that we have $(i/2) < n'$, hence by IH we have:

$$\vdash f\ \ell_1\ (i/1) : \mathbb{C}(\sigma) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, \lfloor \log_2 n' \rfloor, \lceil \log_2 n' \rceil, \_.\top)$$

By the rules U-BIND and U-SUBC it remains to show:

$$\vdash \mathsf{cstep}_1(\mathsf{cret}(x)) \div \sigma \mid 1 \mid 1 \mid \top$$

and inequalities:

$$1 + \lfloor \log_2 n' \rfloor \doteq \lfloor \log_2(2 \cdot n') \rfloor \leq \lfloor \log_2(2 \cdot n') \rfloor$$

and

$$1 + \lceil \log_2 n' \rceil \doteq \lceil \log_2(2 \cdot n') \rceil \leq \lceil \log_2(2 \cdot n') \rceil$$

The first statement follows by the rules U-STEP and U-RET, and the inequalities hold trivially.

Case $\ell \doteq \mathsf{odd}(\ell_1, \ell_2)$. From $\mathrm{size}(\mathsf{odd}(\ell_1, \ell_2), n)$ we have $\mathrm{size}(\ell_1, n' + 1)$ and $\mathrm{size}(\ell_2, n')$, s.t. $n \doteq 2 \cdot n' + 1$ and $n' > 0$.

The syntax-directed reasoning is almost the same as in the previous case, but we need to show the following inequalities:

1. $\lfloor \log_2(2 \cdot n' + 1) \rfloor \leq 1 + \lfloor \log_2(n' + 1) = \lfloor \log_2(2 \cdot n' + 2) \rfloor \rfloor$
2. $1 + \lceil \log_2(n' + 1) \rceil = \lceil \log_2(2 \cdot n' + 2) \rceil \leq \lceil \log_2(2 \cdot n' + 1) \rceil$
3. $\lfloor \log_2(2 \cdot n' + 1) \rfloor \leq 1 + \lfloor \log_2 n' \rfloor = \lfloor \log_2(2 \cdot n') \rfloor$
4. $1 + \lceil \log_2 n' \rceil = \lceil \log_2(2 \cdot n') \rceil \leq \lceil \log_2(2 \cdot n' + 1) \rceil$

The inequalities (1) and (4) hold trivially. We discuss (2) and (3).

Let $k \geq 1$ be such that $2^k < 2 \cdot n' + 1 < 2^{k+1}$ (the inequalities are strict and such a k exists, since $2 \cdot n' + 1$ is odd and $n' > 0$). Then we also have $2^k \leq 2 \cdot n' < 2^{k+1}$ and $2^k < 2 \cdot n' + 2 \leq 2^{k+1}$. Then $k < \log_2(2 \cdot n' + 1) < k + 1$, $k \leq \log_2(2 \cdot n') < k + 1$, and $k < \log_2(2 \cdot n' + 2) \leq k + 1$. Hence $\lfloor \log_2(2 \cdot n') \rfloor = \lfloor \log_2(2 \cdot n' + 1) \rfloor = k$, and $\lceil \log_2(2 \cdot n' + 2) \rceil = \lceil \log_2(2 \cdot n' + 1) \rceil = k + 1$.

6.5 Minimum list element using the insertion sort (unary lazy data-structures)

The following example is inspired by a very similar example in Danielsson (2008), and shows unary cost reasoning on lazy lists. We show the cost of taking the *minimum element of a list*, by first running the *insertion sort* and then taking the *first list element*: in a lazy setting, this has cost n, where n is the length of the list.

First, we define the *lazy list type* $\mathrm{list}_{\mathbf{L}}$ and the refinement (predicate) $\mathrm{len}_{\mathbf{L}}$:

$$\mathrm{list}_{\mathbf{L}} \quad = \quad \mathsf{nil} + \mathsf{cons}(\mathbb{Z} \times \mathbb{C}(\mathrm{list}_{\mathbf{L}}))$$

$$\begin{aligned} \forall \ell, c.\ \mathrm{len}_{\mathbf{L}}(\ell, c, 0) &\Leftrightarrow \ell \doteq \mathsf{nil} \\ \forall \ell, c, n.\ \mathrm{len}_{\mathbf{L}}(\ell, c, n+1) &\Leftrightarrow \exists h, t.\ \ell \doteq \mathsf{cons}(h, t) \wedge \mathbb{C}_{\mathbf{u}}(t, 0, c, t.\mathrm{len}_{\mathbf{L}}(t, c, n)) \end{aligned}$$

Note that the tail of the list is a monadic $\mathbb{C}(-)$ type. The refinement $\mathrm{len}_{\mathbf{L}}(\ell, c, n)$ denotes that a lazy list ℓ, of length n, where c is an *upper bound* on the *cost of forcing* each tail of ℓ. This closely mimics the development of Danielsson (2008).

Next, we define the functions that we analyze: $insert_{\mathbf{L}}$ (lazy insert into a sorted list, part of the insertion sort), $isort_{\mathbf{L}}$ (lazy insertion sort), and $min_{\mathbf{L}}$ (lazy min function):

$$
\begin{array}{l}
insert_{\mathbf{L}} \triangleq \lambda x.\mathsf{rec}\, f(\ell). \\
\quad \mathsf{match}\ \ell\ \mathsf{with} \\
\quad\quad \mathsf{nil} \mapsto \{\mathsf{cret}(\mathsf{cons}(x, \{\mathsf{cret}(\mathsf{nil})\}))\}; \\
\quad\quad \mathsf{cons} \mapsto \lambda h, t. \\
\quad\quad\quad \{\mathsf{cbind}(x \leq h, \{b\}.\mathsf{cbind}(\\
\quad\quad\quad\quad \mathsf{match}\ b\ \mathsf{with} \\
\quad\quad\quad\quad\quad \mathsf{tt} \mapsto \{\mathsf{cret}(\mathsf{cons}(x, \{\mathsf{cret}(\mathsf{cons}(h, t))\}))\}; \\
\quad\quad\quad\quad\quad \mathsf{ff} \mapsto \{\mathsf{cret}(\mathsf{cons}(h, \{\mathsf{cbind}(t, \{t'\}.\, \mathsf{cbind}(f\ t', \{t''\}.\, \mathsf{cret}(t'')))\}))\}, \\
\quad\quad\quad\quad \{z\}.\, \mathsf{cret}(z)))\}
\end{array}
$$

