A type-theory for higher-order amortized cost analysis

Vineet Rajani
vrajani@mpi-sws.org

Marco Gaboardi
gaboardi@bu.edu

Deepak Garg
dg@mpi-sws.org

Jan Hoffmann
jhoffmann@cmu.edu

Technical report MPI-SWS-2020-004
May 2020

Contents

Abstract

This paper presents λ-amor, a new type-theoretic framework for *amortized cost analysis* of higher-order functional programs. λ-amor introduces a new modality for representing *potentials* – costs that have been accounted for, but not yet incurred, which are central to amortized analysis. Additionally, λ-amor relies on standard type-theoretic concepts like affineness (to prevent duplication of potentials, which could lead to unsoundness), refinement types (for expressiveness) and an indexed cost monad (for actual costs). Although λ-amor is proved sound using a rather simple logical relation, it is very expressive. In fact, we use it as a meta-framework to embed other existing type theories for cost analysis, ranging from call-by-value to call-by-name evaluation, and from effect to co-effect tracking of costs. Using one of our embeddings, we also show that λ-amor is relatively complete for all terminating PCF programs.

Keywords: amortized cost analysis, type theory, relative completeness

1 Introduction

Cost analysis refers to the static verification of an upper bound on the evaluation cost of a program,[1] measured in some (application-dependent) abstract unit such as the number of reduction steps, the number of function calls or the number of case analyses during execution of the program. Typically, cost analysis requires a sound static or type-theoretic approximation of the *worst-case* cost. However, in many cases, specifically, for operations on data structures with internal state, it is more useful to talk of the *amortized* or average cost of a sequence of n invocations of an operation, since some invocations may pay a huge cost to change the internal state in a way that *reduces* the cost of subsequent invocations [32]. This kind of an average cost analysis is called an amortized cost analysis. Elementary examples that rely on amortized analysis are a FIFO queue implemented using a pair of functional (LIFO) lists, Fibonacci heaps and the union-find data structure with path compression.

There are type systems for cost analysis, both amortized [16, 20, 19, 17, 18, 11, 26, 23] and non-amortized [6, 8, 12, 3, 24, 9, 10]. Our broad goal in this paper is to develop a type-theoretic framework to unify as many of these lines as possible. In doing so, we also extend the expressiveness of existing frameworks for amortized cost analysis. We call our framework λ-amor and examine its properties, including its expressiveness, in detail. In the remainder of this introduction, we give a high-level description of λ-amor and its properties, followed by a comparison to existing work.

Basics of amortized cost analysis. To motivate λ-amor's design, we first describe the typical structure of amortized cost analysis. Suppose we want to prove that the average cost of running an operation on a data structure is c, when the actual cost of the operation depends on the internal state of the data structure. To do this, we find a function ϕ that maps the state s of the data structure to a non-negative number, called a *potential*, and show (using a type theory like λ-amor) that an invocation of the operation that changes the data structure from state s_i to s_{i+1} has a cost upper-bounded by $\phi(s_i) - \phi(s_{i+1}) + c$. It immediately follows that a sequence of n operations starting in state s_0 with $\phi(s_0) = 0$ has a total cost upper-bounded by $(\phi(s_0) - \phi(s_1) + c) + ... + (\phi(s_{n-1}) - \phi(s_n) + c)$. This is a telescopic series that equals $\phi(s_0) - \phi(s_n) + n \cdot c$, which in turn is upper-bounded by $n \cdot c$ since $\phi(s_0) = 0$ and $\phi(s_n)$ is non-negative. Hence, the average cost of each operation is no more than c, as required. The value $\phi(s)$ is called the potential associated with the state s. This potential is needed for verification only, i.e., it is ghost state and it does not exist at run time. The type theory is used to prove only that the cost of an individual operation is upper-bounded by $\phi(s_i) - \phi(s_{i+1}) + c$ (the rest is trivial).

Based on this intuition, we describe the requirements for a type theory to support amortized cost analysis, and how λ-amor satisfies these requirements.

1) The type theory must include some construct to associate the ghost potential to the type of a data structure. To this end, λ-amor introduces a new type constructor written $[p]\,\tau$, which ascribes a value of type τ with associated potential p. Here, p is a non-negative number.

2) Since the potential p is related to the state s (p equals $\phi(s)$), the type τ of the data structure must reflect its state to sufficient precision, to allow relating p and τ meaningfully. λ-amor uses standard *refinement types* [33] for this. For instance, $L^n\ \tau$ is the type of lists of length n, and $[2n]\,(L^n\ \tau)$ is the type of lists of length n with associated potential $2n$. Note how the refinement n relates the an aspect of the state (the length of the list) to the potential associated with it.

3) The type theory must represent execution costs since we need to establish upper-bounds on them. Costs can be represented as either an effect (monad) or a coeffect (comonad). Prior literature has considered both options [11, 16, 9, 10]. Somewhat arbitrarily, λ-amor uses effects: We include an indexed monad $\mathbb{M}\,\kappa\,\tau$ to represent a computation that has a cost κ, which is also a non-negative number. However, we show that coeffect based cost analysis can be *embedded* or simulated in λ-amor using potentials.

4) The type theory must prevent duplication of any type that has a potential associated with it, else analysis may not be sound. For example, if a typing derivation duplicates the potential $\phi(s_i)$ even once, then

[1] Cost analysis can also be used to establish lower bounds but, here, as in most prior work, the focus is on upper bounds.

the operation's real cost may be up to $2 \cdot \phi(s_i) - \phi(s_{i+1}) + c$, and the amortized analysis described earlier breaks completely. Hence, all potential-carrying types must be treated *affinely* in the type theory. Accordingly, λ-amor is an affine type theory with the usual operators of affine logic like $\otimes$, & and $\oplus$. Duplicable resources are explicitly represented using the standard exponential !. To improve expressiveness, we allow the exponential ! to be indexed, following the dependent sub-exponential from Bounded Linear Logic [15].

Summary of λ-amor and our results. Overall, λ-amor is a λ-calculus equipped with a type system that has the four features mentioned above – the construct $[p]\,\tau$ to associate potential to a type, type refinements, the indexed cost monad $\mathbb{M}\,\kappa\,\tau$, and affineness with an indexed exponential !.[2] We give λ-amor a call-by-name semantics with eager evaluation for all pairs and sums. Although a call-by-name semantics might seem odd for cost analysis (since most languages are call-by-value or call-by-need), note that, here, costs are confined to a monad, so the semantics of pure evaluation are insignificant. We choose call-by-name since it simplifies our technical development. In fact, we show via an embedding how call-by-value amortized analysis can be simulated in λ-amor.

Despite the large number of features, λ-amor is conceptually very simple. We prove it sound using an elementary logical relation that extends Pym's semantics of BI [31]. The key novelty in building this relation is the treatment of potentials, and their interaction with the cost monad (available potential can offset the cost in the monad).

Finally, we show that two existing, state-of-the-art frameworks for (amortized) cost analysis can be embedded faithfully in λ-amor. Our first embedding is that of a core calculus for Resource-aware ML or RAML [16, 19], an implemented, widely-used framework for amortized cost analysis of ML programs. RAML is call-by-value, so this embedding shows how call-by-value (amortized) analysis can be simulated in λ-amor. Other effect-based type systems like the unary fragment of [6] are conceptually simpler than RAML as they do not support amortized analysis, so embedding them in λ-amor is even easier.

Our second embedding is that of $d\ell$PCF [9], a coeffect based type system for PCF, that is relatively complete.[3] This embedding is difficult and shows two things: (a) coeffect-based cost analysis can be simulated in an effect-based system using potentials, and (b) λ-amor is also relatively complete for typing PCF in the sense that all terminating PCF programs can be typed with precise cost in λ-amor, establishing that λ-amor is extremely expressive.

Together, these embeddings show that λ-amor can represent cost analyses from very different settings ranging from call-by-value to call-by-name, and effects vs coeffects. In this sense, we may view it as a unifying framework for (amortized) cost analyses.

Prior work on amortized cost analysis. Besides being a unifying framework for cost analysis as just described, λ-amor also improves the expressiveness of prior work on amortized cost analysis in the call-by-value setting. The state of the art in this setting is the aforementioned RAML. However, RAML has difficulty dealing with the interaction between higher-order values and potential because it is not based on an affine calculus. To understand the issue, consider a Curried function of two arguments, of which the first carries potential that is used to offset the cost of executing the function. Suppose that this function is *applied partially.* The resulting closure must not be duplicable because it captures the potential from the first argument. For this, one needs an affine type system. Since RAML (and its many extensions) are fundamentally not affine, they cannot handle this example correctly, and the best that exists so far is to limit potential to the last argument of a Curried function. In contrast, λ-amor, being fundamentally affine, can handle this example trivially.

There is also work on amortized cost analysis of call-by-need (lazy) programs, e.g., by [11], who formalizes the seminal thesis of Okasaki on the so-called method of debits [30]. Although we cannot embed this line of work *faithfully* in λ-amor due to the fundamental difficulty of simulating call-by-need in call-by-name (this difficulty is not specific to our work), we show how one specific example from Okasaki's/Danielsson's work can also be analyzed in λ-amor's type system. The cost and the potential functions do not change. We are fairly confident that this "porting" generalizes to most examples of Okasaki.

Finally, there is work on using *program logics* for amortized cost analysis [4, 5, 27]. While we *expect* program logics to be very expressive, this line of work *does not actually* show any embeddings of existing frameworks. Hence, unlike our work, this line of work does not take concrete steps towards a common framework for (amortized) cost analysis. Note that, for our purposes, the choice between the use of a type system and program logic is one of personal taste; we could also have chosen to build λ-amor on a program logic instead of a type system and shown similar embeddings.

Organization. To simplify the presentation, we describe λ-amor in two stages. First, we describe λ-amor without indexing on exponentials (Section 2). This suffices for most examples (Section 3) and the embedding of RAML (Section 4), but not the embedding of $d\ell$PCF. Then, we introduce the indexed exponentials (Section 5) and show how to embed $d\ell$PCF (Section 6), thus establishing relative completeness for PCF programs. Appendix A, Appendix B and Appendix C describe the full technical details of λ-amor$^-$, λ-amor (full) and

[2]The name "λ-amor" refers to both the calculus and its type system. The intended use can be disambiguated from the context.
[3]The adjective "relative" means relative to having a refinement domain that is sufficiently expressive.

Types	τ	$::=$	$\mathbf{1} \mid \mathsf{b} \mid \tau_1 \multimap \tau_2 \mid \tau_1 \otimes \tau_2 \mid \tau_1 \,\&\, \tau_2 \mid \tau_1 \oplus \tau_2 \mid !\tau \mid [p]\,\tau \mid \mathbb{M}\,\kappa\,\tau \mid L^n\ \tau$
			$\alpha \mid \forall\alpha : \mathsf{K}.\tau \mid \forall i : \mathsf{S}.\tau \mid \lambda_t i : \mathsf{S}.\tau \mid \tau\ I \mid \exists i : \mathsf{S}.\tau \mid C \Rightarrow \tau \mid C\&\tau$
Expressions	e	$::=$	$v \mid x \mid e_1\ e_2 \mid \langle\langle e_1, e_2\rangle\rangle \mid \mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{\ in\ } e_2 \mid \mathsf{fix}\, x.e \mid$
			$\langle e, e\rangle \mid \mathsf{fst}(e) \mid \mathsf{snd}(e) \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid \mathsf{case}\ e, x.e, y.e \mid$
			$\mathsf{let}\,!\,x = e_1 \mathsf{\ in\ } e_2 \mid e :: e \mid (\mathsf{match}\ e \mathsf{\ with\ } \mid nil \mapsto e_1 \mid h :: t \mapsto e_2) \mid e\ [] \mid$
			$\mathsf{xlet}\, x = e_1 \mathsf{\ in\ } e_2 \mid \mathsf{clet}\, x = e_1 \mathsf{\ in\ } e_2$
Values	v	$::=$	$() \mid c \mid \lambda x.e \mid \langle\langle v_1, v_2\rangle\rangle \mid \langle v, v\rangle \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid\ !\,e \mid nil \mid$
			$\Lambda.e \mid \mathsf{ret}\, e \mid \mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2 \mid \uparrow^\kappa \mid \mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2 \mid \mathsf{store}\, e$
Indices	I, κ, p, n	$::=$	$i \mid N \mid R^+ \mid I + I \mid I - I \mid \lambda_s i : \mathsf{S}.I \mid I\ I$
Constraints	C	$::=$	$I = I \mid I < I \mid C \wedge C$
Sorts	S	$::=$	$\mathbb{N} \mid \mathbb{R}^+ \mid \mathsf{S} \to \mathsf{S}$
Kinds	K	$::=$	$Type \mid \mathsf{S} \to \mathsf{K}$

Figure 1: λ-amor$^-$'s syntax

type-checked encoding of several examples respectively.

Limitations and scope. We focus on the *foundations* of a unifying type theory for (amortized) cost analysis. An implementation of this type theory is beyond the scope of this paper. Nonetheless, we expect that in restricted settings like only polynomial-time analysis, one could use ideas from prior work like RAML to implement λ-amor efficiently. Also, we focus only on additive (non-reusable) costs like time. Cost associated with reusable resources like heap space, which frameworks like RAML consider, are interesting future work.

2 λ-amor$^-$

To simplify the presentation, we first describe λ-amor$^-$, a subset of λ-amor that only considers the standard exponential ! from affine logic, without any indexing (that λ-amor supports).

2.1 Syntax and Semantics

The syntax of λ-amor$^-$ is shown in Fig. 1. We describe the various syntactic categories below.

Indices, sorts, kinds and constraints. λ-amor$^-$ is a refinement type system. (Static) indices, à la DML [33], are used to track information like list lengths, computation costs and potentials. List lengths are represented using natural numbers (sort $\mathbb{N}$). Potentials and costs are both represented using non-negative real numbers (sort $\mathbb{R}^+$). Besides this, we also have index-level function and their application (required for some examples like that in Section 3.1). λ-amor$^-$ also features kinds, denoted by K. *Type* is the kind of standard affine types and $\mathsf{S} \to \mathsf{K}$ represents a kind family indexed by the sort S. Finally, constraints (denoted by C) are predicates $(=, <, \wedge)$ over indices.

Types. λ-amor$^-$ is an affine type system. The most important type is the modal type $[p]\,\tau$, which ascribes values of type τ that have potential p associated with them (as a ghost). We have the multiplicative unit type (denoted $\mathbf{1}$) and an abstract base type (denoted b) to represent types like integers or booleans. Then, there are standard affine types – affine function spaces ($\multimap$), sums ($\oplus$), pairs (both the multiplicative $\otimes$ and the additive &) and the exponential (!), which ascribes expressions that can be duplicated. We include only one representative data type – the length-refined list type $L^n\tau$, where the length n is drawn from the language of indices (described earlier). Other data types can be added if needed.

λ-amor$^-$ also has universal quantification over types and indices denoted by $\forall\alpha : \mathsf{K}.\tau$ and $\forall i : \mathsf{S}.\tau$, respectively, and existential quantification over indices denoted $\exists i : \mathsf{S}.\tau$. The constraint type $C \Rightarrow \tau$ means that *if* constraint C holds *then* the underlying term has the type τ. The dual type $C\&\tau$ means that the constraint C holds *and* the type of the underlying term is τ. For instance, the type of non-empty lists can be written as $\exists n.\,(n > 0)\&(L^n\tau)$. We also have sort-indexed type families, which are type-level functions from sorts to kinds.

Finally, λ-amor$^-$ has the monadic type $\mathbb{M}\,\kappa\,\tau$, which represents computations of cost at most κ. Technically, $\mathbb{M}\,\kappa\,\tau$ is a graded or indexed monad [14]. A non-zero cost can be incurred only by an expression of the monadic type. Following standard convention we call such expressions impure, while expressions of all other types are called pure.

Expressions and values. There are term-level constructors for all types (in the kind *Type*) except for the modal type ($[p]\,\tau$). The inhabitants of type $[p]\,\tau$ are exactly those of type τ since the potential is ghost state without runtime manifestation.

We describe the expression and value forms for some of the types. The term-level constructors for the constraint type ($C \Rightarrow \tau$), type and index-level quantification ($\forall\alpha : \mathsf{K}.\tau$, $\forall i : \mathsf{S}.\tau$) are all denoted $\Lambda.e$. (Note that indices, types and constraints do not occur in terms.) We also have a fixed point operator (fix) which is used to encode recursion.

The monadic type $\mathbb{M}\,\kappa\,\tau$ has several term constructors, including the standard monadic unit ($\mathsf{ret}\, e$) and bind ($\mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2$). The construct $\mathsf{store}\, e$ stores potential with a term and is the introduction form of the type

$\boxed{\text{Forcing reduction, } e \Downarrow^{\kappa} v}$

$$\frac{e \Downarrow v}{\mathsf{ret}\, e \Downarrow^{0} v}\text{ E-return} \qquad \frac{e_1 \Downarrow v_1 \qquad v_1 \Downarrow^{\kappa_1} v_1' \qquad e_2[v_1'/x] \Downarrow v_2 \qquad v_2 \Downarrow^{\kappa_2} v_2'}{\mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2 \Downarrow^{\kappa_1+\kappa_2} v_2'}\text{ E-bind} \qquad \frac{}{\uparrow^{\kappa} \Downarrow^{\kappa} ()}\text{ E-tick}$$

$$\frac{e_1 \Downarrow v_1 \qquad e_2[v_1/x] \Downarrow v_2 \qquad v_2 \Downarrow^{\kappa} v_2'}{\mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2 \Downarrow^{\kappa} v_2'}\text{ E-release} \qquad \frac{e \Downarrow v}{\mathsf{store}\, e \Downarrow^{0} v}\text{ E-store}$$

Figure 2: Selected evaluation rules

$\boxed{\text{Typing judgment: } \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 + \Gamma_2 \vdash \langle\!\langle e_1, e_2 \rangle\!\rangle : (\tau_1 \otimes \tau_2)}\text{ T-tensorI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \otimes \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1, y : \tau_2 \vdash e' : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 + \Gamma_2 \vdash \mathsf{let}\langle\!\langle x, y \rangle\!\rangle = e \mathsf{\ in\ } e' : \tau}\text{ T-tensorE} \qquad \frac{\Psi; \Theta; \Delta; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash !e : !\tau}\text{ T-ExpI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : !\tau \qquad \Psi; \Theta; \Delta; \Omega, x : \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 + \Gamma_2 \vdash \mathsf{let}\, !\, x = e \mathsf{\ in\ } e' : \tau'}\text{ T-ExpE} \qquad \frac{\Psi; \Theta; \Delta; \Omega, x : \tau; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash \mathsf{fix} x.e : \tau}\text{ T-fix}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi; \Theta; \Delta \vdash \Gamma' \sqsubseteq \Gamma \qquad \Psi; \Theta; \Delta \vdash \Omega' \sqsubseteq \Omega \qquad \Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta; \Omega'; \Gamma' \vdash e : \tau'}\text{ T-weaken}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{ret}\, e : \mathbb{M}\, 0\, \tau}\text{ T-ret}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \mathbb{M}\, \kappa_1\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}\, \kappa_2\, \tau_2 \qquad \Theta \vdash \kappa_1 : \mathbb{R}^+ \qquad \Theta \vdash \kappa_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 + \Gamma_2 \vdash \mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2 : \mathbb{M}(\kappa_1 + \kappa_2)\, \tau_2}\text{ T-bind}$$

$$\frac{\Theta \vdash \kappa : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \uparrow^{\kappa} : \mathbb{M}\, \kappa\, \mathbf{1}}\text{ T-tick} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta \vdash p : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{store}\, e : \mathbb{M}\, p\, ([p]\, \tau)}\text{ T-store}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : [p_1]\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}(p_1 + p_2)\, \tau_2 \qquad \Theta \vdash p_1 : \mathbb{R}^+ \qquad \Theta \vdash p_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 + \Gamma_2 \vdash \mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2 : \mathbb{M}\, p_2\, \tau_2}\text{ T-release}$$

Figure 3: Selected typing rules for λ-amor$^-$

$[p]\,\tau$. Dually, $\mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2$ releases potential stored with e_1 and makes it available to offset the cost of e_2. Note that, $\mathsf{store}\, e$ and $\mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2$ are useful only for the type system: they indicate ghost operations, i.e., where potentials should be stored and released, respectively. Operationally, they are uninteresting: $\mathsf{store}\, e$ evaluates exactly like $\mathsf{ret}\, e$, while $\mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2$ evaluates exactly like $\mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2$. Finally, we have a construct for incurring non-zero cost – the "tick" construct denoted $\uparrow^{\kappa}$. This construct indicates that cost κ is incurred where it is placed. Programmers place the construct at appropriate points in a program to model costs incurred during execution, as in prior work [11].

Operational semantics. λ-amor$^-$ is a call-by-name calculus with eager evaluation.[4] We use two evaluation judgments – pure and forcing. The pure evaluation judgment ($e \Downarrow v$) relates a an expression e to the value v it evaluates to. All monadic expressions are treated as values in the pure evaluation. The rules for pure evaluation are standard so we defer them to the Appendix A. The forcing evaluation judgment $e \Downarrow^{\kappa} v$ is a relation between terms of type $\mathbb{M}\, \kappa\, \tau$ and values of type τ. κ is the cost incurred in executing (forcing) e. The rules of this judgment are shown in Fig. 2. E-return states that if e reduces to v in the pure reduction, then $\mathsf{ret}\, e$ forces to v with 0 cost. E-store is exactly like E-return, emphasizing the ghost nature of potential annotations in types. E-bind is the standard monadic composition of e_1 with e_2. The effect (cost) of bind is the sum of the costs of forcing e_1 and e_2. E-release is similar. $\uparrow^{\kappa}$ is the only cost-consuming construct in the language. E-tick says that $\uparrow^{\kappa}$ forces to () and it incurs cost κ.

2.2 Type system

The typing judgment of λ-amor$^-$ is written $\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau$. Here, Ψ is a context mapping type-level variables to their kinds, Θ is a context mapping index-level variables to their sorts, Δ is a context of constraints

[4]Perhaps somewhat surprisingly, even additive (&) pairs are evaluated eagerly. However, since all effects are confined to a monad, this choice does not matter. ! is lazy as in a standard affine λ-calculus.

$$\frac{\Psi;\Theta;\Delta \vdash \tau <: \tau' \qquad \Psi;\Theta;\Delta \vdash p' \leq p}{\Psi;\Theta;\Delta \vdash [p]\,\tau <: [p']\,\tau'}\ \text{sub-potential} \qquad \frac{\Psi;\Theta;\Delta \vdash \tau <: \tau' \qquad \Psi;\Theta;\Delta \vdash \kappa \leq \kappa'}{\Psi;\Theta;\Delta \vdash \mathbb{M}\,\kappa\,\tau <: \mathbb{M}\,\kappa'\,\tau'}\ \text{sub-monad}$$

$$\frac{\Theta \vdash p : \mathbb{R}^+ \qquad \Theta \vdash p' : \mathbb{R}^+}{\Psi;\Theta;\Delta \vdash [p](\tau_1 \multimap \tau_2) <: ([p']\,\tau_1 \multimap [p'+p]\,\tau_2)}\ \text{sub-potArrow} \qquad \frac{}{\Psi;\Theta;\Delta \vdash \tau <: [0]\,\tau}\ \text{sub-potZero}$$

$$\frac{\Psi;\Theta, i : \mathsf{S};\Delta \vdash \tau <: \tau'}{\Psi;\Theta;\Delta \vdash \lambda_t i : \mathsf{S}.\tau <: \lambda_t i : \mathsf{S}.\tau'}\ \text{sub-familyAbs} \qquad \frac{\Theta \vdash I : \mathsf{S}}{\Psi;\Theta;\Delta \vdash (\lambda_t i : \mathsf{S}.\tau)\ I <: \tau[I/i]}\ \text{sub-familyApp1}$$

$$\frac{\Theta \vdash I : \mathsf{S}}{\Psi;\Theta;\Delta \vdash \tau[I/i] <: (\lambda_t i : \mathsf{S}.\tau)\ I}\ \text{sub-familyApp2}$$

Figure 4: Selected subtyping rules

on the index variables, and Ω and Γ are the non-affine and affine typing contexts respectively, both mapping term-level variables to their types. We use the notation $\Gamma_1 + \Gamma_2$ to describe the disjoint union of the affine contexts Γ_1 and Γ_2. Selected typing rules are listed in Fig. 3, and the full set of rules can be found in the Appendix A.

Rules for the affine type constructs of λ-amor are standard. T-tensorI is the type rule for introducing the tensor pair $\langle\langle e_1, e_2\rangle\rangle$ – if e_1 and e_2 are typed τ_1 and τ_2 under affine contexts Γ_1 and Γ_2, respectively, then $\langle\langle e_1, e_2\rangle\rangle$ is typed $(\tau_1 \otimes \tau_2)$ under the context $(\Gamma_1 + \Gamma_2)$. Dually, T-tensorE is the type rule for eliminating the tensor pair – if expression e has type $(\tau_1 \otimes \tau_2)$ in the context Γ_1 and a continuation e' is of type τ' in the context Γ_2 plus both elements of the tensor pair (named x and y here), then the expression $\mathsf{let}\langle\langle x, y\rangle\rangle = e \mathsf{\ in\ } e'$ is of type τ' under the context $(\Gamma_1 + \Gamma_2)$. T-expI ascribes $!e$ the type $!\tau$ if e can be ascribed the type τ under an empty affine context. The subtyping relation ($<:$) used in the rule T-weaken is described below, but we skip describing the standard details of the auxiliary relation $\sqsubseteq$, which is described in the Appendix A. T-expE is the rule for the elimination form of $!\tau$. The important thing here is that the continuation e' has unbounded access to e via the non-affine variable x.

Rules for monadic types are interesting. T-ret types the return of the monad. In the operational semantics, $\mathsf{ret}\, e$ takes a well-typed expression and returns it with 0 cost. Hence, its type $\mathbb{M}\,0\,\tau$ also includes 0 cost. T-bind types the monadic bind, which is basically sequences two computations. The cost in the type of the bind is the sum of the costs of the two computations, again mirroring the operational semantics. T-tick type checks $\uparrow^\kappa$ at type $\mathbb{M}\,\kappa\,\mathbf{1}$ – a monad of unit type with cost κ.

T-store types the $\mathsf{store}\, e$ construct, which is used to associate potential with an expression. If e has type τ, the rule gives $\mathsf{store}\, e$ the type $\mathbb{M}\,\kappa\,([\kappa]\,\tau)$. The way to read this is that if p units of potential are attached to e, then the cost of doing so is p units. Finally, T-release is dual to T-store: It uses the potential p_1 stored with the first expression e_1 to reduce the cost of the continuation by the same amount.

Subtyping. Selected subtyping rules are shown in Fig. 4. As mentioned earlier, λ-amor$^-$ also has type-level functions and applications. Accordingly, We have subtyping rules to convert the type-level application form $((\lambda_t i : \mathsf{S}.\tau)\ I)$ to the substitution form $(\tau[I/i])$ and vice versa. Rule sub-potArrow distributes potential on a function type over the argument and the return value. sub-potZero allows silently casting an expression of type τ to type $[0]\,\tau$. This reinforces the ghost nature of potential. The subtyping of the modal type $[p]\,\tau$ is contra-variant in the potential because it is sound to throw away potential. The subtyping for the monadic type is covariant in both the type and the cost (because the cost in the monadic type is an upper bound). There are additional, standard typing rules for sorts and kinds, which we defer to the Appendix A.

Theorem 1 states the soundness of λ-amor$^-$. Intuitively, it says that, if e is a closed term which has a statically approximated cost of κ units (as specified in the monadic type $\mathbb{M}\,\kappa\,\tau$) and forcing it actually consumes κ' units of cost, then $\kappa' \leq \kappa$. We prove this theorem using a logical relation in Section 2.3.

Theorem 1 (Soundness)**.** $\forall e, v, \kappa, \kappa', \tau \in \mathit{Type}.\ \vdash e : \mathbb{M}\,\kappa\,\tau \wedge e \Downarrow^{\kappa'} v \implies \kappa' \leq \kappa$

2.3 Model of types

To prove the soundness of λ-amor$^-$, we develop a logical-relation model of its types. The model is an extension of Pym's semantics of BI [31] with potentials, the cost monad, and type refinements. We also step-index the model [1] to break a circularity in its definition, arising from impredicative quantification over types, as in the work of [29]. Because we use step-indices, we also have augmented operational semantics that count the number of rules (denoted T) used during evaluation. The revised judments are written $e \Downarrow_T v$ (pure) and $e \Downarrow_T^\kappa v$ (forcing). The expected details are in the Appendix A. Note that there is no connection between T and κ in the forcing judgment – the former is purely an artifact of our metatheoretic proofs, while the latter is induced by $\uparrow$ constructs in the program. Our use of step-indices, also written T, is standard and readers not familiar with them may simply ignore them. The model (Fig. 5) is defined using four relations: a value relation, an

$$
\begin{array}{lcl}
[\![\mathbf{1}]\!] & \triangleq & \{(p, T, ())\} \\
[\![\mathsf{b}]\!] & \triangleq & \{(p, T, v) \mid v \in [\![\mathsf{b}]\!]\} \\
[\![L^0\tau]\!] & \triangleq & \{(p, T, nil)\} \\
[\![L^{s+1}\tau]\!] & \triangleq & \{(p, T, v :: l) | \exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v) \in [\![\tau]\!] \wedge (p_2, T, l) \in [\![L^s\tau]\!]\} \\
[\![\tau_1 \otimes \tau_2]\!] & \triangleq & \{(p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \mid \exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v_1) \in [\![\tau_1]\!] \wedge (p_2, T, v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \ \&\ \tau_2]\!] & \triangleq & \{(p, T, \langle v_1, v_2\rangle) \mid (p, T, v_1) \in [\![\tau_1]\!] \wedge (p, T, v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \oplus \tau_2]\!] & \triangleq & \{(p, T, \mathsf{inl}(v)) \mid (p, T, v) \in [\![\tau_1]\!]\} \cup \{(p, T, \mathsf{inr}(v)) \mid (p, T, v) \in [\![\tau_2]\!]\} \\
[\![!\tau]\!] & \triangleq & \{(p, T, !e) \mid (0, T, e) \in [\![\tau]\!]_{\mathcal{E}}\} \\
[\![\tau_1 \multimap \tau_2]\!] & \triangleq & \{(p, T, \lambda x.e) \mid \forall p', e', T'{<}T\ .(p', T', e') \in [\![\tau_1]\!]_{\mathcal{E}} \implies (p + p', T', e[e'/x]) \in [\![\tau_2]\!]_{\mathcal{E}}\} \\
[\![[n]\,\tau]\!] & \triangleq & \{(p, T, v) \mid \exists p'.p' + n \leq p \wedge (p', T, v) \in [\![\tau]\!]\}\} \\
[\![\mathbb{M}\,\kappa\,\tau]\!] & \triangleq & \{(p, T, v) \mid \forall \kappa', v', T'{<}T\ .v \Downarrow_{T'}^{\kappa'} v' \implies \exists p'.\kappa' + p' \leq p + \kappa \wedge (p', T - T', v') \in [\![\tau]\!]\} \\
[\![\forall\alpha.\tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid \forall \tau', T'{<}T\ .(p, T', e) \in [\![\tau[\tau'/\alpha]]\!]_{\mathcal{E}}\} \\
[\![\forall i.\tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid \forall I, T'{<}T\ .(p, T', e) \in [\![\tau[I/i]]\!]_{\mathcal{E}}\} \\
[\![C \Rightarrow \tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid .\models C \implies (p, T, e) \in [\![\tau]\!]_{\mathcal{E}}\} \\
[\![C \& \tau]\!] & \triangleq & \{(p, T, v) \mid .\models C \wedge (p, T, v) \in [\![\tau]\!]\} \\
[\![\exists s.\tau]\!] & \triangleq & \{(p, T, v) \mid \exists s'.(p, T, v) \in [\![\tau[s'/s]]\!]\} \\
[\![\lambda_t i.\tau]\!] & \triangleq & f \text{ where } \forall I.\ f\ I = [\![\tau[I/i]]\!] \\
[\![\tau\ I]\!] & \triangleq & [\![\tau]\!]\ I \\
\\
[\![\tau]\!]_{\mathcal{E}} & \triangleq & \{(p, T, e) \mid \forall\ T'{<}T, v.e \Downarrow_{T'} v \implies (p, T - T', v) \in [\![\tau]\!]\} \\
\\
[\![\Gamma]\!]_{\mathcal{E}} & \triangleq & \{(p, T, \gamma) \mid \exists f : \mathcal{V}ars \to \mathcal{P}ots. \\
 & & (\forall x \in dom(\Gamma).\ (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x \in dom(\Gamma)} f(x) \leq p)\} \\
[\![\Omega]\!]_{\mathcal{E}} & \triangleq & \{(0, T, \delta) \mid (\forall x \in dom(\Omega).(0, T, \delta(x)) \in [\![\tau]\!]_{\mathcal{E}})\}
\end{array}
$$

Figure 5: Model of λ-amor$^-$ types

expression relation and substitution relations for the affine and non-affine context. The first two are mutually recursive on the lexicographic order $\langle$step index (T), type (τ), value $<$ expression$\rangle$.

Value relation. The value relation (denoted by $[\![.]\!]$) gives an interpretation to λ-amor$^-$ types (of kind $Type$) as sets of triples of the form (p, T, v). Importantly, the potential p is an upper-bound on the ambient potential *required* to construct the value v. It must include potential associated with the (types of) subexpressions of v.

We describe interesting cases of this relation. The interpretation for the list type is defined by a further induction on list size. For a list of size 0 the value relation contains a *nil* value with any potential (since *nil* captures no potential). For a list of size $s+1$, the value relation is defined inductively on s, similar to the tensor pair, which we describe next. For a tensor $(\otimes)$ pair, both components can be used. Therefore, the potential required to construct a tensor pair is at least the sum of the potentials needed to construct the two components. On the other hand, for a with (&) pair, either but not both of the components can be used by the context. So the potential needed for a & pair should be sufficient for each component separately. The type $!\tau$ contains $!e$ when e is in τ. The important aspect here is that the potential associated with e must be 0, otherwise we would have immediate unsoundness due to replication of potential, as described in Section 1.

Next, we explain the interpretation of the arrow type $\tau_1 \multimap \tau_2$: $\lambda x.e$ is in this type with potential p if for any substitution e' (of type τ_1) that comes with potential p', the total potential $p + p'$ is sufficient for the body $e[e'/x]$ (of type τ_2).

The step indices T play an important role only in the interpretation of the polymorphic type $\forall\alpha.\tau$. Since the type-level parameter α may be substituted by any type, potentially one even larger than τ, the relation would not be well-founded by induction on types alone. Here, we rely on step-indices, noting that substituting α with a type consumes at least one step in our operational semantics, so the relation for τ (with the substitution) needs to be defined only at a smaller step index. This follows prior work [29].

Next, we come to the new, interesting types for potential and the cost monad. The potential type $[n]\,\tau$ contains v with required potential p if p is sufficient to account for n *and* the potential required for v. (Note that the same value v is in the interpretation of both τ and $[n]\,\tau$.) The graded monadic type $\mathbb{M}\,\kappa\,\tau$ contains the (impure) value v with required potential p if p and κ *together* suffice to cover the cost κ' of actually forcing v and the potential p' required for the resulting value, i.e., if $p + \kappa \geq \kappa' + p'$. The ambient potential p and the cost κ on the monad appear together in a sum, which explains why the typing rule T-release can *offset* cost on the monad using potential.

The interpretation of a type family $\lambda_t i.\tau$ is a type-level function, as expected. The interpretation of type-level application is an application of such a function. The remaining cases of the value relation of Fig. 5 should be self-explanatory.

Expression relation. The expression relation, denoted $[\![.]\!]_{\mathcal{E}}$, maps a type to a set of triples of the form (p, T, e). Its definition is fairly simple and standard: we simply check if the value v obtained by pure evaluation of e is in the value relation of the same type. The potential does not change during pure evaluation, but we need to adjust the step index correctly.

Substitution relations. Finally, we describe the substitution relations for both the affine context (Γ) and the non-affine context (Ω). Each relation maps the context to a set of valid substitutions for the context, paired with a step index and a potential. The two key points about the interpretation of Γ are: 1) The substitution γ should map each variable to a value of the correct type (semantically), and 2) The required potential p for the context should be more than the sum of the required potentials for the substitutions of each of the variables. The interpretation of the non-affine context Ω is much simpler. It only demands that the substituted value is in the interpretation of the correct type with 0 potential.

Soundness. As is standard for logical-relations models, the main meta-theoretic property is the "fundamental theorem" (Theorem 2). The theorem basically says that if e is well-typed in some contexts at type τ, then the application of any subtitutions in the semantic interpretation of the contexts map e into the semantic interpretation of τ. The important, interesting aspect of the theorem in λ-amor$^-$ is that the potential needed for e (after substitution) equals the potential coming from the context, p_l. This the crux of the soundness of (amortized) cost analysis in λ-amor$^-$.

Theorem 2 (Fundamental theorem for λ-amor$^-$)**.** $\forall \Theta, \Omega, \Gamma, e, \tau, T, p_l, \gamma, \delta, \sigma, \iota.$
$\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \wedge (p_l, T, \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}} \wedge (0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}} \implies$
$(p_l, T, e\ \gamma\delta) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}.$

Here, ι, σ, δ and γ denote substitutions for the index context Θ, the type context Ψ, the non-affine context Ω and the affine context Γ, respectively. This theorem is proved by induction on the given typing judgment with a subinduction on the step-index for the case of fix. The supplementary material has the entire proof.

Theorem 1 is a direct corollary of this fundamental theorem. We can derive several additional corollaries about execution cost directly from this fundamental theorem. For instance, for open terms which only partially use the input potential and save the rest with the result, we can derive Corollary 3. Here, e is a thunk that expects as input a unit argument, but with some associated potential q. When applied, e returns a computation (of 0 cost) that forces to a value with a residual potential q'. The corollary says that if the context Γ provides a potential p_l, then forcing e (with the substitution γ) incurs a cost J and produces a value v that requires potential p_v such that $J \leq (q + p_l) - (q' + p_v)$. This expression may look complex, but it is simply a difference of the incoming potentials of e (q and p_l) and the outgoing potentials of e (q' and p_v). In Section 4, we show an interesting use of this corollary for deriving an alternate proof of soundness of univariate RAML via its embedding in λ-amor$^-$.

Corollary 3. $\forall \Gamma, e, q, q', \tau, p_l, \gamma, J, v_t, v.$
$.; .; .; .; \Gamma \vdash e : [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\,\tau) \wedge (p_l, \_, \gamma) \in [\![\Gamma]\!]_{\mathcal{E}} \wedge e\ ()\ \gamma \Downarrow v_t \Downarrow^J v \implies$
$\exists p_v.\ (p_v, \_, v) \in [\![\tau]\!] \wedge J \leq (q + p_l) - (q' + p_v)$

3 Examples

Next, we show three nontrivial examples of cost analysis in λ-amor$^-$. Complete type derivations for the examples can be found in the supplementary material. That material also has two additional, simple examples (the standard list map and append functions) that we omit here.

3.1 Church encoding

Our first example types Church numerals and operations on them. Typing these constructions require nontrivial use of type and index families. The type we give to Church numerals is both general and expressive enough to encode and give precise cost to operations like addition, multiplication and exponentiation.

For exposition, we first consider the types of Church numerals without any cost. To recap, Church numerals encode natural numbers as function applications. For example, a Church zero is defined as $\lambda f.\lambda x.x$ (with zero applications), a Church one as $\lambda f.\lambda x.f\ x$ (with one application), a Church two as $\lambda f.\lambda x.f\ (f\ x)$ (with two applications) and so on. To type a Church numeral, we must specify a type for f. We assume that we have an $\mathbb{N}$-indexed family of types α, and that f maps $\alpha\ i$ to $\alpha\ (i+1)$ for every i. Then, the nth Church numeral, given such a function f, maps $\alpha\ 0$ to $\alpha\ n$.

Next, we consider costs. Just for illustration, we are interested in counting a unit cost for every *function application.* We want to encode the precise costs of operations like addition, multiplication and exponentiation in types. Classically, these operations are defined compositionally. For example, addition of m and n is defined by applying m to the successor function (for f) and n. This iterates the successor function m times over n. To type this, the type of f in the Church nats must be general enough. For this, we use a cost family C from $\mathbb{N}$ to $\mathbb{R}$. The cost of applying f depends on the index of the argument (called j_n below). Then, given such a f, the nth Church numeral maps $\alpha\ 0$ to $\alpha\ n$ with cost $C\ 0 + \ldots + C\ (n-1) + n$, where each $C\ i$ is the cost of using f

the ith time, and the last n is the cost of the n applications in the definition of the nth Church numeral. Our type for Church numerals, called Nat below, captures exactly this intuition.

$$\begin{aligned}&\mathsf{Nat} = \lambda_t n.\forall\alpha : \mathbb{N} \to \mathit{Type}.\forall C : \mathbb{N} \to \mathbb{R}^+.\\ &\qquad !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap\\ &\qquad \mathbb{M}\,0\,((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n-1) + n)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ n))\end{aligned}$$

Below, we describe a term for the Church one, denoted $\overline{1}$, that has type $\mathsf{Nat}\ 1$. For notational simplification, we define $e_1 \uparrow^1 e_2 \triangleq (\mathsf{bind} - = \uparrow^1 \mathsf{in}\ \mathsf{ret}(e_1\ e_2))$, which applies e_1 to e_2 and additionally incurs a cost of 1 unit.

$$\begin{aligned}&\overline{1} : \mathsf{Nat}\ 1\\ &\overline{1} \triangleq \Lambda.\Lambda.\lambda f.\,\mathsf{ret}\ (\lambda x.\ \mathsf{let}\,!f_u = f\ \mathsf{in}\ \mathsf{let}\,\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ \mathsf{bind}\,a = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\langle y_1, a\rangle\rangle)\end{aligned}$$

The term $\overline{1}$ takes the input pair x of type $\alpha\ 0 \otimes [((C\ 0) + 1)]\,\mathbf{1}$ and binds its two components to y_1 and y_2. It then releases the potential $((C\ 0) + 1)$ in y_2, stores $C\ 0$ of the released potential in a, and applies the input function f_u to $\langle\langle y_1, a\rangle\rangle$, incurring a cost of 1 unit. This incurred cost models the cost of the application (which we want to count). It is offset by the remaining 1 potential that was released from y_2.

Next, we show the encoding for Church addition. Church addition is defined using a successor function ($succ$), which is also defined and type-checked in λ-amor$^-$, but whose details we elide here. It is just enough to know that the cost of $succ$ under our cost model is two units.

$$succ : \forall n.\,[2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n] \multimap \mathbb{M}\,0\,\mathsf{Nat}[n+1])$$

An encoding of Church addition (add) in λ-amor$^-$ is shown below. The type of add takes the required potential (which is $4 * n_1 + 2$ here) along with two Church naturals ($\mathsf{Nat}\ n_1$ and $\mathsf{Nat}\ n_2$) as arguments and computes their sum. The potential of $(4*n_1+2)$ units corresponds to the precise cost of performing the Church addition under our cost model. The whole type is parameterized on n_1 and n_2. Ignoring the decorations, add simply applies $\overline{N_1}$ to $succ$ and $\overline{N_2}$, as expected.

$$\begin{aligned}&add : \forall n_1, n_2.\,[(4*n_1+2)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))))\\ &add \triangleq\ \Lambda.\Lambda.\lambda p.\,\mathsf{ret}(\lambda\overline{N_1}.\,\mathsf{ret}(\lambda\overline{N_2}.\ \ \mathsf{release} - = p\ \mathsf{in}\ \ \mathsf{bind}\,a = E_1\ \mathsf{in}\ E_2))\\ &E_1 \triangleq \overline{N_1}\ \ []\ \ []\ \ \uparrow^1 !(\Lambda.\lambda t.\,\mathsf{let}\,\langle\langle y_1, y_2\rangle\rangle = t\ \mathsf{in}\ \ \mathsf{release} - = y_2\ \mathsf{in}\\ &\qquad\qquad \mathsf{bind}\,b_1 = (\mathsf{bind}\,b_2 = \mathsf{store}()\ \mathsf{in}\ (succ\ []\ b_2))\ \mathsf{in}\ \ b_1 \uparrow^1 y_1)\\ &E_2 \triangleq \mathsf{bind}\,b = \mathsf{store}()\ \mathsf{in}\ a\ \uparrow^1 \langle\langle\overline{N_2}, b\rangle\rangle\end{aligned}$$

Listing 1: Encoding of the Church addition in λ-amor$^-$

We have similarly encoded Church multiplication and exponentiation in λ-amor$^-$. We are unaware of such a general encoding of Church numeral in a pure monadic cost framework without potentials.

3.2 Eager functional queue

Eager functional FIFO queues are often implemented using two LIFO stacks represented as standard functional lists, say l_1 and l_2. Enqueue is implemented as a push (cons) on l_1. Dequeue is implemented as a pop (head) from l_2 if it is non-empty. However, if l_2 is empty, then the contents of l_1 are transferred, one at a time, to l_2 and the new l_2 is popped. The transfer from l_1 to l_2 reverses l_1, thus changing the stack's LIFO semantics to a queue's FIFO semantics. We describe the analysis of this eager queue in λ-amor$^-$ queue. Here, we incur a unit cost for every list cons operation.

