Transfinite Step-Indexing for Termination

Technical Report

Simon Spies, Neel Krishnaswami, Derek Dreyer

October 27, 2020

Contents

1 The Language λ_{CHAN} 2

2 Ordinals 5

3 Resources 7

4 Logical Relation 11

4.1 Kripke Logical Relation 14

4.2 Resource Interpretation Properties 15

4.3 Properties of the Type Interpretations 20

4.4 Semantic Typing 24

4.5 Compatibility Lemmas 26

$$
\begin{array}{llcl}
\text{Values} & v & ::= & \ell \mid () \mid n \mid b \mid \lambda x.e \mid (v_1, v_2) \\
\text{Expressions} & e & ::= & x \mid \ell \mid () \mid n \mid b \mid e_1; e_2 \mid \lambda x.e \mid e_1\ e_2 \mid e_1 \dotplus e_2 \mid \mathsf{iter}(e, e_0, x.e_S) \\
 & & \mid & (e_1, e_2) \mid \mathsf{let}\ (x, y) = e_1\ \mathsf{in}\ e_2 \mid \mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 \\
 & & \mid & \mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e \mid \mathsf{get}(e_1, e_2) \mid \mathsf{put}(e_1, e_2) \\
\text{Types} & A, B & ::= & \mathbb{1} \mid \mathbb{B} \mid \mathbb{N} \mid A \otimes B \mid A \multimap B \mid \mathsf{Get}\ A \mid \mathsf{Put}\ A \\
\text{Type Contexts} & \Gamma, \Delta & ::= & \cdot \mid \Gamma, x : A \\
\text{Heap Values} & hv & ::= & \mathsf{E} \mid \mathsf{V}(v) \mid \mathsf{C}(v) \\
\text{Heaps} & h & ::= & \cdot \mid \ell \mapsto hv, h
\end{array}
$$

Figure 1: The language λ_{CHAN}

1 The Language λ_{CHAN}

In the present work, we consider the language λ_{CHAN} given in Figure 1, an extension of the simply-typed λ-calculus with an implementation of asynchronous channels. A fresh channel can be created with $\mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e$, where x is a handle for receiving values over the new channel and y is a handle for sending values. The operation put can be used to send values and the operation get to receive values.

We assign expressions a type A in the linear type system $\Gamma \vdash e : A$, defined in Figure 2. Besides the base types $\mathbb{1}, \mathbb{B}, \mathbb{N}$, we have linear pairs $A \otimes B$ and linear functions $A \multimap B$. The type $\mathsf{Get}\ A$ is used for the receive handle of a channel, indicating that the channel will transfer a value of type A. Similarly, the type $\mathsf{Put}\ A$ is used for the send handle. We allow values of arbitrary types A to be transferred through channels, including channel handles themselves and functions possibly capturing channel handles. The context Γ is a linear context without an ordering of the variables. We write Γ, Δ for the disjoint union of the contexts Γ and Δ.

We equip the language with a single-threaded, heap-based operational semantics, given in Figure 3. Operationally, each channel is represented by a single location ℓ in the heap, storing either nothing E, a value $\mathsf{V}(v)$, or a continuation $\mathsf{C}(\lambda x.e)$. Initially, the heap stores the empty heap value E. If a value is sent over channel ℓ with $\mathsf{put}(\ell, v)$, then the value is stored in memory as $\mathsf{V}(v)$. If subsequently $\mathsf{get}(\ell, \lambda x.e)$ is executed, then the continuation $\lambda x.e$ is invoked with argument v and the state in the heap is restored to E. If from the initial state $\mathsf{get}(\ell, \lambda x.e)$ is executed, then the continuation is stored in the heap as $\mathsf{C}(\lambda x.e)$ and invoked with argument v once a corresponding $\mathsf{put}(\ell, v)$ is called. Besides the operations on channels, the operational semantics allows standard, pure reductions for the simply typed λ-calculus. Reductions are allowed to occur in any evaluation context K, making it a call-by-value, left-to-right operational semantics.

We write $h[\ell \mapsto hv]$ for the heap which returns hv for argument ℓ and $h(\ell')$ for any argument $\ell' \neq \ell$. Analogously, we use the notation to update finite and infinite maps with a new binding in the remainder of this work. We assume substitution is capture avoiding, write $e[v/x]$ for the single-point substitution replacing x with v in e, and $e[\theta]$ for the parallel substitution replacing each free variable x in e with θx.

$$\boxed{\Gamma \vdash e : A}$$

$$\frac{}{x : A \vdash x : A} \qquad \frac{\Gamma \vdash e : B}{\Gamma, x : A \vdash e : B} \qquad \frac{}{\cdot \vdash () : \mathbb{1}} \qquad \frac{\Gamma \vdash e_1 : \mathbb{1} \qquad \Delta \vdash e_2 : A}{\Gamma, \Delta \vdash e_1; e_2 : A} \qquad \frac{}{\cdot \vdash b : \mathbb{B}}$$

$$\frac{\Gamma \vdash e : \mathbb{B} \qquad \Delta \vdash e_1 : A \qquad \Delta \vdash e_2 : A}{\Gamma, \Delta \vdash \mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : A} \qquad \frac{}{\cdot \vdash n : \mathbb{N}} \qquad \frac{\Gamma \vdash e_1 : \mathbb{N} \qquad \Delta \vdash e_2 : \mathbb{N}}{\Gamma, \Delta \vdash e_1 \dotplus e_2 : \mathbb{N}}$$

$$\frac{\Gamma \vdash e : \mathbb{N} \qquad \Delta \vdash e_0 : A \qquad x : A \vdash e_S : A}{\Gamma, \Delta \vdash \mathsf{iter}(e, e_0, x.e_S) : A} \qquad \frac{\Gamma \vdash e_1 : A_1 \qquad \Delta \vdash e_2 : A_2}{\Gamma, \Delta \vdash (e_1, e_2) : A_1 \otimes A_2}$$

$$\frac{\Gamma \vdash e_1 : A_1 \otimes A_2 \qquad \Delta, x : A_1, y : A_2 \vdash e_2 : B}{\Gamma, \Delta \vdash \mathsf{let}\ (x, y) = e_1\ \mathsf{in}\ e_2 : B} \qquad \frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash \lambda x.e : A \multimap B}$$

$$\frac{\Gamma \vdash e_1 : A \multimap B \qquad \Delta \vdash e_2 : A}{\Gamma, \Delta \vdash e_1\ e_2 : B} \qquad \frac{\Gamma \vdash e_1 : \mathsf{Get}\, A \qquad \Delta \vdash e_2 : A \multimap \mathbb{1}}{\Gamma, \Delta \vdash \mathsf{get}(e_1, e_2) : \mathbb{1}}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{Put}\, A \qquad \Delta \vdash e_2 : A}{\Gamma, \Delta \vdash \mathsf{put}(e_1, e_2) : \mathbb{1}} \qquad \frac{\Gamma, x : \mathsf{Get}\, A, y : \mathsf{Put}\, A \vdash e : B}{\Gamma \vdash \mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e : B}$$

Figure 2: Typing Rules

$$
\begin{array}{llcl}
\text{Evaluation Contexts} & K & ::= & \cdot \mid K; e \mid K\ e' \mid v\ K \mid (K, e) \mid (v, K) \mid \mathsf{let}\ (x, y) = K\ \mathsf{in}\ e' \\
 & & \mid & K \dotplus e \mid v \dotplus K \mid \mathsf{iter}(K, e, x.e') \mid \mathsf{iter}(v, K, x.e) \\
 & & \mid & \mathsf{if}\ K\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 \mid \mathsf{get}(K, e') \mid \mathsf{get}(v, K) \mid \mathsf{put}(K, e') \mid \mathsf{put}(v, K)
\end{array}
$$

$$
\frac{e \rightsquigarrow_{\mathrm{p}} e'}{(e, h) \rightsquigarrow (e', h)}
\qquad\qquad
\frac{(e, h) \rightsquigarrow_{\mathrm{c}} (e', h')}{(e, h) \rightsquigarrow (e', h')}
\qquad\qquad
\frac{(e, h) \rightsquigarrow (e', h')}{(K[e], h) \rightsquigarrow (K[e'], h')}
$$

Pure Reduction

$$
\begin{array}{ll}
(); e \rightsquigarrow_{\mathrm{p}} e & \mathsf{let}\ (x, y) = (v_1, v_2)\ \mathsf{in}\ e \rightsquigarrow_{\mathrm{p}} e[v_1/x, v_2/y] \\
(\lambda x.e)\ v \rightsquigarrow_{\mathrm{p}} e[v/x] & n \dotplus m \rightsquigarrow_{\mathrm{p}} n + m \\
\mathsf{if\ true\ then}\ e_1\ \mathsf{else}\ e_2 \rightsquigarrow_{\mathrm{p}} e_1 & \mathsf{iter}(0, v, x.e) \rightsquigarrow_{\mathrm{p}} v \\
\mathsf{if\ false\ then}\ e_1\ \mathsf{else}\ e_2 \rightsquigarrow_{\mathrm{p}} e_2 & \mathsf{iter}(n + 1, v, x.e) \rightsquigarrow_{\mathrm{p}} \mathsf{iter}(n, e[v/x], x.e)
\end{array}
$$

Channel Reduction

$$
\begin{array}{rll}
(\mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e, h) & \rightsquigarrow_{\mathrm{c}} (e[\ell/x, \ell/y], h[\ell \mapsto \mathsf{E}]) & \text{if } \ell \notin \operatorname{dom} h \\
(\mathsf{get}(\ell, \lambda x.e), h) & \rightsquigarrow_{\mathrm{c}} ((), h[\ell \mapsto \mathsf{C}(\lambda x.e)]) & \text{if } h\ell = \mathsf{E} \\
(\mathsf{get}(\ell, \lambda x.e), h) & \rightsquigarrow_{\mathrm{c}} (e[v/x], h[\ell \mapsto \mathsf{E}]) & \text{if } h\ell = \mathsf{V}(v) \\
(\mathsf{put}(\ell, v), h) & \rightsquigarrow_{\mathrm{c}} ((), h[\ell \mapsto \mathsf{V}(v)]) & \text{if } h\ell = \mathsf{E} \\
(\mathsf{put}(\ell, v), h) & \rightsquigarrow_{\mathrm{c}} (e[v/x], h[\ell \mapsto \mathsf{E}]) & \text{if } h\ell = \mathsf{C}(\lambda x.e)
\end{array}
$$

Figure 3: Operational Semantics

2 Ordinals

In this work, we only consider ordinals strictly smaller than ω^ω. Each such ordinal, denoted by α, β, γ in the following, can be expressed in its Cantor normal form. That is, for each ordinal $\alpha < \omega^\omega$ there is some $k \in \mathbb{N}$ and coefficients $a_0, \ldots, a_k \in \mathbb{N}$ such that $\alpha = \sum_{i=0}^{k} a_i \omega^i \triangleq \omega^k a_k + \cdots + \omega^0 a_0$. We combine ordinals using *natural addition* [3].

Definition 2.1 (Natural Addition). *Let $\alpha = \sum_{i=0}^{k} \omega^i a_i$ and $\beta = \sum_{i=0}^{l} \omega^i b_i$. We define natural addition by $\alpha \oplus \beta \triangleq \sum_{i=0}^{\max(k,l)} \omega^i (a_i + b_i)$ where $a_i \triangleq 0$ for $i = k+1, \ldots, l$ and $b_i \triangleq 0$ for $i = l+1, \ldots, k$.*

Lemma 2.1. *Natural addition is associative and commutative, and 0 is an identity. Natural addition is compatible with $<$ on ordinals, meaning $\alpha < \beta$ implies $\alpha \oplus \gamma < \beta \oplus \gamma$ for all α, β, γ. Natural addition is cancellative, meaning $\alpha \oplus \gamma = \beta \oplus \gamma$ implies $\alpha = \beta$.*

Proof. Let $\alpha = \sum_{i=0}^{m} \omega^i a_i$, $\beta = \sum_{i=0}^{n} \omega^i b_i$, and $\gamma = \sum_{i=0}^{p} \omega^i c_i$. We define $k \triangleq \max(m, n, p)$ and extend α, β, and γ with zeros by $a_i \triangleq 0$ for $i = m+1, \ldots, k$ and $b_i \triangleq 0$ for $i = n+1, \ldots, k$ and $c_i \triangleq 0$ for $i = p+1, \ldots, k$. As $\omega^i \cdot 0 = 0$ and $0 + \alpha = \alpha$ for all i and α, we have $\alpha = \sum_{i=0}^{k} \omega^i a_i$, $\beta = \sum_{i=0}^{k} \omega^i b_i$, and $\gamma = \sum_{i=0}^{k} \omega^i c_i$.

1. *Associativity.* $\alpha \oplus (\beta \oplus \gamma) = \alpha \oplus \sum_{i=0}^{k} \omega^i (b_i + c_i) = \sum_{i=0}^{k} \omega^i (a_i + (b_i + c_i)) = \sum_{i=0}^{k} \omega^i ((a_i + b_i) + c_i) = \left(\sum_{i=0}^{k} \omega^i (a_i + b_i)\right) \oplus \gamma = (\alpha \oplus \beta) \oplus \gamma$.
2. *Commutativity.* $\alpha \oplus \beta = \sum_{i=0}^{k} \omega^i (a_i + b_i) = \sum_{i=0}^{k} \omega^i (b_i + a_i) = \beta \oplus \alpha$.
3. *Identity.* $\alpha \oplus 0 = \sum_{i=0}^{k} \omega^i (a_i + 0) = \sum_{i=0}^{k} \omega^i a_i = \alpha$.
4. *Compatibility.* As $\alpha < \beta$, we have $\alpha \neq \beta$. Let j be the largest number such that $a_j \neq b_j$. Clearly $a_j < b_j$ as otherwise $\beta > \alpha$. Since $\sum_{i=0}^{j-1} \omega^i (a_i + c_i) < \omega^j$, we have:

$$\begin{aligned}
\alpha \oplus \gamma &= \textstyle\sum_{i=0}^{k} \omega^i (a_i + c_i) \\
&= \left(\textstyle\sum_{i=j+1}^{k} \omega^i (a_i + c_i)\right) + \omega^j (a_j + c_j) + \left(\textstyle\sum_{i=0}^{j-1} \omega^i (a_j + c_j)\right) \\
&= \left(\textstyle\sum_{i=j+1}^{k} \omega^i (b_i + c_i)\right) + \omega^j (a_j + c_j) + \left(\textstyle\sum_{i=0}^{j-1} \omega^i (a_i + c_i)\right) \\
&< \left(\textstyle\sum_{i=j+1}^{k} \omega^i (b_i + c_i)\right) + \omega^j (1 + a_j + c_j) \\
&\leq \left(\textstyle\sum_{i=j+1}^{k} (b_i + c_i) \omega^i\right) + \omega^j (b_j + c_j) \\
&\leq \textstyle\sum_{i=0}^{k} \omega^i (b_i + c_i) = \beta \oplus \gamma
\end{aligned}$$

5. *Cancellation.* We first show that if $\sum_{i=0}^{k} \omega^i m_i = \sum_{i=0}^{k} \omega^i n_i$, then $m_i = n_i$ for all $i = 0, \ldots, k$ by induction on k. For $k = 0$ the claim is trivial. For $k > 0$, we first show $m_k = n_k$. By way of contradiction assume $m_k \neq n_k$. Without loss of generality, let $m_k > n_k$. Then $\sum_{i=0}^{k} \omega^i m_i \geq \omega^k m_k \geq \omega^k (n_k + 1) = \omega^k n_k + \omega^k > \omega^k n_k + \sum_{i=0}^{k-1} \omega^i n_i =$

$\sum_{i=0}^{k} \omega^i n_i$, a contradiction. Thus $m_k = n_k$. By left cancellation of ordinal addition, we obtain $\sum_{i=0}^{k-1} \omega^i m_i = \sum_{i=0}^{k-1} \omega^i n_i$. By induction $m_i = n_i$ for all $i = 0, \ldots, k-1$.

Let $\alpha \oplus \gamma = \beta \oplus \gamma$. By assumption $\sum_{i=0}^{k} \omega^i (a_i + c_i) = \alpha \oplus \gamma = \beta \oplus \gamma = \sum_{i=0}^{k} \omega^i (b_i + c_i)$. Thus, we have $a_i + c_i = b_i + c_i$ for all $i = 0, \ldots, k$. The claim follows with right cancellation of $+$ on natural numbers.

□

Natural addition gives rise to *natural multiplication*, a commutative, associative multiplication operation which distributes over natural addition. In the present work, we resort to two special cases, multiplication by natural numbers and multiplication by ω.