$$
\begin{array}{l}
isort_{\mathbf{L}} \triangleq \mathsf{rec}\, f(\ell). \\
\quad \mathsf{match}\ \ell\ \mathsf{with} \\
\quad\quad \mathsf{nil} \mapsto \{\mathsf{cret}(\mathsf{nil})\}; \\
\quad\quad \mathsf{cons} \mapsto \lambda h, t.\ \{\mathsf{cbind}(f\ t, \{t'\}.\, \mathsf{cbind}(insert_{\mathbf{L}}\ h\ t', \{t''\}.\, \mathsf{cret}(t'')))\}
\end{array}
$$

$$
min_{\mathbf{L}} \triangleq \lambda \ell.\ \{\mathsf{cbind}(isort_{\mathbf{L}}\ \ell, \{\ell'\}.\, \mathsf{cret}(\mathsf{match}\ \ell'\ \mathsf{with}\ \ \mathsf{cons} \mapsto \lambda h, t.\ h))\}
$$

The functions have the standard implementations, except for the monadic constructs. The cost is incurred only by the comparison operation $\leq$ in $insert_{\mathbf{L}}$, for which we assume the following:

$$
\leq : \mathbb{Z} \to \mathbb{Z} \to \mathbb{C}(\mathrm{bool}) \mid \forall x_1, x_2.\ \mathbb{C}_{\mathbf{u}}(\mathbf{r}\ x_1\ x_2, 1, 1, \_.\top)
$$

Note that here we do not reason about the values of the list, hence we ignore the functional properties of $\leq$. We point out that in Danielsson (2008) all operations incur cost; we could do the same, but decided on this simpler setting for easier presentation.

We start by showing the following claim about the $insert_{\mathbf{L}}$ function:

$$
insert_{\mathbf{L}} : \mathbb{Z} \to \mathrm{list}_{\mathbf{L}} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \forall x.\ \top \Rightarrow \forall \ell.\ \top \Rightarrow \forall n.\ \mathrm{len}_{\mathbf{L}}(\ell, n, n) \Rightarrow \mathbb{C}_{\mathbf{u}}(\mathbf{r}\ x\ \ell, 0, 1, r.\mathrm{len}_{\mathbf{L}}(r, n{+}1, n{+}1))
$$

Informally, the claim states than for an input list ℓ of length n, with the tail-forcing cost n, the result will incur cost 1 and the resulting list will be of length $n+1$, with the tail-forcing cost $n+1$.

Proof. We first use the rules U-ABS, U-LETREC, and U-MATCH. For U-MATCH we have to consider two cases.

Case $\ell \doteq \mathsf{nil}$. From $\mathrm{len}_{\mathbf{L}}(\ell, n, n)$ we have $n \doteq 0$. We need to show:

$$
\{\mathsf{cret}(\mathsf{cons}(x, \{\mathsf{cret}(\mathsf{nil})\}))\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, 0, 1, \mathbf{r}.\mathrm{len}_{\mathbf{L}}(\mathbf{r}, 1, 1))
$$

By the rules U-MONAD, U-RET, and U-SUBC, and L^{C} (to show $\mathrm{len}_{\mathbf{L}}(\mathsf{nil}, 1, 0)$) we have:

$$
\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, 0, 1, \mathbf{r}.\mathrm{len}_{\mathbf{L}}(\mathbf{r}, 1, 0))
$$

By the similar reasoning we also obtain the required goal.

$\underline{\ell \doteq \mathsf{cons}(h,t).}$ From $\mathrm{len}_{\mathbf{L}}(\ell, n, n)$ we have $\mathbb{C}_{\mathrm{u}}(t, 0, n, t'.\mathrm{len}_{\mathbf{L}}(t', n, n'))$ and $n \doteq n' + 1$. Since we have $\mathbb{C}_{\mathrm{u}}(x \leq h, 1, 1, \_.\top)$, by the rules U-MONAD, U-RET, and U-BIND, and the rule U-SUBC, it remains to show:

$$\mathsf{match}\ b\ \mathsf{with}\ \cdots : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \mathbf{r}.\mathrm{len}_{\mathbf{L}}(\mathbf{r}, n+1, n+1))$$

By the rule U-MATCH we need to consider two sub-cases. The sub-case $b \doteq \mathsf{tt}$ is similar to the nil case above, hence we discuss only the sub-case $b \doteq \mathsf{ff}$.

After applying the rules U-MONAD and U-RET, it remains to show:

$$\mathsf{cons}(h, \{\mathsf{cbind}(t, \{t'\}.\, \mathsf{cbind}(f\ t', \{t''\}.\, \mathsf{cret}(t'')))\}) \mid \mathrm{len}_{\mathbf{L}}(\mathbf{r}, n+1, n+1)$$

This reduces to:

$$\mathsf{cbind}(t, \{t'\}.\, \mathsf{cbind}(f\ t', \{t''\}.\, \mathsf{cret}(t''))) \mid 0 \mid n+1 \mid \mathrm{len}_{\mathbf{L}}(\mathbf{r}, 0, n+1, n)$$

Using the assumption $\mathbb{C}_{\mathrm{u}}(t, 0, n, t'.\mathrm{len}_{\mathbf{L}}(t', n, n'))$ and the rule U-BIND, it remains to show:

$$\mathsf{cbind}(f\ t', \{t''\}.\, \mathsf{cret}(t'')) \mid 0 \mid 1 \mid \mathrm{len}_{\mathbf{L}}(\mathbf{r}, 0, n+1, n)$$

By IH we have:

$$f\ t' \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 1, t''.\mathrm{len}_{\mathbf{L}}(t'', n+1, n'+1))$$

Since $n' + 1 \doteq n$, the result follows by the rules U-BIND and U-RET.