Note that the worst-case cost of dequeue is linear in the size of l_1. However, the *amortized* cost of dequeue is actually constant. This analysis works by accounting the cost of transferring an element from l_1 to l_2 right when the time is enqueued in l_1. This works because an enqueued element can be transferred at most once. Concretely, the enqueue operation has a cost (it requires a potential) of 3 units, of which 1 is used by the enqueue operation itself and the remaining 2 are stored with the element in the list l_1 to be used later in the dequeue operation if required. This is reflected in the type of enqueue. The term for enqueue is straightforward, so we skip it here.

$$enq : \forall m, n.\,[3]\,\mathbf{1} \multimap \tau \multimap L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau) \otimes L^m\tau)$$

Observe how each element of the first list l_1 in both the input and the output has a potential 2 associated with it. The dequeue operation (denoted by dq below) is a bit more involved. The constraints in the type of dequeue reflect a) dequeue can only be performed on a non-empty queue, i.e., if $m+n > 0$ and b) the sum of the lengths of the resulting list is only 1 less than the length of the input lists, i.e., $\exists m', n'.((m'+n'+1) = (m+n))$. The full type and the term for the dequeue operation are described in Listing 2. Dequeue uses a function $move$, which moves elements from the first list to the second. We skip the description of $move$. Type-checked terms for enq, dq and $move$ are in the the Appendix C.

$dq : \forall m, n.(m + n > 0) \Rightarrow L^m([2]\,\tau) \multimap L^n\tau \multimap \mathbb{M}\,0\,(\exists m', n'.((m' + n' + 1) = (m + n))\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$
$dq \triangleq\ \Lambda.\Lambda.\Lambda.\lambda\ l_1\ l_2.\ \mathsf{match}\ l_2\ \mathsf{with}$
$\quad |nil \mapsto \mathsf{bind}\, l_r = move\ [][]\ l_1\ nil\ \mathsf{in}$
$\qquad \mathsf{match}\ l_r\ \mathsf{with}$
$\qquad\quad |nil \mapsto \mathsf{fix} x.x$
$\qquad\quad |h_r :: l'_r \mapsto \mathsf{ret}\,\Lambda.\langle\langle nil, l'_r\rangle\rangle$
$\quad |h_2 :: l'_2 \mapsto \mathsf{ret}\,\Lambda.\langle\langle l_1, l'_2\rangle\rangle$

Listing 2: Dequeue operation for eager functional queue in λ-amor$^-$

3.3 Okasaki's implicit queue

Next we describe an encoding of a lazy data structure, namely, Okasaki's implicit queue [30]. This is a FIFO functional queue, which supports log-time random access. Okasaki shows an analysis of this queue for memoizing (lazy) evaluation using his method of debits. [11] shows how to formalize this analysis in Agda. Here, we replicate Danielsson's analysis in λ-amor$^-$. Although the cost bounds we obtain are the same as Danielsson's, we note that our bound does *not* apply to Okasaki's/Danielsson's lazy evaluation scheme. Rather, it applies to our monadic sequential execution, where we specifically encode operations to incur a unit cost at every case analysis on a queue. It turns out that Okasaki's potential invariants work as-is for this cost model as well.

An implicit queue is either a shallow (trivial) queue of zero or one elements, or a deep (nontrivial) queue consisting of three parts called front, middle and rear. The front has one or two elements, the middle is a recursive, suspended implicit queue of *pairs* of elements, and the rear has zero or one elements. Because the recursive structure (the middle) is a queue of pairs, random lookup on the whole queue is logarithmic in time.

Overall, the queue can be described as a datatype of 6 constructors, corresponding to the two shallow cases and the four deep cases. Hereon, we assume a polymorphic queue type $Queue(\tau)$ with these 6 constructors, called *C0*–*C5* has been added as a primitive to the language. The types of these constructors are shown below. Note that each type encodes the potential needed to apply the constructor (e.g., a potential of 2 units is needed to apply *C4*). These potentials match those of Okasaki.

$C0 : Queue\ \tau$
$C2 : [1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\tau \otimes Queue(\tau \otimes \tau)) \multimap Queue\ \tau$
$C4 : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,((\tau \otimes \tau) \otimes Queue(\tau \otimes \tau)) \multimap Queue\ \tau$
$C1 : \tau \multimap Queue\ \tau$
$C3 : [0]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\tau \otimes Queue(\tau \otimes \tau) \otimes \tau) \multimap Queue\ \tau$
$C5 : [1]\,\mathbf{1} \multimap \mathbb{M}\,0\,((\tau \otimes \tau) \otimes Queue(\tau \otimes \tau) \otimes \tau) \multimap Queue\ \tau$

Figure 6: Constructors of Okasaki's implicit queue

We have implemented and analyzed the expected snoc, head and tail functions for this queue in λ-amor$^-$. While we defer the details of these functions to the supplementary material, in Listing 3 we show the helper function $headTail$, which extracts both the head and the tail of a queue and is the crux of our implementation.[5] This function has an amortized cost of 3 units as indicated by its type. At the top-level, the function releases the 3 units of potential and immediately uses 1 unit to case analyze the form of the queue. The remaining 2 units are used – partially or fully – in the various cases. For example, when the input queue is $C1\ x$, we return a pair of x and the empty queue ($C0$). The 2 units of potential are simply discarded in this case.

The cases *C2*–*C5* are interesting. We describe here only the case *C3*. From Fig. 6 we know that the suspension in *C3* needs 0 units of potential to be forced. So we store 0 units of potential in p' and the remaining 2 units of potential in p_o, which is used later. We then force the suspension (named x) to obtain the front (f), middle (m) and rear (r). The front f is just returned as the head while the tail is constructed using the constructor *C5*. Inside the suspension of *C5* we have 1 unit of additional potential available to us via p''. We use this 1 unit of potential from p'' along with the 2 units of potential from p_o, which we constructed earlier, to obtain a total of 3 units of potential, which allows us to make a recursive call to $headTail$ on the middle. This gives us the head and tail of the middle. The rest of this case is straightforward.

Case *C2* is similar, while cases *C4* and *C5* do not involve recursive calls.

$headTail : [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,(\alpha \otimes Queue\ \ \alpha)$
$headTail \triangleq \mathsf{fix}\ \ HT.\lambda p.\Lambda.\lambda\ \ q.$
$- = \mathsf{release}\, p\ \mathsf{in}\ \ - = \uparrow^1\ \mathsf{in\ ret}$
$\ \mathsf{case}\ \ q\ \ \mathsf{of}$
$\quad |C0 \mapsto \mathsf{fix} x.x$
$\quad |C1\ \ x \mapsto \mathsf{ret}\langle\langle x, C0\rangle\rangle$
$\quad |C2\ \ x \mapsto$
$\qquad \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\, p_o = \mathsf{store}()\ \mathsf{in}$
$\qquad\quad \mathsf{bind}\, x' = x\ \ p'\ \mathsf{in\ let}\langle\langle f, m\rangle\rangle = x'\ \mathsf{in}$
$\qquad\qquad \mathsf{ret}\langle\langle f, (C4\ \ (\lambda p''.- = \mathsf{release}\, p_o\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in\ bind}\, p_r = \mathsf{store}()\ \mathsf{in}\ HT\ \ p_r\ \ []\ \ m))\rangle\rangle$
$\quad |C3\ \ x \mapsto$

[5] Okasaki does not need a corresponding function since he works in a non-affine setting. In some of his implementation he uses the same queue twice, once to extract its head and once to extract its tail. In our setting, a queue is affine so it cannot be used twice. Hence, we define this combined function.

$$
\begin{array}{l}
\quad \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\, p_o = \mathsf{store}()\ \mathsf{in} \\
\quad\quad \mathsf{bind}\, x' = x\ \ p'\ \mathsf{in}\ \ \mathsf{let}\langle\langle fm, r\rangle\rangle = x'\ \mathsf{in}\ \ \mathsf{let}\langle\langle f, m\rangle\rangle = fm\ \mathsf{in} \\
\quad\quad \mathsf{ret}\langle\langle f, (\mathit{C5}\ \ (\lambda p''.- = \mathsf{release}\, p_o\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\, ht = HT\ \ p'''\ \ []\ \ m\ \mathsf{in}\ \mathsf{ret}\langle\langle ht, r\rangle\rangle))\rangle\rangle \\
|\mathit{C4}\ \ x \mapsto \\
\quad \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, x' = x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\langle f, m\rangle\rangle = x'\ \mathsf{in}\ \mathsf{let}\langle\langle f_1, f_2\rangle\rangle = f\ \mathsf{in} \\
\quad \mathsf{ret}\langle\langle f_1, \mathit{C2}\ \ (\lambda p''.\, \mathsf{ret}\langle\langle f_2, m\rangle\rangle)\rangle\rangle \\
|\mathit{C5}\ \ x \mapsto \\
\quad \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, x' = x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\langle fm, r\rangle\rangle = x'\ \mathsf{in}\ \mathsf{let}\langle\langle f, m\rangle\rangle = fm\ \mathsf{in}\ \mathsf{let}\langle\langle f_1, f_2\rangle\rangle = f\ \mathsf{in} \\
\quad\quad \mathsf{ret}\langle\langle f_1, (\mathit{C3}\ \ (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle))\rangle\rangle
\end{array}
$$

Listing 3: Function to obtain head and tail

The Appendix C contains full typing derivations for the $headTail$, $head$, $tail$ and $snoc$ operations.

4 Embedding Univariate RAML

In this section we describe an embedding of Resource Aware ML (RAML) [19, 16] into λ-amor$^-$. RAML is an *effect-based* type system for amortized analysis of OCaml programs using the method of potentials [32, 7]. The main motivation for this embedding is to show that: 1) λ-amor$^-$ can also perform *effect-based* cost analysis like RAML and thus can be used to analyze all examples that have been tried on RAML and 2) λ-amor$^-$, despite being a *call-by-name* framework, can embed RAML which is a *call-by-value* framework.

We describe an embedding of Univariate RAML [19, 16] (which subsumes Linear RAML [20]) into λ-amor$^-$. We leave embedding multivariate RAML [17] to future work but anticipate no fundamental difficulties in doing so.

4.1 A brief primer on Univariate RAML

We give a brief primer of Univariate RAML [19, 16] here. The key feature of Univariate RAML is an ability to encode univariate polynomials in the size of the input data as potential functions. Such functions are expressed as non-negative linear combinations of binomial coefficients $\binom{n}{k}$, where n is the size of the input data structure and k is some natural number. Vector annotations on the list type $L^{\vec{q}}\tau$, for instance, are used as a representation of such univariate polynomials. The underlying potential on a list of size n and type $L^{\vec{q}}\tau$ can then be described as $\phi(\vec{q}, n) \triangleq \sum_{1 \leq i \leq k} \binom{n}{i} q_i$ where $\vec{q} = \{q_1 \dots q_k\}$. The authors of RAML show using the properties of binomial coefficients, that such a representation is amenable to an inductive characterization of polynomials which plays a crucial role in setting up the typing rules of their system. If $\vec{q} = \{q_1 \dots q_k\}$ is the potential vector associated with a list then $\lhd(\vec{q}) = \{q_1 + q_2, q_2 + q_3, \dots q_{k-1} + q_k, q_k\}$ is the potential vector associated with the tail of that list. Trees follow a treatment similar to lists. Base types (unit, bools, ints) have zero potential and the potential of a pair is just the sum of the potentials of the components. A snippet of the definition of the potential function $\Phi(a : A)$ (from [16]) is described below.

$$
\begin{array}{lcl|lcl}
\Phi(a : A) & = & 0 \quad \text{where } A = \{\mathit{unit}, \mathit{int}, \mathit{bool}\} & \Phi([] : L^{\vec{q}}A) & = & 0 \\
\Phi((a_1, a_2) : (A_1, A_2)) & = & \Phi(a_1 : A_1) + \Phi(a_2 : A_2) & \Phi((a :: \ell) : L^{\vec{q}}A) & = & q_1 + \Phi(a : A) + \Phi(\ell : L^{\lhd\vec{q}}A) \\
 & & & \text{where } \vec{q} = \{q_1 \dots q_k\} & &
\end{array}
$$

A type system is built around this basic idea with a typing judgment of the form $\Sigma; \Gamma \vdash^q_{q'} e_r : \tau$ where Γ is a typing context mapping free variables to their types, Σ is a context for function signatures mapping a function name to a type (this is separate from the typing context because RAML only has first-order functions that are declared at the top-level), q and q' denote the statically approximated available and remaining potential before and after the execution of e_r, respectively, and τ is the zero-order type of e_r. Vector annotations are specified on list and tree types (as mentioned above). Types of first-order functions follow an intuition similar to the typing judgment above. $\tau_1 \xrightarrow{q/q'} \tau_2$ denotes the type of a first-order RAML function which takes an argument of type τ_1 and returns a value of type τ_2. q units of potential are needed before this function can be applied and q' units of potential are left after this function has been applied. Intuitively, the cost of the function is upper-bounded by (q+potential of the input) - (q'+potential of the result). Fig. 7 describe typing rules for function application and list cons. The *app* rule type-checks the function application with an input and remaining potential of $(q + K_1^{app})$ and $(q' - K_2^{app})$[6] units, respectively. RAML divides the cost of application between K_1^{app} and K_2^{app} units. Of the available $q + K_1^{app}$ units, q units are required by the function itself and K_1^{app} units are consumed before the application is performed. Likewise, of the remaining $q' - K_2^{app}$ units, q' units are made available from the function and K_2^{app} units are consumed after the application is performed. The *cons* rule requires an input potential of $q + p_1 + K^{cons}$ units of which p_1 units are added to the potential of the resulting list and K^{cons} units are consumed as the cost of performing this operation.

[6]Every time a subtraction like $(I - J)$ appears, RAML implicitly assumes that there is a side condition $(I - J) \geq 0$.

$$\frac{\tau_1 \overset{q/q'}{\to} \tau_2 \in \Sigma(f)}{\Sigma; x:\tau_1 \vdash^{q+K_1^{app}}_{q'-K_2^{app}} f\ x:\tau_2}\ \text{app} \qquad \frac{\vec{p}=(p_1,\ldots,p_k)}{\Sigma; x_h:\tau, x_t: L^{(\lhd\ \vec{p})}\tau \vdash^{q+p_1+K^{cons}}_{q} cons(x_h,x_t): L^{\vec{p}}\tau}\ \text{cons}$$

Figure 7: Selected type rules of Univariate RAML from [16]

Soundness of the type system is defined by Theorem 4. Soundness is defined for *top-level* RAML programs (formalized later in Definition 6), which basically consist of first-order function definitions (denoted by F) and the "main" expression e, which uses those functions. Stack (denoted by V) and heap (denoted by H) are used to provide bindings for free variables and locations in e.

Theorem 4 (Univariate RAML's soundness). $\forall H, H', V, \Gamma, \Sigma, e, \tau, {}^s v, p, p', q, q', t.$
$P = F, e$ is a RAML top-level program and
$H \models V : \Gamma \wedge \Sigma, \Gamma \vdash^q_{q'} e : \tau \wedge V, H \vdash^p_{p'} e \Downarrow_t {}^s v, H' \implies p - p' \leq (\Phi_{H,V}(\Gamma) + q) - (q' + \Phi_H({}^s v : \tau))$

4.2 Type-directed translation of Univariate RAML into λ-amor$^-$

As mentioned above, types in Univariate RAML include types for unit, booleans, integers, lists, trees, pairs and first-order functions. Without loss of generality we introduce two simplifications: a) we abstract RAML's bool and int types into an arbitrary base type denoted by b and b) we just choose to work with the list type only ignoring trees. These simplifications only make the development more concise as we do not have to deal with the redundancy of treating similar types again and again.

The translation from Univariate RAML to λ-amor$^-$ is type-directed. We describe the type translation function (denoted by $(\!|.|\!)$) from RAML types to λ-amor$^-$ types in Fig. 8.

$$\begin{array}{lcl} (\!|unit|\!) & = & \mathbf{1} \\ (\!|\mathsf{b}|\!) & = & !\mathsf{b} \\ (\!|L^{\vec{q}}\ \tau|\!) & = & \exists s.([\phi(\vec{q},s)]\,\mathbf{1} \otimes L^s(\!|\tau|\!)) \end{array} \quad \Bigg| \quad \begin{array}{lcl} (\!|(\tau_1,\tau_2)|\!) & = & ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \\ (\!|\tau_1 \overset{q/q'}{\to} \tau_2|\!) & = & [q]\,\mathbf{1} \multimap (\!|\tau_1|\!) \multimap \mathbb{M}\,0\,([q']\,(\!|\tau_2|\!)) \end{array}$$

Figure 8: Type translation of Univariate RAML

Since RAML allows for full replication of unit and base types, we translate RAML's base type, b, into $!\mathsf{b}$ of λ-amor$^-$. But translation of the unit type does not need a !, as $\mathbf{1}$ and $!\mathbf{1}$ are isomorphic in λ-amor$^-$. Unlike the unit and base type of RAML, the list type does have some potential associated with it, indicated by $\vec{q}$. Therefore, we translate RAML's list type into a pair type composed of a modal unit type carrying the required potential and a λ-amor$^-$ list type. Since the list type in λ-amor$^-$ is refined with size, we add an existential on the pair to quantify the size of the list. The potential captured by the unit type must equal the potential associated with the RAML list (this is indicated by the function $\phi(\vec{q}, s)$). The function $\phi(\vec{q}, s)$ corresponds to the one that RAML uses to compute the total potential associated with a list of s elements, which we described above. Note the difference in how potentials are managed in RAML vs how they are managed in the translation. In RAML, the potential for an element gets added to the potential of the tail with every cons operation and, dually only the, potential of the head element is consumed in the match operation. The translation, however, does not assign potential on a per-element basis, instead the aggregate potential is captured using the ϕ function and the translations of the cons and the match expressions work by adding or removing potential from this aggregate. We believe a translation which works with per element potential is also feasible but we would need an additional index to identify the elements of the list in the list data type.

We translate a RAML pair type into a tensor ($\otimes$) pair. This is in line with how pairs are treated in RAML (both elements of the pair are available on elimination). Finally, a function type $\tau_1 \overset{q/q'}{\to} \tau_2$ in RAML is translated into the function type $[q]\,\mathbf{1} \multimap (\!|\tau_1|\!) \multimap \mathbb{M}\,0\,([q']\,(\!|\tau_2|\!))$. As in RAML, the translated function type also requires a potential of q units for application and a potential of q' units remains after the application. The monadic type is required because we cannot release/store potential without going into the monad. The translation of typing contexts is defined pointwise using the type translation function.

We use this type translation function to produce a translation for Univariate RAML expressions by induction on RAML's typing judgment. The translation judgment is $\Sigma; \Gamma \vdash^q_{q'} e_r : \tau \quad \rightsquigarrow \quad e_a$. It basically means that a well-typed RAML expression e_r is translated into a λ-amor$^-$ expression e_a. The translated expression is of the type $[q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q'](\!|\tau|\!))$. We only describe the app rule here (Fig. 9). Since we know that the desired term must have the type $[q + K_1^{app}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q' - K_2^{app}](\!|\tau|\!))$, the translated term is a function which takes an argument, u, of the desired modal type and releases the potential to make it available for consumption. The continuation then consumes K_1^{app} potential that leaves $q - K_1^{app}$ potential remaining for $\mathsf{bind}\ P = \mathsf{store}()\ \mathsf{in}\ E_1$. We then store q units of potential with the unit and use it to perform a function application. We get a result of type $\mathbb{M}\,0\,([q']\,(\!|\tau_2|\!))$. We release these q' units of potential and consume K_2^{app} units from it. This leaves us with a remaining potential of $q' - K_2^{app}$ units. We store this remaining potential with f_2 and box it up in a monad to get the desired type. Translations of other RAML terms (which we do not describe here) follow a similar approach. The entire translation is intuitive and relies extensively on the ghost operations store and release at appropriate places.

$$\frac{\tau_1 \overset{q/q'}{\to} \tau_2 \in \Sigma(f)}{\Sigma; x:\tau_1 \vdash^{q+K_1^{app}}_{q'-K_2^{app}} f\ x : \tau_2 \leadsto \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K_1^{app}} \text{ in } \mathsf{bind}\ P = \mathsf{store}() \text{ in } E_1} \text{ app}$$

$$E_1 = \mathsf{bind}\ f_1 = (f\ P\ x) \text{ in } \mathsf{release}\ f_2 = f_1 \text{ in } \mathsf{bind} - = \uparrow^{K_2^{app}} \text{ in } \mathsf{bind}\ f_3 = \mathsf{store}\ f_2 \text{ in } \mathsf{ret}\ f_3$$

Figure 9: Expression translation for the app case: Univariate RAML to λ-amor$^-$

We show that the translation is type-preserving by proving that the obtained λ-amor$^-$ terms are well-typed (Theorem 5). The proof of this theorem works by induction on RAML's type derivation.

Theorem 5 (Type preservation: Univariate RAML to λ-amor$^-$)**.** *If $\Sigma;\Gamma \vdash^q_{q'} e : \tau$ in Univariate RAML then there exists e' such that $\Sigma;\Gamma \vdash^q_{q'} e : \tau \leadsto e'$ and there is a derivation of $.;.;.;(\!|\Sigma|\!),(\!|\Gamma|\!) \vdash e' : [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q'](\!|\tau|\!))$ in λ-amor$^-$.*

As mentioned earlier, RAML only has first-order functions which are defined at the top-level. So, we need to lift this translation to the top-level. Definition 6 defines the top-level RAML program along with the translation.

Definition 6 (Top level RAML program translation)**.** *Given a top-level RAML program*
$P \triangleq F, e_{main}$ where $F \triangleq f1(x) = e_{f1}, \ldots, fn(x) = e_{fn}$ s.t.
$\Sigma, x:\tau_{f1} \vdash^{q_1}_{q_1'} e_{f1} : \tau'_{f1}\ \ldots\ \ \Sigma, x:\tau_{fn} \vdash^{q_n}_{q_n'} e_{fn} : \tau'_{fn}$ and $\Sigma,\Gamma \vdash^q_{q'} e_{main} : \tau$
where $\Sigma = f1 : \tau_{f1} \overset{q_1/q_1'}{\to} \tau'_{f1}, \ldots, fn : \tau_{fn} \overset{q_n/q_n'}{\to} \tau'_{fn}$
The translation of P, denoted by $\overline{P}$, is defined as $(\overline{F}, e_t)$ where
$\overline{F} = \mathsf{fix} f_1.\lambda u.\lambda x.e_{t1}, \ldots, \mathsf{fix} f_n.\lambda u.\lambda x.e_{tn}$ s.t.
$\Sigma, x:\tau_{f1} \vdash^{q_1}_{q_1'} e_{f1} : \tau'_{f1} \leadsto e_{t1}\ \ldots\ \ \Sigma, x:\tau_{fn} \vdash^{q_n}_{q_n'} e_{fn} : \tau'_{fn} \leadsto e_{tn}$ and
$\Sigma,\Gamma \vdash^q_{q'} e_{main} : \tau \leadsto e_t$

4.3 Semantic properties of the translation

Besides type-preservation, we additionally : 1) prove that our translation preserves semantics and cost of the source RAML term and 2) re-derive RAML's soundness result using λ-amor$^-$'s fundamental theorem (Theorem 2) and properties of the translation. This is a sanity check to ensure that our type translation preserves cost meaningfully (otherwise, we would not be able to recover RAML's soundness theorem in this way).

Semantics and cost preservation is formally stated in Theorem 7, which can be read as follows: if e_s is a closed source (RAML) term which translates to a target (λ-amor$^-$) term e_t and if the source expression evaluates to a value (and a heap H, because RAML uses imperative boxed data structures) then the target term after applying to a unit (because the translation is always a function) can be evaluated to a value ${}^t v_f$ via pure ($\Downarrow$) and forcing ($\Downarrow^J$) relations s.t. the source and the target values are the same and the cost of evaluation in the target is at least as much as the cost of evaluation in the source.

Theorem 7 (Semantics and cost preservation)**.** *$\forall H, e, {}^s v, p, p', q, q'$.*
$.;. \vdash^q_{q'} e_s : \mathsf{b} \leadsto e_t \wedge .,. \vdash^p_{p'} e \Downarrow {}^s v, H \implies$
$\exists {}^t v_f, J. e_t() \Downarrow \_ \Downarrow^J {}^t v_f \wedge {}^s v = {}^t v_f \wedge p - p' \leq J$

The proof of Theorem 7 is via a cross-language relation between RAML and λ-amor$^-$ terms. The relation (described in the Appendix A) is complex because it has to relate RAML's imperative data structures (like list which is represented as a chain of pointers in the heap) with λ-amor$^-$'s purely functional datastructures. The fundamental theorem of this relation basically allows us to establish that the source expression and its translation are related, which basically internalizes semantics and cost preservation as required by Theorem 7.

Finally, we re-derive RAML's soundness (Theorem 4) in λ-amor$^-$ using λ-amor$^-$'s fundamental theorem and the properties of the translation. To prove this theorem, we obtain a translated term corresponding to the term e (of Theorem 4) via our translation. Then, using Theorem 7, we show that the cost of forcing the unit application of the target is lower-bounded by $p - p'$. After that, we use Corollary 3 to obtain the upper-bound on $p - p'$ as required in the statement of Theorem 4.

5 λ-amor full (with sub-exponentials)

Recall the Church encoding from Section 3.1. A Church numeral always applies the function argument a finite number of times. However, the type that we assigned to Church numeral specified an unbounded number of copies for the function argument. Similarly, the index j_n can only take n unique values in the range 0 to $n-1$, but it was left unrestricted in the type that we saw earlier. Both these limitations are due to λ-amor$^-$'s lack of ability to specify these constraints at the level of types. These limitations, however, can be avoided by refining the exponential type ($!\tau$). In particular, we show that by using the dependent sub-exponential ($!_{i<n}\tau$) from

Bounded Linear Logic [15] we can not only specify a bound on the number of copies of the underlying term, but can also specify the constraints on the index-level substitutions that are needed in the Church encoding. Morally, $!_{i<n}\tau$ represent n copies of τ in which i is uniquely substituted with all values from 0 to $n-1$.

Such an indexed sub-exponential has been used in the prior work. $d\ell$PCF [9], for instance, uses it to obtain relative completeness of typing for PCF programs, which means every PCF program can be type checked in $d\ell$PCF, where the cost of the PCF program gets internalized in $d\ell$PCF's typing derivation. This is a very powerful result. However, cost analysis in $d\ell$PCF works only for whole programs. This is because $d\ell$PCF does not internalize cost into the types but rather tracks it only on the typing judgment. As a result, in order to verify the cost of e_2 in the let expression, say $\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2$, we would need the whole typing derivation of e_1 as cost is encoded on the judgment in $d\ell$PCF.

Contrast this with λ-amor where cost requirements are described in the types ($\mathbb{M}\,\kappa\,\tau$ for instance). In this case, the cost of e_2 can be verified just by knowing the type of e_1 (the whole typing derivation of e_1 is not required to type check e_2. Therefore e_1 can be verified separately).

We show that by adding such an indexed sub-exponential to λ-amor$^-$, we can not only obtain the same relative completeness[7] result that $d\ell$PCF obtains, but also provide a compositional alternative to the $d\ell$PCF style of cost analysis. We describe the addition of $!_{i<n}\tau$ to λ-amor$^-$ in this section. We call the resulting system λ-amor.

5.1 Changes to the type system: syntax and type rules

We take the same language as described earlier in Section 2 but replace the exponential type with an indexed sub-exponential type. There are no changes to the term syntax or semantics of the language. We just extend the index language with two specific counting functions described below.

$$
\begin{array}{llll}
\text{Index} & I, J, K & ::= & \ldots \mid \sum_{a<J} I \mid \triangle\!\!\!\!\bigcirc_a^{J,K} I \mid \ldots \\
\text{Types} & \tau & ::= & \ldots \mid !_{a<I}\tau \mid \ldots \\
\text{Non-affine context} & \Omega & ::= & . \mid \Omega, x :_{a<I} \tau \\
\text{for term variables} & & &
\end{array}
$$

$$
\Omega_1 + \Omega_2 \triangleq \begin{cases} \Omega_2 & \Omega_1 = . \\ (\Omega_1' + \Omega_2/x), x :_{c<I+J} \tau & \Omega_1 = \Omega_1', x :_{a<I} \tau[a/c] \wedge (x :_{b<J} \tau[I+b/c]) \in \Omega_2 \\ (\Omega_1' + \Omega_2), x :_{a<I} \tau & \Omega_1 = \Omega_1', x :_{a<I} \tau \wedge (x :_{-} -) \notin \Omega_2 \end{cases}
$$

$$
\sum_{a<I} \Omega \triangleq \begin{cases} . & \Omega = . \\ (\sum_{a<I} \Omega), x :_{c<\sum_{a<I} J} \sigma & \Omega = \Omega', x :_{b<J} \sigma[(\sum_{d<a} J[d/a] + b)/c] \end{cases}
$$

Figure 10: Changes to the type system syntax

We describe the changes introduced to the type and index language in Fig. 10. Since the sub-exponential type helps in specifying the number of copies of a term, we find inclusion of two specific counting functions to the index language very useful, both of which have been inspired from prior work [9]. The first one is a function for computing a bounded sum over indices, denoted by $\sum_{a<J} I$. It basically describes summation of I with a ranging from 0 to $J-1$ inclusive, i.e., $I[0/a] + \ldots + I[J-1/a]$. The other function is used for computing the number of nodes in a graph structure like a forest of recursion trees. This is called the forest cardinality and denoted $\triangle\!\!\!\!\bigcirc_a^{J,K} I$. The forest cardinality $\triangle\!\!\!\!\bigcirc_a^{J,K} I$ counts the number of nodes in the forest (described by I) consisting of K trees starting from the Jth node. Nodes are assumed to be numbered in a pre-order fashion. It can be formally defined as in Fig. 11 and is used to count and identify children in the recursion tree of a fix construct.

$$
\begin{array}{lcl}
\triangle\!\!\!\!\bigcirc_a^{I,0} K & = & 0 \\
\triangle\!\!\!\!\bigcirc_a^{I,J+1} K & = & \triangle\!\!\!\!\bigcirc_a^{I,J} K + (\triangle\!\!\!\!\bigcirc_a^{I+1+\triangle\!\!\!\!\bigcirc_a^{I,J} K, K[I+\triangle\!\!\!\!\bigcirc_a^{I,J} K/a]} K)
\end{array}
$$

Figure 11: Formal definition of forest cardinality from [9]

The typing judgment is still the same: $\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau$. However, the definition of Ω is now different. The non-affine context Ω now carries the constraint on the index variable described on the ":" as in $x :_{a<I} \tau$ (Fig. 10). It specifies that there are I copies of x with type τ in which the free a is substituted with unique values in the range from 0 to $I-1$. The non-affine context also differs in the definition of splitting. The definition of + (splitting, also referred to as the binary sum) for Ω allows for the same variable to be present in the two contexts but by allowing splitting over the index ranges. Binary sum of Ω_1 and Ω_2 in λ-amor$^-$ was just a disjoint union of the two contexts. However, here in λ-amor, it permits Ω_1 and Ω_2 to have common variables but their multiplicities should add up. We also introduce a notion of bounded sum for the non-affine context denoted by $\sum_{a<I} \Omega$. Both binary and bounded sum over non-affine contexts are described in Fig. 10.

We only describe the type rules for the sub-exponential and the fixpoint in Fig. 12 as these are the only rules that change. T-subExpI is the rule for the introduction form of the sub-exponential. It says that if an

[7] Use of indexed sub-exponential is just one way of obtaining relative completeness. There could be other approaches, which we do not get into here.

expression e has type τ under a non-affine context Ω and $a < I$ s.t. e does not use any affine resources (indicated by an empty Γ) then $!e$ has type $!_{a<I}\tau$ under context $\sum_{a<I}\Omega$. As before, we can always use the weakening rule to add affine resources to the conclusion. T-subExpE is similar to T-expE defined earlier but additionally it also carries the index constraint coming from the type of e_1 in the context for e_2.

The fixpoint expression ($\mathsf{fix}x.e$) encodes recursion by allowing e to refer to $\mathsf{fix}x.e$ via x. T-fix defines the typing for such a fixpoint construct. It is a refinement of the corresponding rule in Fig. 3. The refinements serve two purposes: 1) they make the total number of recursive calls explicit (this is represented by L) and 2) they identify each instance of the recursive call in a pre-order traversal of the recursion tree. This is represented by the index $(b + 1 + \bigcirc\!\!\!\!\!\triangle_b^{b+1,a} I)$ (representing the ath child of the bth node in the pre-order traversal). Using these two refinements, the T-fix rule in Fig. 12 can be read as follows: if for all I copies of x in the context we can type check e with τ, then we can also type check the top-most instance of $\mathsf{fix}\ x.e$ with type $\tau[0/b]$ (0 denotes the starting node in the pre-order traversal of the entire recursion tree). Contrast the rules described in Fig. 12 with the corresponding rules for λ-amor$^-$ described earlier in Fig. 3.

$$\frac{\Psi; \Theta, a; \Delta, a < I; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \sum_{a<I} \Omega; . \vdash !e : !_{a<I}\tau} \text{ T-subExpI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (!_{a<I}\tau) \qquad \Psi; \Theta; \Delta; \Omega_2, x :_{a<I} \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega_1 + \Omega_2; \Gamma_1 + \Gamma_2 \vdash \mathsf{let}\, !\, x = e \,\mathsf{in}\, e' : \tau'} \text{ T-subExpE}$$

$$\frac{\Psi; \Theta, b; \Delta, b < L; \Omega, x :_{a<I} \tau[(b + 1 + \bigcirc\!\!\!\!\!\triangle_b^{b+1,a} I)/b]; . \vdash e : \tau \qquad L \geq \bigcirc\!\!\!\!\!\triangle_b^{0,1} I}{\Psi; \Theta; \Delta; \sum_{b<L} \Omega; . \vdash \mathsf{fix}\ x.e : \tau[0/b]} \text{ T-fix}$$

Figure 12: Changes to the type rules

We also introduce a new subtyping rule, sub-bSum. sub-bSum helps move the potential from the outside to the inside of a sub-exponential. This is sound because 1) potentials are really ghosts at the term level. Therefore terms of type $[\sum_{a<I} K]\, !_{a<I}\tau$ and $!_{a<I}\, [K]\, \tau$ are both just exponentials and 2) there is only a change in the position but no change of potential in going from $[\sum_{a<I} K]\, !_{a<I}\tau$ to $!_{a<I}\, [K]\, \tau$. We have proved that this new subtyping rule is sound wrt the model of λ-amor types by proving that if τ is a subtype of τ' according to the syntactic subtyping rules then the interpretation of τ is a subset of the interpretation of τ'. This is formalized in Lemma 8. σ and ι represent the substitutions for the type and index variables respectively.

$$\frac{}{\Psi; \Theta; \Delta \vdash [\sum_{a<I} K]\, !_{a<I}\tau <: !_{a<I}\, [K]\, \tau} \text{ sub-bSum}$$

It is noteworthy that sub-bSum is the only rule in λ-amor which specifies how the two modalities, namely, the sub-exponential ($!_{a<I}\tau$) and the modal type ($[p]\,\tau$) interact with each other. People familiar with monads and comonads might wonder, why such an interaction between the sub-exponential and the monad is not required? We believe this is because we can always internalize the cost on the type using the store construct, so just relating exponential and potential modal type suffices. However, studying such interactions could be an interesting direction for future work.

Lemma 8 (Value subtyping lemma). $\forall \Psi, \Theta, \Delta, \tau \in \mathit{Type}, \tau', \sigma, \iota.$
$\Psi; \Theta; \Delta \vdash \tau <: \tau' \wedge . \models \Delta\iota \implies [\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$

5.2 Model of types

We only describe the value relation for the sub-exponential here as the remaining cases of the value relation are exactly the same as before. $(p, T, !e)$ is in the value interpretation at type $!_{a<I}\tau$ iff the potential p suffices for all I copies of e at the *instantiated* types $\tau[i/a]$ for $0 \leq i < I$. The other change to the model is in the interpretation of Ω. This time we have (p, δ) instead of $(0, \delta)$ in the interpretation of Ω s.t. p is sufficient for all copies of all variables in the context. The changes to the model are described in Fig. 13.

$$\begin{array}{lcl}
[\![!_{a<I}\tau]\!] & \triangleq & \{(p, !e) \mid \exists p_0, \ldots, p_{I-1}.p_0 + \ldots + p_{I-1} \leq p \wedge \forall 0 \leq i < I.(p_i, e) \in [\![\tau[i/a]]\!]_{\mathcal{E}}\} \\
[\![\Omega]\!]_{\mathcal{E}} & = & \{(p, \delta) \mid \exists f : \mathit{Vars} \to \mathit{Indices} \to \mathit{Pots}. \\
& & (\forall (x :_{a<I} \tau) \in \Omega. \forall 0 \leq i < I. (f\ x\ i, \delta(x)) \in [\![\tau[i/a]]\!]_{\mathcal{E}}) \wedge \\
& & (\sum_{x:_{a<I}\tau\in\Omega} \sum_{0\leq i<I} f\ x\ i) \leq p\}
\end{array}$$

Figure 13: Changes to the model

We prove the soundness of the model by proving a slightly different fundamental theorem (Theorem 9). There is an additional potential (p_m) coming from the interpretation of Ω (which was 0 earlier).

Theorem 9 (Fundamental theorem). $\forall \Psi, \Theta, \Delta, \Omega, \Gamma, e, \tau \in Type, p_l, p_m, \gamma, \delta, \sigma, \iota.$
$\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \wedge (p_l, \gamma) \in \llbracket \Gamma\ \sigma\iota \rrbracket_{\mathcal{E}} \wedge (p_m, \delta) \in \llbracket \Omega\ \sigma\iota \rrbracket_{\mathcal{E}} \wedge . \models \Delta\ \iota \implies$
$(p_l + p_m, e\ \gamma\delta) \in \llbracket \tau\ \sigma\iota \rrbracket_{\mathcal{E}}$

The proof of the theorem proceeds in a manner similar to that of Theorem 2, i.e., by induction on the typing derivation. Now, in the fix case, we additionally induct on the recursion tree (this also involves generalizing the induction hypothesis to account for the potential of the children of a node in the recursion tree). The Appendix B has the entire proof.

6 Embedding $d\ell$PCF

In this section we describe an embedding of $d\ell$PCF [9] into λ-amor. $d\ell$PCF is a *coeffect-based* type system (contrast this with RAML which is an *effect-based* type system) which has been shown to be relatively complete for cost analysis of PCF programs. The objective of this embedding is to show that λ-amor perform *coeffect-based* cost analysis like $d\ell$PCF and hence is relatively complete for PCF too. Additionally, λ-amor (due to its ability to internalize cost in types) can be seen as a compositional extension of $d\ell$PCF (which can analyze whole programs only), as pointed out earlier.

6.1 A brief primer on $d\ell$PCF

$d\ell$PCF [9] is a call-by-name calculus with an affine type system for doing cost analysis of PCF programs. Terms and types of $d\ell$PCF are described in Fig. 14. $d\ell$PCF works with the standard PCF terms but refines the standard types of PCF a bit to perform cost analysis. The type of natural numbers is refined with two indices $Nat[I, J]$ to capture types for natural numbers in the range $[I, J]$ specified by the indices. Function types are refined with index constraints in the negative position. For instance, $[a < I]\tau_1 \multimap \tau_2$ is the type of a function which when given I copies of an expression (since $d\ell$PCF is call-by-name) of type τ_1 will produce a value of type τ_2. The $[a < I]$ acts both as a constraint on what values a can take and also as a binder for free occurrence of a in τ_1 (but not in τ_2). $[a < I]\tau_1 \multimap \tau_2$ is morally equivalent to $(\tau_1[0/a] \otimes \ldots \otimes \tau_1[I-1/a]) \multimap \tau_2$.

$$
\begin{array}{llcl}
d\ell\text{PCF terms} & t & ::= & n \mid s(t) \mid p(t) \mid ifz\ t\ then\ u\ else\ v \mid \lambda x.t \mid tu \mid \mathsf{fix}\ x.t \\
d\ell\text{PCF types} & \sigma & ::= & Nat[I, J] \mid A \multimap \sigma \\
 & A & ::= & [a < I]\sigma
\end{array}
$$

Figure 14: $d\ell$PCF's syntax of terms and types from [9]

The typing judgment of $d\ell$PCF is given by $\Theta; \Delta; \Gamma \vdash^{\xi}_{C} e_d : \tau$. Θ denotes a context of index variables, Δ denotes a context for index constraints, Γ denotes a context of term variables and C denotes the cost of evaluation of e_d. This cost C is the number of variable lookups in a full execution of e_d. ξ on the turnstile denotes an equational program used for interpreting the function symbols of the index language. Like in the negative position of the function type, multiplicities also show up with the types of the variables in the typing context. The typing rules are designed to track these multiplicities (which is a coeffect in the system). For illustration, we only show the typing rule for function application in Fig. 15. Notice how the cost in the conclusion is lower bounded by the sum of: a) the number of times the argument of e_1 can be used by the body, i.e., I, b) the cost of e_1, i.e., J and c) the cost of I copies of e_2, i.e., $\sum_{a<I} K$. The authors of [9] show that this kind of coeffect tracking in the type system actually suffices to give an upper-bound on the cost of execution on a K_{PCF} machine, a Krivine-style machine [25] for PCF.

$$
\frac{\Theta; \Delta; \Gamma \vdash_J e_1 : ([a < I].\tau_1) \multimap \tau_2 \qquad \Theta, a; \Delta, a < I; \Delta \vdash_K e_2 : \tau_1 \qquad \Gamma' \sqsupseteq \Gamma \oplus \sum_{a<I} \Delta \qquad H \geq I + J + \sum_{a<I} K}{\Theta; \Delta; \Gamma' \vdash_H e_1\ e_2 : \tau_2}\ \text{app}
$$

Figure 15: Typing rule for function application from [9]

States of the K_{PCF} machine consist of triples of the form (t, ρ, θ) where t is a $d\ell$PCF term, ρ is an environment for variable binding and θ is stack of closures. A closure (denoted by $\mathtt{C}$) is simply a pair consisting of a term and an environment. The left side of Fig. 16 describes some evaluation rules of the K_{PCF} machine from [9]. For instance, the application triple $(e_1 e_2, \rho, \theta)$ reduces in one step to e_1, and e_2 along with the current closure is pushed on top of the stack for later evaluation. This is how one would expect an evaluation to happen in a call-by-name scheme. One final ingredient that we need to describe for the soundness of $d\ell$PCF is a notion of the size of a term, denoted by $|t|$. The size of a $d\ell$PCF term is defined in [9] (we describe some of the clauses on the right side of Fig. 16).