Definition 2.2 (Natural Multiplication). *Let* $\alpha = \sum_{i=0}^{k} \omega^i a_i$. *We define* $n \otimes \alpha \triangleq \sum_{i=0}^{k} \omega^i a_i n$ *and* $\omega \otimes \alpha \triangleq \sum_{i=0}^{k} \omega^{i+1} a_i$.

Lemma 2.2. *Natural multiplication has the following properties:*

$$0 \otimes \alpha = 0 \qquad 1 \otimes \alpha = \alpha \qquad (n_1 + n_2) \otimes \alpha = n_1 \otimes \alpha \oplus n_2 \otimes \alpha \qquad n \otimes \alpha \leq \omega \otimes \alpha$$

Proof. Let $\alpha = \sum_{i=0}^{m} \omega^i a_i$.

1. $0 \otimes \alpha = 0$. Immediate from the definition.
2. $1 \otimes \alpha = \alpha$. Immediate from the definition.
3. $(n_1 + n_2) \otimes \alpha = n_1 \otimes \alpha \oplus n_2 \otimes \alpha$. We have $(n_1 + n_2) \otimes \alpha = \sum_{i=0}^{m} \omega^i a_i (n_1 + n_2) = \sum_{i=0}^{m} \omega^i (a_i n_1 + a_i n_2) = (\sum_{i=0}^{m} \omega^i a_i n_1) \oplus (\sum_{i=0}^{m} \omega^i a_i n_2) = (n_1 \otimes \alpha) \oplus (n_2 \otimes \alpha)$.
4. $n \otimes \alpha \leq \omega \otimes \alpha$. If $\alpha = 0$, the claim is trivial. If $\alpha > 0$, then one of the coefficients $a_0, \ldots, a_m$ must be larger than zero. Let k be the largest k such that $a_k > 0$. Then $n \otimes \alpha = \sum_{i=0}^{m} \omega^i a_i n = \sum_{i=0}^{k} \omega^i a_i n < \omega^{k+1} \leq \sum_{i=0}^{k} \omega^{i+1} a_i = \sum_{i=0}^{m} \omega^{i+1} a_i = \omega \otimes \alpha$.

□

3 Resources

$$
\begin{array}{llll}
\text{Capabilities} & p, q & ::= & \mathsf{get}(\ell) \mid \mathsf{put}(\ell) \mid \mathsf{al}(\ell) \\
\text{Capability Sets} & C, D & ::= & \emptyset \mid C \uplus \{p\} \\
\text{Invariant Maps} & \Phi & ::= & \emptyset \mid \Phi, \ell : A \\
\text{Resources} & R & ::= & (C, \Phi, \alpha) \mid \lightning \\
\text{Resource Maps} & \rho & ::= & \emptyset \mid \rho, \ell \mapsto R \\
\text{Step-Indices} & i, j, k & ::= & (\alpha, C)
\end{array}
$$

In the logical relation, we incorporate the linearity of the type system in the form of *resources* [2, 4]. In our setting, each resource R is either invalid, meaning $R = \lightning$, or R is a triple (C, Φ, α) where C is a set of capabilities, Φ an invariant map, and α an ordinal. In the logical relation, we relate values, expressions, and heaps with the resources they *own*. Intuitively, we interpret owning a resource $R = (C, \Phi, \alpha)$ as the knowledge that the invariants $\ell : A \in \Phi$ are enforced in the heap typing relation, the right to execute instructions corresponding to capabilities in C, and the right to allocate α new channels. The capability $\mathsf{get}(\ell)$ corresponds to the right to receive a value on channel ℓ, the capability $\mathsf{put}(\ell)$ to the right to send a value over channel ℓ, and the capability $\mathsf{al}(\ell)$ to the right to physically allocate ℓ, meaning to the right to add ℓ to the heap. To combine two resources R and R', we combine their components.

Definition 3.1 (Resource Addition)**.**

$$
\begin{array}{rll}
(C, \Phi, \alpha) \oplus (C', \Phi', \alpha') \triangleq (C \cup C', \Phi \cup \Phi', \alpha \oplus \alpha') & & \text{if } C \# C' \text{ and } \Phi \,\mathsf{ag}\, \Phi' \\
R \oplus R' \triangleq \lightning & & \text{otherwise}
\end{array}
$$

where we write $C \# C'$, if C and C' are disjoint, meaning $C \cap C' = \emptyset$ and we write $\Phi \,\mathsf{ag}\, \Phi'$, if two invariant maps Φ and Φ' agree, meaning $\forall\, \ell \in \operatorname{dom} \Phi \cap \operatorname{dom} \Phi'.\ \Phi\ell = \Phi'\ell$.

We write $\checkmark R$, if R is a valid resource, meaning $R \neq \lightning$. A resource R is a subresource of R', written $R \sqsubseteq R'$, if there exists some resource R'' with $R' = R \oplus R''$. We denote the empty resource by $\epsilon \triangleq (\emptyset, \emptyset, 0)$. In the following, it will sometimes be convenient to consider resources which only consist of capabilities $R_{\mathsf{cap}(C)} \triangleq (C, \emptyset, 0)$, invariants $R_{\mathsf{inv}(\Phi)} \triangleq (\emptyset, \Phi, 0)$, or ordinals $R_{\mathsf{ord}(\alpha)} \triangleq (\emptyset, \emptyset, \alpha)$. We write $R_{\mathsf{cap}(p_1, \ldots, p_n)}$ for the resource $R_{\mathsf{cap}(\{p_1, \ldots, p_n\})}$ and $R_{\mathsf{inv}(\ell_1 : A_1, \ldots, \ell_n : A_n)}$ for the resource $R_{\mathsf{inv}(\{\ell_1 : A_1, \ldots, \ell_n : A_n\})}$.

In the remainder of this section, we establish some basic properties about the use of invariants and resources in general.

Lemma 3.1.

1. *If $\Phi_1 \,\mathsf{ag}\, \Phi_2$, then $\Phi_1 \cup \Phi_2$ is an invariant map. Further, $\operatorname{dom}(\Phi_1 \cup \Phi_2) = \operatorname{dom} \Phi_1 \cup \operatorname{dom} \Phi_2$ and $(\Phi_1 \cup \Phi_2)\ell = \Phi_1\ell$ for all $\ell \in \operatorname{dom} \Phi_1$ and $(\Phi_1 \cup \Phi_2)\ell = \Phi_2\ell$ for all $\ell \in \operatorname{dom} \Phi_2$.*
2. *$\Phi_1 \,\mathsf{ag}(\Phi_2 \cup \Phi_3)$ and $\Phi_2 \,\mathsf{ag}\, \Phi_3$ iff $\Phi_1 \,\mathsf{ag}\, \Phi_2$ and $\Phi_1 \,\mathsf{ag}\, \Phi_3$ and $\Phi_2 \,\mathsf{ag}\, \Phi_3$.*
3. *If $\Phi_1 \,\mathsf{ag}\, \Phi_2$, then $\Phi_2 \,\mathsf{ag}\, \Phi_1$.*
4. *If $\Phi_1 \subseteq \Phi_2$, then $\Phi_1 \,\mathsf{ag}\, \Phi_2$.*
5. *$\Phi \,\mathsf{ag}\, \emptyset$.*

Proof.

1. Assume $\Phi_1 \operatorname{ag} \Phi_2$, that is $\forall \ell \in \operatorname{dom} \Phi_1 \cap \Phi_2 . \Phi_1 \ell = \Phi_2 \ell$. Then $\Phi_1 \cup \Phi_2$ is a function. Clearly $\operatorname{dom}(\Phi_1 \cup \Phi_2) = \operatorname{dom} \Phi_1 \cup \operatorname{dom} \Phi_2$. If $\ell \in \operatorname{dom} \Phi_1$, then $(\Phi_1 \cup \Phi_2)\ell = \Phi_1 \ell$ regardless of whether $\ell \in \operatorname{dom} \Phi_2$ or not with $\Phi_1 \operatorname{ag} \Phi_2$. Similarly, if $\ell \in \operatorname{dom} \Phi_2$, then $(\Phi_1 \cup \Phi_2)\ell = \Phi_2 \ell$ regardless of whether $\ell \in \operatorname{dom} \Phi_1$ or not with $\Phi_1 \operatorname{ag} \Phi_2$.

2. Let $\Phi_1 \operatorname{ag}(\Phi_2 \cup \Phi_3)$ and $\Phi_2 \operatorname{ag} \Phi_3$. With the first claim we know that $\Phi_2 \cup \Phi_3$ is an invariant map. We show $\Phi_1 \operatorname{ag} \Phi_2$. Let $\ell \in \operatorname{dom} \Phi_1 \cap \operatorname{dom} \Phi_2$. Then $\Phi_1 \ell = (\Phi_2 \cup \Phi_3)\ell = \Phi_2 \ell$ by the first claim. Analogously $\Phi_1 \operatorname{ag} \Phi_3$.

   Let $\Phi_1 \operatorname{ag} \Phi_2$ and $\Phi_1 \operatorname{ag} \Phi_3$ and $\Phi_2 \operatorname{ag} \Phi_3$. Let $\ell \in \operatorname{dom} \Phi_1 \cap (\operatorname{dom} \Phi_2 \cup \operatorname{dom} \Phi_3)$. If $\ell \in \operatorname{dom} \Phi_2$, then $\Phi_1 \ell = \Phi_2 \ell = (\Phi_2 \cup \Phi_3)\ell$ by the first claim. If $\ell \in \operatorname{dom} \Phi_3$, then $\Phi_1 \ell = \Phi_3 \ell = (\Phi_2 \cup \Phi_3)\ell$ by the first claim.

3. Let $\Phi_1 \operatorname{ag} \Phi_2$ and $\ell \in \operatorname{dom} \Phi_2 \cap \operatorname{dom} \Phi_1$. Then $\ell \in \operatorname{dom} \Phi_1 \cap \operatorname{dom} \Phi_2$. The claim follows.

4. Since $\Phi_1 \subseteq \Phi_2$, we have $\Phi_1 \ell = \Phi_2 \ell$ for all $\ell \in \operatorname{dom} \Phi_1 \subseteq \operatorname{dom} \Phi_2$.

5. The set $\operatorname{dom} \Phi \cap \operatorname{dom} \emptyset$ is empty. Thus, the claim holds trivially.

□

Lemma 3.2. *The resource addition $\oplus$ is associative, commutative, and ϵ is an identity. The subresource relation $\sqsubseteq$ is reflexive, transitive, and $\epsilon \sqsubseteq R$ and $R \sqsubseteq R \oplus R'$ for all R, R'. Validity extends to subresources, meaning if $\checkmark R$ and $R' \sqsubseteq R$, then $\checkmark R'$. The subresource relation is compatible with resource addition, meaning if $R \sqsubseteq R'$, then $R \oplus R_f \sqsubseteq R' \oplus R_f$. If $(C, \Phi, \alpha) \sqsubseteq (C', \Phi', \alpha')$, then $C \subseteq C'$ and $\Phi \subseteq \Phi'$ and $\alpha \leq \alpha'$.*

Proof.

1. *Associativity* $\oplus$. Let R_1, R_2, R_3 be resources. We show $R_1 \oplus (R_2 \oplus R_3) = (R_1 \oplus R_2) \oplus R_3$. If both sides are $\lightning$, the claim follows.

   Case $R_1 \oplus (R_2 \oplus R_3) = (C, \Phi, \alpha)$ for some C, Φ, α. Then by definition of $\oplus$ we have $C = C_1 \cup C_2 \cup C_3$ and $\Phi = \Phi_1 \cup \Phi_2 \cup \Phi_3$ and $\alpha = \alpha_1 \oplus \alpha_2 \oplus \alpha_3$ where $R_1 = (C_1, \Phi_1, \alpha_1)$ and $R_2 = (C_2, \Phi_2, \alpha_2)$ and $R_3 = (C_3, \Phi_3, \alpha_3)$ such that $C_1 \#(C_2 \cup C_3)$ and $C_2 \# C_3$ and $\Phi_1 \operatorname{ag}(\Phi_2 \cup \Phi_3)$ and $\Phi_2 \operatorname{ag} \Phi_3$. Thus, $C_1 \# C_2$ and $C_1 \# C_3$. Hence $(C_1 \cup C_2) \# C_3$. Further, by Lemma 3.1 we have $\Phi_1 \operatorname{ag} \Phi_2$ and $\Phi_1 \operatorname{ag} \Phi_3$. Thus, $(\Phi_1 \cup \Phi_2) \operatorname{ag} \Phi_3$. With with Lemma 2.1, we obtain: $(R_1 \oplus R_2) \oplus R_3 = ((C_1 \cup C_2) \cup C_3, (\Phi_1 \cup \Phi_2) \cup \Phi_3, (\alpha_1 \oplus \alpha_2) \oplus \alpha_3) = (C_1 \cup (C_2 \cup C_3), \Phi_1 \cup (\Phi_2 \cup \Phi_3), \alpha_1 \oplus (\alpha_2 \oplus \alpha_3)) = R_1 \oplus (R_2 \oplus R_3)$

   Case $(R_1 \oplus R_2) \oplus R_3 = (C, \Phi, \alpha)$ for some C, Φ, α. Then by definition of $\oplus$ we have $C = C_1 \cup C_2 \cup C_3$ and $\Phi = \Phi_1 \cup \Phi_2 \cup \Phi_3$ and $\alpha = \alpha_1 \oplus \alpha_2 \oplus \alpha_3$ where $R_1 = (C_1, \Phi_1, \alpha_1)$ and $R_2 = (C_2, \Phi_2, \alpha_2)$ and $R_3 = (C_3, \Phi_3, \alpha_3)$ such that $(C_1 \cup C_2) \# C_3$ and $C_1 \# C_2$ and $(\Phi_1 \cup \Phi_2) \operatorname{ag} \Phi_3$ and $\Phi_1 \operatorname{ag} \Phi_2$. Thus, $C_1 \# C_3$ and $C_2 \# C_3$. Hence $C_1 \#(C_2 \cup C_3)$. Further, by Lemma 3.1 $\Phi_1 \operatorname{ag} \Phi_3$ and $\Phi_2 \operatorname{ag} \Phi_3$ and thus $\Phi_1 \operatorname{ag}(\Phi_2 \cup \Phi_3)$. With with Lemma 2.1, we obtain: $R_1 \oplus (R_2 \oplus R_3) = (C_1 \cup (C_2 \cup C_3), \Phi_1 \cup (\Phi_2 \cup \Phi_3), \alpha_1 \oplus (\alpha_2 \oplus \alpha_3)) = ((C_1 \cup C_2) \cup C_3, (\Phi_1 \cup \Phi_2) \cup \Phi_3, (\alpha_1 \oplus \alpha_2) \oplus \alpha_3) = (R_1 \oplus R_2) \oplus R_3$

2. *Commutativity* $\oplus$. Let R_1, R_2 be resources. We show $R_1 \oplus R_2 = R_2 \oplus R_1$. If $R_1 = \lightning$ or $R_2 = \lightning$, the claim is trivial. Let $R_1 = (C_1, \Phi_1, \alpha_1)$ and $R_2 = (C_2, \Phi_2, \alpha_2)$. If not $C_1 \# C_2$, then $R_1 \oplus R_2 = \lightning = R_2 \oplus R_1$. If not Φ_1 ag Φ_2, then by Lemma 3.1 not Φ_2 ag Φ_1 and thus $R_1 \oplus R_2 = \lightning = R_2 \oplus R_1$. Otherwise, $R_1 \oplus R_2 = (C_1 \cup C_2, \Phi_1 \cup \Phi_2, \alpha_1 \oplus \alpha_2) = (C_2 \cup C_1, \Phi_2 \cup \Phi_1, \alpha_2 \oplus \alpha_1) = R_2 \oplus R_1$ with Lemma 2.1.

3. *Identity* $\oplus$. Let R be some resource. We show $R \oplus \epsilon = R$. If $R = \lightning$, the claim is trivial. If $R = (C, \Phi, \alpha)$, we have $R \oplus \epsilon = (C \cup \emptyset, \Phi \cup \emptyset, \alpha \oplus 0) = (C, \Phi, \alpha) = R$ since $C \# \emptyset$ and Φ ag $\emptyset$ by Lemma 3.1.

4. *Reflexivity* $\sqsubseteq$. For any resource R, we have $R \oplus \epsilon = R$. Thus $R \sqsubseteq R$.

5. *Transitivity* $\sqsubseteq$. Let $R_1 \sqsubseteq R_2$ and $R_2 \sqsubseteq R_3$. Then there are R_f and R_g such that $R_1 \oplus R_f = R_2$ and $R_2 \oplus R_g = R_3$. Thus $R_1 \oplus (R_f \oplus R_g) = R_3$ and thus $R_1 \sqsubseteq R_3$.

6. $\epsilon \sqsubseteq R$. Follows with $\epsilon \oplus R = R$.