For the insertion sort we show the following claim:

$$\mathit{isort}_{\mathbf{L}} : \mathrm{list} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \forall \ell.\ \top \Rightarrow \forall n.\ \mathrm{Len}(\ell, n) \to \mathbb{C}_{\mathrm{u}}(\mathbf{r}\ \ell, 0, n, r.\mathrm{len}_{\mathbf{L}}(r, n, n))$$

Informally, the claim states that for a *eager* input list ℓ of length n, the result will incur cost n and the resulting *lazy* list will be of length n, with the tail-forcing cost n.

Proof. We first use the rules U-LETREC and U-MATCH. For U-MATCH we have to consider two cases.

$\underline{\text{Case } \ell \doteq \mathsf{nil}.}$ From $\mathrm{Len}(\ell, n)$ we then have $n \doteq 0$. We need to show:

$$\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0, \mathbf{r}.\mathrm{len}_{\mathbf{L}}(\mathbf{r}, 0, 0))$$

This follows by the rules U-MONAD and U-RET, with L^{C} reasoning for $\mathrm{len}_{\mathbf{L}}(\mathsf{nil}, 0, 0)$.

$\underline{\text{Case } \ell \doteq \mathsf{cons}(h,t).}$ From $\mathrm{Len}(\ell, n)$ we have $\mathrm{Len}(t, n')$, s.t. $n \doteq n' + 1$. We need to show:

$$\{\mathsf{cbind}(f\ t, \{t'\}.\, \mathsf{cbind}(\mathit{insert}_{\mathbf{L}}\ h\ t', \{t''\}.\, \mathsf{cret}(t'')))\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, n, \mathbf{r}.\mathrm{len}_{\mathbf{L}}(\mathbf{r}, n, n))$$

By IH we have:

$$f\ t : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, n', t'.\mathrm{len}_{\mathbf{L}}(t', n', n'))$$

Hence, by the rules U-MONAD, U-BIND, and U-SUBC with $n' + 1 \doteq n$, it remains to show:

$$\mathsf{cbind}(\mathit{insert}_{\mathbf{L}}\ h\ t', \{t''\}.\, \mathsf{cret}(t'')) \mid 0 \mid 1 \mid \mathrm{len}_{\mathbf{L}}(\mathbf{r}, n, n)$$

By the result for $\mathit{insert}_{\mathbf{L}}$ from above, we have:

$$\mathit{insert}_{\mathbf{L}}\ h\ t' \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 1, t''.\mathrm{len}_{\mathbf{L}}(r, n'+1, n'+1))$$

Since $n \doteq n' + 1$, the result follows by the rules U-BIND and U-RET.

Finally, for the minimum function we want to show:

$$min_{\mathbf{L}} : \text{list} \to \mathbb{C}(\mathbb{Z}) \mid \forall \ell.\ \top \Rightarrow \forall n.\ \text{Len}(\ell, n+1) \Rightarrow \mathbb{C}_{\mathbf{u}}(\mathbf{r}, 0, n+1, \_.\top)$$

Informally, the claim states that for an *eager* list ℓ of length $n+1$ (we require non-empty list), the result will incur cost $n+1$.

Proof. We use the rule U-ABS. Then it remains to show:

$$\{\mathsf{cbind}(isort_{\mathbf{L}}\ \ell, \{\ell'\}.\, \mathsf{cret}(\mathsf{match}\ \ell'\ \mathsf{with}\ \mathsf{cons} \mapsto h, \_.\ h))\} : \mathbb{C}(\mathbb{Z}) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, 0, n+1, \_.\top)$$

From the result for $isort_{\mathbf{L}}$, from the above, we have:

$$isort_{\mathbf{L}}\ \ell : \mathbb{C}(\text{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathbf{u}}(\mathbf{r}, 0, n+1, \ell'.\text{len}_{\mathbf{L}}(\ell', n+1, n+1))$$

Hence, by the rules U-MONAD, U-BIND, and U-RET, it remains to show:

$$\mathsf{match}\ \ell'\ \mathsf{with}\ \mathsf{cons} \mapsto \lambda h, \_.\ h : \mathbb{Z} \mid \top$$

This follows trivially by observing that we obtain a contradiction in the nil case of match, since $\text{len}_{\mathbf{L}}(\ell', n+1, n+1)$.