Finally $d\ell$PCF soundness (Theorem 10) states that the execution cost (denoted by n) is upper-bounded by the product of the size of the initial term, t and $(I + 1)$. $d\ell$PCF states the soundness result for base (bounded naturals) types only and soundness for functions is derived as a corollary. $\Downarrow^n$ is a shorthand for $\xrightarrow{n}$ (n-step closure under the K_{PCF} reduction relation).

$$
\begin{array}{lcl}
(e_1\ e_2, \rho, \theta) & \to & (e_1, \rho, (e_2, \rho).\theta) \\
(\lambda x.e, \rho, \mathtt{C}.\theta) & \to & (e_1, \mathtt{C}.\rho, \theta) \\
(x, (t_0, \rho_0) \dots (t_n, \rho_n), \theta) & \to & (t_x, \rho_x, \theta) \\
(\mathsf{fix} x.e, \rho, \theta) & \to & (e, (x, (\mathsf{fix} x.e, \mathtt{C}).\rho, \theta)
\end{array}
\quad\Bigg|\quad
\begin{array}{lcl}
|x| & = & 1 \\
|c| & = & 1 \\
|\lambda x.e| & = & |e| + 1 \\
|e_1\ e_2| & = & |e_1| + |e_2| + 1 \\
|\mathsf{fix} x.e| & = & |e| + 1
\end{array}
$$

Figure 16: K_{PCF} reduction rules (left) and size function (right) from [9]

Theorem 10 ($d\ell$PCF's soundness from [9])**.** $\forall t, I, J, K.$
$\vdash_I t : Nat[J, K] \wedge t \Downarrow^n m \implies n \leq |t| * (I + 1)$

6.2 Type-directed translation of $d\ell$PCF into λ-amor

Without loss of generality, as in RAML's embedding, we abstract the type of naturals and treat them as a general abstract base type b. Like RAML, $d\ell$PCF's embedding is also type directed. The type translation function is described in Fig. 17. $d\ell$PCF's base type is translated into the base type of λ-amor. The function type ($[a < I]\tau_1 \multimap \tau_2$) translates to a function which takes I copies of the monadic translation of τ_1 (following Moggi [28]) and I units of potential (to account for I substitutions during application) as a modal unit type, and returns a monadic type of translation of τ_2. The monad on the return type is essential as a function cannot consume (I units of) potential and still return a pure value. The translation of the typing context is defined pointwise for every variable in the context. Since all variables in the $d\ell$PCF's typing context have comonadic types (carrying multiplicities), $d\ell$PCF's typing context is translated into the non-affine typing context of λ-amor.

$$
\begin{array}{lcl}
(\!|b|\!) & = & b \\
(\!|[a < I]\tau_1 \multimap \tau_2|\!) & = & !_{a<I}\, \mathbb{M}\, 0 (\!|\tau_1|\!) \multimap [I]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau_2|\!)
\end{array}
\quad\Bigg|\quad
\begin{array}{lcl}
(\!|.|\!) & = & . \\
(\!|\Gamma, x : [a < I]\tau|\!) & = & (\!|\Gamma|\!), x :_{a<I} \mathbb{M}\, 0\, (\!|\tau|\!)
\end{array}
$$

Figure 17: Type and context translation for $d\ell$PCF

The translation judgment is of the form $\Theta; \Delta; \Gamma \vdash_I e_d : \tau \quad \leadsto \quad e_a$ where e_a denotes the translated λ-amor term. ξ never changes in any of $d\ell$PCF's typing rules, so for simplification we assume it to be present globally and thus we omit it from the translation judgment. The expression translation of $d\ell$PCF terms is defined by induction on typing judgments (Fig. 18). Notice that in the variable rule (var) we place a deliberate tick construct which consumes one unit of potential. This is done to match the cost model of $d\ell$PCF. Without this accounting our semantics and cost preservation theorem would not hold. The translation of function application and the fixpoint construct make use of a coercion function (*coerce*, which is written in λ-amor itself). It helps convert an application of exponentials into an exponential of application. The coercion function is described in the box along with the expression translation rules in Fig. 18.

$$
\dfrac{\Theta; \Delta \models J \geq 0 \qquad \Theta; \Delta \models I \geq 1 \qquad \Theta; \Delta \vdash \sigma[0/a] <: \tau \qquad \Theta; \Delta \models [a < I]\sigma \Downarrow \qquad \Theta; \Delta \models \Gamma \Downarrow}{\Theta; \Delta; \Gamma, x : [a < I]\tau \vdash_J x : \tau[0/a] \leadsto \lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } x}\ \text{var}
$$

$$
\dfrac{\Theta; \Delta; \Gamma, x : [a < I]\tau_1 \vdash_J e : \tau_2 \leadsto e_t}{\begin{array}{c}\Theta; \Delta; \Gamma \vdash_J \lambda x.e : ([a < I].\tau_1) \multimap \tau_2 \leadsto \\ \lambda p_1.\, \mathsf{ret}\, \lambda y.\lambda p_2.\, \mathsf{let}\, !\, x = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_t\ a\end{array}}\ \text{lam}
$$

$$
\dfrac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash_J e_1 : ([a < I].\tau_1) \multimap \tau_2 \leadsto e_{t1} \\ \Theta, a; \Delta, a < I; \Delta \vdash_K e_2 : \tau_1 \leadsto e_{t2} \qquad \Gamma' \sqsupseteq \Gamma \oplus \sum_{a<I} \Delta \qquad H \geq J + I + \sum_{a<I} K\end{array}}{\begin{array}{c}\Theta; \Delta; \Gamma' \vdash_H e_1\ e_2 : \tau_2 \leadsto \\ \lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce\ !e_{t2}\ c)\ d\end{array}}\ \text{app}
$$

$$
\dfrac{\begin{array}{c}\Theta, b; \Delta, b < L; \Gamma, x : [a < I]\sigma \vdash_K e : \tau \leadsto e_t \qquad \tau[0/a] <: \mu \\ \Theta, a, b; \Delta, a < I, b < L; \Gamma \vdash \tau[(b + 1 + \bigotimes_b^{b+1,a} I)/b] <: \sigma \qquad \Gamma' \sqsubseteq \sum_{b<L} \Gamma \qquad L, M \geq \bigotimes_b^{0,1} I \qquad N \geq M - 1 + \sum_{b<L} K\end{array}}{\Theta; \Delta; \Gamma' \vdash_N \mathsf{fix} x.e : \mu \leadsto E_0}\ \text{T-fix}
$$

$E_0 = \mathsf{fix} Y.\lambda p.\mathsf{release} - = p \text{ in } E_1$, $E_1 = \mathsf{release} - = p \text{ in } E_2$
$E_2 = \mathsf{bind}\, A = \mathsf{store}() \text{ in } \mathsf{let}\, !\, x = (E_{2.1}\ E_{2.2}) \text{ in } \mathsf{bind}\, C = \mathsf{store}() \text{ in } e_t\ C$
$E_{2.1} = coerce\ !Y$, $E_{2.2} = (\lambda u.!())\ A$

$$
\boxed{\begin{array}{l} coerce :\ !_{a<I}(\tau_1 \multimap \tau_2) \multimap !_{a<I}\tau_1 \multimap !_{a<I}\tau_2 \\ coerce\ F\ X \triangleq \mathsf{let}\, !\, f = F \text{ in } \mathsf{let}\, !\, x = X \text{ in} !(f\ x) \end{array}}
$$

Figure 18: Expression translation: $d\ell$PCF to λ-amor

We want to highlight another point about this translation. This is the second instance (the first one was embedding of Church numerals, Section 3.1) where embedding using just a cost monad (without potentials) does not seem to work. To understand this, let us try to translate $d\ell$PCF's function type ($[a < I]\tau_1 \multimap \tau_2$) using only the cost monad and without the potentials. One possible translation of $[a < I]\tau_1 \multimap \tau_2$ is $(!_{a<I}(\!|\tau_1|\!)) \multimap \mathbb{M}\, I\, (\!|\tau_2|\!)$.

The I in the monadic type is used to account for the cost of substitution of the I copies of the argument in $d\ell$PCF. Now, in the rule for function abstraction we have to generate a translated term of the type $\mathbb{M}(J + count(\Gamma))\,(!_{a<I}(\!|\tau_1|\!) \multimap \mathbb{M}\,I\,(\!|\tau_2|\!))$. From the induction hypothesis, we have a term of type $\mathbb{M}(I + J + count(\Gamma))\,(\!|\tau_2|\!)$. A possible term translation can be $\mathsf{ret}\,\lambda y.\,\mathsf{let}\,!\,x = y\,\mathsf{in}\,e_t$. This would require us to type e_t with $\mathbb{M}\,I\,(\!|\tau_2|\!)$ under the given context with a free x. However e_t can only be typed with $\mathbb{M}(I + J + count(\Gamma))\,(\!|\tau_2|\!)$ (which cannot be coerced to the desired type). Hence, the translation with just cost monads does not work. We believe that such a translation can be made to work by adding appropriate coercion axioms for the cost monads.

However, there is an alternate way to make this translation work, using the modal type and that is what we use. The idea is to capture the I units as a *potential* using the modal type of λ-amor (in the negative position) instead of capturing it (in the positive position) as a cost on the monad. Concretely, this means that, instead of translating $[a < I]\tau_1 \multimap \tau_2$ to $(!_{a<I}(\!|\tau_1|\!)) \multimap \mathbb{M}\,I\,(\!|\tau_2|\!)$, we translate it to $(!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$ (as described in Fig. 17 earlier). Likewise, the typing judgment is also translated using the same potential approach (as described in Theorem 11). Following this approach, we obtain a term of type $[(J + I + count(\Gamma))]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$ from the induction hypothesis and we are required to produce a term of type $[J + count(\Gamma)]\,\mathbf{1} \multimap$ $\mathbb{M}\,0((!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!))$ in the conclusion. By using the ghost constructs (namely store and release) to rearrange the given potential of $J + count(\Gamma)$ and I units into a potential of $(J + I + count(\Gamma))$ units, it is clear that we can obtain a term of the desired type from the induction hypothesis. The exact term is described in the lam rule of Fig. 18.

We prove that this translation is type preserving (Theorem 11). In particular, we show that the translated term can be typed at the function type $[I + count(\Gamma)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau|\!)$, where count is defined as $count(\Gamma, x : [a < I]\tau) = count(\Gamma) + I$ (with $count(.) = 0$ as the base case). Since $d\ell$PCF counts cost for each variable lookup in a terminating K_{PCF} reduction, the translated term must have enough potential to make sure that all copies of free variables in the context can be used. This is ensured by having $(I + count(\Gamma))$ potential as input (in the argument position of the translated type): I accounts for the substitutions coming from function applications in the $d\ell$PCF expression and $count(\Gamma)$ accounts for the total number of possible substitutions of context variables. All translated expressions release the input potential coming from the argument. This is later consumed using a tick as in the variable rule or stored with a unit value to be used up by the induction hypothesis.

Theorem 11 (Type preservation: $d\ell$PCF to λ-amor)**.** *If $\Theta; \Delta; \Gamma \vdash_I e : \tau$ in $d\ell$PCF then there exists e' such that $\Theta; \Delta; \Gamma \vdash_I e : \tau \rightsquigarrow e'$ such that there is a derivation of $.; \Theta; \Delta; (\!|\Gamma|\!); . \vdash e' : [I + count(\Gamma)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau|\!)$ in λ-amor.*

6.3 Semantic properties of the translation

Besides type preservation we also prove semantics and cost preservation for the translation. To achieve that, we define a cross-language relation between $d\ell$PCF and λ-amor terms by induction on the $d\ell$PCF types. We show that a source ($d\ell$PCF) term and a unit application of its translation (because translated terms are always function abstractions, as described above) are in the cross-language relation at the type of the source. We defer the details of the cross-language relation to the Appendix B. Finally, to show that the meaning of cost annotations in the source ($d\ell$PCF) is not lost during this translation, we want to re-derive $d\ell$PCF's soundness in λ-amor using the properties of the translation only. But $d\ell$PCF's soundness is defined wrt reduction on a K_{PCF} machine [25], as described earlier. So, we would like to re-derive a proof of Theorem 12[8].

Theorem 12 (Generalized $d\ell$PCF's soundness)**.** *$\forall t, I, \tau, \rho.$*
$\vdash_I (t, \epsilon, \epsilon) : \tau \wedge (t, \epsilon, \epsilon) \xrightarrow{n} (v, \rho, \epsilon) \implies n \leq |t| * (I + 1)$

To prove Theorem 12, we need a way of relating K_{PCF} triples to λ-amor terms. So, we come up with an approach for decompiling K_{PCF} triples into $d\ell$PCF terms (which we can then transitively relate to λ-amor terms via our translation). The decompilation is defined as a function (denoted by $(\!|.|\!)$) from Krivine triples to $d\ell$PCF terms. We first define decompilation for closures ($(\!|.|\!)$ is overloaded), by induction on the environment. For an empty environment the decompilation is simply an identity on the given term. And for an environment of the form $\mathtt{C}_1, \ldots, \mathtt{C}_n$, the decompilation is given by closing off the open parts of the given term. Direct substitution of closures in e won't work, as this will take away all the free variables in e. As a result, the decompiled term would not have any cost due to variable lookups, something which $d\ell$PCF's type system explicitly tracks. And the decompilation won't remain cost-preserving. So instead, we decompile it using lambda abstraction and application as described on the left side of Fig. 19. Using this closure decompilation, we define decompilation for the full Krivine triples. When stack is empty, it is just the decompilation of the underlying closure. And when stack is non-empty, the closures on the stack are applied one after the other on the closed term obtained via the translation of the closure. This is described on the right side of Fig. 19. We prove that the decompilation preserves type, cost and semantics of the Krivine triple.

[8] This is a generalized version of $d\ell$PCF's soundness (Theorem 10), where we prove the cost bound for terms of arbitrary types.

$(\!|(e, [])|\!)$	$\triangleq$	e	$(\!|(e, \rho, [])|\!)$	$\triangleq$	$(\!|(e, \rho)|\!)$
$(\!|(e, \mathtt{C}_1, \ldots, \mathtt{C}_n)|\!)$	$\triangleq$	$(\lambda x_1 \ldots x_n.e)\ (\!|\mathtt{C}_1|\!) \ldots (\!|\mathtt{C}_n|\!)$	$(\!|(e, \rho, \mathtt{C}.\theta)|\!)$	$\triangleq$	$(\!|(\!|(e, \rho)|\!)\ (\!|\mathtt{C}|\!), [], \theta|\!)$

Figure 19: Decompilation of closure (left) and Krivine triple (right)

Finally, we compose the decompilation of Krivine triples to $d\ell$PCF terms with the translation of $d\ell$PCF to λ-amor terms to obtain a composite translation from Krivine triples to λ-amor. We then prove that this composite translation preserves the meaning of cost annotations wrt to the intentional soundness criteria stated in Theorem 12. The proof is quite involved, but due to lack of space we cannot get into the technicalities of that proof.

7 Related work

Literature on cost analysis is very vast; we summarize and compare to only a representative fraction, covering several prominent *styles* of cost analysis.

Type and effect systems. Several type and effect system have been proposed for amortized analysis using the method of potentials. Early approaches [20, 22] allow the potential associated with a value to only be a linear function of the value's size. Univariate RAML [19] generalizes this to polynomial potentials. Multivariate RAML [17] is a substantial generalization where a single potential, that is a polynomial of the sizes of multiple input variables, can be associated to all of them together. These approaches are inherently first-order in their treatment of potentials – closures that capture potentials are disallowed by the type systems. We already showed how to embed Univariate RAML in λ-amor in Section 4, and we believe that the embedding can be extended to Multivariate RAML.

AARA [21, 18] extends RAML with *limited* support for closures and higher-order functions. [18] cannot handle Curry-style functions at all, while [21] can handle Curried functions only when the potential is associated with the last argument. As explained in Section 1, these limitations arise from incomplete support for affineness. In contrast, λ-amor, being affine, does not have such limitations.

Some prior work such as the unary fragment of [6] uses effect-based type systems for *non-amortized* cost analysis. A significant line of work tracing lineage back to at least [8] uses sized types and cost represented in a writer monad for cost analysis. More recently, [12, 3, 24], show how to extend this idea to extract sound recurrences from programs. These recurrences can be solved to establish cost bounds. However, this line of work does not support potentials or amortized analysis. Conceptually, it is simpler than the above-mentioned work on amortized cost analysis (it corresponds to RAML functions where the input and output potentials are both 0) and, hence, can be easily simulated in λ-amor.

Cost analysis using program logics. As an alternative to type systems, a growing line of work uses variants of Hoare logic for amortized cost analysis [4, 5, 27]. The common idea is to represent the potential before and after the execution of a code segment as ghost state in the pre- and post-condition of the segment, respectively. Conceptually, this idea is not very different from how we encode potentials using our $[p]\,\tau$ construct in the inputs and outputs of functions (e.g., in embedding RAML in Section 4). However, unlike λ-amor, prior work shows neither embeddings of existing frameworks, nor any (relative) completeness result. [27] introduce a new concept called *time receipts*, which can be used for lower-bound analysis, something that λ-amor does not support yet.

We note that we could have developed λ-amor and showed our embeddings and the relative completeness result using a program logic in place of a type theory. For our purposes, the difference between the two is only a matter of personal preference.

Cost analysis of lazy programs. Some prior work [11, 26, 23] develops methods for cost analysis of lazy programs. While the semantics of laziness, as in call-by-need, cannot be directly embedded in λ-amor, we can replicate the analysis of some lazy programs, with nearly identical potentials, in the call-by-name setting of λ-amor. We already showed an example of this by verifying Okasaki's implicit queue in Section 3.3, replicating an analysis by [11]. We believe that other examples of Okasaki [30] can be replicated in λ-amor (in a call-by-name setting).

An interesting aspect is that amortized analysis of lazy programs does not necessarily require affineness. [11] circumvents the issue related to duplication of potentials by representing potential using the same indexed monad that represents cost. To represent offsetting of cost by potential, he introduces a primitive coercion "pay" of type $\mathbb{M}(\kappa_1 + \kappa_2)\,\tau \multimap \mathbb{M}\,\kappa_1\,(\mathbb{M}\,\kappa_2\,\tau)$, which, in a way, encodes paying κ_1 part of the cost $\kappa_1 + \kappa_2$ using potential from outside. An interesting question is whether we could use the same idea and do away with our construct for potentials. It turns out that our construct for potentials satisfies additional properties that are needed for embedding $d\ell$PCF. In particular, making $d\ell$PCF's embedding work in Danielsson's style requires a different coercion of type $(\tau' \multimap \mathbb{M}(\kappa_1 + \kappa_2)\,\tau) \multimap \mathbb{M}\,\kappa_1\,(\tau' \multimap \mathbb{M}\,\kappa_2\,\tau)$.

Coeffect-based cost analysis. $d\ell$PCF [9] and $d\ell\text{PCF}_\text{V}$ [10] are coeffect-based type systems for non-amortized cost analysis of PCF programs in the call-by-name and call-by-value settings, respectively. Both systems count the number of variable lookups during execution on an abstract machine (the Krivine machine for call-by-name and the CEK machine for call-by-value [25, 13]). This is easily done by tracking (as a coeffect) the number of uses of each variable in an affine type system with dependent exponentials (λ-amor borrows this exponentials, but does not use coeffects for tracking cost). A common limitation is that these type systems cannot internalize the cost of a program into its type; instead the cost is a function of the typing derivation. We showed in Section 6 that λ-amor can embed $d\ell$PCF and internalize its costs into types. Hence, λ-amor advances beyond $d\ell$PCF. We expect that λ-amor can also embed $d\ell\mathrm{PCF_V}$, but have not tried this embedding yet.

[2] presents Quantitative Type Theory (QTT), which is a dependent type theory with coeffects. QTT and λ-amor are very different in their goals. QTT is focused on studying the interaction between dependent types and coeffects, on the other hand, λ-amor studies coeffects from the perspective of cost analysis. Technically, QTT only considers non-dependent coeffects, as in $x :_n \tau$. In contrast, λ-amor studies coeffects with uniform linear dependencies coming from the dependent sub-exponential of Bounded Linear Logic [15], as in $x :_{a<n} \tau$.

8 Conclusion

We have presented λ-amor, a type-theory for (amortized) cost analysis. λ-amor introduces a new modal type constructor to represent potential at the level of types and uses affine typing. We view λ-amor as a unifying framework for representing cost analyses, as witnessed by faithful embeddings of Univariate RAML and $d\ell$PCF. These encompass a variety of settings, ranging from call-by-value to call-by-name and effect-based to coeffect-based.

A Development for λ-amor$^-$

A.1 Syntax

$$
\begin{array}{llcl}
\text{Expressions} & e & ::= & v \mid e_1\ e_2 \mid \langle\!\langle e_1, e_2\rangle\!\rangle \mid \mathsf{let}\langle\!\langle x, y\rangle\!\rangle = e_1 \mathsf{\ in\ } e_2 \mid \\
 & & & \langle e, e\rangle \mid \mathsf{fst}(e) \mid \mathsf{snd}(e) \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid \mathsf{case}\ e, x.e, y.e \mid \\
 & & & \mathsf{let}\,!\,x = e_1 \mathsf{\ in\ } e_2 \mid e :: e \mid e\ [] \mid e; x.e \\
\text{Values} & v & ::= & x \mid () \mid c \mid \lambda x.e \mid \langle\!\langle v_1, v_2\rangle\!\rangle \mid \langle v, v\rangle \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid\ !\,e \mid nil \mid \\
 & & & \Lambda.e \mid \mathsf{ret}\, e \mid \mathsf{bind}\, x = e_1 \mathsf{\ in\ } e_2 \mid \uparrow^I \mid \mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2 \mid \mathsf{store}\, e \\
 & & & \hfill \text{(No value forms for } [I]\,\tau) \\
\text{Index} & I & ::= & i \mid N \mid R \mid I + I \mid I - I \mid \lambda_s i : \mathsf{S}.I \mid I\ I \\
\text{Sort} & \mathsf{S} & ::= & \mathbb{N} \mid \mathbb{R}^+ \mid \mathsf{S} \to \mathsf{S} \\
\text{Kind} & \mathsf{K} & ::= & Type \mid \mathsf{S} \to \mathsf{K} \\
\text{Types} & \tau & ::= & \mathbf{1} \mid \mathsf{b} \mid \tau_1 \multimap \tau_2 \mid \tau_1 \otimes \tau_2 \mid \tau_1 \,\&\, \tau_2 \mid \tau_1 \oplus \tau_2 \mid\ !\,\tau \mid [I]\,\tau \mid \mathbb{M}\, I\, \tau \mid L^I\ \tau \\
 & & & \alpha \mid \forall \alpha : \mathsf{K}.\tau \mid \forall i : \mathsf{S}.\tau \mid \lambda_t i : \mathsf{S}.\tau \mid \tau\ I \mid \exists i : \mathsf{S}.\tau \mid c \Rightarrow \tau \mid c\&\tau \\
\text{Constraints} & c & ::= & I = I \mid I < I \mid c \wedge c \\
 & & & \\
\text{Lin. context for term variables} & \Gamma & ::= & . \mid \Gamma, x : \tau \\
\text{Unres. context for term variables} & \Omega & ::= & . \mid \Omega, x : \tau \\
\text{Unres. context for index variables} & \Theta & ::= & . \mid \Theta, i : \mathsf{S} \\
\text{Unres. context for type variables} & \Psi & ::= & . \mid \Psi, \alpha : \mathsf{K}
\end{array}
$$

Definition 13 (Binary sum of multiplicity context)**.**

$$
\Omega_1 \oplus \Omega_2 \triangleq \begin{cases} \Omega_2 & \Omega_1 = . \\ (\Omega_1' \oplus \Omega_2), x : \tau & \Omega_1 = \Omega_1', x : \tau \wedge (x : -) \notin \Omega_2 \\ undefined & \Omega_1 = \Omega_1', x : \tau \wedge (x : \tau) \in \Omega_2 \end{cases}
$$

Definition 14 (Binary sum of affine context)**.**

$$
\Gamma_1 \oplus \Gamma_2 \triangleq \begin{cases} \Gamma_2 & \Gamma_1 = . \\ (\Gamma_1' \oplus \Gamma_2), x : \tau & \Gamma_1 = \Gamma_1', x : \tau \wedge (x : -) \notin \Gamma_2 \\ undefined & \Gamma_1 = \Gamma_1', x : \tau \wedge (x : -) \in \Gamma_2 \end{cases}
$$

A.2 Typesystem

$\boxed{\text{Typing } \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}$

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau \vdash x : \tau}\text{ T-var1} \qquad \frac{}{\Psi; \Theta; \Delta; \Omega, x : \tau; \Gamma \vdash x : \tau}\text{ T-var2} \qquad \frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash () : \mathbf{1}}\text{ T-unit}$$

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash c : \mathsf{b}}\text{ T-base} \qquad \frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash nil : L^0\ \tau}\text{ T-nil}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : L^n\tau \qquad \Theta; \Delta \vdash n : \mathbb{N}}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e_1 :: e_2 : L^{n+1}\tau}\text{ T-cons}$$

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : L^n\ \tau \qquad \Psi; \Theta; \Delta, n = 0; \Omega; \Gamma_2 \vdash e_1 : \tau' \\ \Psi; \Theta, i; \Delta, n = i+1; \Omega; \Gamma_2, h : \tau, t : L^i\tau \vdash e_2 : \tau' \qquad \Theta; \Delta \vdash n : \mathbb{N} \qquad \Psi; \Theta; \Delta \vdash \tau'\end{array}}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau'}\text{ T-match}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau[n/s] \qquad \Theta; \Delta \vdash n : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \exists s : \mathsf{S}.\tau}\text{ T-existI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : \exists s.\tau \qquad \Psi; \Theta, s; \Delta; \Omega; \Gamma_2, x : \tau \vdash e' : \tau' \qquad \Psi; \Theta; \Delta \vdash \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e; x.e' : \tau'}\text{ T-existE}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \lambda x.e : (\tau_1 \multimap \tau_2)}\text{ T-lam} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : (\tau_1 \multimap \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e_1\ e_2 : \tau_2}\text{ T-app}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi, \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau'}\text{ T-sub}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi; \Theta; \Delta \models \Gamma' \sqsubseteq \Gamma \qquad \Psi; \Theta; \Delta \models \Omega' \sqsubseteq \Omega}{\Psi; \Theta; \Delta; \Omega'; \Gamma' \vdash e : \tau}\text{ T-weaken}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \langle\!\langle e_1, e_2 \rangle\!\rangle : (\tau_1 \otimes \tau_2)}\text{ T-tensorI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \otimes \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1, y : \tau_2 \vdash e' : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\langle\!\langle x, y \rangle\!\rangle = e\ \mathsf{in}\ e' : \tau}\text{ T-tensorE}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \langle e_1, e_2 \rangle : (\tau_1 \,\&\, \tau_2)}\text{ T-withI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1 \ \& \ \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{fst}(e) : \tau_1}\ \text{T-fst} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1 \ \& \ \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{snd}(e) : \tau_2}\ \text{T-snd} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inl}(e) : \tau_1 \oplus \tau_2}\ \text{T-inl}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inr}(e) : \tau_1 \oplus \tau_2}\ \text{T-inr}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \oplus \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, y : \tau_2 \vdash e_2 : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{case}\ e, x.e_1, y.e_2 : \tau}\ \text{T-case}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash !e : !\tau}\ \text{T-ExpI} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : !\tau \qquad \Psi; \Theta; \Delta; \Omega, x : \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\,!\,x = e\ \mathsf{in}\ e' : \tau'}\ \text{T-ExpE}$$

$$\frac{\Psi, \alpha : \mathsf{K}; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall \alpha : \mathsf{K}.\tau)}\ \text{T-tabs} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall \alpha : \mathsf{K}.\tau) \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [\,] : (\tau[\tau'/\alpha])}\ \text{T-tapp}$$

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall i : \mathsf{S}.\tau)}\ \text{T-iabs} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall i : \mathsf{S}.\tau) \qquad \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [\,] : (\tau[I/i])}\ \text{T-iapp}$$

$$\frac{\Psi; \Theta; \Delta; \Omega, x : \tau; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash \mathsf{fix} x.e : \tau}\ \text{T-fix} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{ret}\, e : \mathbb{M}\, 0\, \tau}\ \text{T-ret}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \mathbb{M}\, n_1\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}\, n_2\, \tau_2 \qquad \Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2 : \mathbb{M}(I_1 + I_2)\, \tau_2}\ \text{T-bind}$$

$$\frac{\Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \uparrow^I : \mathbb{M}\, I\, \mathbf{1}}\ \text{T-tick}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : [I_1]\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}(I_1 + I_2)\, \tau_2 \qquad \Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{release}\, x = e_1\ \mathsf{in}\ \ e_2 : \mathbb{M}\, I_2\, \tau_2}\ \text{T-release}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{store}\, e : \mathbb{M}\, I\, ([I]\, \tau)}\ \text{T-store} \qquad \frac{\Psi; \Theta; \Delta, c; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.\ e : (c \Rightarrow \tau)}\ \text{T-CI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \Rightarrow \tau) \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e[\,] : \tau}\ \text{T-CE} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \& \tau)}\ \text{T-CAndI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \& \tau) \qquad \Psi; \Theta; \Delta, c; \Omega; \Gamma, x : \tau \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{clet}\, x = e\ \mathsf{in}\ e' : \tau'}\ \text{T-CAndE}$$

Figure 20: Typing rules for λ-amor

$$\frac{}{\Psi;\Theta;\Delta \vdash \tau <: \tau}\text{ sub-refl} \qquad \frac{\Psi;\Theta;\Delta \vdash \tau_1' <: \tau_1 \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \multimap \tau_2 <: \tau_1' \multimap \tau_2'}\text{ sub-arrow}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_1' \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \otimes \tau_2 <: \tau_1' \otimes \tau_2'}\text{ sub-tensor} \qquad \frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_1' \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \,\&\, \tau_2 <: \tau_1' \,\&\, \tau_2'}\text{ sub-with}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_1' \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \oplus \tau_2 <: \tau_1' \oplus \tau_2'}\text{ sub-sum} \qquad \frac{\Psi;\Theta;\Delta \vdash \tau <: \tau' \qquad \Theta;\Delta \models I' \leq I}{\Psi;\Theta;\Delta \vdash [I]\,\tau <: [I']\,\tau'}\text{ sub-potential}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau <: \tau' \qquad \Theta;\Delta \models I \leq I'}{\Psi;\Theta;\Delta \vdash \mathbb{M}\,I\,\tau <: \mathbb{M}\,I'\,\tau'}\text{ sub-monad} \qquad \frac{\Psi;\Theta;\Delta \vdash \tau <: \tau'}{\Psi;\Theta;\Delta \vdash !\tau <: !\tau'}\text{ sub-Exp}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau <: \tau'}{\Psi;\Theta;\Delta \vdash L^n\ \tau <: L^n\ \tau'}\text{ sub-list} \qquad \frac{\Psi;\Theta;\Delta, s \vdash \tau <: \tau'}{\Psi;\Theta;\Delta \vdash \exists s.\tau <: \exists s.\tau'}\text{ sub-exist}$$

$$\frac{\Psi,\alpha;\Theta;\Delta \vdash \tau_1 <: \tau_2}{\Psi;\Theta;\Delta \vdash \forall\alpha.\tau_1 <: \forall\alpha.\tau_2}\text{ sub-typePoly} \qquad \frac{\Psi;\Theta,i;\Delta \vdash \tau_1 <: \tau_2}{\Psi;\Theta;\Delta \vdash \forall i.\tau_1 <: \forall i.\tau_2}\text{ sub-indexPoly}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_2 \qquad \Theta;\Delta \models c_2 \implies c_1}{\Psi;\Theta;\Delta \vdash c_1 \Rightarrow \tau_1 <: c_2 \Rightarrow \tau_2}\text{ sub-constraint}$$

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_2 \qquad \Theta;\Delta \models c_1 \implies c_2}{\Psi;\Theta;\Delta \vdash c_1 \& \tau_1 <: c_2 \& \tau_2}\text{ sub-CAnd} \qquad \frac{\Psi;\Theta, i : \mathsf{S};\Delta \vdash \tau <: \tau'}{\Psi;\Theta;\Delta \vdash \lambda_t i : \mathsf{S}.\tau <: \lambda_t i : \mathsf{S}.\tau'}\text{ sub-familyAbs}$$

$$\frac{\Theta;\Delta \vdash I : \mathsf{S}}{\Psi;\Theta;\Delta \vdash \lambda_t i : \mathsf{S}.\tau\ I <: \tau[I/i]}\text{ sub-familyApp1} \qquad \frac{\Theta;\Delta \vdash I : \mathsf{S}}{\Psi;\Theta;\Delta \vdash \tau[I/i] <: \lambda_t i : \mathsf{S}.\tau\ I}\text{ sub-familyApp2}$$

Figure 21: Subtyping

$$\frac{}{\Psi;\Theta;\Delta \vdash \Omega \sqsubseteq .}\text{ sub-mBase} \qquad \frac{x : \tau' \in \Omega_1 \qquad \Psi;\Theta;\Delta \vdash \tau' <: \tau \qquad \Psi;\Theta;\Delta \vdash \Omega_1/x \sqsubseteq \Omega_2}{\Psi;\Theta;\Delta \vdash \Omega_1 \sqsubseteq \Omega_2, x : \tau}\text{ sub-mInd}$$

Figure 22: Ω Subtyping

$$\frac{}{\Psi;\Theta;\Delta \vdash \Gamma \sqsubseteq .}\text{ sub-lBase} \qquad \frac{x : \tau' \in \Gamma_1 \qquad \Psi;\Theta;\Delta \vdash \tau' <: \tau \qquad \Psi;\Theta;\Delta \vdash \Gamma_1/x \sqsubseteq \Gamma_2}{\Psi;\Theta;\Delta \vdash \Gamma_1 \sqsubseteq \Gamma_2, x : \tau}\text{ sub-lBase}$$

Figure 23: Γ Subtyping

$$\frac{}{\Theta, i : \mathsf{S};\Delta \vdash i : \mathsf{S}}\text{ S-var} \qquad \frac{}{\Theta;\Delta \vdash N : \mathbb{N}}\text{ S-nat} \qquad \frac{}{\Theta;\Delta \vdash R : \mathbb{R}^+}\text{ S-real} \qquad \frac{\Theta;\Delta \vdash i : \mathbb{N}}{\Theta;\Delta \vdash i : \mathbb{R}^+}\text{ S-real1}$$

$$\frac{\Theta;\Delta \vdash I_1 : \mathbb{N} \qquad \Theta;\Delta \vdash I_2 : \mathbb{N}}{\Theta;\Delta \vdash I_1 + I_2 : \mathbb{N}}\text{ S-add-Nat} \qquad \frac{\Theta;\Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta;\Delta \vdash I_2 : \mathbb{R}^+}{\Theta;\Delta \vdash I_1 + I_2 : \mathbb{R}^+}\text{ S-add-Real}$$

$$\frac{\Theta;\Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta;\Delta \vdash I_2 : \mathbb{R}^+ \qquad \Theta;\Delta \models I_1 \geq I_2}{\Theta;\Delta \vdash I_1 - I_2 : \mathbb{R}^+}\text{ S-minus-Real} \qquad \frac{\Theta, i : \mathsf{S};\Delta \vdash I : \mathsf{S}'}{\Theta;\Delta \vdash \lambda_s i.I : \mathsf{S} \to \mathsf{S}'}\text{ S-family}$$

Figure 24: Typing rules for sorts

$$\frac{}{\Psi; \Theta; \Delta \vdash \mathbf{1} : \mathit{Type}}\ \text{K-unit} \qquad \frac{}{\Psi; \Theta; \Delta \vdash \mathsf{b} : \mathit{Type}}\ \text{K-base} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta \vdash L^I \tau : \mathsf{K}}\ \text{K-List}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \multimap \tau_2 : \mathsf{K}}\ \text{K-arrow} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \otimes \tau_2 : \mathsf{K}}\ \text{K-tensor}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \,\&\, \tau_2 : \mathsf{K}}\ \text{K-with} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \oplus \tau_2 : \mathsf{K}}\ \text{K-or}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash !\tau : \mathsf{K}}\ \text{K-Exp} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta \vdash [I]\,\tau : \mathsf{K}}\ \text{K-lab}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta \vdash \mathbb{M}\, I\, \tau : \mathsf{K}}\ \text{K-monad} \qquad \frac{\Psi, \alpha : \mathsf{K}'; \Theta; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \forall \alpha.\tau : \mathsf{K}}\ \text{K-tabs} \qquad \frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \forall i.\tau : \mathsf{K}}\ \text{K-iabs}$$

$$\frac{\Psi; \Theta; \Delta, c \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash c \Rightarrow \tau : \mathsf{K}}\ \text{K-constraint} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta \vdash c \& \tau : \mathsf{K}}\ \text{K-consAnd}$$

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \lambda_t i.\tau : \mathsf{S} \to \mathsf{K}}\ \text{K-family} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{S} \to \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta \vdash \tau\ I : \mathsf{K}}\ \text{K-iapp}$$

Figure 25: Kind rules for types

A.3 Semantics

$\boxed{\text{Pure reduction, } e \Downarrow_t v}$ $\boxed{\text{Forcing reduction, } e \Downarrow_t^c v}$

$$\frac{e_1 \Downarrow_{t_1} v \qquad e_2 \Downarrow_{t_2} l}{e_1 :: e_2 \Downarrow_{t_1+t_2+1} v :: l}\ \text{E-cons} \qquad \frac{e_1 \Downarrow_{t_1} nil \qquad e_2 \Downarrow_{t_2} v}{\mathsf{match}\ e_1\ \mathsf{with}\ |nil \mapsto e_2\ |h :: t \mapsto e_3 \Downarrow_{t_1+t_2+1} v}\ \text{E-matchNil}$$

$$\frac{e_1 \Downarrow_{t_1} v_h :: l \qquad e_3[v_h/h][l/t] \Downarrow_{t_2} v}{\mathsf{match}\ e_1\ \mathsf{with}\ |nil \mapsto e_2\ |h :: t \mapsto e_3 \Downarrow_{t_1+t_2+1} v}\ \text{E-matchCons} \qquad \frac{e_1 \Downarrow_{t_1} v \qquad e_2[v/x] \Downarrow_{t_2} v'}{e_1; x.e_2 \Downarrow_{t_1+t_2+1} v'}\ \text{E-exist}$$

$$\frac{e_1 \Downarrow_{t_1} \lambda x.e' \qquad e'[e_2/x] \Downarrow_{t_2} v'}{e_1\ e_2 \Downarrow_{t_1+t_2+1} v'}\ \text{E-app} \qquad \frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2 \Downarrow_{t_2} v_2}{\langle\!\langle e_1, e_2\rangle\!\rangle \Downarrow_{t_1+t_2+1} \langle\!\langle v_1, v_2\rangle\!\rangle}\ \text{E-TI}$$

$$\frac{e \Downarrow_{t_1} \langle\!\langle v_1, v_2\rangle\!\rangle \qquad e'[v_1/x][v_2/y] \Downarrow_{t_2} v}{\mathsf{let}\langle\!\langle x, y\rangle\!\rangle = e\ \mathsf{in}\ e' \Downarrow_{t_1+t_2+1} v}\ \text{E-TE} \qquad \frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2 \Downarrow_{t_2} v_2}{\langle e_1, e_2\rangle \Downarrow_{t_1+t_2+1} \langle v_1, v_2\rangle}\ \text{E-WI} \qquad \frac{e \Downarrow_t \langle v_1, v_2\rangle}{\mathsf{fst}(e) \Downarrow_{t+1} v_1}\ \text{E-fst}$$

$$\frac{e \Downarrow_t \langle v_1, v_2\rangle}{\mathsf{fst}(e) \Downarrow_{t+1} v_2}\ \text{E-snd} \qquad \frac{e \Downarrow_t v}{\mathsf{inl}(e) \Downarrow_{t+1} \mathsf{inl}(v)}\ \text{E-inl} \qquad \frac{e \Downarrow_t v}{\mathsf{inr}(e) \Downarrow_{t+1} \mathsf{inr}(v)}\ \text{E-inr}$$

$$\frac{e \Downarrow_{t_1} \mathsf{inl}(v) \qquad e'[v/x] \Downarrow_{t_2} v'}{\mathsf{case}\ e, x.e', y.e'' \Downarrow_{t_1+t_2+1} \mathsf{inl}(v')}\ \text{E-case1} \qquad \frac{e \Downarrow_{t_1} \mathsf{inr}(v) \qquad e''[v/y] \Downarrow_{t_2} v''}{\mathsf{case}\ e, x.e', y.e'' \Downarrow_{t_1+t_2+1} \mathsf{inl}(v'')}\ \text{E-case2} \qquad \frac{}{!e \Downarrow_0 !e}\ \text{E-expI}$$

$$\frac{e \Downarrow_{t_1} !e'' \qquad e'[e''/x] \Downarrow_{t_2} v}{\mathsf{let}\,!\,x = e\ \mathsf{in}\ e' \Downarrow_{t_1+t_2+1} v}\ \text{E-expE} \qquad \frac{e[\mathsf{fix}x.e/x] \Downarrow_t v}{\mathsf{fix}x.e \Downarrow_{t+1} v}\ \text{E-fix}$$

$$\frac{v \in \{(), x, nil, \lambda y.e, \Lambda.e, \mathsf{ret}\,e, \mathsf{bind}\,x = e_1\ \mathsf{in}\ e_2, \uparrow^\kappa, \mathsf{release}\,x = e_1\ \mathsf{in}\ e_2, \mathsf{store}\,e\}}{v \Downarrow_0 v}\ \text{E-val}$$

$$\frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_2} v}{e\,[\,] \Downarrow_{t_1+t_2+1} v}\ \text{E-tapp} \qquad \frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_2} v}{e\,[\,] \Downarrow_{t_1+t_2+1} v}\ \text{E-iapp} \qquad \frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_1} v}{e\,[\,] \Downarrow_{t_1+t_2+1} v}\ \text{E-CE}$$

$$\frac{e_1 \Downarrow_{t_1} v \qquad e_2[v/x] \Downarrow_{t_2} v'}{\mathsf{clet}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+1} v'}\ \text{E-CandE} \qquad \frac{e \Downarrow_t v}{\mathsf{ret}\,e \Downarrow_{t+1}^0 v}\ \text{E-return}$$

$$\frac{e_1 \Downarrow_{t_1} v_1 \qquad v_1 \Downarrow_{t_2}^{c_1} v_1' \qquad e_2[v_1'/x] \Downarrow_{t_3} v_2 \qquad v_2 \Downarrow_{t_4}^{c_2} v_2'}{\mathsf{bind}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+t_3+t_4+1}^{c_1+c_2} v_2'}\ \text{E-bind} \qquad \frac{}{\uparrow^\kappa \Downarrow_1^\kappa ()}\ \text{E-tick}$$

$$\frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2[v_1/x] \Downarrow_{t_2} v_2 \qquad v_2 \Downarrow_{t_3}^c v_2'}{\mathsf{release}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+t_3+1}^c v_2'}\ \text{E-release} \qquad \frac{e \Downarrow_t v}{\mathsf{store}\,e \Downarrow_{t+1}^0 v}\ \text{E-store}$$

Figure 26: Evaluation rules: pure and forcing

A.4 Model

Definition 15 (Value and expression relation).

$$
\begin{array}{lll}
[\![\mathbf{1}]\!] & \triangleq & \{(p,T,())\} \\
[\![\mathsf{b}]\!] & \triangleq & \{(p,T,v) \mid v \in [\![\mathsf{b}]\!]\} \\
[\![L^0\tau]\!] & \triangleq & \{(p,T,nil)\} \\
[\![L^{s+1}\tau]\!] & \triangleq & \{(p,T,v::l) | \exists p_1,p_2.p_1+p_2 \leq p \wedge (p_1,T,v) \in [\![\tau]\!] \wedge (p_2,T,l) \in [\![L^s\tau]\!]\} \\
[\![\tau_1 \otimes \tau_2]\!] & \triangleq & \{(p,T,\langle\!\langle v_1,v_2\rangle\!\rangle) \mid \exists p_1,p_2.p_1+p_2 \leq p \wedge (p_1,T,v_1) \in [\![\tau_1]\!] \wedge (p_2,T,v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \,\&\, \tau_2]\!] & \triangleq & \{(p,T,\langle v_1,v_2\rangle) \mid (p,T,v_1) \in [\![\tau_1]\!] \wedge (p,T,v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \oplus \tau_2]\!] & \triangleq & \{(p,T,\mathsf{inl}(v)) \mid (p,T,v) \in [\![\tau_1]\!]\} \cup \{(p,T,\mathsf{inr}(v)) \mid (p,T,v) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \multimap \tau_2]\!] & \triangleq & \{(p,T,\lambda x.e) \mid \forall p',e',T'{<}T\,.(p',T',e') \in [\![\tau_1]\!]_{\mathcal{E}} \implies (p+p',T',e[e'/x]) \in [\![\tau_2]\!]_{\mathcal{E}}\} \\
[\![!\tau]\!] & \triangleq & \{(p,T,!e) \mid (0,T,e) \in [\![\tau]\!]_{\mathcal{E}}\} \\
[\![[n]\,\tau]\!] & \triangleq & \{(p,T,v) \mid \exists p'.p'+n \leq p \wedge (p',T,v) \in [\![\tau]\!]\}\} \\
[\![\mathbb{M}\,n\,\tau]\!] & \triangleq & \{(p,T,v) \mid \forall n',v',T'{<}T\,.v \Downarrow_{T'}^{n'} v' \implies \exists p'.n'+p' \leq p+n \wedge (p',T-T',v') \in [\![\tau]\!]\} \\
[\![\forall\alpha.\tau]\!] & \triangleq & \{(p,T,\Lambda.e) \mid \forall\tau',T'{<}T\,.(p,T',e) \in [\![\tau[\tau'/\alpha]]\!]_{\mathcal{E}}\} \\
[\![\forall i.\tau]\!] & \triangleq & \{(p,T,\Lambda.e) \mid \forall I,T'{<}T\,.(p,T',e) \in [\![\tau[I/i]]\!]_{\mathcal{E}}\} \\
[\![c \Rightarrow \tau]\!] & \triangleq & \{(p,T,\Lambda.e) \mid .\models c \implies (p,T,e) \in [\![\tau]\!]_{\mathcal{E}}\} \\
[\![c\&\tau]\!] & \triangleq & \{(p,T,v) \mid .\models c \wedge (p,T,v) \in [\![\tau]\!]\} \\
[\![\exists s.\tau]\!] & \triangleq & \{(p,T,v) \mid \exists s'.(p,T,v) \in [\![\tau[s'/s]]\!]\} \\
[\![\lambda_t i.\tau]\!] & \triangleq & f \textit{ where } \forall I.\,f\ I = [\![\tau[I/i]]\!] \\
[\![\tau\ I]\!] & \triangleq & [\![\tau]\!]\ I \\
\\
[\![\tau]\!]_{\mathcal{E}} & \triangleq & \{(p,T,e) \mid \forall\, T'{<}T, v.e \Downarrow_{T'} v \implies (p,T-T',v) \in [\![\tau]\!]\}
\end{array}
$$

Definition 16 (Interpretation of typing contexts).