7. *Validity.* Assume $\checkmark R$ and $R' \sqsubseteq R$. Then there is some R_f such that $R = R' \oplus R_f$. Since $\checkmark R$, we have $R = (C, \Phi, \alpha)$ for some C, Φ, α. Thus $R' = (C', \Phi', \alpha')$ for some C', Φ', α' since $R = R' \oplus R_f$.

8. *Compatibility.* Let $R \sqsubseteq R'$. Then there is some R'' such that $R \oplus R'' = R'$. Hence $R \oplus R_f \oplus R'' = R \oplus R'' \oplus R_f = R' \oplus R_f$. Thus $R \oplus R_f \sqsubseteq R' \oplus R_f$.

9. *Decomposition.* Let $(C, \Phi, \alpha) \sqsubseteq (C', \Phi', \alpha')$. Then there is some R_f such that $(C', \Phi', \alpha') = (C, \Phi, \alpha) \oplus R_f$. By definition of $\oplus$, we have $R_f = (C_f, \Phi_f, \alpha_f)$ for some C_f, Φ_f, α_f such that $C' = C \cup C_f$ and $\Phi' = \Phi \cup \Phi_f$ and $\alpha' = \alpha \oplus \alpha_f$. Since $0 \leq \alpha_f$, we have $\alpha = \alpha \oplus 0 \leq \alpha \oplus \alpha_f = \alpha'$.

□

Lemma 3.3.

1. $(C, \Phi, \alpha) = R_{cap(C)} \oplus R_{inv(\Phi)} \oplus R_{ord(\alpha)}$
2. *If* $\checkmark R_1 \oplus R_{cap(C)}$ *and* $R_2 \oplus R_{cap(C)} = R_1 \oplus R_{cap(C)}$, *then* $R_2 = R_1$.
3. *If* $\checkmark R_1 \oplus R_{cap(C)}$ *and* $R_2 \oplus R_{cap(C)} \sqsubseteq R_1 \oplus R_{cap(C)}$, *then* $R_2 \sqsubseteq R_1$.
4. *If* $R_2 \oplus R_{ord(\alpha)} = R_1 \oplus R_{ord(\alpha)}$, *then* $R_2 = R_1$.
5. *If* $R_2 \oplus R_{ord(\alpha)} \sqsubseteq R_1 \oplus R_{ord(\alpha)}$, *then* $R_2 \sqsubseteq R_1$.
6. *If* $\Phi \subseteq \Psi$, *then* $R_{inv(\Phi)} \oplus R_{inv(\Psi)} = R_{inv(\Psi)}$.
7. $\checkmark R$ *iff* $\checkmark R \oplus R_{ord(\alpha)}$.

Proof.

1. By definition since $C \# \emptyset$ and Φ ag $\emptyset$ by Lemma 3.1.

2. Assuming $\checkmark R_1 \oplus R_{\mathrm{cap}(C)}$, we have $R_1 = (C_1, \Phi_1, \alpha_1)$ for some C_1, Φ_1, α_1 such that $C_1 \# C$. Further, $R_2 = (C_2, \Phi_2, \alpha_2)$ such that $C_2 \# C$. By assumption $(C_1 \cup C, \Phi_1, \alpha_1) =$

$(C_2 \cup C, \Phi_2, \alpha_2)$. Since $C_1 \# C$ and $C_2 \# C$, we have $C_1 = C_2$.

3. By assumption $R_2 \oplus R_{\mathsf{cap}(C)} \oplus R_f = R_1 \oplus R_{\mathsf{cap}(C)}$. Thus by the second claim, $R_2 \oplus R_f = R_1$. Hence $R_2 \sqsubseteq R_1$.
4. If $R_1 = \lightning$, then $R_2 = \lightning$ and the claim follows. If $R_2 = \lightning$, then $R_1 = \lightning$ and the claim follows. Let $R_1 = (C_1, \Phi_1, \alpha_1)$ and $R_2 = (C_2, \Phi_2, \alpha_2)$ for some $C_1, C_2, \Phi_1, \Phi_2, \alpha_1, \alpha_2$. By assumption $(C_1, \Phi_1, \alpha_1 \oplus \alpha) = (C_2, \Phi_2, \alpha_2 \oplus \alpha)$. The claim follows with right cancellation of natural addition, see Lemma 2.1.
5. By assumption $R_2 \oplus R_{\mathsf{ord}(\alpha)} \oplus R_f = R_1 \oplus R_{\mathsf{ord}(\alpha)}$. Thus by the second claim, $R_2 \oplus R_f = R_1$. Hence $R_2 \sqsubseteq R_1$.
6. If $\Phi \subseteq \Psi$, then Φ ag Ψ by Lemma 3.1. Thus $R_{\mathsf{inv}(\Phi)} \oplus R_{\mathsf{inv}(\Psi)} = R_{\mathsf{inv}(\Phi \cup \Psi)} = R_{\mathsf{inv}(\Psi)}$.
7. Follows by case analysis on R.

□

4 Logical Relation

In this section, we define the logical relation of the language λ_{CHAN}. In Figure 4, we give the definition of the type interpretations. In Section 4.1, we prove that the type interpretations are Kripke logical relations. In Section 4.2 and Section 4.3, we prove several properties about the type interpretations which enable compact proofs of the compatibility lemmas. In Section 4.4, we define the semantic typing judgement $\Gamma \vDash e : A$ and in Section 4.5, we prove the compatibility lemmas required to prove that its a sound model of the syntactic typing judgement $\Gamma \vdash e : A$.

Step-Indices As step-indices, we use pairs of ordinals and sets of capabilities $i, j, k ::= (\alpha, C)$. On step-indices, we define the lexicographic ordering:

$$(\alpha, C) < (\alpha', C') \triangleq \alpha < \alpha' \vee (\alpha = \alpha' \wedge C \subsetneq C')$$

and as usual define $i \leq j \triangleq i < j \vee i = j$.

We define the value relation $\mathcal{V}[\![A]\!]_i$, the heap typing $\mathcal{H}[\![\Phi]\!]_i$, and the expression relation $\mathcal{E}[\![A]\!]_i$ by recursion on the step-index i. For a fixed step-index i, we define the value relation $\mathcal{V}[\![A]\!]_i$ and the expression relation $\mathcal{E}[\![A]\!]_i$ by recursion on the type A. More precisely, at step-index i the heap typing $\mathcal{H}[\![\Phi]\!]_i$ depends on $\mathcal{V}[\![B]\!]_j$ for $j < i$ and arbitrary types B, the value relation $\mathcal{V}[\![A]\!]_i$ depends on $\mathcal{E}[\![B]\!]_j$ for $j \leq i$ and B structurally smaller than A, and the expression relation $\mathcal{E}[\![A]\!]_i$ depends on $\mathcal{V}[\![A]\!]_j$ for $j \leq i$ and on $\mathcal{H}[\![\Phi]\!]_j$ for $j \leq i$ and arbitrary Φ.

Lemma 4.1. *If* $(\alpha, C) \leq (\alpha', C')$, *then* $(\alpha, C \setminus D) \leq (\alpha', C' \setminus D)$.

Proof. If $\alpha < \alpha'$, then $(\alpha, C \setminus D) \leq (\alpha, C) < (\alpha', C' \setminus D)$. If $\alpha = \alpha'$, then $C \subseteq C'$. Hence $C \setminus D \subseteq C' \setminus D$. Thus $(\alpha, C \setminus D) \leq (\alpha, C' \setminus D) = (\alpha', C' \setminus D)$. □

Value Relation In the value relation $\mathcal{V}[\![A]\!]_i$, we relate values that semantically inhabit the type A with the resources they own. Unit, Booleans, and natural numbers only own the empty resource ϵ. Inhabitants of $\mathsf{Get}\, A$ are locations ℓ that own the resource $R_{\mathsf{get}(\ell,A)} \triangleq (\{\mathsf{get}(\ell)\}, \{\ell : A\}, 0)$. Ownership of $\mathsf{get}(\ell)$ entails the right to perform a get on location ℓ and ownership of $\ell : A$ guarantees that the invariant $\ell : A$ is satisfied by the heap in the heap typing relation $\mathcal{H}[\![\Phi]\!]_i$. Analogously, inhabitants of $\mathsf{Put}\, A$ are locations ℓ that own the resource $R_{\mathsf{put}(\ell,A)} \triangleq (\{\mathsf{put}(\ell)\}, \{\ell : A\}, 0)$. For linear pairs $A \otimes B$, we combine the resources owned by the individual components. In the interpretation of the linear function type $A \multimap B$, a function is related to those resources that are required to execute the body safely, provided the resources owned by the argument are added.

Logical State There is no physical difference between a channel in its initial state and a channel after it has been used to exchange a value: in both cases, the channel location stores the value E. However, there is a logical difference, which matters for keeping track of resource

Step-Indices and Logical State

$$\begin{array}{llcl} \text{Step-Indices} & i, j, k & ::= & (\alpha, C) \\ \text{Logical State} & s & ::= & \mathsf{Start} \mid \mathsf{Cont} \mid \mathsf{Val} \mid \mathsf{Done} \\ \text{Logical State Map} & \sigma & ::= & \emptyset \mid \sigma, \ell \mapsto s \end{array}$$

$$\begin{array}{llll} \sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Start}] & \text{if } \ell \notin \operatorname{dom} \sigma & \sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Done}] & \text{if } \sigma\ell = \mathsf{Val} \\ \sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Val}] & \text{if } \sigma\ell = \mathsf{Start} & \sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Done}] & \text{if } \sigma\ell = \mathsf{Cont} \\ \sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Cont}] & \text{if } \sigma\ell = \mathsf{Start} & & \end{array}$$

Value Relation

$$\mathcal{V}\llbracket \mathbb{1} \rrbracket_i \triangleq \{((), \epsilon)\} \qquad \mathcal{V}\llbracket \mathbb{B} \rrbracket_i \triangleq \{(\mathsf{true}, \epsilon), (\mathsf{false}, \epsilon)\} \qquad \mathcal{V}\llbracket \mathbb{N} \rrbracket_i \triangleq \{(n, \epsilon) \mid n \in \mathbb{N}\}$$

$$\mathcal{V}\llbracket \mathsf{Get}\, A \rrbracket_i \triangleq \{(\ell, (\{\mathsf{get}(\ell)\}, \{\ell : A\}, 0))\} \qquad \mathcal{V}\llbracket \mathsf{Put}\, A \rrbracket_i \triangleq \{(\ell, (\{\mathsf{put}(\ell)\}, \{\ell : A\}, 0))\}$$

$$\mathcal{V}\llbracket A \otimes B \rrbracket_i \triangleq \{((v_1, v_2), R_1 \oplus R_2) \mid (v_1, R_1) \in \mathcal{V}\llbracket A \rrbracket_i \text{ and } (v_2, R_2) \in \mathcal{V}\llbracket B \rrbracket_i\}$$

$$\mathcal{V}\llbracket A \multimap B \rrbracket_i \triangleq \{(\lambda x.e, R_e) \mid \forall j \leq i, (v, R_v) \in \mathcal{V}\llbracket A \rrbracket_j.\ (e[v/x], R_v \oplus R_e) \in \mathcal{E}\llbracket B \rrbracket_j\}$$

Heap Typing

$$\mathcal{H}\llbracket \Phi \rrbracket_i \triangleq \left\{ (h, \sigma, \rho) \;\middle|\; \begin{array}{l} \operatorname{dom} \Phi = \operatorname{dom} h = \operatorname{dom} \sigma = \operatorname{dom} \rho \text{ and} \\ \forall \ell : A \in \Phi.\ (h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}\llbracket \ell : A \rrbracket_i \end{array} \right\}$$

$$\begin{aligned} \mathcal{HV}\llbracket \ell : A \rrbracket_{\alpha,C} \triangleq{} & \{(\mathsf{E}, \mathsf{Start}, \epsilon), (\mathsf{E}, \mathsf{Done}, \epsilon)\} \\ & \cup \left\{ (\mathsf{V}(v), \mathsf{Val}, R) \;\middle|\; \exists C'.\ C = C' \uplus \{\mathsf{put}(\ell)\} \text{ and } (v, R) \in \mathcal{V}\llbracket A \rrbracket_{\alpha,C'} \right\} \\ & \cup \left\{ (\mathsf{C}(v), \mathsf{Cont}, R) \;\middle|\; \exists C'.\ C = C' \uplus \{\mathsf{get}(\ell)\} \text{ and } (v, R) \in \mathcal{V}\llbracket A \multimap \mathbb{1} \rrbracket_{\alpha,C'} \right\} \end{aligned}$$

Expression Relation

$$\mathcal{E}\llbracket A \rrbracket_i \triangleq \left\{ (e, R_e) \;\middle|\; \begin{array}{l} \forall j \leq i, R_f, \Phi, h, \sigma, \rho.\ \mathsf{RI}_j(R_e, R_f, \Phi, h, \sigma, \rho) \Rightarrow \\ \exists k \leq j, \Phi' \supseteq \Phi, h', \sigma', \rho', v, R_v.\ \mathsf{RI}_k(R_v, R_f, \Phi', h', \sigma', \rho') \\ \text{and } (e, h) \rightsquigarrow^* (v, h') \text{ and } \sigma \dashrightarrow^* \sigma' \text{ and } (v, R_v) \in \mathcal{V}\llbracket A \rrbracket_k \end{array} \right\}$$

where

$$\begin{aligned} \mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho) \triangleq{} & (h, \sigma, \rho) \in \mathcal{H}\llbracket \Phi \rrbracket_{\alpha,C} \text{ and } \mathsf{used}\, \sigma \,\#\, C \text{ and} \\ & \exists D.C = D \uplus \mathsf{idx}\, \sigma \text{ and } R_e \oplus R_f \oplus \textstyle\bigoplus_{\ell \in \operatorname{dom} \rho} \rho\ell \sqsubseteq (D, \Phi, \alpha) \end{aligned}$$

$$\begin{aligned} \mathsf{idx}(\ell, \mathsf{Val}) &= \{\mathsf{put}(\ell)\} \\ \mathsf{idx}(\ell, \mathsf{Cont}) &= \{\mathsf{get}(\ell)\} \\ \mathsf{idx}(\ell, s) &= \emptyset \qquad \text{othw.} \\ \mathsf{idx}\, \sigma &= \bigcup_{\ell \in \operatorname{dom} \sigma} \mathsf{idx}(\ell, \sigma\ell) \end{aligned} \qquad \begin{aligned} \mathsf{used}(\ell, \mathsf{Done}) &= \{\mathsf{al}(\ell), \mathsf{get}(\ell), \mathsf{put}(\ell)\} \\ \mathsf{used}(\ell, s) &= \{\mathsf{al}(\ell)\} \qquad \text{othw.} \\ \mathsf{used}\, \sigma &= \bigcup_{\ell \in \operatorname{dom} \sigma} \mathsf{used}(\ell, \sigma\ell) \end{aligned}$$

Figure 4: Logical Relation

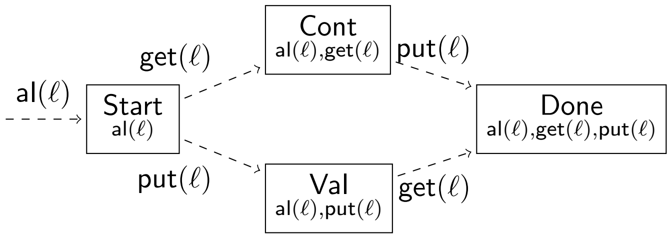
usage in our logical relation. Thus, to distinguish both states, we introduce the notion of the *logical state* of a channel. The logical state of every channel in the heap is tracked in a state map σ. We evolve it according to the transition relation $\sigma \dashrightarrow \sigma'$. Each fresh channel starts in the state Start and is then advanced to Done, either by storing a value or a continuation in between.

Heap Typing The heap typing relation $\mathcal{H}[\![\Phi]\!]_i$ ensures that the values in the heap satisfy the invariants $\ell : A \in \Phi$. More precisely, the heap typing relation $\mathcal{H}[\![\Phi]\!]_i$ relates the logical state σ with the physical heap h and the resources owned by values in the heap ρ such that the invariants in Φ are upheld.

Expression Relation In the expression relation $\mathcal{E}[\![A]\!]_i$, to execute an expression e owning resource R_e, we assume some global invariant map Φ, a current logical state map σ, a current heap h, and a current resource map ρ. To remain compositional, following the approach of Ahmed et al. [1], we additionally assume some *frame* resource R_f, representing the resource owned by a potential context in which e could be executed. We then show that the *resource interpretation* RI is preserved during the execution with a potential decrease in the step-index. That is, we assume the resource interpretation is initially satisfied and show that at the end of the execution the resource interpretation is satisfied for some resource R_v owned by the result v, some extended invariant map Φ', some updated logical state map σ', some updated heap h', and some updated resource map ρ'. After the execution, we ensure that the logical state map was advanced according to the transition relation $\dashrightarrow$ and that the result v is contained in the value relation.