6.6 take and map (relational lazy data-structures)

The following example shows *relational* cost reasoning on lazy lists. To describe concretely our goal, consider the following two functions:

$$\begin{array}{l}
take_{\mathbf{L}} \triangleq \mathsf{rec}\, f(\ell).\ \lambda k. \\
\quad \mathsf{match}\ \ell\ \mathsf{with} \\
\qquad \mathsf{nil} \mapsto \quad \{\mathsf{cret}(\mathsf{nil})\}; \\
\qquad \mathsf{cons} \mapsto \quad \lambda h, t.\ \mathsf{match}\ k\ \mathsf{with} \\
\qquad\qquad\qquad\qquad 0 \mapsto \quad \{\mathsf{cret}(\mathsf{nil})\}; \\
\qquad\qquad\qquad\qquad S \mapsto \quad \lambda k'.\ \{\mathsf{cret}(\mathsf{cons}(h, \{\mathsf{cbind}(t, \\
\qquad\qquad\qquad\qquad\qquad\qquad \{t'\}.\, \mathsf{cbind}(f\ t'\ k', \{z\}.\, \mathsf{cret}(z)))\}))\}
\end{array}$$

$$\begin{array}{l}
map_{\mathbf{L}} \triangleq \mathsf{rec}\, f(\ell).\ \lambda g. \\
\quad \mathsf{match}\ \ell\ \mathsf{with} \\
\qquad \mathsf{nil} \mapsto \quad \{\mathsf{cstep}_1(\mathsf{cret}(\mathsf{nil}))\} \\
\qquad \mathsf{cons} \mapsto \quad \lambda h, t.\ \{\mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(g\ h, \{\mathsf{cbind}(t, \{t'\}.\, \mathsf{cbind}(f\ t'\ g, \{z\}.\, \mathsf{cret}(z)))\})))\}
\end{array}$$

$take_{\mathbf{L}}\ \ell\ k$, for a natural number k, returns the first k elements of a lazy list ℓ (defined in the same way as in the previous example), or the whole list if its length is less than k. $map_{\mathbf{L}}\ \ell\ g$ applies a function g to a lazy list ℓ pointwise. For simplicity, the cost is incurred only in $map_{\mathbf{L}}$ function. Note that these two functions can make different number of recursive calls, even when applied to the same list, hence they are challenging to analyze relationally.

We show that $map_{\mathbf{L}}$ and $take_{\mathbf{L}}$ *commute w.r.t. cost*, i.e. that $take_{\mathbf{L}}$ applied to a (lazy) list after $map_{\mathbf{L}}$, and $map_{\mathbf{L}}$ applied to a list after $take_{\mathbf{L}}$, have the same cost. Note that this holds only in a lazy setting (although they commute functionally, i.e., return the same result, even in an eager setting).

We start by defining a relational version of the refinement $\text{len}_{\mathbf{L}}$, which we discussed in the previous example:

$$\begin{array}{lcl}
\forall \ell_1, \ell_2, c.\ \text{len}^R_{\mathbf{L}}(\ell_1, \ell_2, c, 0) & \Leftrightarrow & \ell_1 \doteq \ell_2 \doteq \mathsf{nil} \\
\forall \ell_1, \ell_2, c, n.\ \text{len}^R_{\mathbf{L}}(\ell_1, \ell_2, c, n+1) & \Leftrightarrow & \exists h_1, t_1, h_2, t_2.\ \ell_1 \doteq \mathsf{cons}(h_1, t_1) \wedge \ell_2 \doteq \mathsf{cons}(h_2, t_2) \\
 & & \quad \wedge\ \mathbb{C}_{\mathrm{r}}(t_1, t_2, c, t_1.t_2.\text{len}^R_{\mathbf{L}}(t_1, t_2, c, n))
\end{array}$$

The refinement $\text{len}^R_{\mathbf{L}}(\ell_1, \ell_2, c, n)$ states that a pair of lazy lists (of the same length n) ℓ_1 and ℓ_2, have a relational cost c of forcing tails pair-wise.

Formally, we show:

$$\ell : \text{list}, k, n : \mathbb{N}, g : \mathbb{N} \to \mathbb{N}; \text{Len}(\ell, n) \vdash$$

$$\{\mathsf{cbind}(\mathit{take}_{\mathbf{L}}\ k\ \ell, \{\ell_1'\}.\ \mathsf{cbind}(\mathit{map}_{\mathbf{L}}\ g\ \ell_1', \{\ell_1''\}.\ \mathsf{cret}(\ell_1'')))\} : \mathbb{C}(\text{list}_{\mathbf{L}}) \sim$$

$$\{\mathsf{cbind}(\mathit{map}_{\mathbf{L}}\ g\ \ell, \{\ell_2'\}.\ \mathsf{cbind}(\mathit{take}_{\mathbf{L}}\ k\ \ell_2', \{\ell_2''\}.\ \mathsf{cret}(\ell_2'')))\} : \mathbb{C}(\text{list}_{\mathbf{L}}) \mid$$

$$\mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_1, 0, \mathbf{r}_1.\mathbf{r}_2.\text{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, 0, \min(k, n))$$

(where $\text{Len}(\ell, n)$ states that ℓ has length n). Informally, we state that the resulting relational cost is 0, that the resulting lists have length $\min(k, n)$, and that the relational cost of forcing the tail-pairs is 0.

Since $\mathit{map}_{\mathbf{L}}$ and $\mathit{take}_{\mathbf{L}}$ can make different number of recursive calls, we show this indirectly. On a very high-level, we prepend (append) the *list identity function* $\mathit{id}_{\mathbf{L}}$ (defined below) to the beginning (end) of the function applications above, hence we compare $\mathit{id}_{\mathbf{L}}$ to $\mathit{map}_{\mathbf{L}}$ (and vice-verse), and $\mathit{take}_{\mathbf{L}}$ to itself. The key point is that $\mathit{id}_{\mathbf{L}}$ and $\mathit{map}_{\mathbf{L}}$ make the same number of recursive calls (they iterate the whole list), hence they are easy to analyze. We formally define this next.

The identity function is defined as follows:

$$\begin{array}{l}
\mathit{id}_{\mathbf{L}} \triangleq \mathsf{rec}\, f(\ell). \\
\ \mathsf{match}\ \ell\ \mathsf{with} \\
\qquad \mathsf{nil} \mapsto \ \{\mathsf{cret}(\mathsf{nil})\}; \\
\qquad \mathsf{cons} \mapsto \ \lambda h, t.\ \{\mathsf{cret}(\mathsf{cons}(h, \{\mathsf{cbind}(t, \{t'\}.\ \mathsf{cbind}(f\ t', \{z\}.\ \mathsf{cret}(z)))\}))\}
\end{array}$$

This crucial property of this function is:

$$;\vdash_{\mathrm{L^C}} \forall \ell.\ \mathit{id}_{\mathbf{L}}\ \ell \doteq \{\mathsf{cret}(\ell)\}$$

This can be shown using the L^{C} induction principle IND.