$$
\begin{array}{lll}
[\![\Gamma]\!]_{\mathcal{E}} & = & \{(p,T,\gamma) \mid \exists f : \mathcal{V}ars \to \mathcal{P}ots. \\
 & & (\forall x \in dom(\Gamma).\,(f(x),T,\gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x\in dom(\Gamma)} f(x) \leq p)\} \\
[\![\Omega]\!]_{\mathcal{E}} & = & \{(0,T,\delta) \mid (\forall x \in dom(\Omega).(0,T,\delta(x)) \in [\![\tau]\!]_{\mathcal{E}})\}
\end{array}
$$

Definition 17 (Type and index substitutions). $\sigma : TypeVar \to Type,\ \iota : IndexVar \to Index$

Lemma 18 (Value monotonicity lemma). $\forall p,p',v,\tau.$
$(p,T,v) \in [\![\tau]\!] \wedge p \leq p' \wedge T' {\leq} T \implies (p',T',v) \in [\![\tau]\!]$

Proof. Proof by induction on τ ☐

Lemma 19 (Expression monotonicity lemma). $\forall p,p',v,\tau.$
$(p,T,e) \in [\![\tau]\!]_{\mathcal{E}} \wedge p \leq p' \wedge T' {\leq} T \implies (p',T',e) \in [\![\tau]\!]_{\mathcal{E}}$

Proof. From Definition 15 and Lemma 61 ☐

Theorem 20 (Fundamental theorem). $\forall\Theta,\Omega,\Gamma,e,\tau,T,p_l,\gamma,\delta,\sigma,\iota.$
$\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e : \tau \wedge (p_l,T,\gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}} \wedge (0,T,\delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}} \implies (p_l,T,e\ \gamma\delta) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}.$

Proof. Proof by induction on the typing judgment

1. T-var1:

$$\frac{}{\Psi;\Theta;\Delta;\Omega;\Gamma,x:\tau \vdash x:\tau}\ \text{T-var1}$$

Given: $(p_l,T,\gamma) \in [\![\Gamma,x:\tau\ \sigma\iota]\!]_{\mathcal{E}}$ and $(0,T,\delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l,T,x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $(p_l,T,\gamma) \in [\![\Gamma,x:\tau\ \sigma\iota]\!]_{\mathcal{E}}$ therefore from Definition 16 we know that $\exists f.(f(x),T,\gamma(x)) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ where $f(x) \leq p_l$

Therefore from Lemma 62 we get $(p_l,T,x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

2. T-var2:

$$\frac{}{\Psi;\Theta;\Delta;\Omega,x:\tau;\Gamma \vdash x:\tau}\ \text{T-var2}$$

Given: $(p_l,T,\gamma) \in [\![\Gamma,\sigma\iota]\!]_{\mathcal{E}}$ and $(0,T,\delta) \in [\![(\Omega,x:\tau)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l,T,x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $(0, T, \delta) \in [\![(\Omega, x : \tau)\ \sigma\iota]\!]_{\mathcal{E}}$ therefore from Definition 16 we know that $(0, T, \delta(x)) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Therefore from Lemma 62 we get $(p_l, T, x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

3. T-unit:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash () : \mathbf{1}}\ \text{T-unit}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, ()\ \delta\gamma) \in [\![\mathbf{1}\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall\, T' {<} T, v'.() \Downarrow_{T'} v' \implies (p_l, T - T', v') \in [\![\mathbf{1}]\!]$

This means given some $T' {<} T, v'$ s.t $() \Downarrow_{T'} v'$ it suffices to prove that
$(p_l, T - T', v') \in [\![\mathbf{1}]\!]$

From (E-val) we know that $T' {=}\ 0$ and $v' = ()$, therefore it suffices to prove that
$(p_l, T, ()) \in [\![\mathbf{1}]\!]$
We get this directly from Definition 15

4. T-base:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash c : \mathsf{b}}\ \text{T-base}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, c) \in [\![\mathsf{b}]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall\, T' {<} T, v'.c \Downarrow_{T'} v' \implies (p_l, T - T', v') \in [\![\mathbf{1}]\!]$

This means given some $T' {<} T, v'$ s.t $c \Downarrow_{T'} v'$ it suffices to prove that
$(p_l, T - T', v') \in [\![\mathbf{1}]\!]$

From (E-val) we know that $T' {=}\ 0$ and $v' = c$, therefore it suffices to prove that
$(p_l, T, c) \in [\![\mathsf{b}]\!]$
We get this directly from Definition 15

5. T-nil:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathit{nil} : L^0\ \tau}\ \text{T-nil}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathit{nil}\ \delta\gamma) \in [\![L^0\ \tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall\, T' {<} T, v'.\mathit{nil} \Downarrow_{T'} v' \implies (p_l, T - T', v') \in [\![L^0\ \tau\ \sigma\iota]\!]$

This means given some $T' {<} T, v'$ s.t $\mathit{nil} \Downarrow_{T'} v'$ it suffices to prove that
$(p_l, T - T', v') \in [\![L^0\ \tau\ \sigma\iota]\!]$

From (E-val) we know that $T' {=}\ 0$ and $v' = \mathit{nil}$, therefore it suffices to prove that
$(p_l, T, \mathit{nil}) \in [\![L^0\ \tau\ \sigma\iota]\!]$
We get this directly from Definition 15

6. T-cons:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : L^n \tau \qquad \Theta \vdash n : \mathbb{N}}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e_1 :: e_2 : L^{n+1} \tau}\ \text{T-cons}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (e_1 :: e_2)\ \delta\gamma) \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t {<} T, v'.(e_1 :: e_2)\ \delta\gamma \Downarrow_t v' \implies (p_l, T - t, v') \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]$

This means given some $t {<} T, v'$ s.t $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v'$, it suffices to prove that
$(p_l, T - t, v') \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]$

From (E-cons) we know that $\exists v_f, l.v' = v_f :: l$

Therefore from Definition 15 it suffices to prove that
$\exists p_1, p_2.p_1 + p_2 \leq p_l \wedge (p_1, T - t, v_f) \in [\![\tau\ \sigma\iota]\!] \wedge (p_2, T - t, l) \in [\![L^n \tau\ \sigma\iota]\!]$ (F-C0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>:
$(p_{l1}, T, e_1\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t1 < T.e_1\ \delta\gamma \Downarrow_{t1} v_f \implies (p_{l1}, T - t1, v_f) \in [\![\tau]\!]$

Since we are given that $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v_f :: l$ therefore fom E-cons we also know that $\exists t1 < t.\ e_1\ \delta\gamma \Downarrow_{t1} v_f$
Since $t1 < t < T$, therefore we have $(p_{l1}, T - t1, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-C1)

<u>IH2</u>:
$(p_{l2}, T, e_2\ \delta\gamma) \in [\![L^n \tau\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t2 < T .e_2\ \delta\gamma \Downarrow_{t2} l \implies (p_{l2}, T - t2, l \in [\![L^n \tau\ \sigma\iota]\!]$

Since we are given that $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v_f :: l$ therefore fom E-cons we also know that $\exists t2 < t - t1.$ $e_2\ \delta\gamma \Downarrow_{t2} l$
Since $t2 < t - t1 < t < T$, therefore we have
$(p_{l2}, T - t2, l) \in [\![L^n \tau\ \sigma\iota]\!]$ (F-C2)

In order to prove (F-C0) we choose p_1 as p_{l1} and p_2 as p_{l2} and it suffices to prove that
$(p_{l1}, T - t, v) \in [\![\tau\ \sigma\iota]\!] \wedge (p_{l2}, T - t, l) \in [\![L^n \tau\ \sigma\iota]\!]$

Since $t = t_1 + t_2 + 1$ therefore from (F-C1) and Lemma 61 we get $(p_{l1}, T - t, v) \in [\![\tau\ \sigma\iota]\!]$
Similarly from (F-C2) and Lemma 61 we also get $(p_{l2}, T - t, l) \in [\![L^n \tau\ \sigma\iota]\!]$

7. T-match:

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : L^n\ \tau \\ \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_1 : \tau' \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, h : \tau, t : L^i \tau \vdash e_2 : \tau' \qquad \Theta \vdash n : \mathbb{N} \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}\end{array}}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau'} \text{ T-match}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f.(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-M0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![L^n \tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t' < T .e\ \delta\gamma \Downarrow_{t'} v_1 \implies (p_{l1}, T - t', v_1) \in [\![L^n \tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t' < t, v_1.e\ \delta\gamma \Downarrow_{t'} v_1$.
Since $t' < t < T$, therefore we have $(p_{l1}, T - t', v_1) \in [\![L^n \tau\ \sigma\iota]\!]$
2 cases arise:

(a) $v_1 = nil$:
In this case we know that $n = 0$ therefore
<u>IH2</u>
$(p_{l2}, T, e_1\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T .e_1\ \delta\gamma \Downarrow_{t_1} v_f \implies (p_{l2}, T - t_1, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t_1 < t.\ e_1\ \delta\gamma \Downarrow_{t_1} v_f$.

Since $t_1 < t < T$ therefore we have
$(p_{l2}, T - t_1, v_f) \in [\![\tau' \ \sigma\iota]\!]$

And from Lemma 61 we get
$(p_{l2} + p_{l1}, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]_{\mathcal{E}}$

And finally since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

(b) $v_1 = v :: l$:
In this case we know that $n > 0$
<u>IH2</u>
$(p_{l2} + p_{l1}, T, e_2 \ \delta\gamma') \in [\![\tau' \ \sigma\iota']\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{h \mapsto v\} \cup \{t \mapsto l\}$
$\iota' = \iota \cup \{i \mapsto n-1\}$

This means from Definition 15 we have
$\forall t_2 < T . e_2 \ \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_{l1}, T - t_2, v_f) \in [\![\tau' \ \sigma\iota']\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t_2 < t.\ e_2\ \delta\gamma' \Downarrow_{t_2} v_f$.
Since $t_2 < t < T$ therefore we have
$(p_{l2} + p_{l1}, T - t_2, v_f) \in [\![\tau' \ \sigma\iota']\!]$

From Lemma 61 we get
$(p_{l2} + p_{l1}, T - t, v_f) \in [\![\tau' \ \sigma\iota']\!]$

And finally since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T - t, v_f) \in [\![\tau' \ \sigma\iota']\!]_{\mathcal{E}}$

And finally since we have $\Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}$ therefore we also have
$(p_l, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

8. T-existI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau[n/s] \qquad \Theta \vdash n : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \exists s : \mathsf{S}.\tau} \text{ T-existI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma \ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega \ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e \ \delta\gamma) \in [\![\exists s.\tau \ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . e \ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f \ \delta\gamma) \in [\![\exists s.\tau \ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $e \ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\exists s.\tau \ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\exists s'.(p_l, T - t, v_f) \in [\![\tau[s'/s] \ \sigma\iota]\!]$ (F-E0)

<u>IH</u>: $(p_l, T, e \ \delta\gamma) \in [\![\tau[n/s] \ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t' < T . e \ \delta\gamma \Downarrow_{t'} v_f \implies (p_l, T - t', v_f) \in [\![\tau[n/s] \ \sigma\iota]\!]$

Since we are given that $e \ \delta\gamma \Downarrow_t v_f$ therefore we get
$(p_l, T - t, v_f) \in [\![\tau[n/s] \ \sigma\iota]\!]$ (F-E1)

To prove (F-E0) we choose s' as n and we get the desired from (F-E1)

9. T-existsE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : \exists s.\tau \qquad \Psi; \Theta, s; \Delta; \Omega; \Gamma_2, x : \tau \vdash e' : \tau' \qquad \Theta \vdash \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e; x.e' : \tau'} \text{ T-existE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2) \ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega) \ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (e; x.e') \ \delta\gamma) \in [\![\tau' \ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (e; x.e') \ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]$

This means given soem $t < T, v_f$ s.t $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-EE0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![\exists s.\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{l1}, T - t_1, v_1) \in [\![\exists s.\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we know that $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-existE we know that $\exists t_1 < t, v_1 . e\ \delta\gamma \Downarrow_{t_1} v_1$.
Therefore we have
$(p_{l1}, T - t_1, v_1) \in [\![\exists s.\tau\ \sigma\iota]\!]$

Therefore from Definition 15 we have
$\exists s'.(p_{l1}, T - t_1, v_1) \in [\![\tau[s'/s]\ \sigma\iota]\!]$ (F-EE1)

<u>IH2</u>
$(p_{l1} + p_{l2}, T, e'\ \delta'\gamma) \in [\![\tau'\ \sigma\iota']\!]_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto e_1\}$ and $\iota' = \iota \cup \{s \mapsto s'\}$

This means from Definition 15 we have
$\forall t_2 < T . e'\ \delta'\gamma \Downarrow_{t_2} v_f \implies (p_{l1} + p_{l2}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since we know that $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-existE we know that $\exists t_2 < t.\ e'\ \delta'\gamma \Downarrow_{t_2} v_f$.
Since $t_2 < t < T$ therefore we have
$(p_{l1} + p_{l2}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

From Lemma 61 we get
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota']\!]$

And finally since we have $\Psi; \Theta \vdash \tau'$ therefore we also have
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$
And we are done

10. T-lam:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \lambda x.e : (\tau_1 \multimap \tau_2)} \text{ T-lam}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\lambda x.e)\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f .(\lambda x.e)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\lambda x.e)\ \delta\gamma \Downarrow_t v_f$. From E-val we know that $t = 0$ and $v_f = (\lambda x.e)\ \delta\gamma$
Therefore it suffices to prove that
$(p_l, T, (\lambda x.e)\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\forall p', e', T' < T . (p', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p_l + p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $p', e', T' < T$ s.t $(p', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that
$(p_l + p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-L1)

From IH we know that
$(p_l + p', T, e\ \delta\gamma') \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{x \mapsto e'\}$

Therefore from Lemma 62 we get the desired

11. T-app:

$$\frac{\Psi;\Theta;\Delta;\Omega_1;\Gamma_1 \vdash e_1 : (\tau_1 \multimap \tau_2) \qquad \Psi;\Theta;\Delta;\Omega_2;\Gamma_2 \vdash e_2 : \tau_1}{\Psi;\Theta;\Delta;\Omega_1 \oplus \Omega_2;\Gamma_1 \oplus \Gamma_2 \vdash e_1\ e_2 : \tau_2} \text{ T-app}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e_1\ e_2\ \delta\gamma) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f.(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau_2\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-A0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e_1\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T .e_1 \Downarrow_{t_1} \lambda x.e \implies (p_{l1}, T - t_1, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$
Since we know that $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-app we know that $\exists t_1 < t.e_1 \Downarrow_{t_1} \lambda x.e$, therefore we have
$(p_{l1}, T - t_1, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

Therefore from Definition 15 we have
$\forall p', e_1, T_1 < T - t_1.(p', T_1, e'_1) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p_{l1} + p', T_1, e[e'_1/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-A1)

<u>IH2</u>
$(p_{l2}, T - t_1 - 1, e_2\ \delta\gamma) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ (F-A2)

Instantiating (F-A1) with p_{l2}, $e_2\ \delta\gamma$ and $T - t_1 - 1$ we get
$(p_{l1} + p_{l2}, T - t_1 - 1, e[e_2\ \delta\gamma/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_2 < T - t_1 - 1.e[e_2\ \delta\gamma/x] \Downarrow_{t_2} v_f \implies (p_{l1} + p_{l2}, T - t_1 - 1 - t_2, v_f) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-app we know that $\exists t_2.e[e_2\ \delta\gamma/x] \Downarrow_{t_2} v_f$ where $t_2 = t - t_1 - 1$, therefore we have
$(p_{l1} + p_{l2}, T - t_1 - t_2 - 1, v_f) \in [\![\tau_2\ \sigma\iota]\!]$ where $p_{l1} + p_{l2} = p_l$
Since from E-app we know that $t = t_1 + t_2 + 1$, therefore we have proved (F-A0)

12. T-sub:

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e : \tau \qquad \Theta \vdash \tau <: \tau'}{\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e : \tau'} \text{ T-sub}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
<u>IH</u> $(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH and Lemma 22

13. T-weaken:

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e : \tau \qquad \Psi;\Theta;\Delta \models \Gamma' <: \Gamma \qquad \Psi;\Theta;\Delta \models \Omega' <: \Omega}{\Psi;\Theta;\Delta;\Omega';\Gamma' \vdash e : \tau} \text{ T-weaken}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma')\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega')\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $(p_l, T, \gamma) \in [\![(\Gamma')\sigma\iota]\!]_{\mathcal{E}}$ therefore from Lemma 23 we also have $(p_l, T, \gamma) \in [\![(\Gamma)\sigma\iota]\!]_{\mathcal{E}}$

Similarly since we are given that $(0, T, \delta) \in [\![(\Omega')\sigma\iota]\!]_{\mathcal{E}}$ therefore from Lemma 25 we also have $(0, T, \delta) \in [\![(\Omega)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH</u>:
$(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH

14. T-tensorI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \langle\langle e_1, e_2 \rangle\rangle : (\tau_1 \otimes \tau_2)} \text{T-tensorI}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T . \langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle \implies (p_l, T - t, \langle\langle v_{f1}, v_{f2} \rangle\rangle \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$

This means given some $t < T$ s.t $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ it suffices to prove that
$(p_l, T - t, \langle\langle v_{f1}, v_{f2} \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$ (F-TI0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>:
$(p_{l1}, T, e_1\ \delta\gamma) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_1 < T . e_1\ \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_{l1}, T - t_1, v_{f1}) \in [\![\tau_1\ \sigma\iota]\!]$

Since we are given that $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ therefore fom E-TI we know that $\exists t_1 < t. e_1\ \delta\gamma \Downarrow_{t_1} v_{f1}$
Hence we have $(p_{l1}, T - t_1, v_{f1}) \in [\![\tau_1\ \sigma\iota]\!]$ (F-TI1)

<u>IH2</u>:
$(p_{l2}, T, e_2\ \delta\gamma) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_2 < T . e_2\ \delta\gamma \Downarrow_{t_2} v_{f2} \implies (p_{l2}, T - t_2, v_{f2} \in [\![\tau_2\ \sigma\iota]\!]$

Since we are given that $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ therefore fom E-TI we also know that $\exists t_2 < t. e_2\ \delta\gamma \Downarrow_{t_2} v_{f2}$ s.t
Since $t_2 < t < T$ therefore we have
$(p_{l2}, T - t_2, v_{f2}) \in [\![\tau_2\ \sigma\iota]\!]$ (F-TI2)

Applying Lemma 61 on (F-TI1) and (F-TI2) and by using Definition 15 we get the desired.

15. T-tensorE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \otimes \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1, y : \tau_2 \vdash e' : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e' : \tau} \text{T-tensorE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-TE0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e\ \delta\gamma \Downarrow_{t_1} \langle\langle v_1, v_2 \rangle\rangle\ \delta\gamma \implies (p_{l1}, T - t_1, \langle\langle v_1, v_2 \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$

Since we know that $(\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-TE we know that $\exists t_1 < t, v_1, v_2 . e\ \delta\gamma \Downarrow_{t_1} \langle\langle v_1, v_2 \rangle\rangle$. Therefore we have
$(p_{l1}, T - t_1, \langle\langle v_1, v_2 \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$
From Definition 15 we know that
$\exists p_1, p_2 . p_1 + p_2 \le p_{l1} \wedge (p_1, T, v_1) \in [\![\tau_1\ \sigma\iota]\!] \wedge (p_2, T, v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-TE1)

<u>IH2</u>
$(p_{l2} + p_1 + p_2, T, e'\ \delta\gamma') \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

where
$\gamma' = \gamma \cup \{x \mapsto v_1\} \cup \{y \mapsto v_2\}$

This means from Definition 15 we have
$\forall t_2 < T . e' \; \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_1 + p_2, T - t_2, v_f) \in [\![\tau \; \sigma\iota]\!]$

Since we know that $(\mathsf{let}\langle\!\langle x, y\rangle\!\rangle = e \; \mathsf{in} \; e') \; \delta\gamma \Downarrow_t v_f$ therefore from E-TE we know that $\exists t_2 < t . e' \; \delta\gamma' \Downarrow_{t_2} v_f$.
Therefore we have
$(p_{l2} + p_1 + p_2, T - t_2, v_f) \in [\![\tau \; \sigma\iota]\!]$

From Lemma 61 we get
$(p_l, T - t, v_f) \in [\![\tau \; \sigma\iota]\!]_{\mathcal{E}}$
And we are done

16. T-withI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \langle e_1, e_2\rangle : (\tau_1 \;\&\; \tau_2)} \text{ T-withI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega \; \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \langle e_1, e_2\rangle \; \delta\gamma) \in [\![(\tau_1 \;\&\; \tau_2) \; \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T . \langle e_1, e_2\rangle \; \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2}\rangle \implies (p_l, T - t, \langle v_{f1}, v_{f2}\rangle \in [\![(\tau_1 \;\&\; \tau_2) \; \sigma\iota]\!]$

This means given $\langle e_1, e_2\rangle \; \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2}\rangle$ it suffices to prove that
$(p_l, T - t, \langle v_{f1}, v_{f2}\rangle) \in [\![(\tau_1 \;\&\; \tau_2) \; \sigma\iota]\!]$ (F-WI0)

<u>IH1</u>:
$(p_l, T, e_1 \; \delta\gamma) \in [\![\tau_1 \; \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_1 < T . e_1 \; \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_l, T - t_1, v_{f1}) \in [\![\tau_1 \; \sigma\iota]\!]$

Since we are given that $\langle e_1, e_2\rangle \; \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2}\rangle$ therefore fom E-WI we know that $\exists t_1 < t . e_1 \; \delta\gamma \Downarrow_{t_1} v_{f1}$
Since $t_1 < t < T$, therefore we have
$(p_l, T - t_1, v_{f1}) \in [\![\tau_1 \; \sigma\iota]\!]$ (F-WI1)

<u>IH2</u>:
$(p_l, T, e_2 \; \delta\gamma) \in [\![\tau_2 \; \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_2 < T . e_2 \; \delta\gamma \Downarrow_{t_2} v_{f2} \implies (p_l, T - t_2, v_{f2} \in [\![\tau_2 \; \sigma\iota]\!]$

Since we are given that $\langle e_1, e_2\rangle \; \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2}\rangle$ therefore fom E-WI we also know that $\exists t_2 < t . e_2 \; \delta\gamma \Downarrow_{t_2} v_{f2}$
Since $t_2 < t < T$, therefore we have
$(p_l, T - t_2, v_{f2}) \in [\![\tau_2 \; \sigma\iota]\!]$ (F-WI2)

Applying Lemma 61 on (F-W1) and (F-W2) we get the desired.

17. T-fst:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1 \;\&\; \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{fst}(e) : \tau_1} \text{ T-fst}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma) \; \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega \; \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{fst}(e)) \; \delta\gamma) \in [\![\tau_1 \; \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{fst}(e)) \; \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau_1 \; \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{fst}(e)) \; \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau_1 \; \sigma\iota]\!]$ (F-F0)

<u>IH</u>
$(p_l, T, e \; \delta\gamma) \in [\![(\tau_1 \;\&\; \tau_2) \; \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e \; \delta\gamma \Downarrow_{t_1} \langle v_1, v_2\rangle \; \delta\gamma \implies (p_l, T - t_1, \langle v_1, v_2\rangle) \in [\![(\tau_1 \;\&\; \tau_2) \; \sigma\iota]\!]$

Since we know that $(\mathsf{fst}(e)) \; \delta\gamma \Downarrow_t v_f$ therefore from E-fst we know that $\exists t_1 < t . v_1, v_2 . e \; \delta\gamma \Downarrow_{t_1} \langle v_1, v_2\rangle$.
Since $t_1 < t < T$, therefore we have

$(p_l, T - t_1, \langle v_1, v_2 \rangle) \in [\![(\tau_1 \,\&\, \tau_2)\ \sigma\iota]\!]$

From Definition 15 we know that
$(p_l, T - t_1, v_1) \in [\![\tau_1\ \sigma\iota]\!]$

Finally using Lemma 61 we also have
$(p_l, T - t, v_1) \in [\![\tau_1\ \sigma\iota]\!]$
Since from E-fst we know that $v_f = v_1$, therefore we are done.

18. T-snd:
Similar reasoning as in T-fst case above.

19. T-inl:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inl}(e) : \tau_1 \oplus \tau_2} \text{ T-inl}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathsf{inl}(e)\ \delta\gamma) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T . \mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v) \implies (p_l, T - t, \mathsf{inl}(v) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]$

This means given some $t < T$ s.t $\mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v)$ it suffices to prove that
$(p_l, T - t, \mathsf{inl}(v)) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]$ (F-IL0)

<u>IH</u>:
$(p_l, T, e_1\ \delta\gamma) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_1 < T . e_1\ \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_l, T - t_1, v_{f1}) \in [\![\tau_1\ \sigma\iota]\!]$

Since we are given that $\mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v)$ therefore fom E-inl we know that $\exists t_1 < t . e\ \delta\gamma \Downarrow_{t_1} v$
Hence we have $(p_l, T - t_1, v) \in [\![\tau_1\ \sigma\iota]\!]$
From Lemma 61 we get $(p_l, T - t, v) \in [\![\tau_1\ \sigma\iota]\!]$

And finally from Definition 15 we get (F-IL0)

20. T-inr:
Similar reasoning as in T-inr case above.

21. T-case:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \oplus \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, y : \tau_2 \vdash e_2 : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{case}\ e, x.e_1, y.e_2 : \tau} \text{ T-case}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-C0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2} . p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t' < T . e\ \delta\gamma \Downarrow_{t'} v_1\ \delta\gamma \implies (p_{l1}, T - t', v_1) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]$

Since we know that $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-case we know that $\exists t' < t, v_1 . e\ \delta\gamma \Downarrow_{t'} v_1$.
Since $t' < t < T$, therefore we have
$(p_{l1}, T - t', v_1) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]$
2 cases arise:

(a) $v_1 = \mathsf{inl}(v)$:
<u>IH2</u>
$(p_{l2} + p_{l1}, T - t', e_1\ \delta\gamma') \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
where

$\gamma' = \gamma \cup \{x \mapsto v\}$

This means from Definition 15 we have
$\forall t_1 <T -t'.e_1\ \delta\gamma' \Downarrow_{t_1} v_f \implies (p_{l2}, T -t' - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-case we know that $\exists t_1.e_1\ \delta\gamma' \Downarrow v_f$ where $t_1 = t - t' - 1$.
Since $t_1 = t - t' - 1 <T -t'$ therefore we have
$(p_{l2}, T -t' - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

From Lemma 61 we get
$(p_{l2} + p_{l1}, T -t, v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

And finally since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T -t, v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

(b) $v_1 = \mathsf{inr}(v)$:
Similar reasoning as in the inl case above.

22. T-ExpI:

$$\frac{\Psi; \Theta; \Delta; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash !e : !\tau} \text{ T-ExpI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, !e\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t <T .(!e)\ \delta\gamma \Downarrow_t (!e)\ \delta\gamma \implies (p_l, T -t, (!e)\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]$

This means given some $t <T$ s.t $(!e)\ \delta\gamma \Downarrow_t (!e)\ \delta\gamma$ it suffices to prove that
$(p_l, T -t, (!e)\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$(0, T -t, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

<u>IH</u>: $(0, T -t, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH

23. T-ExpE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : !\tau \qquad \Psi; \Theta; \Delta; \Omega, x : \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\,!\,x = e\ \mathsf{in}\ e' : \tau'} \text{ T-ExpE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t <T, v_f.(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f \implies (p_l, T -t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t <T, v_f$ s.t $(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T -t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-E0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 <T .e\ \delta\gamma \Downarrow_{t_1} !e_1\ \delta\gamma \implies (p_{l1}, T -t_1, !e_1\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f$ therefore from (E-ExpE) we know that $\exists t_1 < t, e_1.e\ \delta\gamma \Downarrow_{t_1} !e_1\ \delta\gamma$.
Since $t_1 < t <T$, therefore we have
$(p_{l1}, T -t_1, !e_1\ \delta\gamma) \in [\![!\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$(0, T -t_1, e_1\ \delta\gamma) \in [\![\tau]\!]_{\mathcal{E}}$ (F-E1)

<u>IH2</u>

$(p_{l2}, T - t_1, e' \ \delta'\gamma) \in [\![\tau' \ \sigma\iota]\!]_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto e_1\}$

This means from Definition 15 we have
$\forall t_2 < T - t_1 . e' \ \delta'\gamma \Downarrow_{t_2} v_f \implies (p_{l2}, T - t_1 - t_2, v_f) \in [\![\tau' \ \sigma\iota]\!]$

Since we know that $(\mathsf{let}\,!\, x = e \ \mathsf{in} \ e') \ \delta\gamma \Downarrow_t v_f$ therefore from (E-ExpE) we know that $\exists t_2 . e' \ \delta'\gamma \Downarrow v_f$ where $t_2 = t - t_1 - 1$.
Since $t_2 = t - t_1 - 1 < T - t_1$, therefore we have
$(p_{l2}, T - t_1 - t_2, v_f) \in [\![\tau' \ \sigma\iota]\!]$

From Lemma 62 we get
$(p_{l2} + p_{l1}, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]$

And finally since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T - t, v_f) \in [\![\tau' \ \sigma\iota]\!]$
And we are done

24. T-tabs:

$$\frac{\Psi, \alpha : \mathsf{K}; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall \alpha : \mathsf{K}.\tau)} \text{ T-tabs}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega \ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\Lambda.e) \ \delta\gamma) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\Lambda.e) \ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\Lambda.e) \ \delta\gamma \Downarrow_t v_f$. From E-val we know that $t = 0$ and $v_f = (\Lambda.e) \ \delta\gamma$
Therefore it suffices to prove that
$(p_l, T, (\Lambda.e) \ \delta\gamma) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\forall \tau', T' < T . (p_l, T', e) \in [\![\tau[\tau'/\alpha] \ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $\tau', T' < T$ it suffices to prove that
$(p_l, T', e) \in [\![\tau[\tau'/\alpha] \ \sigma\iota]\!]_{\mathcal{E}}$ (F-TAB0)

From IH we know that
$(p_l, T, e \ \delta\gamma) \in [\![\tau \ \sigma'\iota]\!]_{\mathcal{E}}$
where
$\sigma' = \gamma \cup \{\alpha \mapsto \tau'\}$

Therefore from Lemma 62 we get the desired

25. T-tapp:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall \alpha.\tau) \qquad \Psi; \Theta\Delta \vdash \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e \ [] : (\tau[\tau'/\alpha])} \text{ T-tapp}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega \ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e \ [] \ \delta\gamma) \in [\![\tau[\tau'/\alpha] \ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (e \ []) \ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau[\tau'/\alpha] \ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e \ []) \ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau[\tau'/\alpha] \ \sigma\iota]\!]$ (F-A0)

<u>IH</u>
$(p_l, T, e \ \delta\gamma) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e \Downarrow_{t_1} \Lambda.e \implies (p_l, T - t_1, \Lambda.e) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]$
Since we know that $(e \ []) \ \delta\gamma \Downarrow_t v_f$ therefore from E-tapp we know that $\exists t_1 < t . e \Downarrow_{t_1} \Lambda.e$, therefore we have
$(p_l, T - t_1, \Lambda.e) \in [\![(\forall \alpha.\tau) \ \sigma\iota]\!]$

Therefore from Definition 15 we have

$\forall \tau'', T_1{<}T - t_1.(p_l, T_1, e) \in [\![\tau[\tau''/\alpha]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-A1)

Instantiating (F-A1) with the given τ' and $T - t_1 - 1$ we get
$(p_l, T - t_1 - 1, e) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 we have
$\forall t_2 < T - t_1 - 1.e \Downarrow_{t_2} v_f \implies (p_l, T - t_1 - t_2 - 1, v_f) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]$

Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-tapp we know that $\exists t_2.e \Downarrow_{t_2} v_f$ where $t_2 = t - t_1 - 1$
Since $t_2 = t - t_1 - 1 < T - t_1 - 1$, therefore we have
$(p_l, T - t_1 - t_2 - 1, v_f) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]$ and we are done.

26. T-iabs:

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall i : \mathsf{S}.\tau)} \text{ T-iabs}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\Lambda.e)\ \delta\gamma) \in [\![(\forall i.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f.(\Lambda.e)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![(\forall i.\tau)\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\Lambda.e)\ \delta\gamma \Downarrow_t v_f$. From E-val we know that $t = 0$ and $v_f = (\Lambda.e)\ \delta\gamma$
Therefore it suffices to prove that
$(p_l, T, (\Lambda.e)\ \delta\gamma) \in [\![(\forall i.\tau)\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\forall I, T'{<}T\ .(p_l, T', e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $I, T'{<}T$ it suffices to prove that
$(p_l, T', e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-IAB0)

From IH we know that
$(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota']\!]_{\mathcal{E}}$
where
$\iota' = \gamma \cup \{i \mapsto I\}$

Therefore from Lemma 62 we get the desired

27. T-iapp:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall i : \mathsf{S}.\tau) \qquad \Psi; \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [] : (\tau[I/i])} \text{ T-iapp}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, e\ []\ \delta\gamma) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f.(e\ [])\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau[I/i]\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e\ [])\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau[I/i]\ \sigma\iota]\!]$ (F-A0)

<u>IH</u>
$(p_l, T, e\ \delta\gamma) \in [\![(\forall i.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T\ .e \Downarrow_{t_1} \Lambda.e \implies (p_l, T - t_1, \Lambda.e) \in [\![(\forall i.\tau)\ \sigma\iota]\!]$
Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-tapp we know that $\exists t_1 < t.e \Downarrow_{t_1} \Lambda.e$, therefore we have
$(p_l, T - t_1, \Lambda.e) \in [\![(\forall i.\tau)\ \sigma\iota]\!]$

Therefore from Definition 15 we have
$\forall I, T_1{<}T - t_1.(p_l, T_1, e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-IAP1)

Instantiating (F-IAP1) with the given I and $T - t_1 - 1$ we get
$(p_l, T - t_1 - 1, e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 we have
$\forall t_2 < T - t_1 - 1.e \Downarrow_{t_2} v_f \implies (p_l, T - t_1 - t_2 - 1, v_f) \in [\![\tau[I/i]\ \sigma\iota]\!]$

Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-iapp we know that $\exists t_2.e \Downarrow_{t_2} v_f$ where $t_2 = t - t_1 - 1$
Since $t_2 = t - t_1 - 1 < T - t_1 - 1$, therefore we have
$(p_l, T - t_1 - t_2 - 1, v_f) \in [\![\tau[I/i]\ \sigma\iota]\!]$ and we are done.

28. T-CI:

$$\frac{\Psi; \Theta; \Delta, c; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (c \Rightarrow \tau)} \text{ T-CI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l, T, \Lambda.e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall v, t < T\ .\Lambda.e\ \delta\gamma \Downarrow_t v \implies (p_l, T - t, v) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

This means given some $v, t < T$ s.t $\Lambda.e\ \delta\gamma \Downarrow_t v$ and from (E-val) we know that $v = \Lambda.e\ \delta\gamma$ and $t = 0$ therefore it suffices to prove that
$(p_l, T, \Lambda.e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$. \models c\ \iota \implies (p_l, T, e\ \delta\gamma) \in [\![\tau\sigma\iota]\!]_{\mathcal{E}}$

This means given that $. \models c\ \iota$ it suffices to prove that
$(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

<u>IH</u> $(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH

29. T-CE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \Rightarrow \tau) \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e[] : \tau} \text{ T-CE}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l, T, e\ []\ \delta\gamma) \in [\![(\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall v_f, t < T\ .(e\ [])\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![(\tau)\ \sigma\iota]\!]$

This means given some $v_f, t < T$ s.t $(e\ [])\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![(\tau)\ \sigma\iota]\!]$ (F-Tap0)

<u>IH</u>
$(p_l, T, e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall v', t' < T\ .e\ \delta\gamma \Downarrow v' \implies (p_l + p_m, v') \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$
Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-CE we know that $\exists t' < t.e\delta\gamma \Downarrow_{t'} \Lambda.e'$, an since $t' < t < T$ therefore we have
$(p_l, T - t', \Lambda.e') \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

Therefore from Definition 15 we have
$. \models c\ \iota \implies (p_l, T - t', e'\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given $\Theta; \Delta \models c$ and $. \models \Delta\ \iota$ therefore we know that $. \models c\ \iota$. Hence we get
$(p_l, T - t', e'\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall v'_f, t'' < T - t'.(e')\ \delta\gamma \Downarrow_{t''} v'_f \implies (p_l, T - t' - t'', v'_f) \in [\![(\tau)\ \sigma\iota]\!]$ (F-CE1)

Since from E-CE we know that $e'\delta\gamma \Downarrow_t v_f$ therefore we know that $\exists t''.e'\ \delta\gamma \Downarrow_{t''} v_f$ s.t $t = t' + t'' + 1$
Therefore instantiating (F-CE1) with the given v_f and t'' we get
$(p_l, T - t' - t'', v_f) \in [\![(\tau)\ \sigma\iota]\!]$
Since $t = t' + t'' + 1$ therefore from Lemma 61 we get the desired.

30. T-CAndI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \& \tau)} \text{ T-CAndI}$$

Given: $(p_l, T\ \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T\ \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$

To prove: $(p_l, e\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall v_f, t < T . e\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]$

This means given some $v_f, t < T$ s.t $e\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![c\&\tau\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$. \models c\iota \wedge (p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we are given that $. \models \Delta\iota$ and $\Theta; \Delta \models c$ therefore it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-CAI0)

<u>IH</u>: $(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t' < T . e\ \delta\gamma \Downarrow_{t'} v_f \implies (p_l, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we are given that $e\ \delta\gamma \Downarrow_t v_f$ therefore we get
$(p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-CAI1)

We get the desired from (F-CAI1)

31. T-CAndE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (c\&\tau) \qquad \Psi; \Theta; \Delta, c; \Omega; \Gamma_2, x : \tau \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{clet}\, x = e \text{ in } e' : \tau'} \text{ T-CAndE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, (\mathsf{clet}\, x = e \text{ in } e')\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall v_f, t < T . (\mathsf{clet}\, x = e \text{ in } e')\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given soem $v_f, t < T$ s.t $(\mathsf{clet}\, x = e \text{ in } e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-CAE0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2} . p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{l1}, T - t_1 v_1) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we know that $(\mathsf{clet}\, x = e \text{ in } e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-CAndE we know that $\exists v_1, t_1 < t . e\ \delta\gamma \Downarrow_{t_1} v_1$.
Therefore we have
$(p_{l1}, T - t_1, v_1) \in [\![c\&\tau\ \sigma\iota]\!]$

Therefore from Definition 15 we have
$. \models c\iota \wedge (p_{l1}, T - t_1, v_1) \in [\![\tau\ \sigma\iota]\!]$ (F-CAE1)

<u>IH2</u>
$(p_{l2} + p_{l1}, T, e'\ \delta\gamma') \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{x \mapsto v_1\}$

This means from Definition 15 we have
$\forall t_2 < T . e'\ \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_{l1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since we know that $(\mathsf{clet}\, x = e \text{ in } e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-CAndE we know that $\exists t_2 < t . e'\ \delta'\gamma \Downarrow_{t_2} v_f$.
Therefore we have
$(p_{l2} + p_{l1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since $p_l = p_{l1} + p_{l2}$ therefore we get
$(p_l, T - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

And finally from From Lemma 61 we get
$(p_l, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$
And we are done.

32. T-fix:

$$\frac{\Psi; \Theta; \Delta; \Omega, x : \tau; . \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; . \vdash \mathsf{fix} x.e : \tau} \text{ T-fix}$$

Given: $(0, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(0, T, (\mathsf{fix} x.e)\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ (F-FX0)

We induct on T
Base case, $T = 1$:
It suffices to prove that $(0, 1, (\mathsf{fix} x.e)\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]$
This means from Definition 15 it suffices to prove
$\forall t < 1.(\mathsf{fix} x.e)\ \delta\gamma \Downarrow_t v \implies (0, 1 - t, v) \in [\![\tau]\!]$

This further means that given $t < 1$ s.t $(\mathsf{fix} x.e)\ \delta\gamma \Downarrow_t v$ it suffices to prove that $(0, 1 - t, v) \in [\![\tau]\!]$

Since from E-fix we know that minimum value of t can be 1 therefore $t < 1$ is not possible and the goal holds vacuously.

Inductive case:
IH: $(0, T - 1, (\mathsf{fix} x.e)\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 16 we have
$(0, T - 1, \delta') \in [\![\Omega, x : \tau\ \sigma\iota]\!]_{\mathcal{E}}$ where $\delta' = \delta \cup \{x \mapsto \mathsf{fix} x.e\ \delta\}$

Applying Definition 15 on (F-FX0) it suffices to prove that
$\forall t < T .(\mathsf{fix} x.e)\ \delta\gamma \Downarrow_t v_f \implies (0, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

This means given some $t < T$ s.t $\mathsf{fix} x.e\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(0, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-FX0.0)

Now from IH of outer induction we have
$(0, T - 1, e\ \delta'\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
This means from Definition 15 we have
$\forall t' < T - 1.e\ \delta'\gamma \Downarrow_{t'} v_f \implies (0, T - 1 - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we know that $\mathsf{fix} x.e\ \delta\gamma \Downarrow_t v_f$ therefore from E-fix we know that $\exists t' = t - 1$ s.t $e\ \delta'\gamma \Downarrow_{t'} v_f$

Since $t < T$ therefore $t' = t - 1 < T - 1$ hence we have
$(0, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$
Therefore we are done

33. T-ret:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{ret}\, e : \mathbb{M}\, 0\, \tau} \text{ T-ret}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathsf{ret}\, e\ \delta\gamma) \in [\![\mathbb{M}\, 0\, \tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f.(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t, v_f) \in [\![\mathbb{M}\, 0\, \tau\ \sigma\iota]\!]$

It means we are given some $t < T, v_f$ s.t $(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_t v_f$. From E-val we know that $t = 0$ and $v_f = (\mathsf{ret}\, e)\ \delta\gamma$.
Therefore it suffices to prove that
$(p_l, T, (\mathsf{ret}\, e)\ \delta\gamma) \in [\![\mathbb{M}\, 0\, \tau\ \sigma\iota]\!]$

From Definition 15 it further suffices to prove that
$\forall t' < T .(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f \implies \exists p'.n' + p' \leq p_l \wedge (p', T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

This means given some $t' < T$ s.t $(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f$ it suffices to prove that
$\exists p'.n' + p' \leq p_l \wedge (p', T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

From (E-ret) we know that $n' = 0$ therefore we choose p' as p_l and it suffices to prove that
$(p_l, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-R0)

IH
$(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T .(e)\ \delta\gamma \Downarrow_t v_f \implies (p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{ret}\, e)\ \delta\gamma \Downarrow^0_{t'} v_f$ therefore from (E-ret) we know that $\exists t_1.e\ \delta\gamma \Downarrow_{t_1} v_f$

Since $t_1 < t < T$ therefore we have
$(p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

And finally from Lemma 61 we get
$(p_l, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$
and we are done.