The resource interpretation $\mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho)$ serves three purposes: First, it ensures that the heap is well-typed given the current invariant map Φ with $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$. Second, with $R_e \oplus R_f \oplus \bigoplus_{\ell \in \mathrm{dom}\,\rho} \rho\ell \sqsubseteq (D, \Phi, \alpha)$ we ensure *ownership* of resources has the intuitive meaning. For the resources owned by different program components, the invariant maps must all agree and be contained in the global invariant map Φ, the capability sets must be pairwise disjoint and contained in the step-index since $D \subseteq C$, and the sum of all ordinals must be at most α. The latter guarantees that we can compositionally decrease the global step-index by locally decreasing the ordinal in the resource of an expression. For example, if the expression resource is $R_e = R' \oplus R_{\mathrm{ord}(1)}$, we can decrease α and allocate new capabilities by giving up the resource $R_{\mathrm{ord}(1)}$.

Third, the resource interpretation enforces that capabilities are used according to the following state transition system with tokens [5], where "tokens" in our case are capabilities:



Below every state are the capabilities currently *owned* by that state. Formally, for a state map σ, the capabilities currently owned by σ are given by $\mathsf{idx}\,\sigma \cup \mathsf{used}\,\sigma$. The functions $\mathsf{idx}\,\sigma$ and $\mathsf{used}\,\sigma$ distinguish between two modes in which capabilities can be owned by σ, depending on whether they are still contained in the step-index or not. We transfer the capability $\mathsf{put}(\ell)$ (resp. $\mathsf{get}(\ell)$) to $\mathsf{idx}\,\sigma$ at the point where a value (resp. continuation) is stored in the heap. We move them to $\mathsf{used}\,\sigma$ at the point where the continuation or value is retrieved during the execution. If the state map σ owns capabilities, then no program component can own those capabilities since $R_e \oplus R_f \oplus \bigoplus_{\ell \in \operatorname{dom} \rho} \rho\ell \sqsubseteq (D, \Phi, \alpha)$ where $D \# \mathsf{idx}\,\sigma$ and $D \# \mathsf{used}\,\sigma$.

Lemma 4.2.

1. $\mathsf{idx}(\ell, s) \# \mathsf{used}(\ell, s)$
2. *If* $\ell \neq \ell'$*, then* $\mathsf{idx}(\ell, s) \# \mathsf{used}(\ell', s')$.
3. $R_{\mathit{cap}(\mathsf{idx}\,\sigma)} = \bigoplus_{\ell \in \operatorname{dom} \sigma} R_{\mathit{cap}(\mathsf{idx}(\ell, \sigma\ell))}$.
4. *If* $\ell \in \operatorname{dom} \sigma$*, then* $\mathsf{al}(\ell) \in \mathsf{used}\,\sigma$.
5. $\mathsf{al}(\ell) \notin \mathsf{idx}(\ell, s)$ *and* $\mathsf{al}(\ell) \notin \mathsf{idx}\,\sigma$.

Proof.

1. Immediate from the definition of idx and used.
2. Immediate from the definition of idx and used since the capabilities in $\mathsf{idx}(\ell, s)$ only mention the location ℓ and the capabilities in $\mathsf{used}(\ell', s')$ only mention the location ℓ'.
3. For any location ℓ, the set $\mathsf{idx}(\ell, \sigma\ell)$ contains only capabilities with the location ℓ. Thus

$$\mathsf{idx}(\ell, \sigma\ell) \# \bigcup_{\ell' \in L, \ell \neq \ell'} \mathsf{idx}(\ell', \sigma\ell')$$

   for all $\ell \in \operatorname{dom} \sigma$ and $L \subseteq \operatorname{dom} \sigma$. Thus, $\bigoplus_{\ell \in \operatorname{dom} \sigma} R_{\mathsf{cap}(\mathsf{idx}(\ell, \sigma\ell))}$ is valid and:

$$\bigoplus_{\ell \in \operatorname{dom} \sigma} R_{\mathsf{cap}(\mathsf{idx}(\ell, \sigma\ell))} = R_{\mathsf{cap}(\bigcup_{\ell \in \operatorname{dom} \sigma} \mathsf{idx}(\ell, \sigma\ell))} = R_{\mathsf{cap}(\mathsf{idx}\,\sigma)}$$

4. We have $\mathsf{al}(\ell) \in \mathsf{used}(\ell, s)$ for any s. Thus, if $\ell \in \operatorname{dom} \sigma$, we have $\mathsf{al}(\ell) \in \mathsf{used}(\ell, \sigma\ell) \subseteq \bigcup_{\ell \in \operatorname{dom} \sigma} \mathsf{used}(\ell, \sigma\ell) = \mathsf{used}\,\sigma$.
5. Immediate from the definition of idx.

□

4.1 Kripke Logical Relation

The logical logical relation is a Kripke relation in the step-index. More precisely, the value relation $\mathcal{V}[\![A]\!]_i$ and the expression relation $\mathcal{E}[\![A]\!]_i$ are closed under smaller step-indices. The heap typing relation $\mathcal{H}[\![\Phi]\!]_i$ is almost closed under smaller step-indices as the following lemma shows:

Lemma 4.3. *Let $i = (\alpha, C) \geq (\beta, D) = j$.*

1. $\mathcal{E}[\![A]\!]_i \subseteq \mathcal{E}[\![A]\!]_j$
2. $\mathcal{V}[\![A]\!]_i \subseteq \mathcal{V}[\![A]\!]_j$
3. *If* $(hv, s, R) \in \mathcal{HV}[\![\ell : A]\!]_i$ *and* $\mathsf{idx}(\ell, s) \subseteq D$*, then* $(hv, s, R) \in \mathcal{HV}[\![\ell : A]\!]_j$.
4. *If* $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_i$ *and* $\mathsf{idx}\,\sigma \subseteq D$*, then* $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_j$.

Proof.

1. The claim $\mathcal{E}[\![A]\!]_i \subseteq \mathcal{E}[\![A]\!]_j$ is immediate from the definition: For any $k \leq j$, we have $k \leq i$ by transitivity. The claim then follows by assumption.
2. By induction on A. The claim is trivial for $\mathbb{1}$, $\mathbb{B}$, $\mathbb{N}$, $\mathsf{Get}\ A$, and $\mathsf{Put}\ A$.

   For $A \otimes B$, let $(v, R) \in \mathcal{V}[\![A \otimes B]\!]_i$. Then there are v_1, v_2 and R_1, R_2 such that $v = (v_1, v_2)$, $R = R_1 \oplus R_2$, $(v_1, R_1) \in \mathcal{V}[\![A]\!]_i$, and $(v_2, R_2) \in \mathcal{V}[\![B]\!]_i$. By induction, we know $(v_1, R_1) \in \mathcal{V}[\![A]\!]_j$ and $(v_2, R_2) \in \mathcal{V}[\![B]\!]_j$. Hence $((v_1, v_2), R_1 \oplus R_2) \in \mathcal{V}[\![A \otimes B]\!]_j$.

   For $A \multimap B$ the claim follows immediately from the definition of $\mathcal{V}[\![A \multimap B]\!]$: Let $(\lambda x.e, R_e) \in \mathcal{V}[\![A \multimap B]\!]_i$ and $k \leq j$ and $(v, R_v) \in \mathcal{V}[\![A]\!]_k$. Then $k \leq i$. Hence, by $(\lambda x.e, R_e) \in \mathcal{V}[\![A \multimap B]\!]_i$, we have $(e[v/x], R_v \oplus R_e) \in \mathcal{E}[\![B]\!]_k$.
3. By case analysis on s. Trivial for Start and Done.

   For Val, we have $hv = \mathsf{V}(v)$ and $C = C' \uplus \{\mathsf{put}(\ell)\}$ for some C' and v such that $(v, R) \in \mathcal{V}[\![A]\!]_{\alpha, C'}$. Since $\mathsf{put}(\ell) \in \mathsf{idx}(\ell, \mathsf{Val}) \subseteq D$, we know that $D = D' \uplus \{\mathsf{put}(\ell)\}$ for some D'. Since $(\beta, D) \leq (\alpha, C)$, we have $(\beta, D') \leq (\alpha, C')$ by Lemma 4.1. By the second claim, it follows that $(v, R) \in \mathcal{V}[\![A]\!]_{\beta, D'}$.

   For Cont, we have $hv = \mathsf{C}(v)$ and $C = C' \uplus \{\mathsf{get}(\ell)\}$ for some C' and v such that $(v, R) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\alpha, C'}$. Since $\mathsf{get}(\ell) \in \mathsf{idx}(\ell, \mathsf{Cont}) \subseteq D$, we know that $D = D' \uplus \{\mathsf{get}(\ell)\}$ for some D'. Since $(\beta, D) \leq (\alpha, C)$, we have $(\beta, D') \leq (\alpha, C')$ by Lemma 4.1. By the second claim, it follows that $(v, R) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\beta, D'}$.
4. Follows immediately from the first claim since $\mathsf{idx}(\ell, s) \subseteq \mathsf{idx}\,\sigma$ for all $\ell \mapsto s \in \sigma$.

□

4.2 Resource Interpretation Properties

In the compatibility lemmas for the channel operations, we update the resource interpretation during the execution. The lemmas in this section capture how we update the resource interpretation.

For the resource interpretation $\mathsf{RI}_i(R_e, R_f, \Phi, h, \sigma, \rho)$ we use the following equivalent formulation in the proofs:

Lemma 4.4. $\mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho)$ *iff*

$$(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C} \text{ and } \mathsf{used}\,\sigma \mathrel{\#} C \text{ and } \mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{cap(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$$

where $\mathsf{sum}(\rho) \triangleq \bigoplus_{\ell \in \mathrm{dom}\,\rho} \rho\ell$

Proof. For the forward direction, let $\mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho)$. Then $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$ and $\mathsf{used}\,\sigma \mathrel{\#} C$ and $C = D \uplus \mathsf{idx}\,\sigma$ and $R_e \oplus R_f \oplus \mathsf{sum}(\rho) \sqsubseteq (D, \Phi, \alpha)$. By Lemma 3.2, we have:

$$(R_e \oplus R_f \oplus \mathsf{sum}(\rho)) \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (D, \Phi, \alpha) \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} = (C, \Phi, \alpha)$$

The claim follows with commutativity of $\oplus$.

For the backward direction, let $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$ and $\mathsf{used}\,\sigma \mathrel{\#} C$ and $\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$. Then $\mathsf{idx}\,\sigma \subseteq C$ by Lemma 3.2. Thus $C = D \uplus \mathsf{idx}\,\sigma$ for some capability set D. Hence, we have $\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (D, \Phi, \alpha) \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)}$. Thus, by Lemma 3.3, we have $\mathsf{sum}(\rho) \oplus R_e \oplus R_f \sqsubseteq (D, \Phi, \alpha)$. The claim follows with commutativity of $\oplus$. □

We can move resources from the expression into the frame and back:

Lemma 4.5. $\mathsf{RI}_i(R_1 \oplus R_2, R_f, \Phi, h, \sigma, \rho)$ *iff* $\mathsf{RI}_i(R_1, R_2 \oplus R_f, \Phi, h, \sigma, \rho)$

Proof. Follows immediately from associativity of $\oplus$:

$$\mathsf{sum}(\rho) \oplus (R_1 \oplus R_2) \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} = \mathsf{sum}(\rho) \oplus R_1 \oplus (R_2 \oplus R_f) \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)}$$

□

The resource interpretation is affine, in particular in the ordinal resources. For example, we can trade in the right to allocate ω fresh locations for the right to allocate n fresh locations for some $n \in \mathbb{N}$.

Lemma 4.6. *If* $\mathsf{RI}_{\beta,C}(R_e \oplus R_{ord(\alpha)}, R_f, \Phi, h, \sigma, \rho)$ *and* $\alpha' \leq \alpha$*, then there is some* $\beta' \leq \beta$ *such that* $\mathsf{RI}_{\beta',C}(R_e \oplus R_{ord(\alpha')}, R_f, \Phi, h, \sigma, \rho)$.

Proof. By Lemma 3.2, we have $R_{\mathsf{ord}(\alpha)} \sqsubseteq \mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{ord}(\alpha)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta)$ and thus $\beta = \beta_f \oplus \alpha$ for some β_f. We define $\beta' \triangleq \beta_f \oplus \alpha'$. Then $\beta' \leq \beta$ by Lemma 2.1. Thus, $(\beta', C) \leq (\beta, C)$. By Lemma 3.2, we have $\mathsf{idx}\,\sigma \subseteq C$. Since $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\beta,C}$, we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\beta',C}$ by Lemma 4.3. Lastly:

$$\begin{aligned}
&\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{ord}(\alpha)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{ord}(\alpha)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta_f) \oplus R_{\mathsf{ord}(\alpha)} && (\beta = \beta_f \oplus \alpha) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta_f) && (\text{Lemma 3.3}) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{ord}(\alpha')} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta_f) \oplus R_{\mathsf{ord}(\alpha')} && (\text{Lemma 3.2}) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{ord}(\alpha')} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta') && (\beta' = \beta_f \oplus \alpha')
\end{aligned}$$

□

Lemma 4.7. *If* $\mathsf{RI}_{\alpha,C}(R_1 \oplus R_2, R_f, \Phi, h, \sigma, \rho)$*, then* $\mathsf{RI}_{\alpha,C}(R_1, R_f, \Phi, h, \sigma, \rho)$*.*

Proof. By assumption, we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$ and $\mathsf{used}\,\sigma \mathbin{\#} C$ and $\mathsf{sum}(\rho) \oplus R_1 \oplus R_2 \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$. Thus $\mathsf{sum}(\rho) \oplus R_1 \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$ and the claim follows. □

We can trade in one ordinal resource for the capabilities associated with some fresh channel ℓ.

Lemma 4.8. *If* $\mathsf{RI}_{\alpha,C}(R_e \oplus R_{\mathit{ord}(1)}, R_f, \Phi, h, \sigma, \rho)$*, then there is some* ℓ *and* β *with* $\alpha = \beta \oplus 1$ *and* $\mathsf{RI}_{\beta, C \uplus \{\mathsf{al}(\ell), \mathsf{put}(\ell), \mathsf{get}(\ell)\}}(R_e \oplus R_{\mathit{cap}(\mathsf{get}(\ell), \mathsf{put}(\ell), \mathsf{al}(\ell))}, R_f, \Phi, h, \sigma, \rho)$*.*

Proof. By Lemma 3.2, we have $R_{\mathrm{ord}(1)} \sqsubseteq \mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathrm{ord}(1)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$ and thus $\alpha = \beta \oplus 1$ for some β. Let ℓ be some fresh location such that $\ell \notin \operatorname{dom} \sigma$ and $\mathsf{al}(\ell), \mathsf{get}(\ell), \mathsf{put}(\ell) \notin C$. We abbreviate $D \triangleq \{\mathsf{al}(\ell), \mathsf{get}(\ell), \mathsf{put}(\ell)\}$. We have:

$$\begin{aligned}
&\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathrm{ord}(1)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathrm{ord}(1)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta) \oplus R_{\mathrm{ord}(1)} && (\alpha = \beta \oplus 1) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta) && (\text{Lemma 3.3}) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{cap}(D)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \beta) \oplus R_{\mathsf{cap}(D)} && (\text{Lemma 3.2}) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{cap}(D)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C \uplus D, \Phi, \beta) && (D \mathbin{\#} C)
\end{aligned}$$

By assumption, we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$. Further, $\mathsf{idx}\,\sigma \subseteq C \uplus D$ by Lemma 3.2. Since $(\beta, C \uplus D) < (\alpha, C)$, we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\beta, C \uplus D}$ by Lemma 4.3. Since by assumption $\mathsf{used}\,\sigma \mathbin{\#} C$, we have $\mathsf{used}\,\sigma \mathbin{\#} C \uplus D$ as $D = \{\mathsf{al}(\ell), \mathsf{get}(\ell), \mathsf{put}(\ell)\}$ and $\ell \notin \operatorname{dom} \sigma$. □

In the above lemma, we extend the capability set C in the step-index with the capabilities $\mathsf{al}(\ell), \mathsf{get}(\ell)$, and $\mathsf{put}(\ell)$. Due to the lexicographic ordering, we obtain the resource interpretation at a smaller step-index by decreasing α. We refer to the above lemma as the logical allocation since the heap h is left unchanged. The following lemma allows for the physical allocation of ℓ, meaning trading the capability $\mathsf{al}(\ell)$ in for extending the heap h by the fresh location ℓ.