Using this idea we can rewrite the above state goal to:

$$\ell : \text{list}, k, n : \mathbb{N}, g : \mathbb{N} \to \mathbb{N}; \text{Len}(\ell, n) \vdash$$

$$\{\mathsf{cbind}(\mathit{id}_{\mathbf{L}}\ \ell, \{\ell_1'\}.\ \mathsf{cbind}(\mathit{take}_{\mathbf{L}}\ k\ \ell_1', \{\ell_1''\}.\ \mathsf{cbind}(\mathit{map}_{\mathbf{L}}\ g\ \ell_1'', \{\ell_1'''\}.\ \mathsf{cret}(\ell_1'''))))\} : \mathbb{C}(\text{list}_{\mathbf{L}}) \sim$$

$$\{\mathsf{cbind}(\mathit{map}_{\mathbf{L}}\ g\ \ell, \{\ell_2'\}.\ \mathsf{cbind}(\mathit{take}_{\mathbf{L}}\ k\ \ell_2', \{\ell_2''\}.\ \mathsf{cbind}(\mathit{id}_{\mathbf{L}}\ \ell_2'', \{\ell_2'''\}.\ \mathsf{cret}(\ell_2'''))))\} : \mathbb{C}(\text{list}_{\mathbf{L}}) \mid$$

$$\mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_1, 0, \mathbf{r}_1.\mathbf{r}_2.\text{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, 0, \min(k, n))$$

In order to show this goal, it suffices to show the following, which can be done completely relationally:

$$id_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim map_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{C}(\mathrm{list}_{\mathbf{L}})$$

$$\mid \forall \ell_1, \ell_2.\ \ell_1 \doteq \ell_2 \Rightarrow \forall n.\ \mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, 0, n) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, -1, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, -1, n))$$

and

$$take_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to \mathbb{N} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim take_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to \mathbb{N} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid$$

$$\forall \ell_1, \ell_2.\ \top \Rightarrow \forall n.\ \mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, n) \forall k_1, k_2.\ k_1 \doteq k_2 \Rightarrow$$

$$\mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, -1, \min(n, k_1)))$$

and

$$map_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim id_{\mathbf{L}} : \mathrm{list}_{\mathbf{L}} \to \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid$$

$$\forall \ell_1, \ell_2.\ \top \Rightarrow \forall n.\ \mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, n) \Rightarrow \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 1, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, 0, n))$$

Further, note that the first and the third statement are symmetric; hence we show the first two proofs.

Proof of the first statement. We use the R^{C} rule R-LETREC, and then (async.) rule R-LETREC-R (since $map_{\mathbf{L}}$ starts with $\lambda g.$), followed by R-MATCH. We consider the two cases for ℓ_1, ℓ_2 (in other cases we obtain a contradiction from $\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, 0, n)$.

Case $\ell_1 \doteq \ell_2 \doteq \mathsf{nil}$. From $\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, 0, n)$ we have $n \doteq 0$. We need to show:

$$\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim \{\mathsf{cstep}_1(\mathsf{cret}(\mathsf{nil}))\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, -1, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, -1, 0))$$

This follows trivially.

Case $\ell_1 \doteq \mathsf{cons}(h_1, t_1)$ and $\ell_2 \doteq \mathsf{cons}(h_2, t_2)$. We need to show:

$$\{\mathsf{cret}(h_1, \{\mathsf{cbind}(t_1, \{t_1'\}.\ \cdots)\})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim \{\mathsf{cstep}_1(\mathsf{cret}(h_2, \{\mathsf{cbind}(t_2, \{t_2'\}.\ \cdots)\}))\} :$$

$$\mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, -1, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, n))$$

From $\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, 0, n)$ we have $\mathbb{C}_{\mathrm{r}}(t_1, t_2, 0, t_1'.t_2'.\mathrm{len}^R_{\mathbf{L}}(t_1', t_2', 0, n'))$, s.t. $n \doteq n' + 1$. Hence, by applying R^{C} rules it remains to show:

$$\mathsf{cbind}(f_1\ t_1', \{z_1\}.\, \mathsf{cret}(z_1)) \div \mathrm{list}_{\mathbf{L}} \sim \mathsf{cbind}(f_2\ t_2', \{z_2\}.\, \mathsf{cret}(z_2)) \div \mathrm{list}_{\mathbf{L}} \mid -1 \mid \mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, -1, n')$$

This follows by IH.

Proof of the second statement. We apply the R^{C} rules R-LETREC, R-ABS, and R-MATCH. We consider the two cases for ℓ_1, ℓ_2 (in the other two we obtain a contradiction).

Case $\ell_1 \doteq \ell_2 \doteq \mathsf{nil}$. From $\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, n)$ we have $n \doteq 0$. We need to show:

$$\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim \{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\mathbf{r}_1, \mathbf{r}_2, -1, 0))$$

This follows trivially.