34. T-bind:

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma_1 \vdash e_1 : \mathbb{M}\, n_1\, \tau_1 \qquad \Psi;\Theta;\Delta;\Omega;\Gamma_2, x:\tau_1 \vdash e_2 : \mathbb{M}\, n_2\, \tau_2 \qquad \Theta \vdash n_1 : \mathbb{R}^+ \qquad \Theta \vdash n_2 : \mathbb{R}^+}{\Psi;\Theta;\Delta;\Omega;\Gamma_1 \oplus \Gamma_2 \vdash \mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2 : \mathbb{M}(n_1+n_2)\, \tau_2} \text{ T-bind}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2\ \delta\gamma) \in [\![\mathbb{M}(n_1+n_2)\, \tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v.(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2)\ \delta\gamma \Downarrow_t v \implies (p_l, T - t, v) \in [\![\mathbb{M}(n_1+n_2)\, \tau_2\ \sigma\iota]\!]$

This means given some $t < T, v$ s.t $(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2)\ \delta\gamma \Downarrow_t v$. From E-val we know that $t = 0$ and $v = (\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2\ \delta\gamma)$
Therefore it suffices to prove that
$(p_l, T, (\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2\ \delta\gamma)) \in [\![\mathbb{M}(n_1+n_2)\, \tau_2\ \sigma\iota]\!]$

This means from Definition 15 it suffices to prove that
$\forall t' < T, v_f.(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2\ \delta\gamma) \Downarrow^{s'}_{t'} v_f \implies \exists p'.s' + p' \leq p_l + n \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$

This means given some $t' < T, v_f$ s.t $(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2\ \delta\gamma) \Downarrow^{s'}_{t'} v_f$ and we need to prove that
$\exists p'.s' + p' \leq p_l + n \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-B0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e_1\ \delta\gamma) \in [\![\mathbb{M}(n_1)\, \tau_1\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it means we have
$\forall t_1 < T .(e_1)\ \delta\gamma \Downarrow_{t_1} v_{m1} \implies (p_{l1}, T - t_1, v_{m1}) \in [\![\mathbb{M}(n_1)\, \tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2)\ \delta\gamma \Downarrow^{s'}_{t'} v_f$ therefore from E-bind we know that $\exists t_1 < t', v_{m1}.(e_1)\ \delta\gamma \Downarrow_{t_1} v_{m1}$.
Since $t_1 < t' < T$, therefore we have
$(p_{l1}, T - t_1, v_{m1}) \in [\![\mathbb{M}(n_1)\, \tau_1\ \sigma\iota]\!]$ (F-B1)

This means from Definition 15 we are given that
$\forall t'_1 < T - t_1.v_{m1} \Downarrow^{s_1} v_1 \implies \exists p'_1.s_1 + p'_1 \leq p_{l1} + n_1 \wedge (p'_1, T - t_1 - t'_1, v_1) \in [\![\tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2)\ \delta\gamma \Downarrow_{t'} v_f$ therefore from E-bind we know that $\exists t'_1 < t - t_1.(e_1)\ \delta\gamma \Downarrow^{s_1}_{t'_1} v_1$.
Since $t'_1 < t - t_1 < T - t_1$ therefore means we have
$\exists p'_1.s_1 + p'_1 \leq p_{l1} + n_1 \wedge (p'_1, T - t_1 - t'_1, v_1) \in [\![\tau_1\ \sigma\iota]\!]$ (F-B1)

<u>IH2</u>
$(p_{l2} + p'_1, T - t_1 - t'_1, e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it means we have
$\forall t_2 < T - t_1 - t'_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} v_{m2} \implies (p_{l2} + p'_1, T - t_1 - t'_1 - t_2, v_{m2}) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2)\ \delta\gamma \Downarrow^{s'}_{t'} v_f$ therefore from E-bind we know that $\exists t_2 < t' - t_1 - t'_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} v_{m2}$.
Since $t_2 < t' - t_1 - t'_1 < T - t_1 - t'_1$ therefore we have
$(p_{l2} + p'_1, T - t_1 - t'_1 - t_2, v_{m2}) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

This means from Definition 15 we are given that

$\forall t_2' < T - t_1 - t_1' - t_2.v_{m2} \Downarrow_{t_2'}^{s_2} v_2 \implies \exists p_2'.s_2 + p_2' \leq p_{l2} + p_1' + n_2 \wedge (p_2', T - t_1 - t_1' - t_2', v_2) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_{t'}^{s'} v_f$ therefore from E-bind we know that $\exists t_2' < t' - t_1 - t_1' - t_2, s_2, v_2.v_{m2} \Downarrow_{t_2'}^{s_2} v_2$.
This means we have
$\exists p_2'.s_2 + p_2' \leq p_{l2} + p_1' + n_2 \wedge (p_2', T - t_1 - t_1' - t_2 - t_2', v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-B2)

In order to prove (F-B0) we choose p' as p_2' and it suffices to prove

(a) $s' + p_2' \leq p_l + n$:
Since from (F-B2) we know that
$s_2 + p_2' \leq p_{l2} + p_1' + n_2$

Adding s_1 on both sides we get
$s_1 + s_2 + p_2' \leq p_{l2} + s_1 + p_1' + n_2$

Since from (F-B1) we know that
$s_1 + p_1' \leq p_{l1} + n_1$

therefore we also have
$s_1 + s_2 + p_2' \leq p_{l2} + p_{l1} + n_1 + n_2$

And finally since we know that $n = n_1 + n_2$, $s' = s_1 + s_2$ and $p_l = p_{l1} + p_{l2}$ therefore we get the desired

(b) $(p_2', T - t_1 - t_1' - t_2 - t_2', v_f) \in [\![\tau_2\ \sigma\iota]\!]$:
From E-bind we know that $v_f = v_2$ therefore we get the desired from (F-B2)

35. T-tick:

$$\frac{\Theta \vdash n : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \uparrow^n : \mathbb{M}\, n\, \mathbf{1}} \text{ T-tick}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \uparrow^n\ \delta\gamma) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v.(\uparrow^n)\ \delta\gamma \Downarrow_t v \implies (p_l, T - t, v) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]$

This means we are given some $t < T, v$ s.t $(\uparrow^n)\ \delta\gamma \Downarrow_t v$. From E-val we know that $t = 0$ and $v = (\uparrow^n)\ \delta\gamma$
Therefore it suffices to prove that
$(p_l, T, (\uparrow^n)\ \delta\gamma) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\forall t' < T .(\uparrow^n)\ \delta\gamma \Downarrow_{t'}^{n'} () \implies \exists p'.n' + p' \leq p_l + n \wedge (p', T - t', ()) \in [\![\mathbf{1}]\!]$

This means given some $t' < T$ s.t $(\uparrow^n)\ \delta\gamma \Downarrow_{t'}^{n'} ()$ it suffices to prove that
$\exists p'.n' + p' \leq p_l + n \wedge (p', T - t', ()) \in [\![\mathbf{1}]\!]$

From (E-tick) we know that $n' = n$ therefore we choose p' as p_l and it suffices to prove that
$(p_l, T - t', ()) \in [\![\mathbf{1}]\!]$

We get this directly from Definition 15

36. T-release:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e_1 : [n_1]\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}(n_1 + n_2)\, \tau_2 \qquad \Theta \vdash n_1 : \mathbb{R}^+ \qquad \Theta \vdash n_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2 : \mathbb{M}\, n_2\, \tau_2} \text{ T-release}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2\ \delta\gamma) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v.(\mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_t v \implies (p_l, T - t, v) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

This means given some $t < T, v$ s.t $(\mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow (\mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma$. From E-val we know that $t = 0$ and $v = (\mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2\ \delta\gamma)$
Therefore it suffices to prove that
$(p_l, T, (\mathsf{release}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

This means from Definition 15 it suffices to prove that

$\forall t' < T, v_f.(\mathsf{release}\, x = e_1 \text{ in } e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f \implies \exists p'.s' + p' \leq p_l + n_2 \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$

This means given some $t' < T, v_f$ s.t $(\mathsf{release}\, x = e_1 \text{ in } e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f$ and we need to prove that $\exists p'.s' + p' \leq p_l + n_2 \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-R0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t $(p_{l1}, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1}, T, e_1\ \delta\gamma) \in [\![[n_1]\,\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it means we have
$\forall t_1 < T .(e_1)\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{l1}, T - t_1, v_1) \in [\![[n_1]\,\tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1 \text{ in } e_2)\ \delta\gamma \Downarrow_{t'}^{s'} v_f$ therefore from E-rel we know that $\exists t_1 < t'.(e_1)\ \delta\gamma \Downarrow_{t_1} v_1$.
Since $t_1 < t' < T$, therefore we have
$(p_{l1}, T - t_1, v_1) \in [\![[n_1]\,\tau_1\ \sigma\iota]\!]$

This means from Definition 15 we have
$\exists p'_1.p'_1 + n_1 \leq p_{l1} \wedge (p'_1, T - t_1, v_1) \in [\![\tau_1]\!]$ (F-R1)

<u>IH2</u>
$(p_{l2} + p'_1, T - t_1, e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_1 + n_2)\,\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it means we have
$\forall t_2 < T - t_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} v_{m2} \cup \{x \mapsto v_1\} \implies (p_{l2} + p'_1, T - t_1 - t_2, v_{m2}) \in [\![\mathbb{M}(n_1 + n_2)\,\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1 \text{ in } e_2)\ \delta\gamma \Downarrow_{t'}^{s'} v_f$ therefore from E-rel we know that $\exists t_2 < t - t_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} v_{m2}$. This means we have
$(p_{l2} + p'_1, T - t_1 - t_2, v_{m2}) \in [\![\mathbb{M}(n_1 + n_2)\,\tau_2\ \sigma\iota]\!]$

This means from Definition 15 we are given that
$\forall t'_2 < T - t_1 - t_2.v_{m2} \Downarrow_{t'_2}^{s_2} v_2 \implies \exists p'_2.s_2 + p'_2 \leq p_{l2} + p'_1 + n_1 + n_2 \wedge (p'_2, T - t_1 - t_2 - t'_2, v_2) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1 \text{ in } e_2)\ \delta\gamma \Downarrow_{t'}^{s'} v_f$ therefore from E-rel we know that $\exists t'_2.v_{m2} \Downarrow_{t'_2}^{s_2} v_2$ s.t. $t'_2 = t' - t_1 - t_2 - 1$
Since $t'_2 = t' - t_1 - t_2 < T - t_1 - t_2$, therefore we have
$\exists p'_2.s_2 + p'_2 \leq p_{l2} + p'_1 + n_1 + n_2 \wedge (p'_2, T - t_1 - t_2 - t'_2, v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-R2)

In order to prove (F-R0) we choose p' as p'_2 and it suffices to prove

(a) $s' + p'_2 \leq p_l + n_2$:
Since from (F-R2) we know that
$s_2 + p'_2 \leq p_{l2} + p'_1 + n_1 + n_2$

Since from (F-R1) we know that
$p'_1 + n_1 \leq p_{l1}$

therefore we also have
$s_2 + p'_2 \leq p_{l2} + p_{l1} + p_{m1} + n_2$

And finally since we know that $s' = s_2$, $p_l = p_{l1} + p_{l2}$ and $0 = p_{m1}$ therefore we get the desired

(b) $(p'_2, T - t_1 - t_2 - t'_2, v_f) \in [\![\tau_2\ \sigma\iota]\!]$:
From E-rel we know that $v_f = v_2$ therefore we get the desired from (F-R2)

37. T-store:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta \vdash n : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{store}\, e : \mathbb{M}\, n\, ([n]\,\tau)}\ \text{T-store}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(0, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l, T, \mathsf{store}\, e\ \delta\gamma) \in [\![\mathbb{M}\, n\, ([n]\,\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v.(\mathsf{store}\, e)\ \delta\gamma \Downarrow_t v \implies (p_l, T - t, v) \in [\![\mathbb{M}\, n\, ([n]\,\tau)\ \sigma\iota]\!]$

This means we are given some $t < T, v$ s.t $(\mathsf{store}\, e)\ \delta\gamma \Downarrow_t v$. From E-val we know that $t = 0$ and $v = (\mathsf{store}\, e)\ \delta\gamma$
Therefore it suffices to prove that

$(p_l, T, (\mathsf{store}\, e)\ \delta\gamma) \in [\![\mathbb{M}\, n\, ([n]\, \tau)\ \sigma\iota]\!]$

From Definition 15 it suffices to prove that
$\forall t' < T, v_f, n'.(\mathsf{store}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f \implies \exists p'.n' + p' \leq p_l \wedge (p', T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]$

This means given some $t' < T, v_f$ s.t $(\mathsf{store}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f$ it suffices to prove that
$\exists p'.n' + p' \leq p_l \wedge (p', T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]$

From (E-store) we know that $n' = 0$ therefore we choose p' as $p_l + n$ and it suffices to prove that
$(p_l + n, T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]$

This further means that from Definition 15 we have
$\exists p''.p'' + n \leq p_l + n \wedge (p'', T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

We choose p'' as p_l and it suffices to prove that
$(p_l, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-S0)

<u>IH</u>
$(p_l, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T\ .(e)\ \delta\gamma \Downarrow_{t_1} v_f \implies (p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{store}\, e)\ \delta\gamma \Downarrow_{t'}^{0} v_f$ therefore from (E-store) we know that $\exists t_1 < t'.e\ \delta\gamma \Downarrow_{t_1} v_f$

Since $t_1 < t' < T$ therefore we have
$(p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

and finally from Lemma 61 we have
$(p_l, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

□

Lemma 21 (Value subtyping lemma)**.** $\forall \Psi, \Theta, \tau \in \mathit{Type}, \tau'.$
$\Psi; \Theta; \Delta \vdash \tau <: \tau' \wedge . \models \Delta\iota \implies [\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$

Proof. Proof by induction on the $\Psi; \Theta; \Delta \vdash \tau <: \tau'$ relation

1. sub-refl:

$$\frac{}{\Psi; \Theta; \Delta \vdash \tau <: \tau}\ \text{sub-refl}$$

   To prove: $\forall (p, T, v) \in [\![\tau\ \sigma\iota]\!] \implies (p, T, v) \in [\![\tau\ \sigma\iota]\!]$

   Trivial

2. sub-arrow:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1' <: \tau_1 \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \multimap \tau_2 <: \tau_1' \multimap \tau_2'}\ \text{sub-arrow}$$

   To prove: $\forall (p, T, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!] \implies (p, T, \lambda x.e) \in [\![(\tau_1' \multimap \tau_2')\ \sigma\iota]\!]$

   This means given some $(p, T, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$ we need to prove
   $(p, T, \lambda x.e) \in [\![(\tau_1' \multimap \tau_2')\ \sigma\iota]\!]$

   From Definition 15 we are given that
   $\forall\, T' < T, p', e'.(p', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p + p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SL0)

   Also from Definition 15 it suffices to prove that
   $\forall\, T'' < T, p'', e''.(p'', T'', e'') \in [\![\tau_1'\ \sigma\iota]\!]_{\mathcal{E}} \implies (p + p'', T'', e[e''/x]) \in [\![\tau_2'\ \sigma\iota]\!]_{\mathcal{E}}$

   This means given some $T'' < T, p'', e''$ s.t $(p'', T'', e'') \in [\![\tau_1'\ \sigma\iota]\!]$ we need to prove
   $(p + p', T'', e[e''/x]) \in [\![\tau_2'\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SL1)

   <u>IH1</u>: $[\![\tau_1'\ \sigma\iota]\!] \subseteq [\![\tau_1\ \sigma\iota]\!]$

   Since we have $(p'', T'', e'') \in [\![\tau_1'\ \sigma\iota]\!]$ therefore from IH1 we also have $(p'', T'', e'') \in [\![\tau_1\ \sigma\iota]\!]$

   Therefore instantiating (F-SL0) with p', T'', e'' we get
   $(p + p'', T'', e[e''/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

   And finally from Lemma 22 we get the desired

3. sub-tensor:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \otimes \tau_2 <: \tau_1' \otimes \tau_2'} \text{ sub-tensor}$$

To prove: $\forall (p, T, \langle\langle v_1, v_2 \rangle\rangle) \in \llbracket (\tau_1 \otimes \tau_2)\ \sigma\iota \rrbracket \implies (p, T, \langle\langle v_1, v_2 \rangle\rangle) \in \llbracket (\tau_1' \otimes \tau_2')\ \sigma\iota \rrbracket$

This means given $(p, T, \langle\langle v_1, v_2 \rangle\rangle) \in \llbracket (\tau_1 \otimes \tau_2)\ \sigma\iota \rrbracket$
It suffices prove that
$(p, T, \langle\langle v_1, v_2 \rangle\rangle) \in \llbracket (\tau_1' \otimes \tau_2')\ \sigma\iota \rrbracket$

This means from Definition 15 we are given that
$\exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v_1) \in \llbracket \tau_1\ \sigma\iota \rrbracket \wedge (p_2, T, v_2) \in \llbracket \tau_2\ \sigma\iota \rrbracket$

Also from Definition 15 it suffices to prove that
$\exists p_1', p_2'.p_1' + p_2' \leq p \wedge (p_1', T, v_1) \in \llbracket \tau_1'\ \sigma\iota \rrbracket \wedge (p_2', T, v_2) \in \llbracket \tau_2'\ \sigma\iota \rrbracket$

<u>IH1</u> $\llbracket (\tau_1)\ \sigma\iota \rrbracket \subseteq \llbracket (\tau_1')\ \sigma\iota \rrbracket$
<u>IH2</u> $\llbracket (\tau_2)\ \sigma\iota \rrbracket \subseteq \llbracket (\tau_2')\ \sigma\iota \rrbracket$

Choosing p_1 for p_1' and p_2 for p_2' we get the desired from IH1 and IH2

4. sub-with:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \,\&\, \tau_2 <: \tau_1' \,\&\, \tau_2'} \text{ sub-with}$$

To prove: $\forall (p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1 \,\&\, \tau_2)\ \sigma\iota \rrbracket \implies (p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1' \,\&\, \tau_2')\ \sigma\iota \rrbracket$

This means given $(p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1 \,\&\, \tau_2)\ \sigma\iota \rrbracket$
It suffices prove that
$(p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1' \,\&\, \tau_2')\ \sigma\iota \rrbracket$

This means from Definition 15 we are given that
$(p, T, v_1) \in \llbracket \tau_1\ \sigma\iota \rrbracket \wedge (p, T, v_2) \in \llbracket \tau_2\ \sigma\iota \rrbracket$ (F-SW0)

Also from Definition 15 it suffices to prove that
$(p, T, v_1) \in \llbracket \tau_1'\ \sigma\iota \rrbracket \wedge (p, T, v_2) \in \llbracket \tau_2'\ \sigma\iota \rrbracket$

<u>IH1</u> $\llbracket (\tau_1)\ \sigma\iota \rrbracket \subseteq \llbracket (\tau_1')\ \sigma\iota \rrbracket$
<u>IH2</u> $\llbracket (\tau_2)\ \sigma\iota \rrbracket \subseteq \llbracket (\tau_2')\ \sigma\iota \rrbracket$

We get the desired from (F-SW0), IH1 and IH2

5. sub-sum:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \oplus \tau_2 <: \tau_1' \oplus \tau_2'} \text{ sub-sum}$$

To prove: $\forall (p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket \implies (p, T, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1' \oplus \tau_2')\ \sigma\iota \rrbracket$

This means given $(p, T, v) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket$
It suffices prove that
$(p, T, v) \in \llbracket (\tau_1' \oplus \tau_2')\ \sigma\iota \rrbracket$

This means from Definition 15 two cases arise

(a) $v = \mathsf{inl}(v')$:
This means from Definition 15 we have $(p, T, v') \in \llbracket \tau_1\ \sigma\iota \rrbracket$ (F-SS0)

Also from Definition 15 it suffices to prove that
$(p, T, v') \in \llbracket \tau_1'\ \sigma\iota \rrbracket$

<u>IH</u> $\llbracket (\tau_1)\ \sigma\iota \rrbracket \subseteq \llbracket (\tau_1')\ \sigma\iota \rrbracket$

We get the desired from (F-SS0), IH

(b) $v = \mathsf{inr}(v')$:
Symmetric reasoning as in the inl case

6. sub-list:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash L^n\ \tau <: L^n\ \tau'} \text{ sub-list}$$

To prove: $\forall (p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!].(p, T, v) \in [\![L^n\ \tau'\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!]$ and we need to prove $(p, T, v) \in [\![L^n\ \tau'\ \sigma\iota]\!]$

We induct on $(p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!]$

(a) $(p, T, nil) \in [\![L^0\ \tau\ \sigma\iota]\!]$:
We need to prove $(p, T, nil) \in [\![L^0\ \tau'\ \sigma\iota]\!]$
We get this directly from Definition 15

(b) $(p, T, v' :: l') \in [\![L^{m+1}\ \tau\ \sigma\iota]\!]$:
In this case we are given $(p, T, v' :: l') \in [\![L^{m+1}\ \tau\ \sigma\iota]\!]$
and we need to prove $(p, T, v' :: l') \in [\![L^{m+1}\ \tau'\ \sigma\iota]\!]$

This means from Definition 15 are given
$\exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v') \in [\![\tau\ \sigma\iota]\!] \wedge (p_2, T, l') \in [\![L^m \tau\ \sigma\iota]\!]$ (Sub-List0)

Similarly from Definition 15 we need to prove that
$\exists p'_1, p'_2.p'_1 + p'_2 \leq p \wedge (p'_1, T, v') \in [\![\tau'\ \sigma\iota]\!] \wedge (p_2, T, l') \in [\![L^m \tau'\ \sigma\iota]\!]$

We choose p'_1 as p_1 and p'_2 as p_2 and we get the desired from (Sub-List0) IH of outer induction and IH of innner induction

7. sub-exist:

$$\frac{\Psi; \Theta, s; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash \exists s.\tau <: \exists s.\tau'} \text{ sub-exist}$$

To prove: $\forall (p, T, v) \in [\![\exists s.\tau\ \sigma\iota]\!].(p, T, v) \in [\![\exists s.\tau'\ \sigma\iota]\!]$

This means given some $(p, T, v) \in [\![\exists s.\tau\ \sigma\iota]\!]$ we need to prove $(p, T, v) \in [\![\exists s.\tau'\ \sigma\iota]\!]$

From Definition 15 we are given that
$\exists s'.(p, T, v) \in [\![\tau\sigma\iota[s'/s]]\!]$ (F-exist0)

<u>IH</u>: $[\![(\tau)\ \sigma\iota \cup \{s \mapsto s'\}]\!] \subseteq [\![(\tau')\ \sigma\iota \cup \{s \mapsto s'\}]\!]$

Also from Definition 15 it suffices to prove that
$\exists s''.(p, T, v) \in [\![\tau'\sigma\iota[s''/s]]\!]$

We choose s'' as s' and we get the desired from IH

8. sub-potential:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Psi; \Theta; \Delta \vdash n' \leq n}{\Psi; \Theta; \Delta \vdash [n]\,\tau <: [n']\,\tau'} \text{ sub-potential}$$

To prove: $\forall (p, T, v) \in [\![[n]\,\tau\ \sigma\iota]\!].(p, T, v) \in [\![[n']\,\tau'\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![[n]\,\tau\ \sigma\iota]\!]$ and we need to prove $(p, T, v) \in [\![[n']\,\tau'\ \sigma\iota]\!]$

This means from Definition 15 we are given
$\exists p'.p' + n \leq p \wedge (p', T, v) \in [\![\tau\ \sigma\iota]\!]$ (F-SP0)

And we need to prove
$\exists p''.p'' + n' \leq p \wedge (p'', T, v) \in [\![\tau'\ \sigma\iota]\!]$ (F-SP1)

In order to prove (F-SP1) we choose p'' as p'

Since from (F-SP0) we know that $p' + n \leq p$ and we are given that $n' \leq n$ therefore we also have $p' + n' \leq p$

<u>IH</u> $[\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$

We get the desired directly from IH

9. sub-monad:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Psi; \Theta; \Delta \vdash n \leq n'}{\Psi; \Theta; \Delta \vdash \mathbb{M}\, n\, \tau <: \mathbb{M}\, n'\, \tau'} \text{ sub-monad}$$

To prove: $\forall (p, T, v) \in [\![\mathbb{M}\, n\, \tau\ \sigma\iota]\!].(p, T, v) \in [\![\mathbb{M}\, n'\, \tau'\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\mathbb{M}\, n\, \tau\ \sigma\iota]\!]$ and we need to prove $(p, T, v) \in [\![\mathbb{M}\, n'\, \tau'\ \sigma\iota]\!]$

This means from Definition 15 we are given
$\forall t' < T, n_1, v'.v \Downarrow_{t'}^{n_1} v' \implies \exists p'.n_1 + p' \leq p + n \wedge (p', T - t', v') \in [\![\tau\ \sigma\iota]\!]$ (F-SM0)

Again from Definition 15 we need to prove that
$\forall t'' < T, n_2, v''.v \Downarrow_{t''}^{n_2} v'' \implies \exists p''.n_2 + p'' \leq p + n' \wedge (p'', T - t'', v'') \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t'' < T, v'', n_2$ s.t $v \Downarrow_{t''}^{n_2} v'$ it suffices to prove that
$\exists p''.n_2 + p'' \leq p + n' \wedge (p'', T - t'', v'') \in [\![\tau'\ \sigma\iota]\!]$ (F-SM1)

Instantiating (F-SM0) with t'', n_2, v'' Since $v \Downarrow_{t''}^{n_2} v''$ therefore from (F-SM0) we know that
$\exists p'.n_2 + p' \leq p + n \wedge (p', T - t'', v'') \in [\![\tau\ \sigma\iota]\!]$ (F-SM2)

<u>IH</u> $[\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$

In order to prove (F-SM1) we choose p'' as p' and we need to prove

(a) $n_2 + p'' \leq p + n'$:
Since we are given that $n \leq n'$ therefore we get the desired from (F-SM2)

(b) $(p', v') \in [\![\tau'\ \sigma\iota]\!]$
We get this directly from IH and (F-SM2)

10. sub-Exp:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash !\tau <: !\tau'} \text{ sub-Exp}$$

To prove: $\forall (p, T, v) \in [\![!\tau\ \sigma\iota]\!].(p, T, v) \in [\![!\tau'\ \sigma\iota]\!]$

This means given $(p, T, !e) \in [\![!\tau\ \sigma\iota]\!]$ and we need to prove $(p, T, !e) \in [\![!\tau'\ \sigma\iota]\!]$

This means from Definition 15 we are given
$(0, T, e) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SE0)

Again from Definition 15 we need to prove that
$(0, T, e) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SE1)

<u>IH</u> $[\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$
Therefore from (F-SE0) and IH we get $(0, T, e) \in [\![\tau'\ \sigma\iota]\!]$ and we are done.

11. sub-typePoly:

$$\frac{\Psi, \alpha; \Theta; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall \alpha.\tau_1 <: \forall i.\tau_2} \text{ sub-typePoly}$$

To prove: $\forall (p, T, \Lambda.e) \in [\![(\forall i.\tau_1)\ \sigma\iota]\!].(p, T, \Lambda.e) \in [\![(\forall i.\tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda.e) \in [\![(\forall \alpha.\tau_1)\ \sigma\iota]\!]$ we need to prove $(p, T, \Lambda.e) \in [\![(\forall \alpha.\tau_2)\ \sigma\iota]\!]$

From Definition 15 we are given that
$\forall \tau, T' < T\ .(p, T', e) \in [\![\tau_1[\tau/\alpha]]\!]_{\mathcal{E}}$ (F-STP0)

Also from Definition 15 it suffices to prove that
$\forall \tau', T'' < T\ .(p, T'', e) \in [\![\tau_2[\tau'/\alpha]]\!]_{\mathcal{E}}$

This means given some $\tau', T'' < T$ and we need to prove
$(p, T'', e) \in [\![\tau_2[\tau'/\alpha]]\!]_{\mathcal{E}}$ (F-STP1)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota \cup \{\alpha \mapsto \tau'\}]\!] \subseteq [\![(\tau_2)\ \sigma\iota \cup \{\alpha \mapsto \tau'\}]\!]$

Instantiating (F-STP0) with τ', T'' we get
$(p, T'', e) \in [\![\tau_1[\tau'/\alpha]]\!]_{\mathcal{E}}$

and finally from IH we get the desired.

12. sub-indexPoly:

$$\frac{\Psi; \Theta, i; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall i.\tau_1 <: \forall i.\tau_2} \text{ sub-indexPoly}$$

To prove: $\forall (p, T, \Lambda i.e) \in [\![(\forall i.\tau_1)\ \sigma\iota]\!].(p, T, \Lambda i.e) \in [\![(\forall i.\tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda i.e) \in [\![(\forall i.\tau_1)\ \sigma\iota]\!]$ we need to prove
$(p, T, \Lambda i.e) \in [\![(\forall i.\tau_2)\ \sigma\iota]\!]$

From Definition 15 we are given that
$\forall I, T'{<}T\ .(p, T', e) \in [\![\tau_1[I/i]]\!]_{\mathcal{E}}$ (F-SIP0)

Also from Definition 15 it suffices to prove that
$\forall I', T''{<}T\ .(p, T'', e) \in [\![\tau_2[I'/i]]\!]_{\mathcal{E}}$

This means given some $I', T''{<}T$ and we need to prove
$(p, T'', e) \in [\![\tau_2[I'/i]]\!]_{\mathcal{E}}$ (F-SIP1)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota \cup \{i \mapsto I'\}]\!] \subseteq [\![(\tau_2)\ \sigma\iota \cup \{i \mapsto I'\}]\!]$

Instantiating (F-SIP0) with I', T'' we get
$(p, T'', e) \in [\![\tau_1[I'/i]]\!]_{\mathcal{E}}$

and finally from IH we get the desired

13. sub-constraint:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_2 \implies c_1}{\Psi; \Theta; \Delta \vdash c_1 \Rightarrow \tau_1 <: c_2 \Rightarrow \tau_2} \text{ sub-constraint}$$

To prove: $\forall (p, T, \Lambda.e) \in [\![(c_1 \Rightarrow \tau_1)\ \sigma\iota]\!].(p, T, \Lambda.e) \in [\![(c_2 \Rightarrow \tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda.e) \in [\![(c_1 \Rightarrow \tau_1)\ \sigma\iota]\!]$ we need to prove
$(p, T, \Lambda.e) \in [\![(c_2 \Rightarrow \tau_2)\ \sigma\iota]\!]$

From Definition 15 we are given that
$. \models c_1\iota \implies (p, T, e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SC0)

Also from Definition 15 it suffices to prove that
$. \models c_2\iota \implies (p, T, e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$

This means given some $. \models c_2\iota$ and we need to prove
$(p, T, e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$ (F-SC1)

Since we are given that $\Theta; \Delta \models c_2 \implies c_1$ therefore we know that $. \models c_1\iota$

Hence from (F-SC0) we have
$(p, T, e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SC2)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_2)\ \sigma\iota]\!]$

Therefore we ge the desired from IH and (F-SC2)

14. sub-CAnd:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_1 \implies c_2}{\Psi; \Theta; \Delta \vdash c_1 \& \tau_1 <: c_2 \& \tau_2} \text{ sub-CAnd}$$

To prove: $\forall (p, v) \in [\![(c_1 \& \tau_1)\ \sigma\iota]\!].(p, v) \in [\![(c_2 \& \tau_2)\ \sigma\iota]\!]$

This means given some $(p, v) \in [\![(c_1 \& \tau_1)\ \sigma\iota]\!]$ we need to prove
$(p, v) \in [\![(c_2 \& \tau_2)\ \sigma\iota]\!]$

From Definition 15 we are given that
$. \models c_1\iota \wedge (p, e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SCA0)

Also from Definition 15 it suffices to prove that
$. \models c_2\iota \wedge (p, e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $\Theta; \Delta \models c_2 \implies c_1$ and $. \models c_1\iota$ therefore we also know that $. \models c_2\iota$

Also from (F-SCA0) we have $(p, e) \in [\![\tau_1 \sigma\iota]\!]_{\mathcal{E}}$ (F-SCA1)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_2)\ \sigma\iota]\!]$

Therefore we ge the desired from IH and (F-SCA1)

15. sub-familyAbs:

$$\frac{\Psi; \Theta, i : \mathsf{S} \vdash \tau <: \tau'}{\Psi; \Theta \vdash \lambda_t i : \mathsf{S}.\tau <: \lambda_t i : \mathsf{S}.\tau'} \text{ sub-familyAbs}$$

To prove:
$\forall f \in [\![\lambda_t i : \mathsf{S}.\tau\ \sigma\iota]\!].f \in [\![\lambda_t i : \mathsf{S}.\tau'\ \sigma\iota]\!]$

This means given $f \in [\![\lambda_t i : \mathsf{S}.\tau\ \sigma\iota]\!]$ and we need to prove
$f \in [\![\lambda_t i : \mathsf{S}.\tau'\ \sigma\iota]\!]$

This means from Definition 15 we are given
$\forall I.f\ I \in [\![\tau[I/i]\ \sigma\iota]\!]$ (F-SFAbs0)

This means from Definition 15 we need to prove
$\forall I'.f\ I' \in [\![\tau'[I'/i]\ \sigma\iota]\!]$
This further means that given some I' we need to prove
$f\ I' \in [\![\tau'[I'/i]\ \sigma\iota]\!]$ (F-SFAbs1)

Instantiating (F-SFAbs0) with I' we get
$f\ I' \in [\![\tau[I'/i]\ \sigma\iota]\!]$

From IH we know that $[\![\tau\ \sigma\iota \cup \{i \mapsto I'\ \iota\}]\!] \subseteq [\![\tau'\ \sigma\iota \cup \{i \mapsto I'\ \iota\}]\!]$
And this completes the proof.

16. Sub-tfamilyApp1:

$$\frac{}{\Psi; \Theta; \Delta \vdash \lambda_t i : \mathsf{S}.\tau\ I <: \tau[I/i]} \text{ sub-familyApp1}$$

To prove:
$\forall (p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!].(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$ and we need to prove
$(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$

This means from Definition 15 we are given
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau]\!]\ I\ \sigma\iota$
This further means that we have
$(p, T, v) \in f\ I\ \sigma\iota$ where $f\ I\ \sigma\iota = [\![\tau[I/i]\ \sigma\iota]\!]$

This means we have $(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$
And this completes the proof.

17. Sub-tfamilyApp2:

$$\frac{}{\Psi; \Theta; \Delta \vdash \tau[I/i] <: \lambda_t i : \mathsf{S}.\tau\ I} \text{ sub-familyApp2}$$

To prove: $\forall (p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!].(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$ (Sub-tF0)
And we need to prove
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$

This means from Definition 15 it suffices to prove that
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau]\!]\ I\ \sigma\iota$
It further suffices to prove that
$(p, T, v) \in f\ I\ \sigma\iota$ where $f\ I\ \sigma\iota = [\![\tau[I/i]\ \sigma\iota]\!]$

which means we need to show that
$(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$
We get this directly from (Sub-tF0)

□

Lemma 22 (Expression subtyping lemma)**.** $\forall \Psi, \Theta, \tau, \tau'$.
$\Psi; \Theta \vdash \tau <: \tau' \implies [\![\tau\ \sigma\iota]\!]_{\mathcal{E}} \subseteq [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

Proof. To prove: $\forall (p, T, e) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}} \implies (p, T, e) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $(p, T, e) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, e) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we are given
$\forall t < T, v.e \Downarrow_t v \implies (p, T - t, v) \in [\![\tau\ \sigma\iota]\!]$ (S-E0)

Similarly from Definition 15 it suffices to prove that
$\forall t' < T, v'.e \Downarrow_{t'} v' \implies (p, T - t, v') \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t' < T, v'$ s.t $e \Downarrow_{t'} v'$ it suffices to prove that
$(p, T - t', v') \in [\![\tau'\ \sigma\iota]\!]$

Instantiating (S-E0) with t', v' we get $(p, T - t', v') \in [\![\tau\ \sigma\iota]\!]$

And finally from Lemma 21 we get the desired.

$\square$

Lemma 23 (Γ subtyping lemma)**.** $\forall \Psi, \Theta, \Gamma_1, \Gamma_2, \sigma, \iota.$
$\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2 \implies [\![\Gamma_1 \sigma\iota]\!] \subseteq [\![\Gamma_2 \sigma\iota]\!]$

Proof. Proof by induction on $\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta \vdash \Gamma <: .}\ \text{sub-lBase}$$

To prove: $\forall (p, T, \gamma) \in [\![\Gamma_1 \sigma\iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Gamma_1 \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$
From Definition 16 it suffices to prove that
$\exists f : \mathcal{V}ars \to \mathcal{P}ots. (\forall x \in dom(.). (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x \in dom(.)} f(x) \leq p)$
We choose f as a constant function $f' - = 0$ and we get the desired

2. sub-lInd:

$$\frac{x : \tau' \in \Gamma_1 \qquad \Psi; \Theta \vdash \tau' <: \tau \qquad \Psi; \Theta \vdash \Gamma_1/x <: \Gamma_2}{\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2, x : \tau}\ \text{sub-lBase}$$

To prove: $\forall (p, T, \gamma) \in [\![\Gamma_1 \sigma\iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![\Gamma_2, x : \tau]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Gamma_1 \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![\Gamma_2, x : \tau]\!]_{\mathcal{E}}$

This means from Definition 16 we are given that
$\exists f : \mathcal{V}ars \to \mathcal{P}ots.$
$(\forall x \in dom(\Gamma_1). (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}})$ (L0)
$(\sum_{x \in dom(\Gamma_1)} f(x) \leq p)$ (L1)

Similarly from Definition 16 it suffices to prove that
$\exists f' : \mathcal{V}ars \to \mathcal{P}ots. (\forall y \in dom(\Gamma_2, x : \tau). (f'(y), T, \gamma(y)) \in [\![(\Gamma_2, x : \tau)(y)]\!]_{\mathcal{E}}) \wedge$
$(\sum_{y \in dom(\Gamma_2, x:\tau)} f'(y) \leq p)$

We choose f' as f and it suffices to prove that

(a) $\forall y \in dom(\Gamma_2, x : \tau). (f(y), T, \gamma(y)) \in [\![(\Gamma_2, x : \tau)(y)]\!]_{\mathcal{E}}$:
This means given some $y \in dom(\Gamma_2, x : \tau)$ it suffices to prove that
$(f(y), T, \gamma(y)) \in [\![\tau_2]\!]_{\mathcal{E}}$ where say $(\Gamma_2, x : \tau)(y) = \tau_2$

From Lemma 24 we know that
$y : \tau_1 \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau_1 <: \tau_2$

By instantiating (L0) with the given y
$(f(y), T, \gamma(y)) \in [\![\tau_1]\!]_{\mathcal{E}}$

Finally from Lemma 22 we also get $(f(y), T, \gamma(y)) \in [\![\tau_2]\!]_{\mathcal{E}}$
And we are done

(b) $(\sum_{y \in dom(\Gamma_2, x:\tau)} f(y) \leq p)$:
From (L1) we know that $(\sum_{x \in dom(\Gamma_1)} f(x) \leq p)$ and since from Lemma 24 we know that $dom(\Gamma_2, x : \tau) \subseteq dom(\Gamma_1)$ therefore we also have
$(\sum_{y \in dom(\Gamma_2, x:\tau)} f(y) \leq p)$

$\square$

Lemma 24 (Γ Subtyping: domain containment)**.** $\forall p, \gamma, \Gamma_1, \Gamma_2.$
$\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2 \implies \forall x : \tau \in \Gamma_2.\ x : \tau' \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$

Proof. Proof by induction on $\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta \vdash \Gamma_1 <: .}\ \text{sub-lBase}$$

To prove: $\forall x : \tau' \in (.).x : \tau \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$
Trivial

2. sub-lInd:

$$\frac{x : \tau' \in \Gamma_1 \qquad \Psi; \Theta \vdash \tau' <: \tau \qquad \Psi; \Theta \vdash \Gamma_1/x <: \Gamma_2}{\Psi; \Theta \vdash \Gamma_1 <: \Gamma_2, x : \tau_x}\ \text{sub-lBase}$$

To prove: $\forall y : \tau_1 \in (\Gamma_2, x : \tau_x).y : \tau \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$

This means given some $y : \tau \in (\Gamma_2, x : \tau_x)$ it suffices to prove that
$y : \tau \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$
The follwing cases arise:

- $y = x$:
  In this case we are given that $x : \tau' \in \Gamma_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$
  Therefore we are done
- $y \neq x$:
  Since we are given that $\Psi; \Theta \vdash \Gamma_1/x <: \Gamma_2$ therefore we get the desired from IH

□

Lemma 25 (Ω subtyping lemma)**.** $\forall \Psi, \Theta, \Omega_1, \Omega_2, \sigma, \iota.$
$\Psi; \Theta \vdash \Omega_1 <: \Omega_2 \implies [\![\Omega_1 \sigma \iota]\!] \subseteq [\![\Omega_2 \sigma \iota]\!]$

Proof. Proof by induction on $\Psi; \Theta \vdash \Omega_1 <: \Omega_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta \vdash \Omega <: .}\ \text{sub-mBase}$$

To prove: $\forall (0, T, \delta) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}.(0, T, \delta) \in [\![.]\!]_{\mathcal{E}}$

This means given some $(0, T, \delta) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(0, T, \delta) \in [\![.]\!]_{\mathcal{E}}$
We get the desired directly from Definition 16

2. sub-lInd:

$$\frac{x : \tau' \in \Omega_1 \qquad \Psi; \Theta \vdash \tau' <: \tau \qquad \Psi; \Theta \vdash \Omega_1/x <: \Omega_2}{\Psi; \Theta \vdash \Omega_1 <: \Omega_2, x : \tau}\ \text{sub-mInd}$$

To prove: $\forall (0, T, \delta) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}.(0, T, \delta) \in [\![\Omega_2, x : \tau]\!]_{\mathcal{E}}$

This means given some $(0, T, \delta) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(0, T, \delta) \in [\![\Omega_2, x : \tau]\!]_{\mathcal{E}}$

This means from Definition 16 we are given that
$(\forall x : \tau \in \Omega_1.\, (0, T, \delta(x)) \in [\![\tau]\!]_{\mathcal{E}})$ (L0)

Similarly from Definition 16 it suffices to prove that
$(\forall y : \tau_y \in (\Omega_2, x : \tau).(0, T, \delta(y)) \in [\![\tau_y]\!]_{\mathcal{E}})$

This means given some $y : \tau_y \in (\Omega_2, x : \tau)$ it suffices to prove that
$(0, T, \delta(y)) \in [\![\tau_y]\!]_{\mathcal{E}}$

From Lemma 26 we know that $\exists \tau'.y : \tau' \in \mathit{dom}(\Omega_1) \wedge \Psi; \Theta \vdash \tau' <: \tau_y$

Instantiating (L0) with $y : \tau'$ we get $(0, T, \delta(y)) \in [\![\tau']\!]_{\mathcal{E}}$

And finally from Lemma 22 we get the desired

□

Lemma 26 (Ω Subtyping: domain containment)**.** $\forall \Psi, \Theta, \Omega_1, \Omega_2.$
$\Psi; \Theta \vdash \Omega_1 <: \Omega_2 \implies \forall x : \tau \in \Omega_2.\ x : \tau' \in \Omega_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$

Proof. Proof by induction on $\Psi; \Theta \vdash \Omega_1 <: \Omega_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta \vdash \Omega <: .}\ \text{sub-mBase}$$

To prove: $\forall x : \tau \in (.).x : \tau' \in \Omega \wedge \Psi; \Theta \vdash \tau' <: \tau$
Trivial

2. sub-lInd:

$$\frac{x : \tau' \in \Omega_1 \qquad \Psi; \Theta \vdash \tau' <: \tau \qquad \Psi; \Theta \vdash \Omega_1/x <: \Omega_2}{\Psi; \Theta \vdash \Omega_1 <: \Omega_2, x : \tau}\ \text{sub-mInd}$$

To prove: $\forall y : \tau \in (\Omega_2, x : \tau_x).y : \tau' \in \Omega_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$

This means given some $y : \tau \in (\Omega_2, x : \tau)$ it suffices to prove that
$y : \tau' \in \Omega_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$
The following cases arise:

- $y = x$:
In this case we are given that
$x : \tau' \in \Omega_1 \wedge \Psi; \Theta \vdash \tau' <: \tau$
Therefore we are done
- $y \neq x$:
Since we are given that $\Psi; \Theta \vdash \Omega_1/x <: \Omega_2$ therefore we get the desired from IH

□

Theorem 27 (Soundness 1)**.** $\forall e, n, n', \tau \in Type, t.$
$\vdash e : \mathbb{M}\, n\, \tau \wedge e \Downarrow_t^{n'} v \implies n' \leq n$

Proof. From Theorem 20 we know that $(0, t+1, e) \in [\![\mathbb{M}\, n\, \tau]\!]_{\mathcal{E}}$

From Definition 15 this means we have
$\forall t' < t+1.e \Downarrow_{t'} v' \implies (0, t+1-t'v') \in [\![\mathbb{M}\, n\, \tau]\!]$

From the evaluation relation we know that $e \Downarrow_0 e$ therefore we have
$(0, t+1, e) \in [\![\mathbb{M}\, n\, \tau]\!]$

Again from Definition 15 it means we have
$\forall t'' < t+1.e \Downarrow_{t''}^{n'} v \implies \exists p'.n' + p' \leq 0 + n \wedge (p', t+1-t'', v) \in [\![\tau]\!]$

Since we are given that $e \Downarrow_t^{n'} v$ therefore we have
$\exists p'.n' + p' \leq n \wedge (p', 1, v) \in [\![\tau]\!]$

Since $p' \geq 0$ therefore we get $n' \leq n$

□

Theorem 28 (Soundness 2)**.** $\forall e, n, n', \tau \in Type.$
$\vdash e : [n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau \wedge e\, () \Downarrow_{t_1} - \Downarrow_{t_2}^{n'} v \implies n' \leq n$

Proof. From Theorem 20 we know that $(0, t_1 + t_2 + 2, e) \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$

Therefore from Definition 15 we know that
$\forall t' < t_1 + t_2 + 2, v.e \Downarrow_{t'} v \implies (0, t_1 + t_2 + 2 - t', v) \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]$ (S0)

Since we know that $e\, () \Downarrow_{t_1} -$ therefore from E-app we know that $\exists e'.e \Downarrow_{t_1} \lambda x.e'$

Instantiating (S0) with $t_1, \lambda x.e'$ we get $(0, t_2 + 2, \lambda x.e') \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]$

This means from Definition 15 we have
$\forall p', e', t'' < t_2 + 2.(p', t'', e'') \in [\![[n]\, \mathbf{1}]\!]_{\mathcal{E}} \implies (0 + p', t'', e'[e''/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$ (S1)

Claim: $\forall t.(I, t, ()) \in [\![[I]\, \mathbf{1}]\!]_{\mathcal{E}}$
Proof:
From Definition 15 it suffices to prove that
$() \Downarrow_0 v \implies (I, t, v) \in [\![[I]\, \mathbf{1}]\!]$

Since we know that $v = ()$ therefore it suffices to prove that
$(I, t, v) \in [\![[I]\, \mathbf{1}]\!]$

From Definition 15 it suffices to prove that
$\exists p'.p' + I \leq I \wedge (p', t, v) \in [\![\mathbf{1}]\!]\}$

We choose p' as 0 and we get the desired

Instantiating (S1) with $n, (), t_2+1$ we get $(n, t_2+1, e'[()/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$

This means again from Definition 15 we have
$\forall t' < t_2+1.e'[()/x] \Downarrow_{t'} v' \implies (n, t_2+1-t', v') \in [\![\mathbb{M}\, 0\, \tau]\!]$

From E-val we know that $v' = e'[()/x]$ and $t' = 0$ therefore we have
$(n, t_2+1, e'[()/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]$