Lemma 4.9. *If* $\mathsf{RI}_{\alpha,C}(R_e \oplus R_{\mathit{cap}(\mathsf{al}(\ell))}, R_f, \Phi, h, \sigma, \rho)$*, then*
$\mathsf{RI}_{\alpha, C \setminus \{\mathsf{al}(\ell)\}}(R_e \oplus R_{\mathit{inv}(\ell:A)}, R_f, (\Phi, \ell : A), h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Start}], \rho[\ell \mapsto \epsilon])$*.*

Proof. By Lemma 3.2, we have $R_{\mathsf{cap}(\mathsf{al}(\ell))} \sqsubseteq \mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{al}(\ell))} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$. Hence, we have $\{\mathsf{al}(\ell)\} \subseteq C$ and thus $C = C' \uplus \{\mathsf{al}(\ell)\}$ for some C'. Since $\mathsf{used}\,\sigma \mathbin{\#} C$, we have $\mathsf{al}(\ell) \notin \mathsf{used}\,\sigma$. Thus $\ell \notin \operatorname{dom} \sigma$ by Lemma 4.2. From $(\sigma, h, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$, we know $\operatorname{dom} \Phi = \operatorname{dom} \rho = \operatorname{dom} \sigma$ and thus $\ell \notin \operatorname{dom} \Phi$ and $\ell \notin \operatorname{dom} \rho$. We have:

$$\begin{aligned}
&\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{al}(\ell))} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha) \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{al}(\ell))} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C', \Phi, \alpha) \oplus R_{\mathsf{cap}(\mathsf{al}(\ell))} \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C', \Phi, \alpha) && \text{(Lemma 3.3)} \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{inv}(\ell:A)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C', \Phi, \alpha) \oplus R_{\mathsf{inv}(\ell:A)} && \text{(Lemma 3.2)} \\
\Rightarrow\ &\mathsf{sum}(\rho) \oplus R_e \oplus R_{\mathsf{inv}(\ell:A)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C', (\Phi, \ell : A), \alpha) && (\ell \notin \operatorname{dom} \Phi) \\
\Rightarrow\ &\mathsf{sum}(\rho[\ell \mapsto \epsilon]) \oplus R_e \oplus R_{\mathsf{inv}(\ell:A)} \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}(\sigma[\ell\mapsto\mathsf{Start}]))} \sqsubseteq (C', (\Phi, \ell : A), \alpha) && (*)
\end{aligned}$$

where $(*)$ is true since $\ell \notin \operatorname{dom}\rho$ and $\ell \notin \operatorname{dom}\sigma$. We have $\mathsf{idx}(\sigma[\ell \mapsto \mathsf{Start}]) \subseteq C'$ by Lemma 3.2. Thus, by Lemma 4.3, we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C'}$. Since $(\mathsf{E}, \mathsf{Start}, \epsilon) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C'}$, we have $(h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Start}], \rho[\ell \mapsto \epsilon]) \in \mathcal{H}[\![\Phi, \ell : A]\!]_{\alpha,C'}$. By assumption $\mathsf{used}\,\sigma \# C$. Hence $\mathsf{used}(\sigma[\ell \mapsto \mathsf{Start}]) = \{\mathsf{al}(\ell)\} \uplus \mathsf{used}\,\sigma \# C'$. □

Lemma 4.10. *If* $\mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho)$ *and* $\ell : A \in \Phi$ *and* $(hv, s, R) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$ *and* $\mathsf{used}(\ell, s) \setminus \mathsf{used}(\ell, \sigma\ell) \subseteq D$ *and*

$$\rho\ell \oplus R_e \oplus R_{cap(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{inv(\Phi)} = R \oplus R'_e \oplus R_{cap(\mathsf{idx}(\ell,s))} \oplus R_{cap(D)} \oplus R_{inv(\Phi)},$$

then $\mathsf{RI}_{\alpha,C\setminus D}(R'_e, R_f, \Phi, h[\ell \mapsto hv], \sigma[\ell \mapsto s], \rho[\ell \mapsto R])$.

Proof. By assumption $\mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$. We abbreviate the resources that remain unchanged $R_F := (\bigoplus_{\ell'\in\operatorname{dom}\rho, \ell'\neq\ell} \rho\ell') \oplus R_f \oplus (\bigoplus_{\ell'\in\operatorname{dom}\sigma, \ell'\neq\ell} R_{\mathsf{cap}(\mathsf{idx}(\ell',\sigma\ell'))})$. Then $R_F \oplus \rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \sqsubseteq (C, \Phi, \alpha)$ by Lemma 4.2.

$$\begin{aligned}
&R_F \oplus \rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \sqsubseteq (C, \Phi, \alpha) \\
\Rightarrow\ &R_F \oplus \rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \sqsubseteq (C, \Phi, \alpha) \oplus R_{\mathsf{inv}(\Phi)} && \text{(Lemma 3.2)} \\
\Rightarrow\ &R_F \oplus \rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \sqsubseteq (C, \Phi, \alpha) && \text{(Lemma 3.3)} \\
\Rightarrow\ &R_F \oplus R \oplus R'_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,s))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} \sqsubseteq (C, \Phi, \alpha) \\
\Rightarrow\ &R_F \oplus R \oplus R'_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,s))} \oplus R_{\mathsf{cap}(D)} \sqsubseteq (C, \Phi, \alpha) && \text{(Lemma 3.2)} \\
\Rightarrow\ &R_F \oplus R \oplus R'_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,s))} \oplus R_{\mathsf{cap}(D)} \sqsubseteq (C \setminus D, \Phi, \alpha) \oplus R_{\mathsf{cap}(D)} && (*) \\
\Rightarrow\ &R_F \oplus R \oplus R'_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,s))} \sqsubseteq (C \setminus D, \Phi, \alpha) && \text{(Lemma 3.3)} \\
\Rightarrow\ &\mathsf{sum}(\rho[\ell \mapsto R]) \oplus R'_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}(\sigma[\ell\mapsto s]))} \sqsubseteq (C \setminus D, \Phi, \alpha) && \text{(Lemma 4.2)}
\end{aligned}$$

where $(*)$ follows with $D \subseteq C$ which holds given $R_F \oplus R \oplus R'_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,s))} \oplus R_{\mathsf{cap}(D)} \sqsubseteq (C, \Phi, \alpha)$ by Lemma 3.2.

By assumption, we have $C \# \mathsf{used}\,\sigma$. We show $(C \setminus D) \# \mathsf{used}(\sigma[\ell \mapsto s])$. By way of contradiction, assume $p \in C \setminus D$ and $p \in \mathsf{used}(\sigma[\ell \mapsto s])$. Since $C \# \mathsf{used}\,\sigma$, we have $p \notin \mathsf{used}\,\sigma$. Thus $p \notin \mathsf{used}(\ell, \sigma\ell)$ and $p \in \mathsf{used}(\ell, s)$. Hence $p \in \mathsf{used}(\ell, s) \setminus \mathsf{used}(\ell, \sigma\ell) \subseteq D$, a contradiction.

By assumption we have $(h, \sigma, \rho) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$ and $(hv, s, R) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. Thus, by definition we have $(h[\ell \mapsto hv], \sigma[\ell \mapsto s], \rho[\ell \mapsto R]) \in \mathcal{H}[\![\Phi]\!]_{\alpha,C}$. With Lemma 4.3, it suffices to show $\mathsf{idx}(\sigma[\ell \mapsto s]) \subseteq C \setminus D$ which follows from $\mathsf{sum}(\rho[\ell \mapsto R]) \oplus R'_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}(\sigma[\ell\mapsto s]))} \sqsubseteq (C \setminus D, \Phi, \alpha)$ by Lemma 3.2. □

Once a location has been allocated, the following lemma, Lemma 4.11, characterises how we can update the resource interpretation. From the state Start, the resource interpretation may be updated to the Val state by giving up the resources for the value v and the resource $R_{\mathsf{put}(\ell,A)}$ which contains the capability $\mathsf{put}(\ell)$. The resource R_v associated with the value is transferred to the heap while the capability $\mathsf{put}(\ell)$ is transferred to the logical state and remains part of the step-index. Analogously, we can store a continuation in the heap and advance the logical state to Cont. If the heap currently stores a value with logical state Val, we can advance the state to Done and remove the resource R_v from the resource map. We decrease the current step-index (α, C) to the smaller step-index $(\alpha, C \setminus \{\mathsf{get}(\ell), \mathsf{put}(\ell)\})$. Analogously, we can advance from the state where a continuation is stored to the Done state.

Lemma 4.11. *Let* $\mathsf{RI}_{\alpha,C}(R_e, R_f, \Phi, h, \sigma, \rho)$.

1. *If* $\sigma\ell = \mathsf{Start}$ *and* $\rho\ell = \epsilon$ *and* $R_e = R_v \oplus R_{put(\ell,A)}$ *and* $(v, R_v) \in \mathcal{V}[\![A]\!]_{\alpha,C}$, *then* $\mathsf{RI}_{\alpha,C}(\epsilon, R_f, \Phi, h[\ell \mapsto \mathsf{V}(v)], \sigma[\ell \mapsto \mathsf{Val}], \rho[\ell \mapsto R_v])$.
2. *If* $\sigma\ell = \mathsf{Start}$ *and* $\rho\ell = \epsilon$ *and* $R_e = R_\lambda \oplus R_{get(\ell,A)}$ *and* $(v, R_\lambda) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\alpha,C}$, *then* $\mathsf{RI}_{\alpha,C}(\epsilon, R_f, \Phi, h[\ell \mapsto \mathsf{C}(v)], \sigma[\ell \mapsto \mathsf{Cont}], \rho[\ell \mapsto R_\lambda])$.
3. *If* $\sigma\ell = \mathsf{Val}$ *and* $\rho\ell = R_v$ *and* $R_e = R_\lambda \oplus R_{get(\ell,A)}$, *then* $\mathsf{RI}_{\alpha,C\setminus\{\mathsf{get}(\ell),\mathsf{put}(\ell)\}}(R_v \oplus R_\lambda, R_f, \Phi, h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Done}], \rho[\ell \mapsto \epsilon])$.
4. *If* $\sigma\ell = \mathsf{Cont}$ *and* $\rho\ell = R_\lambda$ *and* $R_e = R_v \oplus R_{put(\ell,A)}$, *then* $\mathsf{RI}_{\alpha,C\setminus\{\mathsf{get}(\ell),\mathsf{put}(\ell)\}}(R_v \oplus R_\lambda, R_f, \Phi, h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Done}], \rho[\ell \mapsto \epsilon])$.

Proof. We prove each of the claims using Lemma 4.10. For each case, we have $\ell : A \in \Phi$ because $R_{\mathsf{get}(\ell,A)} \sqsubseteq R_e$ or $R_{\mathsf{put}(\ell,A)} \sqsubseteq R_e$ from which $\ell : A \in \Phi$ follows by Lemma 3.2 since $R_e \sqsubseteq \mathsf{sum}(\rho) \oplus R_e \oplus R_f \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)} \sqsubseteq (C, \Phi, \alpha)$.

1. We pick $D \triangleq \emptyset = \mathsf{used}(\ell, \mathsf{Val}) \setminus \mathsf{used}(\ell, \mathsf{Start})$. We show $(\mathsf{V}(v), \mathsf{Val}, R_v) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. We have $\mathsf{put}(\ell) \in C$ as $R_{\mathsf{put}(\ell,A)} \sqsubseteq R_e \sqsubseteq (C, \Phi, \alpha)$ by Lemma 3.2. Thus, there is some C' such that $C = C' \uplus \{\mathsf{put}(\ell)\}$. By Lemma 4.3, we have $(v, R_v) \in \mathcal{V}[\![A]\!]_{\alpha,C'}$. Lastly, we have by Lemma 3.2 and Lemma 3.3:

$$\begin{aligned}
&\rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \\
=&\epsilon \oplus R_v \oplus R_{\mathsf{put}(\ell,A)} \oplus \epsilon \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)} \\
=&\epsilon \oplus R_v \oplus R_{\mathsf{cap}(\mathsf{put}(\ell))} \oplus R_{\mathsf{inv}(\ell:A)} \oplus \epsilon \oplus R_{\mathsf{inv}(\Phi)} \\
=&R_v \oplus \epsilon \oplus R_{\mathsf{cap}(\mathsf{put}(\ell))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && (\ell : A \in \Phi, R_{\mathsf{cap}(\emptyset)} = \epsilon) \\
=&R_v \oplus \epsilon \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\mathsf{Val}))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)}
\end{aligned}$$

2. We pick $D \triangleq \emptyset = \mathsf{used}(\ell, \mathsf{Cont}) \setminus \mathsf{used}(\ell, \mathsf{Start})$. We show $(\mathsf{C}(v), \mathsf{Cont}, R_\lambda) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. We have $\mathsf{get}(\ell) \in C$ as $R_{\mathsf{get}(\ell,A)} \sqsubseteq R_e \sqsubseteq (C, \Phi, \alpha)$ by Lemma 3.2. Thus, there is some C' with $C = C' \uplus \{\mathsf{get}(\ell)\}$. By Lemma 4.3, we have $(v, R_\lambda) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\alpha,C'}$.

Lastly, we have by Lemma 3.2 and Lemma 3.3:

$$\begin{aligned}
&\rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \\
=&\epsilon \oplus R_\lambda \oplus R_{\mathsf{get}(\ell,A)} \oplus \epsilon \oplus R_{\mathsf{inv}(\Phi)} \\
=&\epsilon \oplus R_\lambda \oplus R_{\mathsf{cap}(\mathsf{get}(\ell))} \oplus R_{\mathsf{inv}(\ell:A)} \oplus \epsilon \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)} \\
=&R_\lambda \oplus \epsilon \oplus R_{\mathsf{cap}(\mathsf{get}(\ell))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && (\ell : A \in \Phi, R_{\mathsf{cap}(\emptyset)} = \epsilon) \\
=&R_\lambda \oplus \epsilon \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\mathsf{Cont}))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)}
\end{aligned}$$

3. We pick $D \triangleq \{\mathsf{get}(\ell), \mathsf{put}(\ell)\} = \mathsf{used}(\ell, \mathsf{Done}) \setminus \mathsf{used}(\ell, \mathsf{Val})$. By definition, we have $(\mathsf{E}, \mathsf{Done}, \epsilon) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. Lastly, we have by Lemma 3.2 and Lemma 3.3:

$$\begin{aligned}
&\rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \\
=&R_v \oplus R_\lambda \oplus R_{\mathsf{get}(\ell,A)} \oplus R_{\mathsf{cap}(\mathsf{put}(\ell))} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)} \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(\mathsf{get}(\ell),\mathsf{put}(\ell))} \oplus R_{\mathsf{inv}(\ell:A)} \oplus R_{\mathsf{inv}(\Phi)} \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && (\ell : A \in \Phi) \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\mathsf{Done}))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)}
\end{aligned}$$

4. We pick $D \triangleq \{\mathsf{get}(\ell), \mathsf{put}(\ell)\} = \mathsf{used}(\ell, \mathsf{Done}) \setminus \mathsf{used}(\ell, \mathsf{Cont})$. By definition, we have $(\mathsf{E}, \mathsf{Done}, \epsilon) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. Lastly, we have by Lemma 3.2 and Lemma 3.3:

$$\begin{aligned}
&\rho\ell \oplus R_e \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\sigma\ell))} \oplus R_{\mathsf{inv}(\Phi)} \\
=&R_\lambda \oplus R_v \oplus R_{\mathsf{put}(\ell,A)} \oplus R_{\mathsf{cap}(\mathsf{get}(\ell))} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)} \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(\mathsf{get}(\ell),\mathsf{put}(\ell))} \oplus R_{\mathsf{inv}(\ell:A)} \oplus R_{\mathsf{inv}(\Phi)} \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && (\ell : A \in \Phi) \\
=&\epsilon \oplus (R_v \oplus R_\lambda) \oplus R_{\mathsf{cap}(\mathsf{idx}(\ell,\mathsf{Done}))} \oplus R_{\mathsf{cap}(D)} \oplus R_{\mathsf{inv}(\Phi)} && \text{(Def. idx)}
\end{aligned}$$

□

4.3 Properties of the Type Interpretations

The expression relation is closed under larger resources, in particular in the number of locations that can be allocated. We can always weaken the number to a larger bound:

Lemma 4.12. *1. If $(e, R_e \oplus R_{ord(\alpha)}) \in \mathcal{E}[\![A]\!]_i$ and $\alpha \leq \beta$, then $(e, R_e \oplus R_{ord(\beta)}) \in \mathcal{E}[\![A]\!]_i$.*

2. If $(e, R_e) \in \mathcal{E}[\![A]\!]_i$, then $(e, R_e \oplus R) \in \mathcal{E}[\![A]\!]_i$.

Proof. 1. Let $(\gamma, C) \leq i$ and $\mathsf{RI}_{\gamma,C}(R_e \oplus R_{\mathsf{ord}(\beta)}, R_f, \Phi, h, \sigma, \rho)$. Then there is some $\gamma' \leq \gamma$ such that $\mathsf{RI}_{\gamma',C}(R_e \oplus R_{\mathsf{ord}(\alpha)}, R_f, \Phi, h, \sigma, \rho)$ by Lemma 4.6. Since $\gamma' \leq \gamma$, we have $(\gamma', C) \leq (\gamma, C) \leq i$. By assumption, there is some $k \leq (\gamma', C)$ and $\Phi' \supseteq \Phi$ and h', σ', ρ' and $(v, R_v) \in \mathcal{V}[\![A]\!]_k$ such that $\mathsf{RI}_k(R_v, R_f, \Phi', h', \sigma', \rho')$ and $\sigma \dashrightarrow^* \sigma'$ and $(e, h) \rightsquigarrow^* (v, h')$. The claim follows with $k \leq (\gamma', C) \leq (\gamma, C)$.