Case $\ell_1 \doteq \mathsf{cons}(h_1, t_1)$ and $\ell_2 \doteq \mathsf{cons}(h_2, t_2)$. We apply the R^{C} rule R-MATCH. We need to consider two sub-cases for k_1, k_2 (the other two cases yield a contradiction since $k_1 \doteq k_2$); the first case is almost identical to the nil case above. Hence we consider the sub-case when $k_1 \doteq k_2 \doteq k' + 1$. We need to show:

$$\{\mathsf{cret}(\mathsf{cons}(h_1, \{\mathsf{cbind}(t_1, \{t_1'\}, \mathsf{cbind}(f_1\ t_1'\ k', \{z_1\}.\, \mathsf{cret}(z_1)))\}))\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim$$

$$\{\mathsf{cret}(\mathsf{cons}(h_2, \{\mathsf{cbind}(t_2, \{t_2'\}, \mathsf{cbind}(f_2\ t_2'\ k', \{z_2\}.\, \mathsf{cret}(z_1)))\}))\} : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid$$

$$\mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, \mathbf{r}_1.\mathbf{r}_2.\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, \min(n, k_1)))$$

From $\mathrm{len}^R_{\mathbf{L}}(\ell_1, \ell_2, -1, n)$ we have $\mathbb{C}_{\mathrm{r}}(t_1, t_2, -1, t_1'.t_2'.\mathrm{len}^R_{\mathbf{L}}(t_1', t_2', -1, n'))$, s.t. $n \doteq n' + 1$. Since $n \doteq n' + 1$ and $k_1 \doteq k' + 1$, we have $\min(n, k_1) \doteq \min(n', k') + 1$. By IH we also have:

$$f_1\ t_1'\ k' : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \sim f_2\ t_2'\ k' : \mathbb{C}(\mathrm{list}_{\mathbf{L}}) \mid \mathbb{C}_{\mathrm{r}}(\mathbf{r}_1, \mathbf{r}_2, 0, z_1.z_2.\mathrm{len}^R_{\mathbf{L}}(z_1, z_2, -1, \min(n', k')))$$

Hence the goal follows by the rules of R^{C}.

7 Proofs of the examples from the paper

In this section we give the proofs of the examples that are omitted, for space reasons, from the paper.

The remark from the Section 6 about the details in the proofs applies also in this section.

7.1 Insert into a sorted list

Here we give the part of the proof that was omitted from the paper.

We have the following assumptions:

(a) $\mathrm{Sorted}(\mathsf{cons}(h, t))$;

(b) $x \not\leq h$, hence $h < x$ and $h \leq x$;

(c) $\phi[t/\ell][t'/\mathbf{r}] \triangleq \mathrm{Sorted}(t') \wedge \forall y, q.\ \mathrm{LargerThan}(y, \mathsf{cons}(x, t), q) \Rightarrow \mathrm{LargerThan}(y, t', q)$;

The remaining obligation is:

$$\phi[\mathsf{cons}(h, t')/\mathbf{r}] \triangleq \mathrm{Sorted}(\mathsf{cons}(h, t')) \wedge \forall y, q.\ \mathrm{LargerThan}(y, \mathsf{cons}(x, \ell), q) \Rightarrow \mathrm{LargerThan}(y, \mathsf{cons}(h, t'), q)$$

Proof of: $\mathrm{Sorted}(\mathsf{cons}(h, t'))$.
From (a) we have $\mathrm{Sorted}(\mathsf{cons}(h, t)) \Rightarrow \mathrm{Unsorted}(\mathsf{cons}(h, t), 0) \Rightarrow \mathrm{LargerThan}(h, t, 0)$. Further, from (b) $h \leq x$ and $\mathrm{LargerThan}(h, t, 0)$ follows $\mathrm{LargerThan}(h, \mathsf{cons}(x, t), 0)$, and then from (c) $\mathit{LargerThan}(h, t', 0)$; and then $\mathrm{Sorted}(t') \Rightarrow \mathrm{Unsorted}(t', 0)$, hence finally $\mathrm{Sorted}(\mathsf{cons}(h, t'))$, as required.

Proof of: $\forall y, q.\ \mathrm{LargerThan}(y, \mathsf{cons}(x, \ell), q) \Rightarrow \mathrm{LargerThan}(y, \mathsf{cons}(h, t'), q)$.
Assume: $\mathrm{LargerThan}(y, \mathsf{cons}(x, \ell), q) \doteq \mathrm{LargerThan}(y, \mathsf{cons}(x, \mathsf{cons}(h, t)), q)$ for some y and q.
We need to show: $\mathrm{LargerThan}(y, \mathsf{cons}(h, t'), q)$. We case analyze $y \leq x$.

If $y \leq x$ then $\mathrm{LargerThan}(y, \mathsf{cons}(h, t), q)$. We further case analyze $y \leq h$.

If $y \leq h$, then LargerThan(y, t, q) and further LargerThan$(y, \mathsf{cons}(x, t), q)$, and then by (c): LargerThan(y, t', q) and finally we have LargerThan$(y, \mathsf{cons}(h, t'), q)$.

If $\neg(y \leq h)$ (and hence $h \leq y$) then LargerThan$(y, t, q-1)$ (note that $q > 0$) and LargerThan$(y, \mathsf{cons}(x, t), q-1)$, and then by (c) LargerThan$(y, t', q-1)$, and finally we have LargerThan$(y, \mathsf{cons}(h, t'), q)$.

If $\neg(y \leq x)$ (and hence $x \leq y$ and $h \leq y$), then LargerThan$(y, \mathsf{cons}(h, t), q-1)$ and further LargerThan$(y, t, q-2)$ (note that $q > 1$) and LargerThan$(y, \mathsf{cons}(x, t), q-1)$. Then by (c) LargerThan$(y, t', q-1)$ and LargerThan$(y, \mathsf{cons}(h, t), q)$.

7.2 Insertion sort

Here we give the part of the proof that was omitted from the paper.