Again from Definition 15 we have
$\forall t' < t_2+1.e'[()/x] \Downarrow^{n'}_{t'} v'' \implies \exists p'.n'+p' \leq n+0 \wedge (p', t_2+1-t', v'') \in [\![\tau]\!]$

Since we are given that $e \Downarrow_{t_1} - \Downarrow^{n'}_{t_2} v$ therefore we get
$\exists p'.n'+p' \leq n \wedge (p', 1, v'') \in [\![\tau]\!]$

Since $p' \geq 0$ therefore we have $n' \leq n$

□

Corollary 29 (Soundness)**.** $\forall \Gamma, e, q, q', \tau, T, p_l.$
$.; .; .; \Gamma \vdash e : [q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, [q']\, \tau\ \wedge$
$(p_l, T, \gamma) \in [\![\Gamma]\!]_{\mathcal{E}}\ \wedge$
$e\ ()\ \gamma \Downarrow_{t_1} v_t \Downarrow^{J}_{t_2} v\ \wedge$
$t_1 + t_2 < T$
$\implies$
$\exists p_v.(p_v, T - t_1 - t_2, v) \in [\![\tau]\!]\ \wedge\ J \leq (q+p_l) - (q'+p_v)$

Proof. From Theorem 20 we know that $(p_l, T, e) \in [\![[q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, [q']\, \tau]\!]_{\mathcal{E}}$

Therefore from Definition 15 we know that
$\forall\, T'<T, v.e\gamma \Downarrow_{T'} v \implies (p_l, T-T', v) \in [\![[q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, [q']\, \tau]\!]$ (S0)

Since we know that $e\ ()\gamma \Downarrow_{t_1} v_t$ therefore from E-app we know that
$\exists e'.e \Downarrow_{t'_1} \lambda x.e'$ and $e'[()/x] \Downarrow_{t''_1} v_t$ s.t $t'_1 + t''_1 + 1 = t_1$

Instantiating (S0) with $t'_1, \lambda x.e'$ we get $(p_l, T - t'_1, \lambda x.e') \in [\![[q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, [q']\, \tau]\!]$

This means from Definition 15 we have
$\forall p', T' < (T - t'_1), e'.(p', T', e'') \in [\![[q]\, \mathbf{1}]\!]_{\mathcal{E}} \implies (p_l + p', T', e'[e''/x]) \in [\![\mathbb{M}\, 0\, [q']\, \tau]\!]_{\mathcal{E}}$ (S1)

Claim: $\forall\, T\, .(I, T, ()) \in [\![[I]\, \mathbf{1}]\!]_{\mathcal{E}}$
Proof:
From Definition 15 it suffices to prove that
$\forall\, T''<T, v.() \Downarrow_{T''} v \implies (I, T-T'', v) \in [\![[I]\, \mathbf{1}]\!]$

From (E-val) we know that $T''= 0$ and $v = ()$ therefore it suffices to prove that
$(I, T, ()) \in [\![[I]\, \mathbf{1}]\!]$

From Definition 15 it further suffices to prove that
$\exists p'.p' + I \leq I \wedge (p', T, ()) \in [\![\mathbf{1}]\!]\}$

We choose p' as 0 and we get the desired

□

Using the claim we know that we have $(q, T - t'_1 - 1, ()) \in [\![[q]\, \mathbf{1}]\!]_{\mathcal{E}}$
Instantiating (S1) with $q, T - t'_1 - 1, ()$ and using the claim proved above we get
$(p_l + q, T - t'_1 - 1, e'[()/x]) \in [\![\mathbb{M}\, 0\, [q']\, \tau]\!]_{\mathcal{E}}$

This means again from Definition 15 we have
$\forall\, T_1 < T - t'_1 - 1.e'[()/x] \Downarrow v' \implies (p_l + q, T - t'_1 - 1 - T_1, v') \in [\![\mathbb{M}\, 0\, [q']\, \tau]\!]$

Instantiating with t''_1, v_t and since $t_1 < T$, therefore we also have $t''_1 < T - t'_1$.
Also since we are given that $e()\gamma \Downarrow_{t_1} v_t$, therefore we know that $v' = v_t$. Thus, we have
$(p_l + q, T - t'_1 - 1 - t''_1, v_t) \in [\![\mathbb{M}\, 0\, [q']\, \tau]\!]$

Again from Definition 15 we have
$\forall v'', t'_2 < T - t'_1 - t''_1 - 1.v_t \Downarrow^{J}_{t'_2} v'' \implies \exists p'.J + p' \leq p_l + q \wedge (p', T - t'_1 - t''_1 - 1 - t'_2, v'') \in [\![[q']\, \tau]\!]$

Instantiating with v, t_2 and since $t_2 < T - t'_1 - t''_1 - 1$ and $e \Downarrow_{t_1} v_t \Downarrow^{J}_{t_2} v$ therefore we get
$\exists p'.J + p' \leq p_l + q \wedge (p', T - t'_1 - t''_1 - 1 - t_2, v) \in [\![[q']\, \tau]\!]$ (S2)

Since we have $(p', T - t'_1 - t''_1 - 1 - t_2, v) \in [\![[q']\,\tau]\!]$ therefore from Definition 15 we have
$\exists p'_1 . p'_1 + q' \leq p' \wedge (p'_1, T - t'_1 - t''_1 - 1 - t_2, v) \in [\![\tau]\!]\}$ (S3)

In order to prove $\exists p_v.(p_v, T - t_1 - t_2, v) \in [\![\tau]\!] \wedge J \leq (q + p_l) - (q' + p_v)$ we choose p_v as p'_1 and we need to prove

1. $(p'_1, T - t_1 - t_2, v) \in [\![\tau]\!]$:
   Since from (S3) we have $(p'_1, T - t'_1 - t''_1 - 1 - t_2, v) \in [\![\tau]\!]$ and since $t'_1 + t''_1 + 1 = t_1$ therefore also have $(p'_1, T - t_1 - t_2, v) \in [\![\tau]\!]$
2. $J \leq (q + p_l) - (q' + p_v)$:
   From (S2) and (S3) we get
   $J \leq (p_l + q) - (q' + p'_1)$

□

A.5 Embedding Univariate RAML

Univariate RAML's type syntax

$$\begin{array}{llll} \text{Types} & \tau & ::= & \mathsf{b} \mid L^{\vec{q}}\,\tau \mid (\tau_1, \tau_2) \\ & A & ::= & \tau \xrightarrow{q/q'} \tau \end{array}$$

Type translation

$$\begin{array}{lcl} (\!|unit|\!) & = & \mathbf{1} \\ (\!|\mathsf{b}|\!) & = & !\mathsf{b} \\ (\!|L^{\vec{q}}\,\tau|\!) & = & \exists s.([\phi(\vec{q}, s)]\,\mathbf{1} \otimes L^s (\!|\tau|\!)) \\ (\!|(\tau_1, \tau_2)|\!) & = & ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \\ (\!|\tau_1 \xrightarrow{q/q'} \tau_2|\!) & = & ([q]\,\mathbf{1} \multimap (\!|\tau_1|\!) \multimap \mathbb{M}\,0\,[q']\,(\!|\tau_2|\!)) \end{array}$$

Type context translation

$$\begin{array}{lcl} (\!|.|\!) & = & . \\ (\!|\Gamma, x : \tau|\!) & = & (\!|\Gamma|\!), x : (\!|\tau|\!) \end{array}$$

Function context translation

$$\begin{array}{lcl} (\!|.|\!) & = & . \\ (\!|\Sigma, x : \tau|\!) & = & (\!|\Sigma|\!), x : (\!|\tau|\!) \end{array}$$

Judgment translation

$$\boxed{\Sigma; \Gamma \vdash^{q}_{q'} e_r : \tau \quad \rightsquigarrow \quad .; .; (\!|\Sigma|\!); (\!|\Gamma|\!) \vdash e_a : [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q'](\!|\tau|\!))}$$

Definition 30. $\phi(\vec{q}, n) \triangleq \sum_{1 \leq i \leq k} \binom{n}{i} q_i$ *as defined in [16, 19]*

Expression translation

$$\frac{}{\Sigma; . \vdash^{q+K^{unit}}_{q} () : unit \rightsquigarrow \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{unit}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{ret}(a)} \text{ unit}$$

$$\frac{}{\Sigma; . \vdash^{q+K^{base}}_{q} c : \mathsf{b} \rightsquigarrow \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{base}} \text{ in } \mathsf{bind}\, a = \mathsf{store}(!c) \text{ in } \mathsf{ret}(a)} \text{ base}$$

$$\frac{}{\Sigma; x : \tau \vdash^{q+K^{var}}_{q} x : \tau \rightsquigarrow \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{var}} \text{ in } \mathsf{bind}\, a = \mathsf{store}\, x \text{ in } \mathsf{ret}(a)} \text{ var}$$

$$\frac{\tau_1 \xrightarrow{q/q'} \tau_2 \in \Sigma(f)}{\Sigma; x : \tau_1 \vdash^{q+K^{app}_1}_{q'-K^{app}_2} f\ x : \tau_2 \rightsquigarrow \lambda u.E_0} \text{ app}$$

where
$E_0 = \mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{app}_1} \text{ in } \mathsf{bind}\, P = \mathsf{store}() \text{ in } E_1$
$E_1 = \mathsf{bind}\, f_1 = (f\ P\ x) \text{ in } \mathsf{release}\, f_2 = f_1 \text{ in } \mathsf{bind} - = \uparrow^{K^{app}_2} \text{ in } \mathsf{bind}\, f_3 = \mathsf{store}\, f_2 \text{ in } \mathsf{ret}\, f_3$

$$\frac{}{\Sigma; \emptyset \vdash^{q+K^{nil}}_{q} nil : L^{\vec{p}}\tau \rightsquigarrow \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{nil}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil \rangle\!\rangle \text{ in } \mathsf{ret}(b)} \text{ nil}$$

$$\frac{\vec{p} = (p_1, \ldots, p_k)}{\Sigma; x_h : \tau, x_t : L^{(\lhd\ \vec{p})}\tau \vdash_{q}^{q+p_1+K^{cons}} cons(x_h, x_t) : L^{p}\tau \rightsquigarrow \quad \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0} \text{ cons}$$

where
$E_0 = x_t; x.\, \mathsf{let}\langle\langle x_1, x_2\rangle\rangle = x \text{ in } E_1$
$E_1 = \mathsf{release} - = x_1 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\langle a, x_h :: x_2\rangle\rangle \text{ in } \mathsf{ret}(b)$

$$\frac{\Sigma; \Gamma \vdash_{q'+K_2^{matN}}^{q-K_1^{matN}} e_1 : \tau' \rightsquigarrow e_{a1} \qquad \vec{p} = (p_1, \ldots, p_k) \qquad \Sigma; \Gamma, h : \tau, t : L^{(\lhd\ \vec{p})}\tau \vdash_{q'+K_2^{matN}}^{q+p_1-K_1^{matC}} e_2 : \tau' \rightsquigarrow e_{a2}}{\Sigma; \Gamma; x : L^{p}\tau \vdash_{q'}^{q} \mathsf{match}\ x \text{ with } |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau' \rightsquigarrow \quad \lambda u.E_0} \text{ match}$$

where
$E_0 = \mathsf{release} - = u \text{ in } E_{0.1}$
$E_{0.1} = x; a.\, \mathsf{let}\langle\langle x_1, x_2\rangle\rangle = a \text{ in } E_1$
$E_1 = \mathsf{match}\ x_2 \text{ with } |nil \mapsto E_2\ |h :: l_t \mapsto E_3$
$E_2 = \mathsf{bind} - = \uparrow^{K_1^{matN}} \text{ in } E_{2.1}$
$E_{2.1} = \mathsf{bind}\, b = \mathsf{store}() \text{ in } E_2'$
$E_2' = \mathsf{bind}\, c = (e_{a1}\ b) \text{ in } E_{2.1}'$
$E_{2.1}' = \mathsf{release}\, d = c \text{ in } E_{2.2}'$
$E_{2.2}' = \mathsf{bind} - = \uparrow^{K_2^{matN}} \text{ in } E_{2.3}'$
$E_{2.3}' = \mathsf{release} - = x_1 \text{ in } \mathsf{store}\, d$
$E_3 = \mathsf{bind} - = \uparrow^{K_1^{matC}} \text{ in } E_{3.1}$
$E_{3.1} = \mathsf{release} - = x_1 \text{ in } E_{3.2}$
$E_{3.2} = \mathsf{bind}\, b = \mathsf{store}() \text{ in } E_{3.3}$
$E_{3.3} = \mathsf{bind}\, t = \mathsf{ret}\langle\langle b, l_t\rangle\rangle \text{ in } E_{3.4}$
$E_{3.4} = \mathsf{bind}\, d = \mathsf{store}() \text{ in } E_{3.5}$
$E_{3.5} = \mathsf{bind}\, f = e_{a2}\ d \text{ in } E_{3.6}$
$E_{3.6} = \mathsf{release}\, g = f \text{ in } E_{3.7}$
$E_{3.7} = \mathsf{bind} - = \uparrow^{K_2^{matC}} \text{ in } \mathsf{store}\, g$

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = \tau_1 = \tau_2 = \mathbf{1}}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-unit}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\mathbf{1},\mathbf{1},\mathbf{1}}\ z \text{ in } \mathsf{let}\langle\langle x, y\rangle\rangle = a \text{ in } e_a\ u$

$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} : (\!|\mathbf{1}|\!) \multimap \mathbb{M}\,0\,((\!|\mathbf{1}|\!) \otimes (\!|\mathbf{1}|\!))$
$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} \triangleq \lambda u.\, \mathsf{ret}\langle\langle !(), !()\rangle\rangle$

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = \tau_1 = \tau_2 = \mathsf{b}}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-base}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\mathsf{b},\mathsf{b},\mathsf{b}}\ z \text{ in } \mathsf{let}\langle\langle x, y\rangle\rangle = a \text{ in } e_a\ u$

$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} : (\!|\mathsf{b}|\!) \multimap \mathbb{M}\,0\,((\!|\mathsf{b}|\!) \otimes (\!|\mathsf{b}|\!))$
$coerce_{\mathsf{b},\mathsf{b},\mathsf{b}} \triangleq \lambda u.\, \mathsf{let}\,!\,u' = u \text{ in } \mathsf{ret}\langle\langle !u', !u'\rangle\rangle$

$$\frac{\begin{array}{c}\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \\ \tau = L^{\vec{p}}\tau'' \qquad \tau_1 = L^{\vec{p_1}}\tau_1'' \qquad \tau_2 = L^{\vec{p_2}}\tau_2'' \qquad \tau'' = \tau_1'' \curlyvee \tau_2'' \qquad \vec{p} = \vec{p_1} + \vec{p_2}\end{array}}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-list}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\tau,\tau_1,\tau_2}\ z \text{ in } \mathsf{let}\langle\langle x, y\rangle\rangle = a \text{ in } e_a\ u$

$coerce_{L^{\vec{p}}\tau, L^{\vec{p_1}}\tau_1, L^{\vec{p_2}}\tau_2} :!((\!|\tau|\!) \multimap \mathbb{M}\,0\,(\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \multimap (\!|L^{\vec{p}}\tau|\!) \multimap \mathbb{M}\,0\,(\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!)$
$coerce_{L^{\vec{p}}\tau, L^{\vec{p_1}}\tau_1, L^{\vec{p_2}}\tau_2} \triangleq \mathsf{fix} f.\lambda\_g.\lambda e.\, \mathsf{let}\,!\,g' = g \text{ in } e; x.\, \mathsf{let}\langle\langle p, l\rangle\rangle = x \text{ in } E_0$

where

$E_0 \triangleq \mathsf{release} - = p \text{ in } E_1$
$E_1 \triangleq \mathsf{match}\ l \text{ with } |nil \mapsto E_{2.1}\ |h :: t \mapsto E_3$
$E_{2.1} \triangleq \mathsf{bind}\, z_1 = \mathsf{store}() \text{ in } E_{2.2}$
$E_{2.2} \triangleq \mathsf{bind}\, z_2 = \mathsf{store}() \text{ in } E_{2.3}$
$E_{2.3} \triangleq \mathsf{ret}\langle\!\langle \langle\!\langle z_1, nil \rangle\!\rangle, \langle\!\langle z_2, nil \rangle\!\rangle \rangle\!\rangle$
$E_3 \triangleq \mathsf{bind}\, H = g'\ h \text{ in } E_{3.1}$
$E_{3.1} \triangleq \mathsf{bind}\, o_t = () \text{ in } E_{3.2}$
$E_{3.2} \triangleq \mathsf{bind}\, T = f\ g\ \langle\!\langle o_t, t \rangle\!\rangle \text{ in } E_4$
$E_4 \triangleq \mathsf{let}\langle\!\langle H_1, H_2 \rangle\!\rangle = H \text{ in } E_5$
$E_5 \triangleq \mathsf{let}\langle\!\langle T_1, T_2 \rangle\!\rangle = T \text{ in } E_6$
$E_6 \triangleq T_1; tp_1.\, \mathsf{let}\langle\!\langle p'_1, l'_1 \rangle\!\rangle = tp_1 \text{ in } E_{7.1}$
$E_{7.1} \triangleq T_2; tp_2.\, \mathsf{let}\langle\!\langle p'_2, l'_2 \rangle\!\rangle = tp_2 \text{ in } E_{7.2}$
$E_{7.2} \triangleq \mathsf{release} - = p'_1 \text{ in } E_{7.3}$
$E_{7.3} \triangleq \mathsf{release} - = p'_2 \text{ in } E_{7.4}$
$E_{7.4} \triangleq \mathsf{bind}\, o_1 = \mathsf{store}() \text{ in } E_{7.5}$
$E_{7.5} \triangleq \mathsf{bind}\, o_2 = \mathsf{store}() \text{ in } E_8$
$E_8 \triangleq \mathsf{ret}\langle\!\langle \langle\!\langle o_1, H_1 :: T_1 \rangle\!\rangle, \langle\!\langle o_2, H_2 :: T_2 \rangle\!\rangle \rangle\!\rangle$

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash^q_{q'} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = (\tau_a, \tau_b) \qquad \tau_1 = (\tau'_a, \tau'_b) \qquad \tau_2 = (\tau''_a, \tau''_b)}{\Sigma; \Gamma, z : \tau \vdash^q_{q'} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{Share-pair}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{(\tau_a,\tau_b),(\tau'_a,\tau'_b),(\tau''_a,\tau''_b)}\ z \text{ in } \mathsf{let}\langle\!\langle x, y \rangle\!\rangle = a \text{ in } e_a\ u$

$coerce_{(\tau_a,\tau_b),(\tau'_a,\tau_b),(\tau''_a,\tau''_b)} : !(\llparenthesis \tau_a \rrparenthesis \multimap \mathbb{M}\,0\, \llparenthesis \tau'_a \rrparenthesis \otimes \llparenthesis \tau''_a \rrparenthesis) \multimap !(\llparenthesis \tau_b \rrparenthesis \multimap \mathbb{M}\,0\, \llparenthesis \tau'_b \rrparenthesis \otimes \llparenthesis \tau''_b \rrparenthesis) \multimap \llparenthesis (\tau_a, \tau_b) \rrparenthesis \multimap \mathbb{M}\,0\, \llparenthesis (\tau'_a, \tau'_b) \rrparenthesis \otimes \llparenthesis (\tau''_a, \tau''_b) \rrparenthesis$
$coerce_{(\tau_a,\tau_b),(\tau'_a,\tau'_b),(\tau''_a,\tau''_b)} \triangleq \lambda\_g_1.\lambda\_g_2.\lambda p.\, \mathsf{let}\, !\langle\!\langle p_1, p_2 \rangle\!\rangle = p \text{ in } E_0$
where
$E_0 \triangleq \mathsf{let}\, !\, g'_1 = g_1 \text{ in } E_1$
$E_1 \triangleq \mathsf{let}\, !\, g'_2 = g_2 \text{ in } E_2$
$E_2 \triangleq \mathsf{bind}\, P'_1 = g'_1 p_1 \text{ in } E_3$
$E_3 \triangleq \mathsf{bind}\, P'_2 = g'_2 p_2 \text{ in } E_4$
$E_4 \triangleq \mathsf{let}\, !\langle\!\langle p'_{11}, p'_{12} \rangle\!\rangle = P'_1 \text{ in } E_5$
$E_5 \triangleq \mathsf{let}\, !\langle\!\langle p'_{21}, p'_{22} \rangle\!\rangle = P'_2 \text{ in } E_6$
$E_6 \triangleq \mathsf{ret}\langle\!\langle p'_{11}, p'_{21} \rangle\!\rangle, \langle\!\langle p'_{12}, p'_{22} \rangle\!\rangle$

$$\frac{\Sigma; \Gamma \vdash^q_{q'} e : \tau \rightsquigarrow e_a \qquad \tau <: \tau'}{\Sigma; \Gamma \vdash^q_{q'} e : \tau' \rightsquigarrow e_a} \text{Sub}$$

$$\frac{\Sigma; \Gamma, x : \tau_1 \vdash^q_{q'} e : \tau \rightsquigarrow e_a \qquad \tau'_1 <: \tau_1}{\Sigma; \Gamma, x : \tau'_1 \vdash^q_{q'} e : \tau \rightsquigarrow e_a} \text{Super}$$

$$\frac{\Sigma; \Gamma \vdash^p_{p'} e : \tau \rightsquigarrow e_a \qquad q \geq p \qquad q - p \geq q' - p'}{\Sigma; \Gamma \vdash^q_{q'} e : \tau \rightsquigarrow \lambda o.E_0} \text{Relax}$$

where
$E_0 = \mathsf{release} - = o \text{ in } E_1$
$E_1 = \mathsf{bind}\, a = \mathsf{store}() \text{ in } E_2$
$E_2 = \mathsf{bind}\, b = e_a\ a \text{ in } E_3$
$E_3 = \mathsf{release}\, c = b \text{ in } \mathsf{store}\, c$

$$\frac{\Sigma; \Gamma_1 \vdash^{q - K^{let}_1}_{p} e_1 : \tau_1 \rightsquigarrow e_{a1} \qquad \Sigma; \Gamma_2, x : \tau_1 \vdash^{p - K^{let}_2}_{q' + K^{let}_3} e_2 : \tau_1 \rightsquigarrow e_{a2}}{\Sigma; \Gamma_1, \Gamma_2 \vdash^q_{q'} \mathsf{let}\ x = e_1\ in\ e_2 : \tau \rightsquigarrow E_t} \text{Let}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u \text{ in } E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K^{let}_1} \text{ in } E_2$
$E_2 = \mathsf{bind}\, a = \mathsf{store}() \text{ in } E_3$
$E_3 = \mathsf{bind}\, b = e_{a1}\ a \text{ in } E_4$

$E_4 = \mathsf{release}\, x = b \;\mathsf{in}\; E_5$
$E_5 = \mathsf{bind} - = \uparrow^{K_2^{let}} \;\mathsf{in}\; E_6$
$E_6 = \mathsf{bind}\, c = \mathsf{store}() \;\mathsf{in}\; E_7$
$E_7 = \mathsf{bind}\, d = e_{a2}\; c \;\mathsf{in}\; E_8$
$E_8 = \mathsf{release}\, f = d \;\mathsf{in}\; E_9$
$E_9 = \mathsf{bind} - = \uparrow^{K_3^{let}} \;\mathsf{in}\; E_{10}$
$E_{10} = \mathsf{bind}\, g = \mathsf{store}\, f \;\mathsf{in}\; \mathsf{ret}\, g$

$$\frac{}{\Sigma; x_1 : \tau_1, x_2 : \tau_2 \vdash_{q}^{q+K^{pair}} (x_1, x_2) : (\tau_1, \tau_2) \leadsto E_t} \text{ pair}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u \;\mathsf{in}\; E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K^{pair}} \;\mathsf{in}\; E_2$
$E_2 = \mathsf{bind}\, a = \mathsf{store}(x_1, x_2) \;\mathsf{in}\; \mathsf{ret}\, a$

$$\frac{\tau = (\tau_1, \tau_2) \qquad \Sigma, \Gamma, x_1 : \tau_1, x_2 : \tau_2 \vdash_{q'+K_2^{matP}}^{q-K_1^{matP}} e : \tau' \leadsto e_t}{\Sigma; \Gamma, x : \tau \vdash_{q}^{q} \textit{match } x \textit{ with } (x_1, x_2) \to e : \tau' \leadsto E_t} \text{ matP}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u \;\mathsf{in}\; E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K_1^{matP}} \;\mathsf{in}\; E_2$
$E_2 = \mathsf{let}\langle\!\langle x_1, x_2 \rangle\!\rangle = x \;\mathsf{in}\; E_3$
$E_3 = \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; E_4$
$E_4 = \mathsf{bind}\, b = e_t\; a \;\mathsf{in}\; E_5$
$E_5 = \mathsf{release}\, c = b \;\mathsf{in}\; E_6$
$E_6 = \mathsf{bind} - = \uparrow^{K_2^{matP}} \;\mathsf{in}\; E_7$
$E_7 = \mathsf{bind}\, d = \mathsf{store}\, c \;\mathsf{in}\; \mathsf{ret}\, d$

$$\frac{\Sigma; \Gamma \vdash_{q'}^{q} e : \tau \leadsto e_a}{\Sigma; \Gamma, x : \tau' \vdash_{q'}^{q} e : \tau \leadsto e_a} \text{ Augment}$$

A.5.1 Type preservation

Theorem 31 (Type preservation: Univariate RAML to λ-amor)**.** *If $\Sigma; \Gamma \vdash_{q'}^{q} e : \tau$ in Univariate RAML then there exists e' such that $\Sigma; \Gamma \vdash_{q'}^{q} e : \tau \leadsto e'$ such that there is a derivation of $.; .; .; (\!|\Sigma|\!), (\!|\Gamma|\!) \vdash e' : [q]\,\mathbf{1} \multimap \mathbb{M}\, 0\, ([q'] (\!|\tau|\!))$ in λ-amor.*

Proof. By induction on $\Sigma; \Gamma \vdash_{q'}^{q} e : \tau$

1. unit:

$$\frac{}{\Sigma; . \vdash_{q}^{q+K^{unit}} () : \textit{unit} \leadsto \lambda u.\mathsf{release} - = u \;\mathsf{in}\; \mathsf{bind} - = \uparrow^{K^{unit}} \;\mathsf{in}\; \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; \mathsf{ret}(a)} \text{ unit}$$

$E_0 = \lambda u.\mathsf{release} - = u \;\mathsf{in}\; \mathsf{bind} - = \uparrow^{K^{unit}} \;\mathsf{in}\; \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; \mathsf{ret}(a)$
$E_1 = \mathsf{release} - = u \;\mathsf{in}\; \mathsf{bind} - = \uparrow^{K^{unit}} \;\mathsf{in}\; \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; \mathsf{ret}(a)$
$T_0 = [q + K^{unit}]\,\mathbf{1} \multimap \mathbb{M}\, 0\, ([q] (\!|\textit{unit}|\!))$
$T_1 = [q + K^{unit}]\,\mathbf{1}$
$T_2 = \mathbb{M}(q + K^{unit})\, ([q]\,\mathbf{1})$
$T_{2.1} = \mathbb{M}(q)\, ([q]\,\mathbf{1})$
$T_3 = \mathbb{M}\, K^{unit}\, \mathbf{1}$
$T_4 = \mathbb{M}\, 0\, ([q]\,\mathbf{1})$
$T_5 = \mathbb{M}\, q\, ([q]\,\mathbf{1})$
D1:

$$\frac{\frac{}{.; .; .; (\!|\Sigma|\!); . \vdash \mathsf{store}() : T_5} \qquad \frac{}{.; .; .; (\!|\Sigma|\!); a : [q]\,\mathbf{1} \vdash \mathsf{ret}(a) : T_4}}{.; .; .; (\!|\Sigma|\!); . \vdash \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; \mathsf{ret}(a) : T_5}$$

D0:

$$\frac{}{.; .; .; (\!|\Sigma|\!); . \vdash \uparrow^{K^{unit}} : T_3}$$

D0.0:

$$\frac{D0 \qquad D1}{.;.;.;(\!|\Sigma|\!);. \vdash \mathsf{bind}\, - = \uparrow^{K^{unit}} \mathsf{in\ bind}\, a = \mathsf{store}()\ \mathsf{in\ ret}(a) : T_2}\ \text{T-bind}$$

Main derivation:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);u : T_1 \vdash u : T_1}\ \text{T-var} \qquad D0.0}{.;.;.;(\!|\Sigma|\!);u : T_1 \vdash E_1 : T_4}\ \text{T-release}}{.;.;.;(\!|\Sigma|\!);. \vdash E_0 : T_0}\ \text{T-lam}$$

2. base:

$$\frac{}{\Sigma;. \vdash_q^{q+K^{base}} c : \mathsf{b} \rightsquigarrow \lambda u.\mathsf{release}\, - = u\ \mathsf{in\ \ bind}\, - = \uparrow^{K^{base}} \mathsf{in\ bind}\, a = \mathsf{store}(!c)\ \mathsf{in\ ret}(a)}\ \text{base}$$

$E_0 = \lambda u.\mathsf{release}\, - = u\ \mathsf{in\ \ bind}\, - = \uparrow^{K^{base}} \mathsf{in\ bind}\, a = \mathsf{store}(!c)\ \mathsf{in\ ret}(a)$
$E_1 = \mathsf{release}\, - = u\ \mathsf{in\ \ bind}\, - = \uparrow^{K^{base}} \mathsf{in\ bind}\, a = \mathsf{store}(!c)\ \mathsf{in\ ret}(a)$
$T_0 = [q + K^{base}]\, \mathbf{1} \multimap \mathbb{M}\, 0\, ([q](\!|\mathsf{b}|\!))$
$T_1 = [q + K^{base}]\, \mathbf{1}$
$T_2 = \mathbb{M}(q + K^{base})\, ([q]\, !\mathsf{b})$
$T_{2.1} = \mathbb{M}(q)\, ([q]\, !\mathsf{b})$
$T_3 = \mathbb{M}\, K^{base}\, (\!|\mathbf{1}|\!)$
$T_4 = \mathbb{M}\, 0\, ([q]\, !\mathsf{b})$
$T_5 = \mathbb{M}\, q\, ([q]\, !\mathsf{b})$
D1:

$$\frac{\dfrac{}{.;.;.;(\!|\Sigma|\!);. \vdash \mathsf{store}(!c) : T_5} \qquad \dfrac{}{.;.;.;(\!|\Sigma|\!);a : [q]\, !\mathsf{b} \vdash \mathsf{ret}(a) : T_4}}{.;.;.;(\!|\Sigma|\!);. \vdash \mathsf{bind}\, a = \mathsf{store}(!c)\ \mathsf{in\ ret}(a) : T_{2.1}}$$

D0:

$$\frac{}{.;.;.;(\!|\Sigma|\!);. \vdash \uparrow^{K^{base}} : T_3}$$

D0.0:

$$\frac{D0 \qquad D1}{.;.;.;(\!|\Sigma|\!);. \vdash \mathsf{bind}\, - = \uparrow^{K^{base}} \mathsf{in\ bind}\, a = \mathsf{store}(!c)\ \mathsf{in\ ret}(a) : T_2}\ \text{T-bind}$$

Main derivation:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);u : T_1 \vdash u : T_1}\ \text{T-var} \qquad D0.0}{.;.;.;(\!|\Sigma|\!);u : T_1 \vdash E_1 : T_4}\ \text{T-release}}{.;.;.;(\!|\Sigma|\!);. \vdash E_0 : T_0}\ \text{T-lam}$$

3. var:

$$\frac{}{\Sigma; x : \tau \vdash_q^{q+K^{var}} x : \tau \rightsquigarrow \lambda u.\, \mathsf{bind}\, - = \uparrow^{K^{var}} \mathsf{in\ ret}(x)}\ \text{var}$$

$E_0 = \lambda u.\mathsf{release}\, - = u\ \mathsf{in\ \ bind}\, - = \uparrow^{K^{var}} \mathsf{in\ bind}\, a = \mathsf{store}\, x\ \mathsf{in\ ret}(a)$
$E_1 = \mathsf{release}\, - = u\ \mathsf{in\ \ bind}\, - = \uparrow^{K^{var}} \mathsf{in\ bind}\, a = \mathsf{store}\, x\ \mathsf{in\ ret}(a)$
$T_0 = [q + K^{var}]\, \mathbf{1} \multimap \mathbb{M}\, 0\, ([q](\!|\tau|\!))$
$T_1 = [q + K^{var}]\, \mathbf{1}$
$T_2 = \mathbb{M}\, 0\, ([q + K^{var}](\!|\tau|\!))$
$T_3 = \mathbb{M}\, K^{var}\, (\!|\mathbf{1}|\!)$
$T_4 = \mathbb{M}\, 0\, ([q](\!|\tau|\!))$
$T_5 = \mathbb{M}\, q\, ([q](\!|\tau|\!))$
D1:

$$\frac{\dfrac{}{.;.;.;(\!|\Sigma|\!);x : (\!|\tau|\!) \vdash \mathsf{store}\, x : T_5} \qquad \dfrac{}{.;.;.;(\!|\Sigma|\!);a : [q]\, (\!|\tau|\!) \vdash \mathsf{ret}(a) : T_4}}{.;.;.;(\!|\Sigma|\!);x : (\!|\tau|\!) \vdash \mathsf{bind}\, a = \mathsf{store}\, x\ \mathsf{in\ ret}(a) : T_5}$$

D0:

$$\frac{}{.;.;.;(\!|\Sigma|\!);x : (\!|\tau|\!) \vdash \uparrow^{K^{var}} : T_3}$$

D0.0:

$$\frac{D0 \qquad D1}{.;.;.;(\!|\Sigma|\!);x : (\!|\tau|\!) \vdash \mathsf{bind}\, - = \uparrow^{K^{var}} \mathsf{in\ bind}\, a = \mathsf{store}\, x\ \mathsf{in\ ret}(a) : T_2}\ \text{T-bind}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);u:T_1\vdash u:T_1}\,\text{T-var}\qquad D0.0}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau|\!),u:T_1\vdash E_1:T_4}\,\text{T-release}}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau|\!)\vdash E_0:T_0}\,\text{T-lam}$$

4. app:

$$\dfrac{\tau_1 \overset{q/q'}{\to} \tau_2 \in \Sigma(f)}{\Sigma; x:\tau_1 \vdash^{q+K_1^{app}}_{q'-K_2^{app}} f\ x : \tau_2 \rightsquigarrow \lambda u.E_0}\,\text{app}$$

where
$E_0 = \mathsf{release} - = u \mathsf{\ in\ \ bind} - = \uparrow^{K_1^{app}} \mathsf{\ in\ bind\ } P = \mathsf{store}() \mathsf{\ in\ } E_1$
$E_1 = \mathsf{bind\ } f_1 = (f\ P\ x) \mathsf{\ in\ release\ } f_2 = f_1 \mathsf{\ in\ \ bind} - = \uparrow^{K_2^{app}} \mathsf{\ in\ bind\ } f_3 = \mathsf{store\ } f_2 \mathsf{\ in\ ret\ } f_3$
$E_{1.1} = \mathsf{release\ } f_2 = f_1 \mathsf{\ in\ \ bind} - = \uparrow^{K_2^{app}} \mathsf{\ in\ bind\ } f_3 = \mathsf{store\ } f_2 \mathsf{\ in\ ret\ } f_3$
$E_{1.2} = \mathsf{bind} - = \uparrow^{K_2^{app}} \mathsf{\ in\ bind\ } f_3 = \mathsf{store\ } f_2 \mathsf{\ in\ ret\ } f_3$
$E_{1.3} = \mathsf{bind\ } f_3 = \mathsf{store\ } f_2 \mathsf{\ in\ ret\ } f_3$
$E_{1.4} = \mathsf{store\ } f_2$
$E_{1.5} = \mathsf{ret\ } f_3$
$E_{0.1} = \mathsf{bind} - = \uparrow^{K_1^{app}} \mathsf{\ in\ bind\ } F = f \mathsf{\ in\ } E_1$

$T_0 = [q + K_1^{app}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q' - K_2^{app}](\!|\tau|\!))$
$T_{0.1} = [q + K_1^{app}]\,\mathbf{1}$
$T_{0.2} = \mathbb{M}\,0\,([q' - K_2^{app}](\!|\tau_2|\!))$
$T_1 = \mathbb{M}(q + K_1^{app})\,\mathbf{1}$
$T_{1.2} = \mathbb{M}\,0\,[q' - K_2^{app}]\,(\!|\tau_2|\!)$
$T_2 = \mathbb{M}(K_1^{app})\,\mathbf{1}$
$T_3 = \mathbb{M}(q)\,(\!|\tau_2|\!)$
$T_4 = \mathbb{M}\,q\,((\!|\tau_1|\!) \multimap \mathbb{M}\,0\,[q']\,(\!|\tau_2|\!))$
$T_{4.1} = ((\!|\tau_1|\!) \multimap \mathbb{M}\,0\,[q']\,(\!|\tau_2|\!))$
$T_{4.2} = \mathbb{M}\,0\,[q']\,(\!|\tau_2|\!)$
$T_{4.3} = [q']\,(\!|\tau_2|\!)$
$T_{4.4} = \mathbb{M}(\,q' - K_2^{app})[q' - K_2^{app}]\,(\!|\tau_2|\!)$
$T_{4.41} = \mathbb{M}(\,q')[q' - K_2^{app}]\,(\!|\tau_2|\!)$
$T_{4.5} = [q' - K_2^{app}]\,(\!|\tau_2|\!)$
$T_{4.6} = \mathbb{M}\,0\,[q' - K_2^{app}]\,(\!|\tau_2|\!)$

D2.3:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);f_2:(\!|\tau_2|\!)\vdash E_{1.4}:T_{4.4}}\qquad \dfrac{}{.;.;.;(\!|\Sigma|\!);f_3:T_{4.5}\vdash E_{1.5}:T_{4.6}}}{.;.;.;(\!|\Sigma|\!);f_2:(\!|\tau_2|\!),f_3:T_{4.5}\vdash E_{1.3}:T_{4.4}}$$

D2.2:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);.\vdash \uparrow^{K_2^{app}}:\mathbb{M}\,K_2^{app}\,\mathbf{1}}\qquad D2.3}{.;.;.;(\!|\Sigma|\!);f_2:(\!|\tau_2|\!)\vdash E_{1.2}:T_{4.41}}$$

D2.1:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);f_1:T_{4.3}\vdash f_1:T_{4.3}}\qquad D2.2}{.;.;.;(\!|\Sigma|\!);f_1:T_{4.3}\vdash E_{1.1}:T_{1.2}}$$

D2:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!),P:[q]\,\mathbf{1}\vdash f\ \ P\ \ x:T_{4.2}}\qquad D2.1}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!),P:[q]\,\mathbf{1}\vdash E_1:T_{1.2}}$$

D1:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);.\vdash \mathsf{store}():\mathbb{M}\,q\,[q]\,\mathbf{1}}\qquad D2}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!)\vdash \mathsf{bind\ } P=\mathsf{store}()\mathsf{\ in\ } E_1:T_{1.2}}$$

D0:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!)\vdash \uparrow^{K_1^{app}}:T_1}\qquad D1}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!)\vdash E_{0.1}:T_{1.2}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);u:T_{0.1}\vdash u:T_{0.1}}\text{T-var}\qquad D0}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!),u:T_{0.1}\vdash E_0:T_{0.2}}}{.;.;.;(\!|\Sigma|\!);x:(\!|\tau_1|\!)\vdash \lambda u.E_0:T_0}$$

5. nil:

$$\dfrac{}{\Sigma;\emptyset\vdash_q^{q+K^{nil}} nil : L^{\vec{p}}\tau \rightsquigarrow \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{nil}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)}\text{nil}$$

$E_0 = \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{nil}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)$
$E_1 = \mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{nil}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)$
$E_2 = \mathsf{bind} - = \uparrow^{K^{nil}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)$
$E_3 = \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)$
$E_4 = \mathsf{bind}\, b = \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle \text{ in } \mathsf{ret}(b)$
$E_5 = \mathsf{ret}(b)$

$T_0 = [q + K^{nil}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q]\,\exists n.\phi(\vec{p}, n) \otimes \text{list}[n](\!|\tau|\!))$
$T_1 = [(q + K^{nil})]\,\mathbf{1}$
$T_2 = \mathbb{M}\,0\,([q]\,\exists n.\,[\phi(\vec{p}, n)]\,\mathbf{1} \otimes \text{list}[n](\!|\tau|\!))$
$T_3 = \mathbb{M}(q + K^{nil})\,([q]\,\exists n.\,[\phi(\vec{p}, n)]\,\mathbf{1} \otimes \text{list}[n](\!|\tau|\!))$
$T_4 = \mathbb{M}\,K^{nil}\,\mathbf{1}$
$T_5 = \mathbb{M}(q)\,([q]\,\exists n.\,[\phi(\vec{p}, n)]\,\mathbf{1} \otimes \text{list}[n](\!|\tau|\!))$
$T_{5.1} = ([q]\,\exists n.\,[\phi(\vec{p}, n)]\,\mathbf{1} \otimes \text{list}[n](\!|\tau|\!))$
$T_6 = \mathbb{M}(0)\,([q]\,\exists n.\,[\phi(\vec{p}, n)]\,\mathbf{1} \otimes \text{list}[n](\!|\tau|\!))$

D4:

$$\dfrac{\dfrac{\dfrac{\phi(\vec{p},0)=0}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash a:[0]\,\mathbf{1}}\qquad \dfrac{}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash nil:\text{list}[0](\!|\tau|\!)}}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash \langle\!\langle a, nil\rangle\!\rangle : T_6[0/n]}}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash \langle\!\langle a, nil\rangle\!\rangle : T_6}$$

D3:

$$\dfrac{}{.;.;.;(\!|\Sigma|\!);b:T_{5.1}\vdash E_5:T_6}$$

D2:

$$\dfrac{\dfrac{D4}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash \mathsf{store}\langle\!\langle a, nil\rangle\!\rangle : T_5}\qquad D3}{.;.;.;(\!|\Sigma|\!);a:[0]\,\mathbf{1}\vdash E_4:T_5}$$

D1:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);.\vdash \mathsf{store}():\mathbb{M}\,0\,[0]\,\mathbf{1}}\qquad D2}{.;.;.;(\!|\Sigma|\!);.\vdash E_3:T_5}$$

D0:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);.\vdash \uparrow^{K^{nil}}:T_4}\qquad D1}{.;.;.;(\!|\Sigma|\!);.\vdash E_2:T_3}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);u:T_1\vdash u:T_1}\qquad D0}{.;.;.;(\!|\Sigma|\!);u:T_1\vdash E_1:T_2}}{.;.;.;(\!|\Sigma|\!);.\vdash E_0:T_0}$$

6. cons:

$$\dfrac{\vec{p}=(p_1,\ldots,p_k)}{\Sigma;x_h:\tau,x_t:L^{(\lhd\,\vec{p})}\tau\vdash_q^{q+p_1+K^{cons}} cons(x_h,x_t):L^{p}\tau \rightsquigarrow \quad \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0}\text{cons}$$

where

$E_0 = x_t; x.\,\mathsf{let}\langle\!\langle x_1, x_2\rangle\!\rangle = x \text{ in } E_1$
$E_1 = \mathsf{release} - = x_1 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{store}\langle\!\langle a, x_h :: x_2\rangle\!\rangle$

$T_0 = [q + p_1 + K^{cons}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q]\,\exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!))$
$T_1 = [q + p_1 + K^{cons}]\,\mathbf{1}$
$T_2 = \mathbb{M}\,0\,([q]\,\exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!))$
$T_{2.1} = \mathbb{M}(q + p_1)\,([q]\,\exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!))$
$T_{2.2} = \mathbb{M}(q + p_1 + \phi(\lhd\vec{p}, s))\,([q]\,\exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!))$
$T_{2.3} = \mathbb{M}(q)\,([q]\,\exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!))$
$T_{2.4} = \exists n'.\,[\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!)$
$T_{2.5} = [\phi(\vec{p}, n')]\,\mathbf{1} \otimes L^{n'}(\!|\tau|\!)$
$T_3 = [(p_1 + \phi(\lhd\vec{p}, s))]\,\mathbf{1}$
$T_l = \exists s.([\phi(\lhd\vec{p}, s)]\,\mathbf{1} \otimes L^{s}(\!|\tau|\!))$
$T_{l1} = ([\phi(\lhd\vec{p}, s)]\,\mathbf{1} \otimes L^{s}(\!|\tau|\!))$
$T_{l2} = [\phi(\lhd\vec{p}, s)]\,\mathbf{1}$
$T_{l3} = L^{s}(\!|\tau|\!)$

D1.4:

$$\dfrac{\dfrac{\dfrac{}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_2 : T_{l3}, a : T_3 \vdash \langle\!\langle a, x_h :: x_2\rangle\!\rangle : T_{2.5}[(s+1)/n']} \qquad \dfrac{}{s : \mathbb{N} \vdash s + 1 : \mathbb{N}}}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_2 : T_{l3}, a : T_3 \vdash \langle\!\langle a, x_h :: x_2\rangle\!\rangle : T_{2.4}}}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_2 : T_{l3}, a : T_3 \vdash \mathsf{store}\langle\!\langle a, x_h :: x_2\rangle\!\rangle : T_{2.3}}$$