2. Let $j \leq i$ and $\mathsf{RI}_j(R_e \oplus R, R_f, \Phi, h, \sigma, \rho)$. Then $\mathsf{RI}_j(R_e, R_f, \Phi, h, \sigma, \rho)$ by Lemma 4.7. The claim follows by the assumption $(e, R_e) \in \mathcal{E}\llbracket A \rrbracket_i$.

□

The following lemma is useful in proofs of standard compatibility lemmas. First, every value in the value relation is already contained in the expression relation. Second, we can take steps which do not manipulate the heap, defined by $e' \leadsto e \triangleq \forall h.(e, h) \leadsto (e', h)$, to prove that an expression is contained in the logical relation. Third, we can reason about composite expressions $K[e]$ by reasoning about e and the remaining expression $K[v]$ after e has been executed, if we abstract over the result v.

Lemma 4.13. *1. $\mathcal{V}\llbracket A \rrbracket_i \subseteq \mathcal{E}\llbracket A \rrbracket_i$.*

2. If $(e, R_e) \in \mathcal{E}\llbracket A \rrbracket_i$ and $e' \leadsto^* e$, then $(e', R_e) \in \mathcal{E}\llbracket A \rrbracket_i$.

3. If $(e, R_e) \in \mathcal{E}\llbracket A \rrbracket_i$ and $\forall j \leq i, (v, R_v) \in \mathcal{V}\llbracket A \rrbracket_j.\ (K[v], R_v \oplus R_K) \in \mathcal{E}\llbracket B \rrbracket_j$, then $(K[e], R_e \oplus R_K) \in \mathcal{E}\llbracket B \rrbracket_i$.

Proof. 1. Let $(v, R_v) \in \mathcal{V}\llbracket A \rrbracket_i$ and $j \leq i$ and $\mathsf{RI}_j(R_v, R_f, \Phi, h, \sigma, \rho)$. Then $(v, R_v) \in \mathcal{V}\llbracket A \rrbracket_j$ by Lemma 4.3. The claim follows with $(v, h) \leadsto^* (v, h)$ and $\sigma \leadsto^* \sigma$.

2. Let $(e, R_e) \in \mathcal{E}\llbracket A \rrbracket_i$ and $e' \leadsto^* e$ and $j \leq i$ and $\mathsf{RI}_j(R_v, R_f, \Phi, h, \sigma, \rho)$. By assumption, there exist $k \leq j$ and $\Phi' \supseteq \Phi$ and $h', \sigma', \rho', v, R_v$ such that $\mathsf{RI}_k(R_v, R_f, \Phi', h', \sigma', \rho')$ and $(v, R_v) \in \mathcal{V}\llbracket A \rrbracket_k$ and $(e, h) \leadsto^* (v, h')$. The claim follows with $(e', h) \leadsto^* (e, h) \leadsto^* (v, h')$.

3. Let $j \leq i$ and $\mathsf{RI}_j(R_e \oplus R_K, R_f, \Phi, h, \sigma, \rho)$. By Lemma 4.5, we have $\mathsf{RI}_j(R_e, R_K \oplus R_f, \Phi, h, \sigma, \rho)$. By assumption, there exist some $j' \leq j$ and $\Phi' \supseteq \Phi$ and $h', \sigma', \rho', v, R_v$ such that $\mathsf{RI}_{j'}(R_v, R_K \oplus R_f, \Phi', h', \sigma', \rho')$ and $\sigma \dashrightarrow^* \sigma'$ and $(v, R_v) \in \mathcal{V}\llbracket A \rrbracket_{j'}$ and $(e, h) \leadsto^* (v, h')$. By Lemma 4.5, $\mathsf{RI}_{j'}(R_v \oplus R_K, R_f, \Phi', h', \sigma', \rho')$ follows. By assumption, $(K[v], R_v \oplus R_K) \in \mathcal{E}\llbracket B \rrbracket_j$. Thus, there exists some $k \leq j'$ and $\Phi'' \supseteq \Phi'$ and $h'', \sigma'', \rho'', v', R_v'$ such that $\mathsf{RI}_k(R_v', R_f, \Phi'', h'', \sigma'', \rho'')$ and $\sigma' \dashrightarrow^* \sigma''$ and $(v', R_v') \in \mathcal{V}\llbracket B \rrbracket_k$ and $(K[v], h') \leadsto^* (v', h'')$. We have $(K[e], h) \leadsto^* (K[v], h') \leadsto^* (v', h'')$ and $\sigma \dashrightarrow^* \sigma' \dashrightarrow^* \sigma''$ and $k \leq j' \leq j$.

□

The following four lemmas are the compatibility lemmas on closed expressions for creating channels, for receiving values over them, for sending values over them, and for closing them.

Lemma 4.14.

$$\frac{\forall \ell.(e[\ell/x, \ell/y], R_e \oplus R_{\mathit{get}(\ell,A)} \oplus R_{\mathit{put}(\ell,A)}) \in \mathcal{E}\llbracket B \rrbracket_i}{(\mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e, R_e \oplus R_{\mathit{ord}(1)}) \in \mathcal{E}\llbracket B \rrbracket_i}$$

Proof. Let $(\alpha, C) \leq i$ and $\mathsf{RI}_{\alpha,C}(R_e \oplus R_{\mathsf{ord}(1)}, R_f, \Phi, h, \sigma, \rho)$. By Lemma 4.8, we have

$$\mathsf{RI}_{\beta, C \uplus \{\mathsf{al}(\ell), \mathsf{put}(\ell), \mathsf{get}(\ell)\}}(R_e \oplus R_{\mathsf{cap}(\mathsf{al}(\ell), \mathsf{get}(\ell), \mathsf{put}(\ell))}, R_f, \Phi, h, \sigma, \rho)$$

for some β with $\alpha = \beta \oplus 1$ and some location ℓ. We have $(C \uplus \{\mathsf{al}(\ell), \mathsf{put}(\ell), \mathsf{get}(\ell)\}) \mathrel{\#} \mathsf{used}\,\sigma$ by the definition of RI and thus $\ell \notin \operatorname{dom}\sigma$ by Lemma 4.2. With $(h, \sigma, \rho) \in \mathcal{H}\llbracket\Phi\rrbracket_{\alpha,C}$, we further have $\ell \notin \operatorname{dom} h$ and $\ell \notin \operatorname{dom}\Phi$ and $\ell \notin \operatorname{dom}\rho$. By Lemma 4.9, we have:

$$\mathsf{RI}_{\beta, C \uplus \{\mathsf{get}(\ell), \mathsf{put}(\ell)\}}(R_e \oplus \underbrace{R_{\mathsf{cap}(\mathsf{get}(\ell),\mathsf{put}(\ell))} \oplus R_{\mathsf{inv}(\ell:A)}}_{=R_{\mathsf{get}(\ell,A)} \oplus R_{\mathsf{put}(\ell,A)}}, R_f, (\Phi, \ell : A), h', \sigma', \rho')$$

for $h' \triangleq h[\ell \mapsto \mathsf{E}]$ and $\sigma' \triangleq \sigma[\ell \mapsto \mathsf{Start}]$ and $\rho' \triangleq \rho[\ell \mapsto \epsilon]$. By assumption $(e[\ell/x, \ell/y], R_e \oplus R_{\mathsf{get}(\ell,A)} \oplus R_{\mathsf{put}(\ell,A)}) \in \mathcal{E}\llbracket B\rrbracket_i$. Since $\beta < \alpha$ by Lemma 2.1, we have $(\beta, C \uplus \{\mathsf{get}(\ell), \mathsf{put}(\ell)\}) < (\alpha, C) \leq i$. Hence, there is some $(\beta', C') \leq (\beta, C \uplus \{\mathsf{get}(\ell), \mathsf{put}(\ell)\})$ and $\Phi' \supseteq (\Phi, \ell : A)$ with

$$\mathsf{RI}_{\beta',C'}(R_v, R_f, \Phi', h'', \sigma'', \rho'')$$

for some h'', σ'', ρ'', $(v, R_v) \in \mathcal{V}\llbracket B\rrbracket_{\beta',C'}$. Further, $\sigma' \dashrightarrow^* \sigma''$ and $(e[\ell/x, \ell/y], h') \rightsquigarrow^* (v, h'')$. We have $(\beta', C') \leq (\beta, C \uplus \{\mathsf{put}(\ell), \mathsf{get}(\ell)\}) \leq (\alpha, C)$ and $\Phi' \supseteq (\Phi, \ell : A) \supseteq \Phi$. Since $\ell \notin \operatorname{dom}\sigma$ and $\ell \notin \operatorname{dom} h$, we have $\sigma \dashrightarrow \sigma' \dashrightarrow^* \sigma''$ and $(\mathsf{let}\ (x, y) = \mathsf{chan}()\ \mathsf{in}\ e, h) \rightsquigarrow (e[\ell/x, \ell/y], h') \rightsquigarrow^* (v, h'')$. □

Lemma 4.15.

$$\frac{(v_{ch}, R_{ch}) \in \mathcal{V}\llbracket\mathsf{Get}\ A\rrbracket_i \qquad (v_\lambda, R_\lambda) \in \mathcal{V}\llbracket A \multimap \mathbb{1}\rrbracket_i}{(\mathsf{get}(v_{ch}, v_\lambda), R_{ch} \oplus R_\lambda) \in \mathcal{E}\llbracket\mathbb{1}\rrbracket_i}$$

Proof. Let $(\alpha, C) \leq i$ and $\mathsf{RI}_{\alpha,C}(R_{\mathrm{ch}} \oplus R_\lambda, R_f, \Phi, h, \sigma, \rho)$. By $(v_{\mathrm{ch}}, R_{\mathrm{ch}}) \in \mathcal{V}\llbracket\mathsf{Get}\ A\rrbracket_i$, we know $v_{\mathrm{ch}} = \ell$ and $R_{\mathrm{ch}} = R_{\mathsf{get}(\ell,A)}$ for some ℓ. By Lemma 3.2, we have $R_{\mathsf{get}(\ell,A)} \sqsubseteq (C, \Phi, \alpha)$ and $\checkmark R_{\mathsf{get}(\ell,A)} \oplus R_{\mathsf{cap}(\mathsf{idx}\,\sigma)}$ and thus $\mathsf{get}(\ell) \in C$ and $\mathsf{get}(\ell) \notin \mathsf{idx}\,\sigma$ and $\ell : A \in \Phi$. Further, we have $\mathsf{used}\,\sigma \mathrel{\#} C$. Thus, $\mathsf{get}(\ell) \notin \mathsf{used}\,\sigma$ follows. Further, unfolding the definition of $\mathsf{RI}_{\alpha,C}$, we have $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}\llbracket\ell : A\rrbracket_{\alpha,C}$. Since $\mathsf{get}(\ell) \notin \mathsf{used}\,\sigma$ and $\mathsf{get}(\ell) \notin \mathsf{idx}\,\sigma$, either $\sigma\ell = \mathsf{Start}$ or $\sigma\ell = \mathsf{Val}$.

1. Let $\sigma\ell = \mathsf{Start}$. By $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}\llbracket\ell : A\rrbracket_{\alpha,C}$, we have $h\ell = \mathsf{E}$ and $\rho\ell = \epsilon$. By Lemma 4.3, we have $(v_\lambda, R_\lambda) \in \mathcal{V}\llbracket A \multimap \mathbb{1}\rrbracket_{\alpha,C}$. Hence, with Lemma 4.11 we have:

   $$\mathsf{RI}_{\alpha,C}(\epsilon, R_f, \Phi, h[\ell \mapsto \mathsf{C}(v_\lambda)], \sigma[\ell \mapsto \mathsf{Cont}], \rho[\ell \mapsto R_\lambda])$$

   We define $h' \triangleq h[\ell \mapsto \mathsf{C}(v_\lambda)]$, $\sigma' \triangleq \sigma[\ell \mapsto \mathsf{Cont}]$, and $\rho' \triangleq \rho[\ell \mapsto R_\lambda]$. By definition $(\mathsf{get}(\ell, v_\lambda), h) \rightsquigarrow^* ((), h')$ and $\sigma \dashrightarrow^* \sigma'$. Further, by definition $((), \epsilon) \in \mathcal{V}\llbracket\mathbb{1}\rrbracket_{\alpha,C}$.

2. Let $\sigma\ell = \mathsf{Val}$. By $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}\llbracket\ell : A\rrbracket_{\alpha,C}$, we have $C = C_1 \uplus \{\mathsf{put}(\ell)\}$ for some C_1 and $h\ell = \mathsf{V}(v)$ and $\rho\ell = R_v$ for some $(v, R_v) \in \mathcal{V}\llbracket A\rrbracket_{\alpha,C_1}$. Since $\mathsf{get}(\ell) \in C$, there is some C_2 such that $C_2 = C_1 \uplus \{\mathsf{get}(\ell)\}$. By Lemma 4.3 $(v, R_v) \in \mathcal{V}\llbracket A\rrbracket_{\alpha,C_2}$. Thus, by definition of the value relation $\mathcal{V}\llbracket A \multimap \mathbb{1}\rrbracket_i$, we have $v_\lambda = \lambda x.e$ for some x and e such that $(e[v/x], R_v \oplus R_\lambda) \in \mathcal{E}\llbracket\mathbb{1}\rrbracket_{\alpha,C_2}$. By Lemma 4.11, we obtain:

   $$\mathsf{RI}_{\alpha,C_2}(R_v \oplus R_\lambda, R_f, \Phi, h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Done}], \rho[\ell \mapsto \epsilon])$$

   Thus, there are some $(\beta, C_3) \leq (\alpha, C_2)$ and $\Phi' \supseteq \Phi$, h', σ', ρ' and $(v', R_v') \in \mathcal{V}\llbracket\mathbb{1}\rrbracket_{\beta,C_3}$, such that $\mathsf{RI}_{\beta,C_3}(R_v', R_f, \Phi', h', \sigma', \rho')$ and $(e[v/x], h[\ell \mapsto \mathsf{E}]) \rightsquigarrow^* (v', h')$ and $\sigma[\ell \mapsto \mathsf{Done}] \dashrightarrow^* \sigma'$.

We have $(\beta, C_3) \leq (\alpha, C_2) \leq (\alpha, C)$ and $\Phi' \supseteq \Phi$. Further, we have $(\mathsf{get}(\ell, \lambda x.e), h) \rightsquigarrow (e[v/x], h[\ell \mapsto \mathsf{E}]) \rightsquigarrow^* (v', h')$. Lastly, $\sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Done}] \dashrightarrow^* \sigma'$.

□

Lemma 4.16.

$$\frac{(v_{ch}, R_{ch}) \in \mathcal{V}[\![\mathsf{Put}\, A]\!]_i \qquad (v, R_v) \in \mathcal{V}[\![A]\!]_i}{(\mathsf{put}(v_{ch}, v), R_{ch} \oplus R_v) \in \mathcal{E}[\![\mathbb{1}]\!]_i}$$

Proof. Let $(\alpha, C) \leq i$ and $\mathsf{RI}_{\alpha,C}(R_{\mathrm{ch}} \oplus R_v, R_f, \Phi, h, \sigma, \rho)$. By $(v_{\mathrm{ch}}, R_{\mathrm{ch}}) \in \mathcal{V}[\![\mathsf{Put}\, A]\!]_i$, we know $v_{\mathrm{ch}} = \ell$ and $R_{\mathrm{ch}} = R_{\mathrm{put}(\ell,A)}$ for some ℓ. By Lemma 3.2, we have $R_{\mathrm{put}(\ell,A)} \sqsubseteq (C, \Phi, \alpha)$ and $\checkmark R_{\mathrm{put}(\ell,A)} \oplus R_{\mathrm{cap}(\mathrm{idx}\,\sigma)}$ and thus $\mathsf{put}(\ell) \in C$ and $\mathsf{put}(\ell) \notin \mathsf{idx}\,\sigma$ and $\ell : A \in \Phi$. Further, we have $\mathsf{used}\,\sigma \mathbin{\#} C$. Thus, $\mathsf{put}(\ell) \notin \mathsf{used}\,\sigma$ follows. Further, unfolding the definition of $\mathsf{RI}_{\alpha,C}$, we have $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$. Since $\mathsf{put}(\ell) \notin \mathsf{used}\,\sigma$ and $\mathsf{put}(\ell) \notin \mathsf{idx}\,\sigma$, either $\sigma\ell = \mathsf{Start}$ or $\sigma\ell = \mathsf{Cont}$.