We have the following assumptions:

(a) $\phi'_1 \triangleq \text{Sorted}(t'_1) \wedge \forall y, q.\ \text{LargerThan}(y, t_1, q) \Rightarrow \text{LargerThan}(y, t'_1, q)$;

(b) $\phi'_2 \triangleq \text{Sorted}(t'_2) \wedge \forall y, q.\ \text{LargerThan}(y, t_2, q) \Rightarrow \text{LargerThan}(y, t'_2, q)$;
(in the paper $\phi'_1 \wedge \phi'_2 \triangleq \phi' \triangleq \phi[t_1/\ell_1][t_2/\ell_2][t'_1/\mathbf{r}_1][t'_2/\mathbf{r}_2]$)

(c) $\phi''_1 \triangleq \text{Sorted}(z_1) \wedge \forall y, q.\ \text{LargerThan}(y, \mathsf{cons}(h_1, t'_1), q) \Rightarrow \text{LargerThan}(y, z_1, q)$;

(d) $\phi''_2 \triangleq \text{Sorted}(z_2) \wedge \forall y, q.\ \text{LargerThan}(y, \mathsf{cons}(h_2, t'_2), q) \Rightarrow \text{LargerThan}(y, z_2, q)$;
(in the paper $\triangleq \phi''_1 \wedge \phi''_2 \triangleq \phi''$)

The remaining obligation is:

$$\begin{array}{rcll} \phi[z_1/\mathbf{r}_1][z_2/\mathbf{r}_2] & \triangleq & \phi_1 \wedge \phi_2 & \text{where:} \\ \phi_1 & \triangleq & \text{Sorted}(z_1) \wedge \forall y, q.\ \text{LargerThan}(y, \ell_1, q) \Rightarrow \text{LargerThan}(y, z_1, q) & \\ \phi_1 & \triangleq & \text{Sorted}(z_2) \wedge \forall y, q.\ \text{LargerThan}(y, \ell_2, q) \Rightarrow \text{LargerThan}(y, z_2, q) & \end{array}$$

Sorted(z_1) and Sorted(z_2) follow directly from (c) and (d). The remaining goals are symmetric, so we show only the first one: $\forall y, q.\ \text{LargerThan}(y, \ell_1, q) \Rightarrow \text{LargerThan}(y, z_1, q)$.
Assume: LargerThan(y, ℓ_1, q) for some y and q.
To show: LargerThan(y, z_1, q).

Note that the assumption is equivalent to LargerThan$(y, \mathsf{cons}(h_1, t_1), q)$, since $\ell_1 \doteq \mathsf{cons}(h_1, t_1)$. We case analyze $y \leq h_1$.

Assume $y \leq h_1$. Then LargerThan(y, t_1, q), and by (a) we have LargerThan(y, t'_1, q), and then LargerThan$(y, \mathsf{cons}(h_1, t'_1), q)$. Finally, by (c) we have LargerThan(y, z_1, q).

Assume $\neg(y \leq h_1)$. Then LargerThan$(y, t_1, q-1)$ (note that it has to be $q > 0$), and by (a) we have LargerThan$(y, t'_1, q-1)$, and then LargerThan$(y, \mathsf{cons}(h_1, t'_1), q)$. Then finally by (c) we have LargerThan(y, z_1, q).

7.3 Fixed-width counter

Here we discuss the proof of the fixed-width counter example, which is discussed in the paper.

We consider the following programs:

$$\begin{array}{lll} \mathit{incg} \triangleq \mathsf{rec}\, f(\ell).\ \mathsf{match}\ \ell\ \mathsf{with} & & \\ \quad \mathsf{nil} \mapsto \{\mathsf{cret}(\mathsf{nil})\}; & & \\ \quad \mathsf{cons} \mapsto \lambda x, t.\ \mathsf{if}\ x < D-1 & \mathsf{then}\ \{\mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(x+1,t)))\} & \\ & \mathsf{else}\ \{\mathsf{cbind}(f\ t, \{t'\}.\, \mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(0,t'))))\} & \end{array}$$

$$\begin{array}{ll} \mathit{rinc} \triangleq \mathsf{rec}\, f(k).\lambda\ell.\ \mathsf{match}\ k\ \mathsf{with} & 0 \mapsto \{\mathsf{cret}(\ell)\}; \\ & S \mapsto \lambda k'.\ \{\mathsf{cbind}(\mathit{incg}\ \ell, \{\ell'\}.\, \mathsf{cbind}(f\ k'\ \ell', \{\ell''\}.\, \mathsf{cret}(\ell'')))\} \end{array}$$

And the following potential predicate:

$$\forall \ell, n.\ \Phi(\ell, n) \Leftrightarrow (\ell \doteq \mathsf{nil} \wedge n \doteq 0) \vee (\exists t, i, n'.\ \ell \doteq \mathsf{cons}(i,t) \wedge \Phi(t,n') \wedge n \doteq n' + \frac{i}{D-1} \wedge i \leq D-1)$$

Then, formally, we want to show the following two claims:

$$\vdash \mathit{incg} : \mathrm{list} \to \mathbb{C}(\mathrm{list}) \mid \forall \ell.\ \top \Rightarrow \forall n.\ \Phi(\ell, n) \Rightarrow \exists n'.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}\ \ell, 0, n + \frac{D}{D-1} - n', \mathbf{r}.\Phi(\mathbf{r}, n'))$$

$$\vdash \mathit{rinc} : \mathrm{nat} \to \mathrm{list} \to \mathbb{C}(\mathrm{list}) \mid \forall k.\ \forall n, \ell.\ \Phi(\ell, n) \Rightarrow \exists n'.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}\ k\ \ell, 0, \frac{D}{D-1} \cdot k + n - n', \mathbf{r}.\Phi(\mathbf{r}, n'))$$

Proof of the first claim. We start by using the rules U-LETREC and U-MATCH. We consider two cases for ℓ.