D1.3:

$$\dfrac{\dfrac{}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); . \vdash \mathsf{store}() : \mathbb{M}(p_1 + \phi(\lhd\vec{p}, s))\,[p_1 + \phi(\lhd\vec{p}, s)]\,\mathbf{1}} \qquad D1.4}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_2 : T_{l3} \vdash \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{store}\langle\!\langle a, x_h :: x_2\rangle\!\rangle : T_{2.2}}$$

D1.2:

$$\dfrac{\dfrac{}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_1 : T_{l2} \vdash x_1 : T_{l2}} \qquad D1.3}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_1 : T_{l2}, x_2 : T_{l3} \vdash E_1 : T_{2.1}}$$

D1.1:

$$\dfrac{\dfrac{}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x : T_{l1} \vdash x : T_{l1}} \qquad D1.2}{.; s : \mathbb{N}; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x : T_{l1} \vdash \mathsf{let}\langle\!\langle x_1, x_2\rangle\!\rangle = x \text{ in } E_1 : T_{2.1}}$$

D1:

$$\dfrac{\dfrac{}{.; .; .; .; (\!|\Sigma|\!); x_t : T_l \vdash x_t : T_l} \qquad D1.1}{.; .; .; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_t : T_l \vdash E_0 : T_{2.1}}$$

D0:

$$\dfrac{\dfrac{}{.; .; .; .; (\!|\Sigma|\!); . \vdash \uparrow^{K^{cons}}} \qquad D1}{.; .; .; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_t : T_l \vdash \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0 : T_{2.1}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.; .; .; .; (\!|\Sigma|\!); u : T_1 \vdash u : T_1} \qquad D0}{.; .; .; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_t : T_l, u : T_1 \vdash \mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0 : T_2}}{.; .; .; .; (\!|\Sigma|\!); x_h : (\!|\tau|\!), x_t : T_l \vdash \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0 : T_0}$$

7. match:

$$\dfrac{\begin{array}{c}\Sigma; \Gamma \vdash^{q - K_1^{matN}}_{q' + K_2^{matN}} e_1 : \tau' \rightsquigarrow e_{a1} \\ \vec{p} = (p_1, \ldots, p_k) \qquad \Sigma; \Gamma, h : \tau, t : L^{(\lhd\,\vec{p})}\tau \vdash^{q + p_1 - K_1^{matC}}_{q' + K_2^{matC}} e_2 : \tau' \rightsquigarrow e_{a2}\end{array}}{\Sigma; \Gamma, x : L^{p}\tau \vdash^{q}_{q'} \mathsf{match}\ x\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau' \rightsquigarrow \qquad \lambda u.E_0}\ \text{match}$$

where
$E_0 = \mathsf{release} - = u \text{ in } E_{0.1}$
$E_{0.1} = x; a.\,\mathsf{let}\langle\!\langle x_1, x_2\rangle\!\rangle = a \text{ in } E_1$
$E_1 = \mathsf{match}\ x_2\ \mathsf{with}\ |nil \mapsto E_2\ |h :: l_t \mapsto E_3$
$E_2 = \mathsf{bind} - = \uparrow^{K_1^{matN}} \text{ in } E_{2.1}$

$E_{2.1} = \mathsf{bind}\, b = \mathsf{store}()\ \mathsf{in}\ E'_2$
$E'_2 = \mathsf{bind}\, c = (e_{a1}\ b)\ \mathsf{in}\ E'_{2.1}$
$E'_{2.1} = \mathsf{release}\, d = c\ \mathsf{in}\ E'_{2.2}$
$E'_{2.2} = \mathsf{bind} - = \uparrow^{K_2^{matN}}\ \mathsf{in}\ E'_{2.3}$
$E'_{2.3} = \mathsf{release} - = x_1\ \mathsf{in}\ \mathsf{store}\, d$
$E_3 = \mathsf{bind} - = \uparrow^{K_1^{matC}}\ \mathsf{in}\ E_{3.1}$
$E_{3.1} = \mathsf{release} - = x_1\ \mathsf{in}\ E_{3.2}$
$E_{3.2} = \mathsf{bind}\, b = \mathsf{store}()\ \mathsf{in}\ E_{3.3}$
$E_{3.3} = \mathsf{bind}\, t = \mathsf{ret}\langle\!\langle b, l_t\rangle\!\rangle\ \mathsf{in}\ E_{3.4}$
$E_{3.4} = \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ E_{3.5}$
$E_{3.5} = \mathsf{bind}\, f = e_{a2}\ d\ \mathsf{in}\ E_{3.6}$
$E_{3.6} = \mathsf{release}\, g = f\ \mathsf{in}\ E_{3.7}$
$E_{3.7} = \mathsf{bind} - = \uparrow^{K_2^{matC}}\ \mathsf{in}\ \mathsf{store}\, g$

$T_0 = [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\,(\!|\tau'|\!))$
$T_1 = [q]\,\mathbf{1}$
$T_2 = \mathbb{M}\,0\,([q']\,(\!|\tau'|\!))$
$T_{2.0} = \mathbb{M}\,q'\,([q']\,(\!|\tau'|\!))$
$T_{2.1} = \mathbb{M}\,q\,([q']\,(\!|\tau'|\!))$
$T_{2.10} = \mathbb{M}(q - K_1^{matC})\,([q']\,(\!|\tau'|\!))$
$T_{2.11} = \mathbb{M}(q - K_1^{matN})\,([q']\,(\!|\tau'|\!))$
$T_{2.12} = \mathbb{M}(q - K_1^{matN})\,([(q - K^{matN})]\,\mathbf{1})$
$T_{2.13} = ([(q - K_1^{matN})]\,\mathbf{1})$
$T_3 = \mathbb{M}(q - K_1^{matC} + p_1 + \phi(\lhd\vec{p}, i))\,[q']\,(\!|\tau'|\!)$
$T_{3.0} = \mathbb{M}(q - K_1^{matC} + p_1)\,[q']\,(\!|\tau'|\!)$
$T_{3.1} = \mathbb{M}\,0\,[q']\,(\!|\tau'|\!)$
$T_{3.2} = \mathbb{M}(q' + K_2^{matC})\,[q']\,(\!|\tau'|\!)$
$T_{4.0} = \mathbb{M}(\phi(\lhd\vec{p}, i))\,\mathbf{1}$
$T_{4.10} = \mathbb{M}\,0\,T_{4.1}$
$T_{4.1} = \exists s'.([(\phi(\lhd\vec{p}, s'))]\,\mathbf{1} \otimes L^{s'}(\!|\tau|\!))$
$T_{4.11} = ([(\phi(\lhd\vec{p}, i))]\,\mathbf{1} \otimes L^{i}(\!|\tau|\!))$
$T_{4.12} = ([(\phi(\lhd\vec{p}, i))]\,\mathbf{1}$
$T_{4.13} = L^{i}(\!|\tau|\!))$
$T_{4.2} = \mathbb{M}(q - k_1^{matC} + p_1)\,[(q - k_1^{matC} + p_1)]\,\mathbf{1}$
$T_{4.3} = \mathbb{M}\,0\,[(q' + K_2^{matC})]\,(\!|\tau|\!)$
$T_{4.4} = [(q' + K_2^{matC})]\,(\!|\tau|\!)$
$T_b = [\phi(\lhd\vec{p}, s')]\,\mathbf{1}$
$T_c = \exists s'.([\phi(\lhd\vec{p}, s')]\,\mathbf{1} \otimes L^{s'}(\!|\tau|\!))$
$T_d = [q - K_1^{matC} + p_1]\,\mathbf{1}$
$T_f = T_{4.4}$
$T_g = (\!|\tau|\!)$
$T_l = \exists s.([\phi(\vec{p}, s)]\,\mathbf{1} \otimes L^{s}(\!|\tau|\!))$
$T'_l = ([\phi(\vec{p}, s)]\,\mathbf{1} \otimes L^{s}(\!|\tau|\!))$
$T_{l1} = [\phi(\vec{p}, s)]\,\mathbf{1}$
$T_{l2} = L^{s}(\!|\tau|\!)$
$T_{l3} = \exists s'.([\phi(\lhd\vec{p}, s')]\,\mathbf{1} \otimes L^{s'}(\!|\tau|\!))$
$T_{l4} = L^{i}(\!|\tau|\!)$
$T_{ih1} = [q - K_1^{matN}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q' + K_2^{matN}]\,(\!|\tau'|\!))$
$T_{ih1.1} = \mathbb{M}\,0\,([q' + K_2^{matN}]\,(\!|\tau'|\!))$
$T_{ih1.2} = ([q' + K_2^{matN}]\,(\!|\tau'|\!))$
$T_{ih2} = [q + p_1 - K_1^{matC}]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q' + K_2^{matC}]\,(\!|\tau'|\!))$
$T_{ih2.1} = \mathbb{M}\,0\,([q' + K_2^{matC}]\,(\!|\tau'|\!))$

D3.8:

$$\frac{}{.; s, i; s = i + 1; (\!|\Sigma|\!); g : T_g \vdash \mathsf{store}\, g : \mathbb{M}\,q'\,[q']\,(\!|\tau|\!)}$$

D3.7:

$$\frac{\dfrac{}{.; s, i; s = i + 1; (\!|\Sigma|\!); . \vdash \uparrow^{K_2^{matC}} : \mathbb{M}\,K_2^{matC}\,\mathbf{1}} \quad E_{3.8}}{.; s, i; s = i + 1; (\!|\Sigma|\!); g : T_g \vdash E_{3.7} : T_{3.3}}$$

D3.6:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;f:T_f\vdash f:T_f}\qquad E_{3.7}}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;f:T_f\vdash E_{3.6}:T_{3.2}}$$

D3.5:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,h:\llparenthesis\tau\rrparenthesis,t:T_c,d:T_d\vdash e_{a2}\ d:T_{4.3}}\qquad E_{3.6}}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,h:\llparenthesis\tau\rrparenthesis,t:T_c,d:T_d\vdash E_{3.5}:T_{3.2}}$$

D3.4:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;.\vdash \mathsf{store}():T_{4.2}}\qquad E_{3.5}}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,h:\llparenthesis\tau\rrparenthesis,t:T_c\vdash E_{3.4}:T_{3.1}}$$

D3.31:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;l_t:T_{l4},b:T_b\vdash\langle\langle b,l_t\rangle\rangle:T_{4.11}}}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;l_t:T_{l4},b:T_b\vdash\langle\langle b,l_t\rangle\rangle:T_{4.1}}$$

D3.3:

$$\dfrac{\dfrac{D3.31}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;l_t:T_{l4},b:T_b\vdash \mathsf{ret}\langle\langle b,l_t\rangle\rangle:T_{4.10}}\qquad D_{3.4}}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,h:\llparenthesis\tau\rrparenthesis,l_t:T_{l4},b:T_b\vdash E_{3.3}:T_{3.1}}$$

D3.2:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;.\vdash \mathsf{store}():T_{4.0}}\qquad D3.3}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis;h:\llparenthesis\tau\rrparenthesis,l_t:T_{l4}\vdash E_{3.2}:T_3}$$

D3.1:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;x_1:T_{l1}\vdash x_1:T_{l1}}\qquad D3.2}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,x_1:T_{l1},h:\llparenthesis\tau\rrparenthesis,l_t:T_{l4}\vdash E_{3.1}:T_{2.10}}$$

D3:

$$\dfrac{\dfrac{}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;.\vdash\uparrow^{K_1^{matC}}:\mathbb{M}\,K_1^{matC}\,\mathbf{1}}\qquad D3.1}{.;s,i;s=i+1;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,x_1:T_{l1},h:\llparenthesis\tau\rrparenthesis,l_t:T_{l4}\vdash E_3:T_{2.1}}$$

D2.32:

$$\dfrac{\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;x_1:T_{l1}\vdash x_1:T_{l1}}\qquad\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;d:\llparenthesis\tau'\rrparenthesis\vdash \mathsf{store}\,d:T_{2.0}}}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;x_1:T_{l1},d:\llparenthesis\tau'\rrparenthesis\vdash E'_{2.3}:T_{2.0}}$$

D2.31:

$$\dfrac{\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;.\vdash\uparrow^{K_2^{matN}}:\mathbb{M}\,K_2^{matN}\,\mathbf{1}}\qquad D2.32}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;x_1:T_{l1},d:\llparenthesis\tau'\rrparenthesis\vdash E'_{2.2}:T_{3.2}}$$

D2.3:

$$\dfrac{\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;c:T_{ih1.2}\vdash c:T_{ih1.2}}\qquad D2.31}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;x_1:T_{l1},c:T_{ih1.2}\vdash E'_{2.1}:T_2}$$

D2.22:

$$\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;b:T_{2.13}\vdash b:T_{2.13}}$$

D2.21:

$$\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis\vdash e_{a1}:T_{ih1}}$$

D2.2:

$$\dfrac{D2.21\qquad D2.22}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,b:T_{2.13}\vdash e_{a1}\ b:T_{ih1.1}}$$

D2.20:

$$\dfrac{D2.2\qquad D2.3}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,x_1:T_{l1},b:T_{2.13}\vdash E'_2:T_2}$$

D2.1:

$$\dfrac{\dfrac{}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;.\vdash \mathsf{store}():T_{2.12}}\qquad D2.20}{.;s;s=0;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis,x_1:T_{l1}\vdash E_{2.1}:T_{2.11}}$$

D2:

$$\dfrac{\dfrac{}{.; s; s = 0; \llparenthesis\Sigma\rrparenthesis; . \vdash \uparrow^{K_1^{matN}} : \mathbb{M}\, K_1^{matN}\, \mathbf{1}} \qquad D2.1}{.; s; s = 0; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : T_{l1} \vdash E_2 : T_{2.1}}$$

D1.1:

$$\dfrac{\dfrac{}{.; s; .; \llparenthesis\Sigma\rrparenthesis; x_2 : T_{l2} \vdash x_2 : T_{l2}} \qquad D2 \qquad D3}{.; s; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : T_{l1}, x_2 : T_{l2} \vdash E_1 : T_{2.1}}$$

D1:

$$\dfrac{\dfrac{}{.; s; .; \llparenthesis\Sigma\rrparenthesis; a : T_l' \vdash a : T_l'} \qquad D1.1}{.; s; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, a : T_l' \vdash \mathsf{let}\langle\!\langle x_1, x_2\rangle\!\rangle = a \ \mathsf{in}\ E_1 : T_{2.1}}$$

D0:

$$\dfrac{\dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; x : T_l \vdash x : T_l} \qquad D1}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : T_l \vdash E_{0.1} : T_{2.1}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : T_l, u : T_1 \vdash u : T_1} \qquad D0}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : T_l, u : T_1 \vdash E_0 : T_2}}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : T_l \vdash \lambda u.E_0 : T_0}$$

8. Share:

$$\dfrac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash^q_{q'} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = \tau_1 = \tau_2 = \mathbf{1}}{\Sigma; \Gamma, z : \tau \vdash^q_{q'} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-unit}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\mathbf{1},\mathbf{1},\mathbf{1}}\ z \ \mathsf{in}\ \mathsf{let}\langle\!\langle x, y\rangle\!\rangle = a \ \mathsf{in}\ e_a\ u$

$T_0 = [q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, ([q']\, \llparenthesis\tau'\rrparenthesis)$

D1:

$$\dfrac{\dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, u : [q]\, \mathbf{1}, x : \llparenthesis\tau_1\rrparenthesis, y : \llparenthesis\tau_2\rrparenthesis \vdash e_a : T_0} \qquad \dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; u : [q]\, \mathbf{1} \vdash u : [q]\, \mathbf{1}}}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, u : [q]\, \mathbf{1}, x : \llparenthesis\tau_1\rrparenthesis, y : \llparenthesis\tau_2\rrparenthesis \vdash e_a\ u : \mathbb{M}\, 0\, [q']\, \mathbf{1}}$$

D0:

$$\dfrac{\dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; a : (\llparenthesis\tau_1\rrparenthesis \otimes \llparenthesis\tau_2\rrparenthesis) \vdash a : (\llparenthesis\tau_1\rrparenthesis \otimes \llparenthesis\tau_2\rrparenthesis)} \qquad D1}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, u : [q]\, \mathbf{1}, a : (\llparenthesis\tau_1\rrparenthesis \otimes \llparenthesis\tau_2\rrparenthesis) \vdash \mathsf{let}\langle\!\langle x, y\rangle\!\rangle = a \ \mathsf{in}\ e_a\ u : [q] \multimap \mathbb{M}\, 0\, [q]\, \llparenthesis\tau'\rrparenthesis}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{Dc1 \qquad \dfrac{}{.; .; .; \llparenthesis\Sigma\rrparenthesis; z : \llparenthesis\tau\rrparenthesis \vdash z : \llparenthesis\tau\rrparenthesis}}{.; .; .; \llparenthesis\Sigma\rrparenthesis; z : \llparenthesis\tau\rrparenthesis \vdash coerce_{\mathbf{1},\mathbf{1},\mathbf{1}}\ z : \mathbb{M}\, 0\, (\llparenthesis\tau_1\rrparenthesis \otimes \llparenthesis\tau_2\rrparenthesis)} \qquad D0}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, z : \llparenthesis\tau\rrparenthesis, u : [q]\, \mathbf{1} \vdash E_0 : \mathbb{M}\, 0\, [q']\, \llparenthesis\tau'\rrparenthesis}}{.; .; .; \llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, z : \llparenthesis\tau\rrparenthesis \vdash \lambda u.E_0 : T_0}$$

$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} : \llparenthesis\mathbf{1}\rrparenthesis \multimap \mathbb{M}\, 0\, (\llparenthesis\mathbf{1}\rrparenthesis \otimes \llparenthesis\mathbf{1}\rrparenthesis)$
$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} \triangleq \lambda u.\, \mathsf{ret}\langle\!\langle !(), !()\rangle\!\rangle$

$T_{c0} = \llparenthesis\mathbf{1}\rrparenthesis \multimap \mathbb{M}\, 0\, (\llparenthesis\mathbf{1}\rrparenthesis \otimes \llparenthesis\mathbf{1}\rrparenthesis)$
$T_{c1} = \mathbb{M}\, 0\, (\llparenthesis\mathbf{1}\rrparenthesis \otimes \llparenthesis\mathbf{1}\rrparenthesis)$
$T_{c2} = \llparenthesis\mathbf{1}\rrparenthesis \otimes \llparenthesis\mathbf{1}\rrparenthesis$

Dc1:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{.; .; .; .; . \vdash \langle\!\langle !(), !()\rangle\!\rangle : T_{c2}}}{.; .; .; .; u : \llparenthesis\mathbf{1}\rrparenthesis \vdash \langle\!\langle !(), !()\rangle\!\rangle : T_{c2}}}{.; .; .; .; u : \llparenthesis\mathbf{1}\rrparenthesis \vdash \mathsf{ret}\langle\!\langle !(), !()\rangle\!\rangle : T_{c1}}}{.; .; .; .; . \vdash \lambda u.\, \mathsf{ret}\langle\!\langle !(), !()\rangle\!\rangle : T_{c0}}$$

$$\frac{\Sigma;\Gamma, x:\tau_1, y:\tau_2 \vdash^q_{q'} e:\tau' \leadsto e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau=\tau_1=\tau_2=\mathsf{b}}{\Sigma;\Gamma, z:\tau \vdash^q_{q'} e[z/x,z/y]:\tau' \leadsto E_0}\ \text{Share-base}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = \mathit{coerce}_{\mathsf{b},\mathsf{b},\mathsf{b}}\ z\ \mathsf{in}\ \mathsf{let}\langle\!\langle x,y\rangle\!\rangle = a\ \mathsf{in}\ e_a\ u$

$T_0 = [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,[q']\,\llparenthesis\tau'\rrparenthesis$

D1:

$$\frac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, x:\llparenthesis\tau_1\rrparenthesis, y:\llparenthesis\tau_2\rrparenthesis \vdash e_a : T_0} \qquad \dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; u:[q]\,\mathbf{1} \vdash u:[q]\,\mathbf{1}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, x:\llparenthesis\tau_1\rrparenthesis, y:\llparenthesis\tau_2\rrparenthesis \vdash e_a\ u : \mathbb{M}\,0\,[q']\,\mathbf{1}}$$

D0:

$$\frac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis) \vdash a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis)} \qquad D1}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis) \vdash \mathsf{let}\langle\!\langle x,y\rangle\!\rangle = a\ \mathsf{in}\ e_a\ u : [q] \multimap \mathbb{M}\,0\,[q]\,\llparenthesis\tau'\rrparenthesis}$$

Main derivation:

$$\frac{\dfrac{\dfrac{Dc1 \qquad \dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; z:\llparenthesis\tau\rrparenthesis \vdash z:\llparenthesis\tau\rrparenthesis}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; z:\llparenthesis\tau\rrparenthesis \vdash \mathit{coerce}_{\mathsf{b},\mathsf{b},\mathsf{b}}\ z : \mathbb{M}\,0\,(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis)} \qquad D0}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, z:\llparenthesis\tau\rrparenthesis, u:[q]\,\mathbf{1} \vdash E_0 : \mathbb{M}\,0\,[q']\,\llparenthesis\tau'\rrparenthesis}}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, z:\llparenthesis\tau\rrparenthesis \vdash \lambda u.E_0 : T_0}$$

$\mathit{coerce}_{\mathsf{b},\mathsf{b},\mathsf{b}} : \llparenthesis\mathsf{b}\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\mathsf{b}\rrparenthesis\otimes\llparenthesis\mathsf{b}\rrparenthesis)$
$\mathit{coerce}_{\mathsf{b},\mathsf{b},\mathsf{b}} \triangleq \lambda u.\,\mathsf{let}\,!\,u' = u\ \mathsf{in}\ \mathsf{ret}\langle\!\langle !u', !u'\rangle\!\rangle$

$T_{c0} = \llparenthesis\mathsf{b}\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\mathsf{b}\rrparenthesis\otimes\llparenthesis\mathsf{b}\rrparenthesis)$
$T_{c1} = \mathbb{M}\,0\,(\llparenthesis\mathsf{b}\rrparenthesis\otimes\llparenthesis\mathsf{b}\rrparenthesis)$
$T_{c2} = \llparenthesis\mathsf{b}\rrparenthesis\otimes\llparenthesis\mathsf{b}\rrparenthesis$

Dc2:

$$\frac{\dfrac{}{.;.;.;u':\mathsf{b};. \vdash \langle\!\langle !u', !u'\rangle\!\rangle : T_{c2}}}{.;.;.;u':\mathsf{b};. \vdash \mathsf{ret}\langle\!\langle !u', !u'\rangle\!\rangle : T_{c1}}$$

Dc1:

$$\frac{\dfrac{\dfrac{}{.;.;.;.;u:!\mathsf{b} \vdash u:!\mathsf{b}} \qquad Dc2}{.;.;.;.;u:\llparenthesis\mathsf{b}\rrparenthesis \vdash \mathsf{let}\,!\,u' = u\ \mathsf{in}\ \mathsf{ret}\langle\!\langle !u', !u'\rangle\!\rangle : T_{c1}}}{.;.;.;.;. \vdash \lambda u.\,\mathsf{let}\,!\,u' = u\ \mathsf{in}\ \mathsf{ret}\langle\!\langle !u', !u'\rangle\!\rangle : T_{c0}}$$

$$\frac{\begin{array}{c}\Sigma;\Gamma, x:\tau_1, y:\tau_2 \vdash^q_{q'} e:\tau' \leadsto e_a \\ \tau = L^{\vec{p}}\tau'' \qquad \tau_1 = L^{\vec{p_1}}\tau_1'' \qquad \tau_2 = L^{\vec{p_2}}\tau_2'' \qquad \tau'' = \tau_1'' \curlyvee \tau_2'' \qquad \vec{p} = \vec{p_1} + \vec{p_2}\end{array}}{\Sigma;\Gamma, z:\tau \vdash^q_{q'} e[z/x,z/y]:\tau' \leadsto E_0}\ \text{Share-list}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = \mathit{coerce}_{\tau,\tau_1,\tau_2}\ z\ \mathsf{in}\ \mathsf{let}\langle\!\langle x,y\rangle\!\rangle = a\ \mathsf{in}\ e_a\ u$

$T_0 = [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\,\llparenthesis\tau'\rrparenthesis)$

D1:

$$\frac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, x:\llparenthesis\tau_1\rrparenthesis, y:\llparenthesis\tau_2\rrparenthesis \vdash e_a : T_0} \qquad \dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; u:[q]\,\mathbf{1} \vdash u:[q]\,\mathbf{1}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, x:\llparenthesis\tau_1\rrparenthesis, y:\llparenthesis\tau_2\rrparenthesis \vdash e_a\ u : \mathbb{M}\,0\,[q']\,\mathbf{1}}$$

D0:

$$\frac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis) \vdash a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis)} \qquad D1}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, u:[q]\,\mathbf{1}, a:(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis) \vdash \mathsf{let}\langle\!\langle x,y\rangle\!\rangle = a\ \mathsf{in}\ e_a\ u : [q] \multimap \mathbb{M}\,0\,[q]\,\llparenthesis\tau'\rrparenthesis}$$

Main derivation:

$$\frac{\dfrac{.;.;.;\llparenthesis\Sigma\rrparenthesis; z:\llparenthesis\tau\rrparenthesis \vdash \mathit{coerce}_{\tau,\tau_1,\tau_2}\ z : \mathbb{M}\,0\,(\llparenthesis\tau_1\rrparenthesis\otimes\llparenthesis\tau_2\rrparenthesis) \qquad D0}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, z:\llparenthesis\tau\rrparenthesis, u:[q]\,\mathbf{1} \vdash E_0 : \mathbb{M}\,0\,[q']\,\llparenthesis\tau'\rrparenthesis}}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, z:\llparenthesis\tau\rrparenthesis \vdash \lambda u.E_0 : T_0}$$

$coerce_{L^{\vec{p}}\tau, L^{\vec{p_1}}\tau_1, L^{\vec{p_2}}\tau_2} :!((\!|\tau|\!) \multimap \mathbb{M}\, 0\, (\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \multimap (\!|L^{\vec{p}}\tau|\!) \multimap \mathbb{M}\, 0\, (\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!)$
$coerce_{L^{\vec{p}}\tau, L^{\vec{p_1}}\tau_1, L^{\vec{p_2}}\tau_2} \triangleq \mathsf{fix} f.\lambda\_g.\lambda e.\, \mathsf{let}\,!\, g' = g \text{ in } e; x.\, \mathsf{let}\langle\!\langle p, l\rangle\!\rangle = x \text{ in } E_0$

where
$E_0 \triangleq \mathsf{release}\, - = p \text{ in } \; E_1$

$E_1 \triangleq \mathsf{match}\; l \text{ with } |nil \mapsto E_{2.1} \; |h :: t \mapsto E_3$

$E_{2.1} \triangleq \mathsf{bind}\, z_1 = \mathsf{store}() \text{ in } E_{2.2}$

$E_{2.2} \triangleq \mathsf{bind}\, z_2 = \mathsf{store}() \text{ in } E_{2.3}$

$E_{2.3} \triangleq \mathsf{ret}\langle\!\langle\langle\!\langle z_1, nil\rangle\!\rangle, \langle\!\langle z_2, nil\rangle\!\rangle\rangle\!\rangle$

$E_3 \triangleq \mathsf{bind}\, H = g'\; h \text{ in } E_{3.1}$

$E_{3.1} \triangleq \mathsf{bind}\, o_t = () \text{ in } E_{3.2}$

$E_{3.2} \triangleq \mathsf{bind}\, T = f\; g\; \langle\!\langle o_t, t\rangle\!\rangle \text{ in } E_4$

$E_4 \triangleq \mathsf{let}\langle\!\langle H_1, H_2\rangle\!\rangle = H \text{ in } E_5$

$E_5 \triangleq \mathsf{let}\langle\!\langle T_1, T_2\rangle\!\rangle = T \text{ in } E_6$

$E_6 \triangleq T_1; tp_1.\, \mathsf{let}\langle\!\langle p_1', l_1'\rangle\!\rangle = tp_1 \text{ in } E_{7.1}$

$E_{7.1} \triangleq T_2; tp_2.\, \mathsf{let}\langle\!\langle p_2', l_2'\rangle\!\rangle = tp_2 \text{ in } E_{7.2}$

$E_{7.2} \triangleq \mathsf{release}\, - = p_1' \text{ in } \; E_{7.3}$

$E_{7.3} \triangleq \mathsf{release}\, - = p_2' \text{ in } \; E_{7.4}$

$E_{7.4} \triangleq \mathsf{bind}\, o_1 = \mathsf{store}() \text{ in } E_{7.5}$

$E_{7.5} \triangleq \mathsf{bind}\, o_2 = \mathsf{store}() \text{ in } E_8$

$E_8 \triangleq \mathsf{ret}\langle\!\langle\langle\!\langle o_1, H_1 :: T_1\rangle\!\rangle, \langle\!\langle o_2, H_2 :: T_2\rangle\!\rangle\rangle\!\rangle$

$T_0 =!((\!|\tau|\!) \multimap \mathbb{M}\, 0\, ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!))) \multimap (\!|L^{\vec{p}}\tau|\!) \multimap \mathbb{M}\, 0\, ((\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!))$
$T_1 =!((\!|\tau|\!) \multimap \mathbb{M}\, 0\, ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)))$
$T_1' = ((\!|\tau|\!) \multimap \mathbb{M}\, 0\, ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)))$
$T_{1.0} = \exists s.([\phi(\vec{p}, s)]\, \mathbf{1} \otimes L^s(\!|\tau|\!))$
$T_{1.1} = ([\phi(\vec{p}, s)]\, \mathbf{1} \otimes L^s(\!|\tau|\!))$
$T_{1.2} = [\phi(\vec{p}, s)]\, \mathbf{1}$
$T_{1.3} = L^s(\!|\tau|\!)$
$T_2 = (\!|L^{\vec{p}}\tau|\!) \multimap \mathbb{M}\, 0\, ((\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!))$
$T_3 = \mathbb{M}\, 0\, ((\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!))$
$T_{3.1} = \mathbb{M}(\phi(\vec{p}, s))\, ((\!|L^{\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\tau_2|\!))$
$T_{3.11} = \mathbb{M}(\phi(\triangleleft\vec{p}, s-1))\, ([(\phi(\triangleleft\vec{p}, s-1))]\, \mathbf{1})$
$T_{3.12} = [(\phi(\triangleleft\vec{p}, s-1))]\, \mathbf{1}$
$T_4 = \mathbb{M}\, 0\, ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!))$
$T_{4.1} = ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!))$
$T_5 = \mathbb{M}\, 0\, ((\!|L^{\triangleleft\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\triangleleft\vec{p_2}}\tau_2|\!))$
$T_{5.1} = ((\!|L^{\triangleleft\vec{p_1}}\tau_1|\!) \otimes (\!|L^{\triangleleft\vec{p_2}}\tau_2|\!))$
$T_{5.2} = (\!|L^{\triangleleft\vec{p_1}}\tau_1|\!) = \exists s_1'.([\phi(\triangleleft\vec{p_1}, s_1')]\, \mathbf{1} \otimes L^{s_1'}(\!|\tau_1|\!))$
$T_{5.21} = ([\phi(\triangleleft\vec{p_1}, s_1')]\, \mathbf{1} \otimes L^{s_1'}(\!|\tau_1|\!))$
$T_{5.22} = [\phi(\triangleleft\vec{p_1}, s_1')]\, \mathbf{1}$
$T_{5.23} = L^{s_1'}(\!|\tau_1|\!)$
$T_{5.3} = (\!|L^{\triangleleft\vec{p_2}}\tau_2|\!) = \exists s_2'.([\phi(\triangleleft\vec{p_2}, s_2')]\, \mathbf{1} \otimes L^{s_2'}(\!|\tau_2|\!))$
$T_{5.31} = ([\phi(\triangleleft\vec{p_2}, s_2')]\, \mathbf{1} \otimes L^{s_2'}(\!|\tau_2|\!))$

$T_{5.32} = [\phi(\triangleleft\vec{p_2}, s'_2)]\,\mathbf{1}$
$T_{5.33} = L^{s'_2}(\!|\tau_2|\!)$
$P_1 = \vec{p_1}\downarrow_1 + \phi(\triangleleft\vec{p_1}, s'_1)$
$P_2 = \vec{p_2}\downarrow_1 + \phi(\triangleleft\vec{p_2}, s'_2)$
$T_6 = \mathbb{M}\,P_1\,([P_1]\,\mathbf{1})$
$T_{6.1} = [P_1]\,\mathbf{1}$
$T_7 = \mathbb{M}\,P_2\,([P_2]\,\mathbf{1})$
$T_{7.1} = [P_2]\,\mathbf{1}$
$T_{8.0} = \mathbb{M}(\vec{p}\downarrow_1)\,((\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!))$
$T_{8.1} = \mathbb{M}(\vec{p}\downarrow_1 + P_1)\,((\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!))$
$T_{8.2} = \mathbb{M}(\vec{p}\downarrow_1 + P_1 + P_2)\,((\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!))$
$T_{8.3} = \mathbb{M}(\vec{p_2}\downarrow_1 + P_2)\,((\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!))$
$T_{8.4} = \mathbb{M}\,0\,((\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!))$
$T_{8.41} = (\!|L^{\vec{p_1}}\,\tau_1|\!) \otimes (\!|L^{\vec{p_2}}\,\tau_2|\!)$
$T_{8.5} = (\!|L^{\vec{p_1}}\,\tau_1|\!)$
$T_{8.51} = \exists s_1.([\phi(\vec{p_1}, s_1)]\,\mathbf{1} \otimes L[s_1](\!|\tau_1|\!))$
$T_{8.52} = ([\phi(\vec{p_1}, s'_1)]\,\mathbf{1} \otimes L^{s'_1}(\!|\tau_1|\!))$
$T_{8.6} = (\!|L^{\vec{p_2}}\,\tau_2|\!)$
$T_{8.61} = \exists s_2.([\phi(\vec{p_2}, s_2)]\,\mathbf{1} \otimes L[s_2](\!|\tau_2|\!))$
$T_{8.62} = ([\phi(\vec{p_2}, s'_2)]\,\mathbf{1} \otimes L^{s'_2}(\!|\tau_2|\!))$

D1.82:

$$\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_2 : (\!|\tau_2|\!), l'_2 : T_{5.33}, o_2 : T_{7.1} \vdash \langle\!\langle o_2, H_2 :: l'_2 \rangle\!\rangle : T_{8.62}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_2 : (\!|\tau_2|\!), l'_2 : T_{5.33}, o_2 : T_{7.1} \vdash \langle\!\langle o_2, H_2 :: l'_2 \rangle\!\rangle : T_{8.61}}$$

D1.81:

$$\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), l'_1 : T_{5.23}, o_1 : T_{6.1} \vdash \langle\!\langle o_1, H_1 :: l'_1 \rangle\!\rangle : T_{8.52}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), l'_1 : T_{5.23}, o_1 : T_{6.1} \vdash \langle\!\langle o_1, H_1 :: l'_1 \rangle\!\rangle : T_{8.51}}$$

D1.8:

$$\dfrac{\dfrac{\dfrac{D1.81 \qquad D1.182}{\begin{array}{c} .; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33}, o_1 : T_{6.1}, o_2 : T_{7.1} \vdash \\ \langle\!\langle \langle\!\langle o_1, H_1 :: l'_1 \rangle\!\rangle, \langle\!\langle o_2, H_2 :: l'_2 \rangle\!\rangle \rangle\!\rangle : T_{8.41} \end{array}}}{\begin{array}{c} .; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33}, o_1 : T_{6.1}, o_2 : T_{7.1} \vdash \\ \mathsf{ret}\langle\!\langle \langle\!\langle o_1, H_1 :: l'_1 \rangle\!\rangle, \langle\!\langle o_2, H_2 :: l'_2 \rangle\!\rangle \rangle\!\rangle : T_{8.4} \end{array}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33}, o_1 : T_{6.1}, o_2 : T_{7.1} \vdash E_8 : T_{8.4}}$$

D1.75:

$$\dfrac{\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; . \vdash \mathsf{store}() : T_7} \qquad D1.8}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33}, o_1 : T_{6.1} \vdash \mathsf{bind}\, o_2 = \mathsf{store}()\ \mathsf{in}\ E_8 : T_{8.3}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33}, o_1 : T_{6.1} \vdash E_{7.5} : T_{8.3}}$$

D1.74:

$$\dfrac{\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; . \vdash \mathsf{store}() : T_6} \qquad D1.75}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33} \vdash \mathsf{bind}\, o_1 = \mathsf{store}()\ \mathsf{in}\ E_{7.5} : T_{8.2}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, l'_2 : T_{5.33} \vdash E_{7.4} : T_{8.2}}$$

D1.73:

$$\dfrac{\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; p'_2 : T_{5.32} \vdash p'_2 : T_{5.32}} \qquad D1.74}{\begin{array}{c} .; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, p'_2 : T_{5.32}, l'_2 : T_{5.33} \vdash \\ \mathsf{release} - = p'_2\ \mathsf{in}\ E_{7.4} : T_{8.1} \end{array}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), l'_1 : T_{5.23}, p'_2 : T_{5.32}, l'_2 : T_{5.33} \vdash E_{7.3} : T_{8.1}}$$

D1.72:

$$\dfrac{\dfrac{\dfrac{}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; p'_1 : T_{5.22} \vdash p'_1 : T_{5.22}} \qquad D1.73}{\begin{array}{c} .; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), p'_1 :, l'_1 :, p'_2 : T_{5.32}, l'_2 : T_{5.33} \vdash \\ \mathsf{release} - = p'_1\ \mathsf{in}\ E_{7.3} : T_{8.0} \end{array}}}{.; s'_2, s'_1, s; .; g' : T'_1, f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), p'_1 :, l'_1 :, p'_2 : T_{5.32}, l'_2 : T_{5.33} \vdash E_{7.2} : T_{8.0}}$$

D1.711:

$$\dfrac{\dfrac{}{.; s_2', s_1', s; .; g' : T_1', f : T_0; tp_2 : T_{5.31} \vdash tp_2 : T_{5.31}}\; D1.72}{\begin{array}{c} .; s_2', s_1', s; .; g' : T_1', f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), p_1' : T_{5.22}, l_1' : T_{5.23}, tp_2 : T_{5.31} \vdash \\ \mathsf{let}\langle\!\langle p_2', l_2' \rangle\!\rangle = tp_2 \mathsf{~in~} E_{7.2} : T_{8.0} \end{array}}$$

D1.71:

$$\dfrac{\dfrac{\dfrac{}{.; s_1', s; .; g' : T_1', f : T_0; T_2 : T_{5.3} \vdash T_2 : T_{5.3}}\; D1.711}{\begin{array}{c} .; s_1', s; .; g' : T_1', f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T_2 : T_{5.3}, p_1' : T_{5.22}, l_1' : T_{5.23} \vdash \\ T_2; tp_2. \mathsf{let}\langle\!\langle p_2', l_2' \rangle\!\rangle = tp_2 \mathsf{~in~} E_{7.2} : T_{8.0} \end{array}}}{.; s_1', s; .; g' : T_1', f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T_2 : T_{5.3}, p_1' : T_{5.22}, l_1' : T_{5.23} \vdash E_7 : T_{8.0}}$$

D1.61:

$$\dfrac{\dfrac{}{.; s_1', s; .; g' : T_1', f : T_0; tp_1 : T_{5.21} \vdash tp_1 : T_{5.21}}\; D1.71}{.; s_1', s; .; g' : T_1', f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T_2 : T_{5.3}, tp_1 : T_{5.21} \vdash \mathsf{let}\langle\!\langle p_1', l_1' \rangle\!\rangle = tp_1 \mathsf{~in~} E_7 : T_{8.0}}$$

D1.6:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; T_1 : T_{5.2} \vdash T_1 : T_{5.2}}}{\begin{array}{c} .; s; .; g' : T_1', f : T_0; p : T_{1.2}, H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T_1 : T_{5.2}, T_1 : T_{5.3} \vdash \\ T_1; tp_1. \mathsf{let}\langle\!\langle p_1', l_1' \rangle\!\rangle = tp_1 \mathsf{~in~} E_7 : T_{8.0} \end{array}}\; D1.61}{.; s; .; g' : T_1', f : T_0; H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T_1 : T_{5.2}, T_1 : T_{5.3} \vdash E_6 : T_{8.0}}$$

D1.5:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; T : T_{5.1} \vdash T : T_{5.1}}\; D1.6}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T : T_{5.1} \vdash \mathsf{let}\langle\!\langle T_1, T_2 \rangle\!\rangle = T \mathsf{~in~} E_6 : T_{8.0}}}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, H_1 : (\!|\tau_1|\!), H_2 : (\!|\tau_2|\!), T : T_{5.1} \vdash E_5 : T_{8.0}}$$

D1.4:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; H : T_{4.1} \vdash H : T_{4.1}}\; D1.5}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, H : T_{4.1}, T : T_{5.1} \vdash \mathsf{let}\langle\!\langle H_1, H_2 \rangle\!\rangle = H \mathsf{~in~} E_5 : T_{8.0}}}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, H : T_{4.1}, T : T_{5.1} \vdash E_4 : T_{8.0}}$$

D1.3:

$$\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; t : L^{s-1}(\!|\tau|\!), o_t : T_{3.12} \vdash f\ \langle\!\langle o_t, t \rangle\!\rangle : T_5}\; D1.4}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, H : T_{4.1}, t : L^{s-1}(\!|\tau|\!), o_t : T_{3.12} \vdash \mathsf{bind}\, T = f\ \langle\!\langle o_t, t \rangle\!\rangle \mathsf{~in~} E_4 : T_{8.0}}$$

D1.21:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, h : (\!|\tau|\!), t : L^{s-1}(\!|\tau|\!) \vdash \mathsf{store}() : T_{3.11}}\; D1.3}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, h : (\!|\tau|\!), t : L^{s-1}(\!|\tau|\!) \vdash \mathsf{bind}\, o_t = \mathsf{store}() \mathsf{~in~} E_{3.2} : T_{3.1}}}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, h : (\!|\tau|\!), t : L^{s-1}(\!|\tau|\!) \vdash E_{3.1} : T_{3.1}}$$

D1.2:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; h : (\!|\tau|\!) \vdash g'\ h : T_4}\; D1.3}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, h : (\!|\tau|\!), t : L^{s-1}(\!|\tau|\!) \vdash \mathsf{bind}\, H = g'\ h \mathsf{~in~} E_{3.1} : T_{3.1}}}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, h : (\!|\tau|\!), t : L^{s-1}(\!|\tau|\!) \vdash E_3 : T_{3.1}}$$

D1.14:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; z_2 : [0]\,\mathbf{1} \vdash z_2 : [0]\,\mathbf{1}} \qquad \dfrac{}{.; s; .; g' : T_1', f : T_0; z_2 : [0]\,\mathbf{1} \vdash nil : L^0(\!|\tau_2|\!)}}{.; s; .; g' : T_1', f : T_0; z_2 : [0]\,\mathbf{1} \vdash \langle\!\langle z_2, nil \rangle\!\rangle : ([0]\,\mathbf{1} \otimes L^0(\!|\tau_2|\!))}}{.; s; .; g' : T_1', f : T_0; z_2 : [0]\,\mathbf{1} \vdash \langle\!\langle z_2, nil \rangle\!\rangle : \exists s'.([s']\,\mathbf{1} \otimes L^{s'}(\!|\tau_2|\!))}$$

D1.13:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash z_1 : [0]\,\mathbf{1}} \qquad \dfrac{}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash nil : L^0(\!|\tau_1|\!)}}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash \langle\!\langle z_1, nil \rangle\!\rangle : ([0]\,\mathbf{1} \otimes L^0(\!|\tau_1|\!))}}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash \langle\!\langle z_1, nil \rangle\!\rangle : \exists s'.([s']\,\mathbf{1} \otimes L^{s'}(\!|\tau_1|\!))}$$

D1.12:

$$\dfrac{\dfrac{\dfrac{D1.13 \qquad D1.14}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1}, z_2 : [0]\,\mathbf{1} \vdash \langle\!\langle \langle\!\langle z_1, nil \rangle\!\rangle, \langle\!\langle z_2, nil \rangle\!\rangle \rangle\!\rangle : T_{3.2}}}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1}, z_2 : [0]\,\mathbf{1} \vdash \mathsf{ret}\langle\!\langle \langle\!\langle z_1, nil \rangle\!\rangle, \langle\!\langle z_2, nil \rangle\!\rangle \rangle\!\rangle : T_{3.1}}}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1}, z_2 : [0]\,\mathbf{1} \vdash E_{2.3} : T_{3.1}}$$

D1.11:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; . \vdash \mathsf{store}() : \mathbb{M}\,0\,[0]\,\mathbf{1}} \qquad D1.12}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash \mathsf{bind}\; z_2 = \mathsf{store}() \;\mathsf{in}\; E_{2.3} : T_{3.1}}}{.; s; .; g' : T_1', f : T_0; z_1 : [0]\,\mathbf{1} \vdash E_{2.2} : T_{3.1}}$$