1. Let $\sigma\ell = \mathsf{Start}$. By $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$, we have $h\ell = \mathsf{E}$ and $\rho\ell = \epsilon$. By Lemma 4.3, we have $(v, R_v) \in \mathcal{V}[\![A]\!]_{\alpha,C}$. Hence, with Lemma 4.11 we have:

   $$\mathsf{RI}_{\alpha,C}(\epsilon, R_f, \Phi, h[\ell \mapsto \mathsf{V}(v)], \sigma[\ell \mapsto \mathsf{Val}], \rho[\ell \mapsto R_v])$$

   We define $h' \triangleq h[\ell \mapsto \mathsf{V}(v)]$, $\sigma' \triangleq \sigma[\ell \mapsto \mathsf{Val}]$, and $\rho' \triangleq \rho[\ell \mapsto R_v]$. By definition $(\mathsf{put}(\ell, v), h) \rightsquigarrow^* ((), h')$ and $\sigma \dashrightarrow^* \sigma'$. Further, by definition $((), \epsilon) \in \mathcal{V}[\![\mathbb{1}]\!]_{\alpha,C}$.

2. Let $\sigma\ell = \mathsf{Cont}$. By $(h\ell, \sigma\ell, \rho\ell) \in \mathcal{HV}[\![\ell : A]\!]_{\alpha,C}$, we have $C = C_1 \uplus \{\mathsf{get}(\ell)\}$ for some C_1 and $h\ell = \mathsf{C}(v_\lambda)$ and $\rho\ell = R_\lambda$ for some $(v_\lambda, R_\lambda) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\alpha,C_1}$. Since $\mathsf{put}(\ell) \in C$, there is some C_2 such that $C_2 = C_1 \uplus \{\mathsf{put}(\ell)\}$. By Lemma 4.3, we have $(v, R_v) \in \mathcal{V}[\![A]\!]_{\alpha,C_2}$. Thus, by definition of the value relation $\mathcal{V}[\![A \multimap \mathbb{1}]\!]_{\alpha,C_1}$ we have $v_\lambda = \lambda x.e$ for some x and e such that $(e[v/x], R_v \oplus R_\lambda) \in \mathcal{E}[\![\mathbb{1}]\!]_{\alpha,C_2}$. By Lemma 4.11, we obtain:

   $$\mathsf{RI}_{\alpha,C_2}(R_v \oplus R_\lambda, R_f, \Phi, h[\ell \mapsto \mathsf{E}], \sigma[\ell \mapsto \mathsf{Done}], \rho[\ell \mapsto \epsilon])$$

   Thus, there are some $(\beta, C_3) \leq (\alpha, C_2)$ and $\Phi' \supseteq \Phi$, h', σ', ρ' and $(v', R_v') \in \mathcal{V}[\![\mathbb{1}]\!]_{\beta,C_3}$, such that $\mathsf{RI}_{\beta,C_3}(R_v', R_f, \Phi', h', \sigma', \rho')$ and $(e[v/x], h[\ell \mapsto \mathsf{E}]) \rightsquigarrow^* (v', h')$ and $\sigma[\ell \mapsto \mathsf{Done}] \dashrightarrow^* \sigma'$.

   We have $(\beta, C_3) \leq (\alpha, C_2) \leq (\alpha, C)$ and $\Phi' \supseteq \Phi$. Further, we have $(\mathsf{put}(\ell, v), h) \rightsquigarrow (e[v/x], h[\ell \mapsto \mathsf{E}]) \rightsquigarrow^* (v', h')$. Lastly, $\sigma \dashrightarrow \sigma[\ell \mapsto \mathsf{Done}] \dashrightarrow^* \sigma'$.

□

For the remaining expressions, properties about their behavior follow from the lemmas above. We showcase iteration. For iteration $\mathsf{iter}(n, e', x.e)$, the function that is iterated $\lambda x.e$ cannot make use of capabilities of existing channels. Otherwise, the capabilities would have to be duplicated for repeated iteration. The function may however allocate fresh channels. If α is an upper bound on the number of locations that are allocated by $\lambda x.e$, regardless of the argument, then $n \otimes \alpha$ is an upper bound on the number of locations that are allocated when iterating the function n-times. If the number of iterations is not known before the execution, we can bound it by $\omega \otimes \alpha$.

Lemma 4.17.

$$\frac{(e', R'_e) \in \mathcal{E}[\![A]\!]_i \qquad (\lambda x.e, R_{ord(\alpha)}) \in \mathcal{V}[\![A \multimap A]\!]_i}{(\mathsf{iter}(n, e', x.e), R'_e \oplus R_{ord(n \otimes \alpha)}) \in \mathcal{E}[\![A]\!]_i}$$

Proof. By induction on n for arbitrary i, e', and R'_e. With Lemma 4.13, it suffices to show $(\mathsf{iter}(n, v, x.e), R_v \oplus R_{\mathrm{ord}(n\otimes\alpha)}) \in \mathcal{E}[\![A]\!]_i$ for all $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![A]\!]_j$.

1. *Case* $n = 0$. As $R_{\mathrm{ord}(0\otimes\alpha)} = \epsilon$ and $\mathsf{iter}(0, v, x.e) \rightsquigarrow v$, the claim follows with Lemma 4.13 and $(v, R_v) \in \mathcal{V}[\![A]\!]_j$.
2. *Case* $n > 0$. By definition $\mathsf{iter}(n, v, x.e) \rightsquigarrow \mathsf{iter}(n-1, e[v/x], x.e)$. With Lemma 4.13 and Lemma 2.2, it suffices to show $(\mathsf{iter}(n-1, e[v/x], x.e), R_v \oplus R_{\mathrm{ord}(\alpha)} \oplus R_{\mathrm{ord}((n-1)\otimes\alpha)}) \in \mathcal{E}[\![A]\!]_j$. Since $(v, R_v) \in \mathcal{V}[\![A]\!]_j$ and $(\lambda x.e, R_{\mathrm{ord}(\alpha)}) \in \mathcal{V}[\![A \multimap A]\!]_i$ and $j \leq i$, we have $(e[v/x], R_v \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![A]\!]_j$. The claim follows by induction and Lemma 4.3.

□

Lemma 4.18.

$$\frac{(e, R) \in \mathcal{E}[\![\mathbb{N}]\!]_i \qquad (e_0, R_0) \in \mathcal{E}[\![A]\!]_i \qquad (\lambda x.e_S, R_{ord(\alpha_S)}) \in \mathcal{V}[\![A \multimap A]\!]_i}{(\mathsf{iter}(e, e_0, x.e_S), R \oplus R_0 \oplus R_{ord(\omega\otimes\alpha_S)}) \in \mathcal{E}[\![A]\!]_i}$$

Proof. By Lemma 4.13, it suffices to show:

$$(\mathsf{iter}(v, e_0, x.e_S), R_v \oplus R_0 \oplus R_{\mathrm{ord}(\omega\otimes\alpha_S)}) \in \mathcal{E}[\![A]\!]_j$$

for all $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![\mathbb{N}]\!]_j$. By definition of $\mathcal{V}[\![\mathbb{N}]\!]_j$, we have $v = n$ for some $n \in \mathbb{N}$ and $R_v = \epsilon$. Since $n \otimes \alpha_S \leq \omega \otimes \alpha_S$ by Lemma 2.2, by Lemma 4.12 it suffices to show:

$$(\mathsf{iter}(n, e_0, x.e_S), R_0 \oplus R_{\mathrm{ord}(n\otimes\alpha_S)}) \in \mathcal{E}[\![A]\!]_j$$

The claim follows with Lemma 4.3 and Lemma 4.17. □

4.4 Semantic Typing

To define the semantic typing judgement $\Gamma \vDash e : A$, we close the expression with a substitution from the context interpretation $\mathcal{G}[\![\Gamma]\!]_i$:

$$\begin{aligned}
\mathcal{G}[\![\cdot]\!]_i &\triangleq \{(\theta, \epsilon)\}\\
\mathcal{G}[\![\Gamma, x : A]\!]_i &\triangleq \{(\theta, R_\theta \oplus R) \mid (\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i \text{ and } (\theta x, R) \in \mathcal{V}[\![A]\!]_i\}\\
\Gamma \vDash e : A &\triangleq \exists \alpha. \forall i. \forall (\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i. (e[\theta], R_\theta \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![A]\!]_i
\end{aligned}$$

Context Interpretation Before we proceed to prove semantic soundness, we establish the following properties of the context interpretation $\mathcal{G}[\![\Gamma]\!]_i$.

Lemma 4.19. *If $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$, then $R_\theta = R_1 \oplus R_2$ for some resources R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$.*

Proof. By induction on Δ.

1. Case $\cdot$. Then $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i$. The claim follows with $R_1 \triangleq R_\theta$ and $R_2 \triangleq \epsilon$.
2. Case $\Delta, x : A$. Then $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta, x : A]\!]_i$. Thus, $R_\theta = R'_\theta \oplus R$ for some R'_θ, R with $(\theta x, R) \in \mathcal{V}[\![A]\!]_i$ and $(\theta, R'_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. By induction, there are R_1, R'_2 such that $R'_\theta = R_1 \oplus R'_2$ and $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R'_2) \in \mathcal{G}[\![\Delta]\!]_i$. We define $R_2 \triangleq R'_2 \oplus R$. By definition, $(\theta, R_2) \in \mathcal{G}[\![\Delta, x : A]\!]_i$.

□

Lemma 4.20.

1. *If $\theta =_{\mathrm{dom}\,\Gamma} \theta'$ and $(\theta, R) \in \mathcal{G}[\![\Gamma]\!]_i$, then $(\theta', R) \in \mathcal{G}[\![\Gamma]\!]_i$.*
2. *If $i \leq j$, then $\mathcal{G}[\![\Gamma]\!]_j \subseteq \mathcal{G}[\![\Gamma]\!]_i$.*

Proof. Both by induction on Γ. For the second, we use Lemma 4.3. □

Lemma 4.21. *If $\Gamma, x : A$ is defined and $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(v, R_v) \in \mathcal{V}[\![A]\!]_i$, then $(\theta[x \mapsto v], R_\theta \oplus R_v) \in \mathcal{G}[\![\Gamma, x : A]\!]_i$.*

Proof. By definition of $\mathcal{G}[\![\Gamma, x : A]\!]_i$ it suffices to show $(\theta[x \mapsto v], R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i$. As type contexts are linear, we know $x \notin \mathrm{dom}\,\Gamma$. Thus $\theta[x \mapsto v] =_{\mathrm{dom}\,\Gamma} \theta$. The claim follows with Lemma 4.20. □

Semantic Soundness In the following, we prove semantic soundness of the type system $\Gamma \vdash e : A$ with respect to the semantic interpretation $\Gamma \vDash e : A$. The proof proceeds by induction using compatibility lemmas in each case. We prove those compatibility lemmas in Section 4.5. Given the semantic soundness proof, it is then straightforward to derive termination.

Theorem 4.1. *If $\Gamma \vdash e : A$, then $\Gamma \vDash e : A$.*

Proof. By induction on $\Gamma \vdash e : A$ using Lemmas 4.22 to 4.37. □

Corollary 4.1. *If $\cdot \vdash e : A$, then $(e, \cdot) \rightsquigarrow^* (v, h)$ for some value v and heap h.*

Proof. By Lemma 4.1, we have $\cdot \vDash e : A$. Thus, there is some α such that:

$$\forall i. (e[id], \epsilon \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![A]\!]_i$$

since $(id, \epsilon) \in \mathcal{G}[\![\cdot]\!]_i$ for all i by definition. We pick $i \triangleq (\alpha, \emptyset)$. Thus $(e, R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![A]\!]_i$. By definition of $\mathcal{E}[\![A]\!]_i$, it suffices to show $\mathsf{RI}_i(R_{\mathrm{ord}(\alpha)}, \epsilon, \emptyset, \cdot, \emptyset, \emptyset)$. By definition $(\cdot, \emptyset, \emptyset) \in \mathcal{H}[\![\emptyset]\!]_i$. Further, used $\emptyset = \emptyset \mathbin{\#} \emptyset$. Lastly, $\mathsf{sum}(\emptyset) \oplus R_{\mathrm{ord}(\alpha)} \oplus \epsilon \oplus R_{\mathsf{cap}(\mathsf{idx}\,\emptyset)} = R_{\mathrm{ord}(\alpha)} \sqsubseteq (\emptyset, \emptyset, \alpha)$. □

4.5 Compatibility Lemmas

Lemma 4.22.

$$\frac{}{x : A \vDash x : A}$$

Proof. Pick $\alpha \triangleq 0$. Let $(\theta, R) \in \mathcal{G}[\![x : A]\!]_i$. Then $(\theta x, R) \in \mathcal{V}[\![A]\!]_i$. It remains to show $(x[\theta], R \oplus R_{\mathrm{ord}(0)}) \in \mathcal{E}[\![A]\!]_i$. As $(x[\theta], R \oplus R_{\mathrm{ord}(0)}) = (\theta x, R)$, the claim follows with Lemma 4.13. □

Lemma 4.23.

$$\frac{\Gamma \vDash e : B}{\Gamma, x : A \vDash e : B}$$

Proof. By assumption, we have some α_e for e. We pick $\alpha \triangleq \alpha_e$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, x : A]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![x : A]\!]_i$ by Lemma 4.19. It remains to show $(e[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha)} \oplus R_2) \in \mathcal{E}[\![B]\!]_i$. By Lemma 4.12, it suffices to show:

$$(e[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![B]\!]_i$$

which follows by assumption with $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$. □

Lemma 4.24.

$$\frac{}{\cdot \vDash () : \mathbb{1}}$$

Proof. Pick $\alpha \triangleq 0$. Let $(\theta, R) \in \mathcal{G}[\![\cdot]\!]_i$. It remains to show $(()[\theta], R_{\mathrm{ord}(0)}) \in \mathcal{E}[\![\mathbb{1}]\!]_i$. As $(()[\theta], R_{\mathrm{ord}(0)}) = ((), \epsilon) \in \mathcal{V}[\![\mathbb{1}]\!]_i$, the claim follows with Lemma 4.13. □

Lemma 4.25.

$$\frac{\Gamma \vDash e_1 : \mathbb{1} \qquad \Delta \vDash e_2 : A}{\Gamma, \Delta \vDash e_1; e_2 : A}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $(e_1[\theta]; e_2[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_i$.

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![\mathbb{1}]\!]_i$. By Lemma 4.13, it suffices to show:

$$(v_1; e_2[\theta], R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![\mathbb{1}]\!]_j$. By definition of $\mathcal{V}[\![\mathbb{1}]\!]_j$, we know $v_1 = ()$ and $R_{v_1} = \epsilon$. By assumption $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_i$. The claim follows with Lemma 4.13 and Lemma 4.3 given the pure reduction $(); (e_2[\theta]) \rightsquigarrow e_2[\theta]$. □

Lemma 4.26.

$$\frac{}{\cdot \vDash b : \mathbb{B}}$$

Proof. Pick $\alpha \triangleq 0$. Let $(\theta, R) \in \mathcal{G}[\![\cdot]\!]_i$. It remains to show $(b[\theta], R_{\mathrm{ord}(0)}) \in \mathcal{E}[\![\mathbb{B}]\!]_i$. As $(b[\theta], R_{\mathrm{ord}(0)}) = (b, \epsilon) \in \mathcal{V}[\![\mathbb{B}]\!]_i$, the claim follows with Lemma 4.13. □

Lemma 4.27.

$$\frac{\Gamma \vDash e : \mathbb{B} \qquad \Delta \vDash e_1 : A \qquad \Delta \vDash e_2 : A}{\Gamma, \Delta \vDash \mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : A}$$

Proof. By assumption, we have α_e for e, α_1 for e_1, and α_2 for e_2. We pick $\alpha \triangleq \alpha_e \oplus \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. Thus, it remains to show:

$$(\mathsf{if}\ e[\theta]\ \mathsf{then}\ e_1[\theta]\ \mathsf{else}\ e_2[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_e)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_1 \oplus \alpha_2)}) \in \mathcal{E}[\![A]\!]_i$$

By assumption $(e[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_e)}) \in \mathcal{E}[\![\mathbb{B}]\!]_i$. Thus, by Lemma 4.13, it suffices to show:

$$(\mathsf{if}\ v\ \mathsf{then}\ e_1[\theta]\ \mathsf{else}\ e_2[\theta], R_v \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_1 \oplus \alpha_2)}) \in \mathcal{E}[\![A]\!]_j$$

for all $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![\mathbb{B}]\!]_j$. By definition of the value relation, we have $R_v = \epsilon$ and $v = \mathsf{true}$ or $v = \mathsf{false}$.