Case $\ell \doteq \mathsf{nil}$. This forces $n \doteq 0$. We pick $n' := 0$, hence it remains to show:

$$\{\mathsf{cret}(\mathsf{nil})\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, 0 + \frac{D}{D-1} - 0, \mathbf{r}.\Phi(\mathbf{r}, 0))$$

This follows by the rules U-MONAD and U-RET, followed by the rule U-SUBC with $0 \leq 0 + \frac{D}{D-1} - 0$, and reasoning in L^{C} to show $\Phi(\mathsf{nil}, 0)$.

Case $\ell \doteq \mathsf{cons}(x,t)$. We further case-analyze $x < D-1$ (in the inner if-then-else/match).

Sub-case $x < D-1$. From the assumption $\Phi(\ell, n)$ we have $\Phi(t, n'')$ for some n'', s.t. $n \doteq n'' + \frac{x}{D-1}$. We pick $n' := n + \frac{1}{D-1}$, hence it remains to show:

$$\{\mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(x+1,t)))\} : \mathbb{C}(\mathrm{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, n + \frac{D}{D-1} - (n+1), \mathbf{r}.\Phi(\mathbf{r}, n+1))$$

This follows by the rules U-MONAD, U-STEP, and U-RET, followed by the rule U-SUBC with $0 \leq 1 \leq n + \frac{D}{D-1} - (n + \frac{1}{D-1}) \doteq 1$, and reasoning in L^{C} to show $\Phi(\mathsf{cons}(x+1,t), n + \frac{1}{D-1})$. The latter holds since $n'' + \frac{x+1}{D-1} \doteq n'' + \frac{x}{D-1} + \frac{1}{D-1} \doteq n + \frac{1}{D-1}$ and $x + 1 \leq D - 1$.

Sub-case $x \not\leq D-1$. From, $\Phi(\ell, n)$ we have $x \doteq D-1$ and $\Phi(t, n'')$, for some n'', s.t. and $n \doteq n'' + \frac{x}{D-1} \doteq n'' + 1$. We need to show:

$$\{\mathsf{cbind}(f\ t, \{t'\}.\, \mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(0,t'))))\} : \mathbb{C}(\mathrm{list}) \mid \exists n'.\ \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, n + \frac{D}{D-1} - n', \mathbf{r}.\Phi(\mathbf{r}, n'))$$

By IH we obtain:

$$\exists n'.\ \mathbb{C}_{\mathrm{u}}(f\ t, 0, n'' + \frac{D}{D-1} - n', t'.\Phi(t', n'))$$

We use the rule ∃E, and pick $n' := n'$. Since $(n'' + \frac{D}{D-1} - n') + 1 \doteq n + \frac{D}{D-1} - n'$, by the rules U-MONAD and U-BIND, and the rule U-SUBC it remains to show:

$$\mathsf{cstep}_1(\mathsf{cret}(\mathsf{cons}(0, t'))) \div \text{list} \mid 0 \mid 1 \mid \Phi(\mathbf{r}, n')$$

This follows by the rules U-STEP and U-RET, and reasoning in L^{C} to show $\Phi(\mathsf{cons}(0, t'), n')$.

Proof of the second claim. We start with the rules U-LETREC, U-ABS, and U-MATCH. We consider two cases for k.

Case $k \doteq 0$. We pick $n' := n$, hence it remains to show:

$$\{\mathsf{cret}(\ell)\} : \mathbb{C}(\text{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \frac{D}{D-1} \cdot 0 + n - n, \mathbf{r}.\Phi(\mathbf{r}, n))$$

This follows by the rules U-MONAD, U-RET, and U-SUBC with $\frac{D}{D-1} \cdot 0 + n - n \doteq 0$, and L^{C} reasoning to show $\Phi(\ell, n)$.

Case $k \doteq S\ k'$. By the result for *incg*, from above, we have:

$$\exists n'''.\ \mathbb{C}_{\mathrm{u}}(\mathit{incg}\ \ell, 0, n + \frac{D}{D-1} - n''', \ell'.\Phi(\ell', n'''))$$

Further, by IH we have:

$$\exists n''.\ \mathbb{C}_{\mathrm{u}}(f\ k'\ \ell', 0, \frac{D}{D-1} \cdot k' + n''' - n'', \ell''.\Phi(\ell'', n''))$$

We pick $n' := n''$, hence we need to show:

$$\{\mathsf{cbind}(\mathit{inc}\ \ell, \{\ell'\}.\, \mathsf{cbind}(f\ k'\ \ell', \{\ell''\}.\, \mathsf{cret}(\ell'')))\} : \mathbb{C}(\text{list}) \mid \mathbb{C}_{\mathrm{u}}(\mathbf{r}, 0, \frac{D}{D-1} \cdot k + n - n'', \ell''.\Phi(\ell'', n''))$$

Note that $n + \frac{D}{D-1} - n''' + \frac{D}{D-1} \cdot k' + n''' - n'' \doteq \frac{D}{D-1} \cdot k' + \frac{D}{D-1} + n - n'' \doteq \frac{D}{D-1} \cdot k + n - n''$. Hence, the required goal follows by the U-MONAD, U-BIND, U-SUBC, and U-RET, and L^{C} reasoning to show $\Phi(\ell'', n'')$.

References

Alejandro Aguirre, Gilles Barthe, Marco Gaboardi, Deepak Garg, and Pierre-Yves Strub. 2017. A Relational Logic for Higher-Order Programs. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming (ICFP)*. http://arxiv.org/abs/1703.05042

Nils Anders Danielsson. 2008. Lightweight Semiformal Time Complexity Analysis for Purely Functional Data Structures. In *Proceedings of the 35th Symposium on Principles of Programming Languages (POPL)*.

Hongwei Xi. 1999. Dependently typed data structures. In *Proceedings of Workshop of Algorithmic Aspects of Advanced Programming Languages (WAAAPL'99)*. 17–32.