1. Let $v = \mathsf{true}$. By assumption $(e_1[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![A]\!]_i$. By Lemma 4.12 and Lemma 4.3, we have $(e_1[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_1 \oplus \alpha_2)}) \in \mathcal{E}[\![A]\!]_j$. The claim follows with Lemma 4.13 since $\mathsf{if\ true\ then}\ e_1[\theta]\ \mathsf{else}\ e_2[\theta] \rightsquigarrow e_1[\theta]$.
2. Let $v = \mathsf{false}$. By assumption $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_i$. By Lemma 4.12 and Lemma 4.3, we have $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_1 \oplus \alpha_2)}) \in \mathcal{E}[\![A]\!]_j$. The claim follows with Lemma 4.13 since $\mathsf{if\ false\ then}\ e_1[\theta]\ \mathsf{else}\ e_2[\theta] \rightsquigarrow e_2[\theta]$.

□

Lemma 4.28.

$$\frac{}{\cdot \vDash n : \mathbb{N}}$$

Proof. Pick $\alpha \triangleq 0$. Let $(\theta, R) \in \mathcal{G}[\![\cdot]\!]_i$. It remains to show $(n[\theta], R_{\mathrm{ord}(0)}) \in \mathcal{E}[\![\mathbb{N}]\!]_i$. As $(n[\theta], R_{\mathrm{ord}(0)}) = (n, \epsilon) \in \mathcal{V}[\![\mathbb{N}]\!]_i$, the claim follows with Lemma 4.13. □

Lemma 4.29.

$$\frac{\Gamma \vDash e_1 : \mathbb{N} \qquad \Delta \vDash e_2 : \mathbb{N}}{\Gamma, \Delta \vDash e_1 \dotplus e_2 : \mathbb{N}}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $(e_1[\theta] \dotplus e_2[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{N}]\!]_i$.

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![\mathbb{N}]\!]_i$. By Lemma 4.13, it suffices to show:

$$(v_1 \dotplus e_2[\theta], R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{N}]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![\mathbb{N}]\!]_j$. By assumption $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{N}]\!]_i$ and by Lemma 4.3 $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{N}]\!]_j$. Thus, by Lemma 4.13, it suffices to show:

$$(v_1 \dotplus v_2, R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![\mathbb{N}]\!]_k$$

for all $k \leq j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![\mathbb{N}]\!]_k$. By definition of $\mathcal{V}[\![\mathbb{N}]\!]\_$, we know $v_1 = n_1$ and $v_2 = n_2$ and $R_1 = \epsilon$ and $R_2 = \epsilon$ for some $n_1, n_2 \in \mathbb{N}$. As $(n_1 + n_2, \epsilon) \in \mathcal{V}[\![\mathbb{N}]\!]_k$, the claim follows with Lemma 4.13 given the pure reduction $n_1 \dotplus n_2 \rightsquigarrow n_1 + n_2$. □

Lemma 4.30.

$$\frac{\Gamma \vDash e : \mathbb{N} \qquad \Delta \vDash e_0 : A \qquad x : A \vDash e_S : A}{\Gamma, \Delta \vDash \mathsf{iter}(e, e_0, x.e_S) : A}$$

Proof. By assumption, we have α_e for e, α_0 for e_0, and α_S for e_S. We pick $\alpha \triangleq \alpha_e \oplus \alpha_0 \oplus \omega \otimes \alpha_S$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. Thus, it remains to show:

$$\big(\mathsf{iter}(e[\theta], e_0[\theta], x.e_S[\theta[x \mapsto x]]), R_1 \oplus R_{\mathrm{ord}(\alpha_e)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_0)} \oplus R_{\mathrm{ord}(\omega \otimes \alpha_S)}\big) \in \mathcal{E}[\![A]\!]_i$$

We have $(e[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_e)}) \in \mathcal{E}[\![\mathbb{N}]\!]_i$ and $(e_0[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_0)}) \in \mathcal{E}[\![A]\!]_i$ by assumption. Hence, by Lemma 4.18, it suffices to show:

$$(\lambda x.e_S[\theta[x \mapsto x]], R_{\mathrm{ord}(\alpha_S)}) \in \mathcal{V}[\![A \multimap A]\!]_i$$

Let $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![A]\!]_j$. We show $(e_S[\theta[x \mapsto x]][v/x], R_{\mathrm{ord}(\alpha_S)} \oplus R_v) \in \mathcal{E}[\![A]\!]_j$. By definition $(\theta[x \mapsto v], R_v) \in \mathcal{G}[\![x : A]\!]_j$ and thus by assumption $(e_S[\theta[x \mapsto v]], R_v \oplus R_{\mathrm{cap}(\alpha_S)}) \in \mathcal{E}[\![A]\!]_j$. The claim follows with $e_S[\theta[x \mapsto x]][v/x] = e_S[\theta[x \mapsto v]]$. □

Lemma 4.31.

$$\frac{\Gamma \vDash e_1 : A_1 \qquad \Delta \vDash e_2 : A_2}{\Gamma, \Delta \vDash (e_1, e_2) : A_1 \otimes A_2}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 and $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $((e_1[\theta], e_2[\theta]), R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A_1 \otimes A_2]\!]_i$.

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![A_1]\!]_i$. By Lemma 4.13, it suffices to show:

$$((v_1, e_2[\theta]), R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A_1 \otimes A_2]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![A_1]\!]_j$. We have $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A_2]\!]_i$ by assumption and thus $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A_2]\!]_j$ by Lemma 4.3. By Lemma 4.13, it suffices to show:

$$((v_1, v_2), R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![A_1 \otimes A_2]\!]_k$$

for all $k \leq j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![A_2]\!]_k$. The claim follows with Lemma 4.12 and Lemma 4.3. □

Lemma 4.32.

$$\frac{\Gamma \vDash e_1 : A_1 \otimes A_2 \qquad \Delta, x : A_1, y : A_2 \vDash e_2 : B}{\Gamma, \Delta \vDash \mathsf{let}\ (x, y) = e_1\ \mathsf{in}\ e_2 : B}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 and $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show:

$$(\mathsf{let}\ (x, y) = e_1[\theta]\ \mathsf{in}\ e_2[\theta[x \mapsto x, y \mapsto y]], R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![B]\!]_i$$

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![A_1 \otimes A_2]\!]_i$. By Lemma 4.13, it suffices to show:

$$(\mathsf{let}\ (x, y) = v\ \mathsf{in}\ e_2[\theta[x \mapsto x, y \mapsto y]], R_v \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![B]\!]_j$$

for all $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![A_1 \otimes A_2]\!]_j$. By definition of the value relation, we have $v = (v_1, v_2)$ and $R = R_{v_1} \oplus R_{v_2}$ for some $(v_1, R_{v_1}) \in \mathcal{V}[\![A_1]\!]_j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![A_2]\!]_j$. By Lemma 4.13, it suffices to show

$$(e_2[\theta[x \mapsto v_1, y \mapsto v_2]], R_{v_1} \oplus R_{v_2} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![B]\!]_j$$

since $\mathsf{let}\ (x, y) = (v_1, v_2)\ \mathsf{in}\ e_2[\theta[x \mapsto x, y \mapsto y]] \rightsquigarrow e_2[\theta[x \mapsto x, y \mapsto y]][v_1/x, v_2/y] = e_2[\theta[x \mapsto v_1, y \mapsto v_2]]$. By Lemma 4.20, we have $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_j$. Thus, $(\theta[x \mapsto v_1, y \mapsto v_2], R_2 \oplus R_{v_1} \oplus R_{v_2}) \in \mathcal{G}[\![\Delta, x : A_1, y : A_2]\!]_j$ by Lemma 4.21. The claim follows by the assumption for e_2. □

Lemma 4.33.

$$\frac{\Gamma, x : A \vDash e : B}{\Gamma \vDash \lambda x.e : A \multimap B}$$

Proof. By assumption we have some α_e for e. We pick $\alpha \triangleq \alpha_e$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i$. It remains to show $(\lambda x.(e[\theta[x \mapsto x]]), R_\theta \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![A \multimap B]\!]_i$. By Lemma 4.13, $(\lambda x.(e[\theta[x \mapsto x]]), R_\theta \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{V}[\![A \multimap B]\!]_i$ suffices. Let $j \leq i$ and $(v, R_v) \in \mathcal{V}[\![A]\!]_j$. We show:

$$(e[\theta[x \mapsto x]][v/x], R_\theta \oplus R_{\mathrm{ord}(\alpha)} \oplus R_v) \in \mathcal{E}[\![B]\!]_j$$

By Lemma 4.20, we have $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_j$. Hence $(\theta[x \mapsto v], R_\theta \oplus R_v) \in \mathcal{G}[\![\Gamma, x : A]\!]_j$ by Lemma 4.21. By assumption, we have $(e[\theta[x \mapsto v]], R_\theta \oplus R_v \oplus R_{\mathrm{ord}(\alpha)}) \in \mathcal{E}[\![B]\!]_j$. The claim follows with $e[\theta[x \mapsto v]] = e[\theta[x \mapsto x]][v/x]$. □

Lemma 4.34.

$$\frac{\Gamma \vDash e_1 : A \multimap B \qquad \Delta \vDash e_2 : A}{\Gamma, \Delta \vDash e_1\ e_2 : B}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $(e_1[\theta]\ e_2[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![B]\!]_i$.

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![A \multimap B]\!]_i$. By Lemma 4.13, it suffices to show:

$$(v_1\ e_2[\theta], R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![B]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![A \multimap B]\!]_j$. By assumption, we have $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_i$ and thus $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_j$ by Lemma 4.3. Hence, by Lemma 4.13, it suffices to show:

$$(v_1\ v_2, R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![B]\!]_k$$

for all $k \leq j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![A]\!]_k$. By definition of $\mathcal{V}[\![A \multimap B]\!]_j$, we know $v_1 = \lambda x.e$ for some x and e. Further, we have $(e[v_2/x], R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![B]\!]_k$. The claim follows by Lemma 4.13 given the pure reduction $(\lambda x.e)\ v_2 \rightsquigarrow e[v_2/x]$. □

Lemma 4.35.

$$\frac{\Gamma \vDash e_1 : \mathsf{Get}\ A \qquad \Delta \vDash e_2 : A \multimap \mathbb{1}}{\Gamma, \Delta \vDash \mathsf{get}(e_1, e_2) : \mathbb{1}}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $(\mathsf{get}(e_1[\theta], e_2[\theta]), R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{1}]\!]_i$. By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![\mathsf{Get}\ A]\!]_i$. By Lemma 4.13, it suffices to show:

$$(\mathsf{get}(v_1, e_2[\theta]), R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{1}]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![\mathsf{Get}\ A]\!]_j$. By assumption, we have $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A \multimap \mathbb{1}]\!]_i$ and thus by Lemma 4.3 it follows $(e_2[\theta], R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![A \multimap \mathbb{1}]\!]_j$. Hence, by Lemma 4.13, it suffices to show:

$$(\mathsf{get}(v_1, v_2), R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![\mathbb{1}]\!]_k$$

for all $k \leq j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![A \multimap \mathbb{1}]\!]_k$. The claim follows with Lemma 4.15 and Lemma 4.3. □

Lemma 4.36.

$$\frac{\Gamma \vDash e_1 : \mathsf{Put}\ A \qquad \Delta \vDash e_2 : A}{\Gamma, \Delta \vDash \mathsf{put}(e_1, e_2) : \mathbb{1}}$$

Proof. By assumption, we have α_1 for e_1 and α_2 for e_2. We pick $\alpha \triangleq \alpha_1 \oplus \alpha_2$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma, \Delta]\!]_i$. Then $R_\theta = R_1 \oplus R_2$ for some R_1, R_2 such that $(\theta, R_1) \in \mathcal{G}[\![\Gamma]\!]_i$ and $(\theta, R_2) \in \mathcal{G}[\![\Delta]\!]_i$ by Lemma 4.19. It remains to show $(\mathsf{put}(e_1[\theta], e_2[\theta]), R_1 \oplus R_{\mathrm{ord}(\alpha_1)} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{1}]\!]_i$.

By assumption $(e_1[\theta], R_1 \oplus R_{\mathrm{ord}(\alpha_1)}) \in \mathcal{E}[\![\mathsf{Put}\ A]\!]_i$. By Lemma 4.13, it suffices to show:

$$(\mathsf{put}(v_1, e_2[\theta]), R_{v_1} \oplus R_2 \oplus R_{\mathrm{ord}(\alpha_2)}) \in \mathcal{E}[\![\mathbb{1}]\!]_j$$

for all $j \leq i$ and $(v_1, R_{v_1}) \in \mathcal{V}[\![\textsf{Put}\ A]\!]_j$. By assumption, we have $(e_2[\theta], R_2 \oplus R_{\text{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_i$. Thus, $(e_2[\theta], R_2 \oplus R_{\text{ord}(\alpha_2)}) \in \mathcal{E}[\![A]\!]_j$ by Lemma 4.3. Hence, by Lemma 4.13, it suffices to show:

$$(\textsf{put}(v_1, v_2), R_{v_1} \oplus R_{v_2}) \in \mathcal{E}[\![\mathbb{1}]\!]_k$$

for all $k \leq j$ and $(v_2, R_{v_2}) \in \mathcal{V}[\![A]\!]_k$. The claim follows with Lemma 4.16 and Lemma 4.3. □

Lemma 4.37.

$$\frac{\Gamma, x : \textsf{Get}\ A, y : \textsf{Put}\ A \vDash e : B}{\Gamma \vDash \textsf{let}\ (x, y) = \textsf{chan}()\ \textsf{in}\ e : B}$$

Proof. By assumption we have some α_e for e. We pick $\alpha \triangleq \alpha_e \oplus 1$. Let $(\theta, R_\theta) \in \mathcal{G}[\![\Gamma]\!]_i$. We show $(\textsf{let}\ (x, y) = \textsf{chan}()\ \textsf{in}\ (e[\theta[x \mapsto x, y \mapsto y]]), R_\theta \oplus R_{\text{ord}(\alpha_e)} \oplus R_{\text{ord}(1)}) \in \mathcal{E}[\![B]\!]_i$. To show the claim, we use Lemma 4.14. Let ℓ be some location. It remains to show:

$$(e[\theta[x \mapsto x, y \mapsto y]][\ell/x, \ell/y], R_\theta \oplus R_{\text{cap}(\alpha_e)} \oplus R_{\text{get}(\ell,A)} \oplus R_{\text{put}(\ell,A)}) \in \mathcal{E}[\![B]\!]_i$$

By definition $(\ell, R_{\text{get}(\ell,A)}) \in \mathcal{V}[\![\textsf{Get}\ A]\!]_i$ and $(\ell, R_{\text{put}(\ell,A)}) \in \mathcal{V}[\![\textsf{Put}\ A]\!]_i$. Thus, we have:

$$(\theta[x \mapsto \ell, y \mapsto \ell], R_\theta \oplus R_{\text{get}(\ell,A)} \oplus R_{\text{put}(\ell,A)}) \in \mathcal{G}[\![\Gamma, x : \textsf{Get}\ A, y : \textsf{Put}\ A]\!]_i$$

with Lemma 4.21. The claim follows by assumption with $e[\theta[x \mapsto x, y \mapsto y]][\ell/x, \ell/y] = e[\theta[x \mapsto \ell, y \mapsto \ell]]$. □

References

[1] A. Ahmed, M. Fluet, and G. Morrisett. L^3: a linear language with locations. *Fundamenta Informaticae*, 77(4):397–449, 2007.

[2] C. Calcagno, P. W. O'Hearn, and H. Yang. Local action and abstract separation logic. In *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 366–378. IEEE, 2007.

[3] G. Hessenberg. *Grundbegriffe der Mengenlehre*, volume 1. Vandenhoeck & Ruprecht, 1906.

[4] N. R. Krishnaswami, A. Turon, D. Dreyer, and D. Garg. Superficially substructural types. In *Proceedings of the 17th ACM SIGPLAN International Conference on Functional Programming*, pages 41–54, 2012.

[5] A. J. Turon, J. Thamsborg, A. Ahmed, L. Birkedal, and D. Dreyer. Logical relations for fine-grained concurrency. In *POPL*, pages 343–356. ACM New York, NY, USA, 2013.