D1.10:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; . \vdash \mathsf{store}() : \mathbb{M}\,0\,[0]\,\mathbf{1}} \qquad D1.11}{.; s; .; g' : T_1', f : T_0; . \vdash \mathsf{bind}\; z_1 = \mathsf{store}() \;\mathsf{in}\; E_{2.2} : T_{3.1}}}{.; s; .; g' : T_1', f : T_0; . \vdash E_{2.1} : T_{3.1}}$$

D1:

$$\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; l : T_{1.3} \vdash l : T_{1.3}} \qquad D1.10 \qquad D1.2}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, l : T_{1.3} \vdash \mathsf{match}\; l \;\mathsf{with}\; |nil \mapsto E_2 \;|h :: t \mapsto E_3 : T_{3.1}}$$

D0.3:

$$\dfrac{\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; p : T_{1.2} \vdash p : T_{1.2}} \qquad D1}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, l : T_{1.3} \vdash \mathsf{release} - = p \;\mathsf{in}\; E_1 : T_3}}{.; s; .; g' : T_1', f : T_0; p : T_{1.2}, l : T_{1.3} \vdash E_0 : T_3}$$

D0.2:

$$\dfrac{\dfrac{}{.; s; .; g' : T_1', f : T_0; x : T_{1.1} \vdash x : T_{1.1}} \qquad D0.3}{.; s; .; g' : T_1', f : T_0; x : T_{1.1} \vdash \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_3}$$

D0.1:

$$\dfrac{\dfrac{}{.; .; .; g' : T_1', f : T_0; e : (\!| L^p \tau |\!) \vdash e : (\!| L^p \tau |\!)} \qquad D0.2}{.; .; .; g' : T_1', f : T_0; e : (\!| L^p \tau |\!) \vdash e; x.\, \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_3}$$

D0:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{.; .; .; f : T_0; g : T_1 \vdash g : T_1} \qquad D1.1}{.; .; .; f : T_0; g : T_1, e : (\!| L^p \tau |\!) \vdash \mathsf{let}\,!\, g' = g \;\mathsf{in}\; e; x.\, \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_3}}{.; .; .; f : T_0; g : T_1 \vdash \lambda e.\, \mathsf{let}\,!\, g' = g \;\mathsf{in}\; e; x.\, \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_2}}{.; .; .; f : T_0; . \vdash \lambda\_g.\lambda e.\, \mathsf{let}\,!\, g' = g \;\mathsf{in}\; e; x.\, \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_0}}{.; .; .; .; . \vdash \mathsf{fix} f.\lambda\_g.\lambda e.\, \mathsf{let}\,!\, g' = g \;\mathsf{in}\; e; x.\, \mathsf{let}\langle\!\langle p, l \rangle\!\rangle = x \;\mathsf{in}\; E_0 : T_0}$$

$$\dfrac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash^q_{q'} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = (\tau_a, \tau_b) \qquad \tau_1 = (\tau_a', \tau_b') \qquad \tau_2 = (\tau_a'', \tau_b'')}{\Sigma; \Gamma, z : \tau \vdash^q_{q'} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-pair}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\; a = coerce_{(\tau_a,\tau_b),(\tau_a',\tau_b'),(\tau_a'',\tau_b'')}\; z \;\mathsf{in}\; \mathsf{let}\langle\!\langle x, y \rangle\!\rangle = a \;\mathsf{in}\; e_a\; u$

$T_0 = [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\,(\!| \tau' |\!))$

D1:

$$\dfrac{\dfrac{}{.; .; .; (\!|\Sigma|\!); (\!|\Gamma|\!), u : [q]\,\mathbf{1}, x : (\!|\tau_1|\!), y : (\!|\tau_2|\!) \vdash e_a : T_0} \qquad \dfrac{}{.; .; .; (\!|\Sigma|\!); u : [q]\,\mathbf{1} \vdash u : [q]\,\mathbf{1}}}{.; .; .; (\!|\Sigma|\!); (\!|\Gamma|\!), u : [q]\,\mathbf{1}, x : (\!|\tau_1|\!), y : (\!|\tau_2|\!) \vdash e_a\; u : \mathbb{M}\,0\,[q']\,\mathbf{1}}$$

D0:

$$\dfrac{\dfrac{}{.; .; .; (\!|\Sigma|\!); a : ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \vdash a : ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!))} \qquad D1}{.; .; .; (\!|\Sigma|\!); (\!|\Gamma|\!), u : [q]\,\mathbf{1}, a : ((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) \vdash \mathsf{let}\langle\!\langle x, y \rangle\!\rangle = a \;\mathsf{in}\; e_a\; u : [q] \multimap \mathbb{M}\,0\,[q]\,(\!|\tau'|\!)}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; z : \llparenthesis\tau\rrparenthesis \vdash \mathit{coerce}_{(\tau_a,\tau_b),(\tau'_a,\tau'_b),(\tau''_a,\tau''_b)}\ z : \mathbb{M}\,0\,(\llparenthesis\tau_1\rrparenthesis \otimes \llparenthesis\tau_2\rrparenthesis)} \quad D0}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, z : \llparenthesis\tau\rrparenthesis, u : [q]\,\mathbf{1} \vdash E_0 : \mathbb{M}\,0\,[q']\,\llparenthesis\tau'\rrparenthesis}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, z : \llparenthesis\tau\rrparenthesis \vdash \lambda u.E_0 : T_0}$$

$\mathit{coerce}_{(\tau_a,\tau_b),(\tau'_a,\tau_b),(\tau''_a,\tau''_b)}$:
$!(\llparenthesis\tau_a\rrparenthesis \multimap \mathbb{M}\,0\,\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis) \multimap !(\llparenthesis\tau_b\rrparenthesis \multimap \mathbb{M}\,0\,\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis) \multimap \llparenthesis(\tau_a,\tau_b)\rrparenthesis \multimap \mathbb{M}\,0\,\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis$
$\mathit{coerce}_{(\tau_a,\tau_b),(\tau'_a,\tau'_b),(\tau''_a,\tau''_b)} \triangleq \lambda\_g_1.\lambda\_g_2.\lambda p.\,\mathsf{let}\,!\langle\langle p_1,p_2\rangle\rangle = p \mathsf{\ in\ } E_0$
where
$E_0 \triangleq \mathsf{let}\,!\,g'_1 = g_1 \mathsf{\ in\ } E_1$
$E_1 \triangleq \mathsf{let}\,!\,g'_2 = g_2 \mathsf{\ in\ } E_2$
$E_2 \triangleq \mathsf{bind}\,P'_1 = g'_1 p_1 \mathsf{\ in\ } E_3$
$E_3 \triangleq \mathsf{bind}\,P'_2 = g'_2 p_2 \mathsf{\ in\ } E_4$
$E_4 \triangleq \mathsf{let}\,!\langle\langle p'_{11},p'_{12}\rangle\rangle = P'_1 \mathsf{\ in\ } E_5$
$E_5 \triangleq \mathsf{let}\,!\langle\langle p'_{21},p'_{22}\rangle\rangle = P'_2 \mathsf{\ in\ } E_6$
$E_6 \triangleq \mathsf{ret}\langle\langle p'_{11},p'_{21}\rangle\rangle, \langle\langle p'_{12},p'_{22}\rangle\rangle$

$T_0 = !(\llparenthesis\tau_a\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis)) \multimap !(\llparenthesis\tau_b\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis)) \multimap$
$\llparenthesis(\tau_a,\tau_b)\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis)$
$T_{0.31} = !(\llparenthesis\tau_a\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis))$
$T_{0.32} = (\llparenthesis\tau_a\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis))$
$T_{0.4} = !(\llparenthesis\tau_b\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis)) \multimap \llparenthesis(\tau_a,\tau_b)\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis)$
$T_{0.41} = !(\llparenthesis\tau_b\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis))$
$T_{0.42} = (\llparenthesis\tau_b\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis))$
$T_{0.5} = \llparenthesis(\tau_a,\tau_b)\rrparenthesis \multimap \mathbb{M}\,0\,(\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis)$
$T_{0.51} = \llparenthesis(\tau_a,\tau_b)\rrparenthesis$
$T_{0.6} = \mathbb{M}\,0\,(\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis)$
$T_{0.61} = (\llparenthesis(\tau'_a,\tau'_b)\rrparenthesis \otimes \llparenthesis(\tau''_a,\tau''_b)\rrparenthesis)$
$T_1 = \mathbb{M}\,0\,(\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis)$
$T_{1.1} = (\llparenthesis\tau'_a\rrparenthesis \otimes \llparenthesis\tau''_a\rrparenthesis)$
$T_{1.11} = \llparenthesis\tau'_a\rrparenthesis$
$T_{1.12} = \llparenthesis\tau''_a\rrparenthesis$
$T_2 = \mathbb{M}\,0\,(\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis)$
$T_{2.1} = (\llparenthesis\tau'_b\rrparenthesis \otimes \llparenthesis\tau''_b\rrparenthesis)$
$T_{2.11} = \llparenthesis\tau'_b\rrparenthesis$
$T_{2.12} = \llparenthesis\tau''_b\rrparenthesis$

D6:

$$\dfrac{\dfrac{\dfrac{}{\begin{array}{c}.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p'_{11} : T_{1.11}, p'_{12} : T_{1.12}, p'_{21} : T_{2.11}, p'_{22} : T_{2.12} \vdash \\ \langle\langle p'_{11},p'_{21}\rangle\rangle, \langle\langle p'_{12},p'_{22}\rangle\rangle : T_{0.61}\end{array}}}{\begin{array}{c}.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p'_{11} : T_{1.11}, p'_{12} : T_{1.12}, p'_{21} : T_{2.11}, p'_{22} : T_{2.12} \vdash \\ \mathsf{ret}\langle\langle p'_{11},p'_{21}\rangle\rangle, \langle\langle p'_{12},p'_{22}\rangle\rangle : T_{0.6}\end{array}}}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p'_{11} : T_{1.11}, p'_{12} : T_{1.12}, p'_{21} : T_{2.11}, p'_{22} : T_{2.12} \vdash E_6 : T_{0.6}}$$

D5:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; P'_2 : T_{2.1} \vdash P'_2 : T_{2.1}} \quad D6}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42};, P'_2 : T_{2.1}, p'_{11} : T_{1.11}, p'_{12} : T_{1.12} \vdash \mathsf{let}\,!\langle\langle p'_{21},p'_{22}\rangle\rangle = P'_2 \mathsf{\ in\ } E_6 : T_{0.6}}}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42};, P'_2 : T_{2.1}, p'_{11} : T_{1.11}, p'_{12} : T_{1.12} \vdash E_5 : T_{0.6}}$$

D4:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42};, P'_1 : T_{1.1} \vdash P'_1 : T_{1.1}} \quad D5}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; P'_1 : T_{1.1}, P'_2 : T_{2.1} \vdash \mathsf{let}\,!\langle\langle p'_{11},p'_{12}\rangle\rangle = P'_1 \mathsf{\ in\ } E_5 : T_{0.6}}}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; P'_1 : T_{1.1}, P'_2 : T_{2.1} \vdash E_4 : T_{0.6}}$$

D3:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p_2 : \llparenthesis\tau_2\rrparenthesis \vdash g'_2 p_2 : T_2} \quad D4}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p_2 : \llparenthesis\tau_2\rrparenthesis, P'_1 : T_{1.1} \vdash \mathsf{bind}\,P'_2 = g'_2 p_2 \mathsf{\ in\ } E_4 : T_{0.6}}}{.;.;.;f : T_0; g'_1 : T_{0.32}, g'_2 : T_{0.42}; p_2 : \llparenthesis\tau_2\rrparenthesis, P'_1 : T_{1.1} \vdash E_3 : T_{0.6}}$$

D2:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f:T_0;g'_1:T_{0.32},g'_2:T_{0.42};p_1:(\!|\tau_1|\!)\vdash g'_1p_1:T_1} \qquad D3}{.;.;.;f:T_0;g'_1:T_{0.32},g'_2:T_{0.42};p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash \mathsf{bind}\,P'_1=g'_1p_1 \ \mathsf{in}\ E_3:T_{0.6}}}{.;.;.;f:T_0;g'_1:T_{0.32},g'_2:T_{0.42};p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash E_2:T_{0.6}}$$

D1:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f:T_0;g'_1:T_{0.32};g_2:T_{0.41}\vdash g_2:T_{0.41}} \qquad D2}{.;.;.;f:T_0;g'_1:T_{0.32};g_2:T_{0.41},p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash \mathsf{let}\,!\,g'_2=g_2\ \mathsf{in}\ E_2:T_{0.6}}}{.;.;.;f:T_0;g'_1:T_{0.32};g_2:T_{0.41},p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash E_1:T_{0.6}}$$

D0.1:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;f:T_0;g_1:T_{0.31}\vdash g_1:T_{0.31}} \qquad D1}{.;.;.;f:T_0;g_1:T_{0.31},g_2:T_{0.41},p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash \mathsf{let}\,!\,g'_1=g_1\ \mathsf{in}\ E_1:T_{0.6}}}{.;.;.;f:T_0;g_1:T_{0.31},g_2:T_{0.41},p_1:(\!|\tau_1|\!),p_2:(\!|\tau_2|\!)\vdash E_0:T_{0.6}}$$

D0:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{.;.;.;f:T_0;p:T_{0.51}\vdash p:T_{0.51}} \qquad D0.1}{.;.;.;f:T_0;g_1:T_{0.31},g_2:T_{0.41},p:T_{0.51}\vdash \mathsf{let}\,!\langle\!\langle p_1,p_2\rangle\!\rangle=p\ \mathsf{in}\ E_0:T_{0.6}}}{.;.;.;f:T_0;g_1:T_{0.31},g_2:T_{0.41}\vdash \lambda p.\,\mathsf{let}\,!\langle\!\langle p_1,p_2\rangle\!\rangle=p\ \mathsf{in}\ E_0:T_{0.5}}}{.;.;.;f:T_0;g_1:T_{0.31}\vdash .\lambda\_g_2.\lambda p.\,\mathsf{let}\,!\langle\!\langle p_1,p_2\rangle\!\rangle=p\ \mathsf{in}\ E_0:T_{0.4}}}{.;.;.;f:T_0;.\vdash \lambda\_g_1.\lambda\_g_2.\lambda p.\,\mathsf{let}\,!\langle\!\langle p_1,p_2\rangle\!\rangle=p\ \mathsf{in}\ E_0:T_0}}{.;.;.;.;.\vdash \mathsf{fix}f.\lambda\_g_1.\lambda\_g_2.\lambda p.\,\mathsf{let}\,!\langle\!\langle p_1,p_2\rangle\!\rangle=p\ \mathsf{in}\ E_0:T_0}$$

9. Sub:

$$\dfrac{\Sigma;\Gamma\vdash^q_{q'} e:\tau\rightsquigarrow e_a \qquad \tau<:\tau'}{\Sigma;\Gamma\vdash^q_{q'} e:\tau'\rightsquigarrow e_a}\ \text{Sub}$$

Main derivation:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma|\!)\vdash e_a:[q]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q'](\!|\tau|\!))} \qquad \dfrac{\tau<:\tau'}{.;.;.\vdash(\!|\tau|\!)<:(\!|\tau'|\!)}\ \text{Lemma 32}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma|\!)\vdash e_a:[q]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q'](\!|\tau'|\!))}\ \text{T-sub}$$

10. Super:

$$\dfrac{\Sigma;\Gamma,x:\tau_1\vdash^q_{q'} e:\tau\rightsquigarrow e_a \qquad \tau'_1<:\tau_1}{\Sigma;\Gamma,x:\tau'_1\vdash^q_{q'} e:\tau\rightsquigarrow e_a}\ \text{Super}$$

Main derivation:

$$\dfrac{.;.;.;(\!|\Sigma|\!);(\!|\Gamma|\!),x:(\!|\tau_1|\!)\vdash e_a:[q]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q'](\!|\tau|\!)) \qquad \dfrac{\tau'_1<:\tau_1}{.;.;.\vdash(\!|\tau'_1|\!)<:(\!|\tau_1|\!)}\ \text{Lemma 32}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma|\!),x:(\!|\tau'_1|\!)\vdash e_a:[q]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q'](\!|\tau|\!))}\ \text{T-weaken}$$

11. Relax:

$$\dfrac{\Sigma;\Gamma\vdash^p_{p'} e:\tau\rightsquigarrow e_a \qquad q\geq p \qquad q-p\geq q'-p'}{\Sigma;\Gamma\vdash^q_{q'} e:\tau\rightsquigarrow\lambda o.E_0}\ \text{Relax}$$

where
$E_0=\mathsf{release}\,-=o\ \mathsf{in}\ E_1$
$E_1=\mathsf{bind}\,a=\mathsf{store}()\ \mathsf{in}\ E_2$
$E_2=\mathsf{bind}\,b=e_a\,a\ \mathsf{in}\ E_3$
$E_3=\mathsf{release}\,c=b\ \mathsf{in}\ \mathsf{store}\,c$

D2:

$$\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);b:[p'](\!|\tau|\!)\vdash b:[p'](\!|\tau|\!)} \qquad \dfrac{}{.;.;.;(\!|\Sigma|\!);c:(\!|\tau|\!)\vdash \mathsf{store}\,c:\mathbb{M}(q-p+p')\,([q-p+p'](\!|\tau|\!))}}{.;.;.;(\!|\Sigma|\!);b:[p'](\!|\tau|\!)\vdash E_3:\mathbb{M}(q-p)\,([q-p+p'](\!|\tau|\!))}$$

D1.2:

$$\frac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; a : [p]\,\mathbf{1} \vdash a : [p]\,\mathbf{1}}$$

D1.1:

$$\frac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis \vdash e_a : [p]\,\mathbf{1} \multimap \mathbb{M}\,0\,([p']\llparenthesis\tau\rrparenthesis)}\ \text{IH}$$

D1:

$$\frac{\dfrac{D1.1 \qquad D1.2}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, a : [p]\,\mathbf{1} \vdash e_a\ a : \mathbb{M}\,0\,([p']\llparenthesis\tau\rrparenthesis)} \qquad D2}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, a : [p]\,\mathbf{1} \vdash E_2 : \mathbb{M}(q-p)\,([q-p+p']\llparenthesis\tau\rrparenthesis)}$$

D0:

$$\frac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis;. \vdash \mathsf{store}() : \mathbb{M}\,p\,([p]\,\mathbf{1})} \qquad D1}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis \vdash E_1 : \mathbb{M}(q)\,([q-p+p']\llparenthesis\tau\rrparenthesis)}$$

D0.0:

$$\frac{\dfrac{\dfrac{}{q' \leq q-p+p'}\ \text{Given}}{.;.;. \vdash ([q-p+p']\llparenthesis\tau\rrparenthesis) <: ([q']\llparenthesis\tau\rrparenthesis)}}{.;.;. \vdash \mathbb{M}\,0\,([q-p+p']\llparenthesis\tau\rrparenthesis) <: \mathbb{M}\,0\,([q']\llparenthesis\tau\rrparenthesis)}$$

Main derivation:

$$\frac{\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; o : [q]\,\mathbf{1} \vdash o : [q]\,\mathbf{1}} \qquad D0}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, o : [q]\,\mathbf{1} \vdash E_0 : \mathbb{M}\,0\,([q-p+p']\llparenthesis\tau\rrparenthesis)} \qquad D0.0}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis, o : [q]\,\mathbf{1} \vdash E_0 : \mathbb{M}\,0\,([q']\llparenthesis\tau\rrparenthesis)}\ \text{T-sub}}{.;.;.;\llparenthesis\Sigma\rrparenthesis;\llparenthesis\Gamma\rrparenthesis \vdash \lambda o.E_0 : [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\llparenthesis\tau\rrparenthesis)}$$

12. Let:

$$\frac{\Sigma;\Gamma_1 \vdash_{p}^{q-K_1^{let}} e_1 : \tau_1 \rightsquigarrow e_{a1} \qquad \Sigma;\Gamma_2, x:\tau_1 \vdash_{q'+K_3^{let}}^{p-K_2^{let}} e_2 : \tau_1 \rightsquigarrow e_{a2}}{\Sigma;\Gamma_1,\Gamma_2 \vdash_{q'}^{q} \mathsf{let}\ x = e_1\ in\ e_2 : \tau \rightsquigarrow E_t}\ \text{Let}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u \mathsf{\ in\ } E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K_1^{let}} \mathsf{\ in\ } E_2$
$E_2 = \mathsf{bind}\, a = \mathsf{store}() \mathsf{\ in\ } E_3$
$E_3 = \mathsf{bind}\, b = e_{a1}\ a \mathsf{\ in\ } E_4$
$E_4 = \mathsf{release}\, x = b \mathsf{\ in\ } E_5$
$E_5 = \mathsf{bind} - = \uparrow^{K_2^{let}} \mathsf{\ in\ } E_6$
$E_6 = \mathsf{bind}\, c = \mathsf{store}() \mathsf{\ in\ } E_7$
$E_7 = \mathsf{bind}\, d = e_{a2}\ c \mathsf{\ in\ } E_8$
$E_8 = \mathsf{release}\, f = d \mathsf{\ in\ } E_9$
$E_9 = \mathsf{bind} - = \uparrow^{K_3^{let}} \mathsf{\ in\ } E_{10}$
$E_{10} = \mathsf{bind}\, g = \mathsf{store}\, f \mathsf{\ in\ ret}\, g$

$T_0 = [q]\,\mathbf{1} \multimap \mathbb{M}\,0\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.1} = [q]\,\mathbf{1}$
$T_{0.2} = \mathbb{M}\,0\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.3} = \mathbb{M}\,q\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.4} = \mathbb{M}(q-K_1^{let})\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.5} = \mathbb{M}(q-K_1^{let})\,([q-K_1^{let}]\,\mathbf{1})$
$T_{0.51} = [q-K_1^{let}]\,\mathbf{1}$
$T_{0.6} = \mathbb{M}\,0\,[p]\,\llparenthesis\tau_1\rrparenthesis$
$T_{0.61} = [p]\,\llparenthesis\tau_1\rrparenthesis$
$T_{0.7} = \mathbb{M}\,p\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.8} = \mathbb{M}(p-K_2^{let})\,([q']\llparenthesis\tau\rrparenthesis)$
$T_{0.9} = \mathbb{M}(p-K_2^{let})\,([(p-K_2^{let})]\,\mathbf{1})$
$T_{0.91} = [(p-K_2^{let})]\,\mathbf{1}$
$T_1 = \mathbb{M}\,0\,[(q'+{K_3}^{let})]\,\llparenthesis\tau\rrparenthesis$

$T_{1.1} = [(q' + K_3{}^{let})]\,(\!|\tau|\!)$
$T_{1.2} = \mathbb{M}(q' + K_3{}^{let})\,([q'](\!|\tau|\!))$
$T_{1.3} = \mathbb{M}\,q'\,([q'](\!|\tau|\!))$

D10:

$$\frac{}{.;.;.;(\!|\Sigma|\!);g:[q']\,(\!|\tau|\!) \vdash \mathsf{ret}\,g : \mathbb{M}\,0\,[q']\,(\!|\tau|\!)}$$

D9:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);f:(\!|\tau|\!) \vdash \mathsf{store}\,f : T_{1.3}} \quad D10}{.;.;.;(\!|\Sigma|\!);f:(\!|\tau|\!) \vdash \mathsf{bind}\,g = \mathsf{store}\,f\ \mathsf{in}\ \mathsf{ret}\,g : T_{1.3}}}{.;.;.;(\!|\Sigma|\!);f:(\!|\tau|\!) \vdash E_{10} : T_{1.3}}$$

D8:

$$\frac{\dfrac{\dfrac{}{.;.;.;. \vdash \uparrow^{K_3^{let}} : \mathbb{M}\,K_3^{let}\,\mathbf{1}} \quad D9}{.;.;.;(\!|\Sigma|\!);f:(\!|\tau|\!) \vdash \mathsf{bind} - = \uparrow^{K_3^{let}}\ \mathsf{in}\ E_{10} : T_{1.2}}}{.;.;.;(\!|\Sigma|\!);f:(\!|\tau|\!) \vdash E_9 : T_{1.2}}$$

D7:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);d:T_{1.1} \vdash d : T_{1.1}} \quad D8}{.;.;.;(\!|\Sigma|\!);d:T_{1.1} \vdash \mathsf{release}\,f = d\ \mathsf{in}\ E_9 : T_{0.2}}}{.;.;.;(\!|\Sigma|\!);d:T_{1.1} \vdash E_8 : T_{0.2}}$$

D6:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!),c:T_{0.91} \vdash e_{a2}\ c : T_1} \quad D7}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!),c:T_{0.91} \vdash \mathsf{bind}\,d = e_{a2}\ c\ \mathsf{in}\ E_8 : T_{0.2}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!),c:T_{0.91} \vdash E_7 : T_{0.2}}$$

D5:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);. \vdash \mathsf{store}() : T_{0.9}} \quad D6}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!) \vdash \mathsf{bind}\,c = \mathsf{store}()\ \mathsf{in}\ E_7 : T_{0.8}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!) \vdash E_6 : T_{0.8}}$$

D4:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);. \vdash \uparrow^{K_2^{let}} : \mathbb{M}\,K_2^{let}\,\mathbf{1}} \quad D5}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!) \vdash \mathsf{bind} - = \uparrow^{K_2^{let}}\ \mathsf{in}\ E_6 : T_{0.7}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!) \vdash E_5 : T_{0.7}}$$

D3:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);b:T_{0.61} \vdash b : T_{0.61}} \quad D4}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!),b:T_{0.61} \vdash \mathsf{release}\,x = b\ \mathsf{in}\ E_5 : T_{0.2}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_2|\!),b:T_{0.61} \vdash E_4 : T_{0.2}}$$

D2:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),a:T_{0.51} \vdash e_{a1}\ a : T_{0.6}} \quad D3}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!),a:T_{0.51} \vdash \mathsf{bind}\,b = e_{a1}\ a\ \mathsf{in}\ E_4 : T_{0.2}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!),a:T_{0.51} \vdash E_3 : T_{0.2}}$$

D1:

$$\frac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!) \vdash \mathsf{store}() : T_{0.5}} \quad D2}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!) \vdash \mathsf{bind}\,a = \mathsf{store}()\ \mathsf{in}\ E_3 : T_{0.4}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!) \vdash E_2 : T_{0.4}}$$

D0:

$$\frac{\dfrac{\dfrac{}{.;.;.;. \vdash \uparrow^{K_1^{let}} : \mathbb{M}\,K_1^{let}\,\mathbf{1}} \quad D1}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!) \vdash \mathsf{bind} - = \uparrow^{K_1^{let}}\ \mathsf{in}\ E_2 : T_{0.3}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!) \vdash E_1 : T_{0.3}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!),u:T_{0.1}\vdash u:T_{0.1}}\quad D0}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!),u:T_{0.1}\vdash \mathsf{release}-=u\ \mathsf{in}\ E_1:T_{0.2}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!),u:T_{0.1}\vdash E_0:T_{0.2}}}{.;.;.;(\!|\Sigma|\!);(\!|\Gamma_1|\!),(\!|\Gamma_2|\!)\vdash \lambda u.E_0:T_0}$$

13. Pair:

$$\dfrac{}{\Sigma;x_1:\tau_1,x_2:\tau_2\vdash^{q+K^{pair}}_{q}(x_1,x_2):(\tau_1,\tau_2)\rightsquigarrow E_t}\ \text{pair}$$

where
$E_t=\lambda u.E_0$
$E_0=\mathsf{release}-=u\ \mathsf{in}\ E_1$
$E_1=\mathsf{bind}-=\uparrow^{K^{pair}}\ \mathsf{in}\ E_2$
$E_2=\mathsf{bind}\,a=\mathsf{store}(x_1,x_2)\ \mathsf{in}\ \mathsf{ret}\,a$

$T_0=[(q+K^{pair})]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!))$
$T_{0.1}=[(q+K^{pair})]\,\mathbf{1}$
$T_{0.2}=\mathbb{M}\,0\,([q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!))$
$T_{0.3}=\mathbb{M}\,(q+K^{pair})\,([q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!))$
$T_{0.4}=\mathbb{M}\,q\,([q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!))$
D2:

$$\dfrac{}{.;.;.;(\!|\Sigma|\!);a:[q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!)\vdash \mathsf{ret}\,a:\mathbb{M}\,0\,[q]\,(\!|\tau_1|\!)\otimes(\!|\tau_2|\!)}$$

D1:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash\mathsf{store}(x_1,x_2):T_{0.4}}\quad D2}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash\mathsf{bind}\,a=\mathsf{store}(x_1,x_2)\ \mathsf{in}\ \mathsf{ret}\,a:T_{0.4}}}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash E_2:T_{0.4}}$$

D0:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);.\vdash\uparrow^{K^{pair}}:\mathbb{M}\,K^{pair}\,\mathbf{1}}\quad D1}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash\mathsf{bind}-=\uparrow^{K^{pair}}\ \mathsf{in}\ E_2:T_{0.3}}}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash E_1:T_{0.3}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!),u:T_{0.1}\vdash u:T_{0.1}}\quad D0}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!),u:T_{0.1}\vdash\mathsf{release}-=u\ \mathsf{in}\ E_1:T_{0.2}}}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!),u:T_{0.1}\vdash E_0:T_{0.2}}}{.;.;.;(\!|\Sigma|\!);x_1:(\!|\tau_1|\!),x_2:(\!|\tau_2|\!)\vdash\lambda u.E_0:T_0}$$

14. MatP:

$$\dfrac{\tau=(\tau_1,\tau_2)\qquad \Sigma,\Gamma,x_1:\tau_1,x_2:\tau_2\vdash^{q-K_1^{matP}}_{q'+K_2^{matP}}e:\tau'\rightsquigarrow e_t}{\Sigma;\Gamma,x:\tau\vdash^{q}_{q'}\ match\ x\ with\ (x_1,x_2)\to e:\tau'\rightsquigarrow E_t}\ \text{matP}$$

where
$E_t=\lambda u.E_0$
$E_0=\mathsf{release}-=u\ \mathsf{in}\ E_1$
$E_1=\mathsf{bind}-=\uparrow^{K_1^{matP}}\ \mathsf{in}\ E_2$
$E_2=\mathsf{let}\langle\!\langle x_1,x_2\rangle\!\rangle=x\ \mathsf{in}\ E_3$
$E_3=\mathsf{bind}\,a=\mathsf{store}()\ \mathsf{in}\ E_4$
$E_4=\mathsf{bind}\,b=e_t\ a\ \mathsf{in}\ E_5$
$E_5=\mathsf{release}\,c=b\ \mathsf{in}\ E_6$
$E_6=\mathsf{bind}-=\uparrow^{K_2^{matP}}\ \mathsf{in}\ E_7$
$E_7=\mathsf{bind}\,d=\mathsf{store}\,c\ \mathsf{in}\ \mathsf{ret}\,d$

$T_0=[q]\,\mathbf{1}\multimap\mathbb{M}\,0\,([q']\,(\!|\tau'|\!))$
$T_{0.1}=[q]\,\mathbf{1}$

$T_{0.2} = \mathbb{M}\,0\,([q']\,\llparenthesis\tau'\rrparenthesis)$
$T_{0.3} = \mathbb{M}\,q\,([q']\,\llparenthesis\tau'\rrparenthesis)$
$T_{0.4} = \mathbb{M}(q - K_1^{matP})\,([q']\,\llparenthesis\tau'\rrparenthesis)$
$T_{0.5} = \mathbb{M}(q - K_1^{matP})\,([(q - K_1^{matP})]\,\mathbf{1})$
$T_{0.51} = [(q - K_1^{matP})]\,\mathbf{1}$
$T_{0.6} = \mathbb{M}\,0\,([q' + k_2^{matP}]\,\llparenthesis\tau'\rrparenthesis)$
$T_{0.61} = [(q' + k_2^{matP})]\,\llparenthesis\tau'\rrparenthesis$
$T_{0.7} = \mathbb{M}(q' + K_2^{matP})\,[q']\,\llparenthesis\tau'\rrparenthesis$
$T_{0.71} = [q']\,\llparenthesis\tau'\rrparenthesis$
$T_{0.8} = \mathbb{M}\,q'\,([q']\,\llparenthesis\tau'\rrparenthesis)$

D7:

$$\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; d : [q']\,\llparenthesis\tau'\rrparenthesis \vdash \mathsf{ret}\,d : \mathbb{M}\,0\,[q']\,\llparenthesis\tau'\rrparenthesis}$$

D6:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash \mathsf{store}\,c : \mathbb{M}\,q'\,[q']\,\llparenthesis\tau'\rrparenthesis} \quad D7}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash \mathsf{bind}\,d = \mathsf{store}\,c\ \mathsf{in}\ \mathsf{ret}\,d : T_{0.8}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash E_7 : T_{0.8}}$$

D5:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash \uparrow^{K_2^{matP}} : \mathbb{M}\,K_2^{matP}\,\mathbf{1}} \quad D6}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash \mathsf{bind}\,- = \uparrow^{K_2^{matP}}\ \mathsf{in}\ E_7 : T_{0.7}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; c : \llparenthesis\tau'\rrparenthesis \vdash E_6 : T_{0.7}}$$

D4:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; b : T_{0.61} \vdash b : T_{0.61}} \quad D5}{.;.;.;\llparenthesis\Sigma\rrparenthesis; b : T_{0.61} \vdash \mathsf{release}\,c = b\ \mathsf{in}\ E_6 : T_{0.2}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; b : T_{0.61} \vdash E_5 : T_{0.2}}$$

D3:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : \llparenthesis\tau_1\rrparenthesis, x_2 : \llparenthesis\tau_2\rrparenthesis, a : T_{0.51} \vdash e_t\ a : T_{0.6}} \quad D4}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : \llparenthesis\tau_1\rrparenthesis, x_2 : \llparenthesis\tau_2\rrparenthesis, a : T_{0.51} \vdash \mathsf{bind}\,b = e_t\ a\ \mathsf{in}\ E_5 : T_{0.2}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : \llparenthesis\tau_1\rrparenthesis, x_2 : \llparenthesis\tau_2\rrparenthesis, a : T_{0.51} \vdash E_4 : T_{0.2}}$$

D2:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; . \vdash \mathsf{store}() : T_{0.5}} \quad D3}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : \llparenthesis\tau_1\rrparenthesis, x_2 : \llparenthesis\tau_2\rrparenthesis \vdash \mathsf{bind}\,a = \mathsf{store}()\ \mathsf{in}\ E_4 : T_{0.4}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x_1 : \llparenthesis\tau_1\rrparenthesis, x_2 : \llparenthesis\tau_2\rrparenthesis \vdash E_3 : T_{0.4}}$$

D1:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; x : \llparenthesis\tau\rrparenthesis \vdash x : \llparenthesis\tau\rrparenthesis} \quad D2}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis \vdash \mathsf{let}\langle\langle x_1, x_2\rangle\rangle = x\ \mathsf{in}\ E_3 : T_{0.4}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis \vdash E_2 : T_{0.4}}$$

D0:

$$\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; . \vdash \uparrow^{K_1^{matP}} : \mathbb{M}\,K_1^{matP}\,\mathbf{1}} \quad D1}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis \vdash \mathsf{bind}\,- = \uparrow^{K_1^{matP}}\ \mathsf{in}\ E_2 : T_{0.3}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis \vdash E_1 : T_{0.3}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis, u : T_{0.1} \vdash u : T_{0.1}} \quad D0}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis, u : T_{0.1} \vdash \mathsf{release}\,- = u\ \mathsf{in}\ E_1 : T_{0.2}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis, u : T_{0.1} \vdash E_0 : T_{0.2}}}{.;.;.;\llparenthesis\Sigma\rrparenthesis; \llparenthesis\Gamma\rrparenthesis, x : \llparenthesis\tau\rrparenthesis \vdash \lambda u.E_0 : T_0}$$

15. Augment:

$$\dfrac{\Sigma; \Gamma \vdash^q_{q'} e : \tau \rightsquigarrow e_a}{\Sigma; \Gamma, x : \tau' \vdash^q_{q'} e : \tau \rightsquigarrow e_a}\ \text{Augment}$$

Main derivation:

$$\frac{.; .; .; (\!|\Sigma|\!); (\!|\Gamma|\!) \vdash e_a : [q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, ([q'] (\!|\tau|\!))}{.; .; .; (\!|\Sigma|\!); (\!|\Gamma|\!), x : (\!|\tau'|\!) \vdash e_a : [q]\, \mathbf{1} \multimap \mathbb{M}\, 0\, ([q'] (\!|\tau|\!))} \text{ T-weaken}$$

□

Lemma 32 (Subtyping preservation)**.** $\forall \tau, \tau'$.
$\tau <: \tau' \implies (\!|\tau|\!) <: (\!|\tau'|\!)$

Proof. Proof by induction on the $\tau <: \tau'$ relation

1. Base:

$$\frac{}{\mathsf{b} <: \mathsf{b}}$$

Main derivation:

$$\frac{}{.; . \vdash !\mathsf{b} <: !\mathsf{b}}$$

2. Pair:

$$\frac{\tau_1 <: \tau_1' \qquad \tau_2 <: \tau_2'}{(\tau_1, \tau_2) <: (\tau_1', \tau_2')}$$

Main derivation:

$$\frac{\dfrac{}{(\!|\tau_1|\!) <: (\!|\tau_1'|\!)} \text{IH1} \qquad \dfrac{}{(\!|\tau_2|\!) <: (\!|\tau_2'|\!)} \text{IH2}}{((\!|\tau_1|\!) \otimes (\!|\tau_2|\!)) <: ((\!|\tau_1'|\!) \otimes (\!|\tau_2'|\!))}$$

3. List:

$$\frac{\tau_1 <: \tau_2 \qquad \vec{p} \geq \vec{q}}{L^{\vec{p}}\tau_1 <: L^{\vec{q}}\tau_2}$$

Main derivation:

$$\frac{\dfrac{\dfrac{\dfrac{}{\vec{q} \leq \vec{p}} \text{Given}}{\phi(\vec{q}, s) \leq \phi(\vec{p}, s)}}{.; s \vdash [\phi(\vec{p}, s)]\, \mathbf{1} <: [\phi(\vec{q}, s)]\, \mathbf{1}} \qquad \dfrac{\dfrac{}{.; s \vdash (\!|\tau_1|\!) <: (\!|\tau_2|\!)} \text{IH}}{.; s \vdash L^s (\!|\tau_1|\!) <: L^s (\!|\tau_2|\!)}}{\dfrac{.; s \vdash ([\phi(\vec{p}, s)]\, \mathbf{1} \otimes L^s (\!|\tau_1|\!)) <: ([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s (\!|\tau_2|\!))}{.; . \vdash \exists s.([\phi(\vec{p}, s)]\, \mathbf{1} \otimes L^s (\!|\tau_1|\!)) <: \exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s (\!|\tau_2|\!))}}$$

□

A.5.2 Cross-language model: RAMLU to λ-amor

Definition 33 (Logical relation for RAMLU to λ-amor)**.**

$$\begin{array}{lll}
\lfloor unit \rfloor_{\mathcal{V}}^{H} & \triangleq & \{(T, {}^s v, {}^t v) \mid {}^s v \in [\![unit]\!] \wedge {}^t v \in [\![\mathbf{1}]\!] \wedge {}^s v = {}^t v\} \\
\lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H} & \triangleq & \{(T, {}^s v, !{}^t v) \mid {}^s v \in [\![\mathsf{b}]\!] \wedge {}^t v \in [\![\mathsf{b}]\!] \wedge {}^s v = {}^t v\} \\
\lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H} & \triangleq & \{(T, \ell, \langle\!\langle {}^t v_1, {}^t v_2 \rangle\!\rangle) \mid H(\ell) = ({}^s v_1, {}^s v_2) \wedge (T, {}^s v_1, {}^t v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, {}^s v_2, {}^t v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}\} \\
\lfloor L^{\vec{q}} \tau \rfloor_{\mathcal{V}}^{H} & \triangleq & \{(T, \ell_s, \langle\!\langle (), l_t \rangle\!\rangle) \mid (T, \ell_s, l_t) \in \lfloor L\ \tau \rfloor_{\mathcal{V}}^{H}\} \\
where & & \\
\lfloor L\ \tau \rfloor_{\mathcal{V}}^{H} & \triangleq & \{(T, NULL, nil)\}\} \cup \\
& & \{(T, \ell, {}^t v :: l_t) \mid H(\ell) = ({}^s v, \ell_s) \wedge (T, {}^s v, {}^t v) \in \lfloor \tau \rfloor_{\mathcal{V}} \wedge (T, \ell_s, l_t) \in \lfloor L\ \tau \rfloor_{\mathcal{V}}\} \\
\lfloor \tau_1 \overset{q/q'}{\to} \tau_2 \rfloor^{H} & \triangleq & \{(T, f(x) = e_s, \mathsf{fix} f.\lambda u.\lambda x.e_t) \mid \forall {}^s v', {}^t v', T' {<} T\ . \\
& & (T', {}^s v', {}^t v') \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \implies (T', e_s, e_t[()/u][{}^t v'/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor \tau_2 \rfloor_{\mathcal{E}}^{\{x \mapsto {}^s v'\}, H}\} \\
& & \\
\lfloor \tau \rfloor_{\mathcal{E}}^{V,H} & \triangleq & \{(T, e_s, e_t) \mid \forall H', {}^s v, p, p', t {<} T\ .\ V, H \vdash^{p}_{p'} e_s \Downarrow_t {}^s v, H' \implies \\
& & \exists {}^t v_t, {}^t v_f, J. e_t \Downarrow_{-} {}^t v_t \Downarrow^{J}_{-} {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J\}
\end{array}$$

Definition 34 (Interpretation of typing context)**.**

$$\lfloor \Gamma \rfloor_{\mathcal{V}}^{H} \quad = \quad \{(T, V, \delta_t) \mid \forall x : \tau \in dom(\Gamma).(T, V(x), \delta_t(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}\}$$

Definition 35 (Interpretation of function context).

$$\lfloor \Sigma \rfloor_{\mathcal{V}}^{H} \quad = \quad \{(T, \delta_{sf}, \delta_{tf}) \mid (\forall f : (\tau_1 \overset{q/q'}{\to} \tau_2) \in dom(\Sigma).(T, \delta_{sf}(f)\ \delta_{sf}, \delta_{tf}(f)\ \delta_{tf}) \in \lfloor (\tau_1 \overset{q/q'}{\to} \tau_2) \rfloor^{H})\}$$

Lemma 36 (Monotonicity for values). $\forall^{s} v, {}^{t}v, T, \tau, H.$
$(T, {}^{s}v, {}^{t}v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H} \implies \forall\, T' \leq T\ .(T', {}^{s}v, {}^{t}v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$

Proof. Given: $(T, {}^{s}v, {}^{t}v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$
To prove: $\forall\, T' \leq T\ .(T', {}^{s}v, {}^{t}v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$

This means given some $T' \leq T$ it suffices to prove that
$(T', {}^{s}v, {}^{t}v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$
By induction on τ

1. $\tau = unit$:
   In this case we are given that $(T, {}^{s}v, {}^{t}v) \in \lfloor unit \rfloor_{\mathcal{V}}^{H}$
   and we need to prove $(T', {}^{s}v, {}^{t}v) \in \lfloor unit \rfloor_{\mathcal{V}}^{H}$
   We get the desired trivially from Definition 33
2. $\tau = \mathsf{b}$:
   In this case we are given that $(T, {}^{s}v, !^{t}v') \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
   and we need to prove $(T', {}^{s}v, !^{t}v') \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
   We get the desired trivially from Definition 33
3. $\tau = L^{\vec{p}}\tau'$:
   In this case we are given that $(T, {}^{s}v, {}^{t}v) \in \lfloor L^{\vec{p}}\tau' \rfloor_{\mathcal{V}}^{H}$

   Here let ${}^{s}v = \ell_s$ and ${}^{t}v = \langle\!\langle (), {}^{t}v_h :: l_t \rangle\!\rangle$
   and we have $(T, \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$ (MV-L1)

   And we need to prove $(T', \ell_s, {}^{t}v_h :: l_t) \in \lfloor L^{\vec{p}}\tau' \rfloor_{\mathcal{V}}^{H}$

   Therefore it suffices to prove that $(T', \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$

   We induct on $(T, \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$
   - $(T, NULL, nil) \in \lfloor L^{\vec{p}}\tau' \rfloor_{\mathcal{V}}^{H}$:
     In this case we need to prove that $(T', NULL, nil) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$
     We get this directly from Definition 33
   - $(T, \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$:
     Since from (MV-L1) we are given that $(T, \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$
     therefore from Definition 33 we have
     $H(\ell_s) = ({}^{s}v_h, \ell_{st}) \wedge (T, {}^{s}v_h, {}^{t}v_h) \in \lfloor \tau' \rfloor_{\mathcal{V}} \wedge (T, \ell_{st}, l_t) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$ (MV-L2)

     In this case we need to prove that $(T', \ell_s, {}^{t}v_h :: l_t) \in \lfloor L\tau' \rfloor_{\mathcal{V}}^{H}$
     From Definition 33 it further it suffices to prove that
     – $H(\ell_s) = ({}^{s}v_h, \ell_{st})$:
       Directly from (MV-L2)
     – $(T', {}^{s}v_h, {}^{t}v_h) \in \lfloor \tau' \rfloor_{\mathcal{V}}$:
       From (MV-L2) and outer induction
     – $(T', \ell_{st}, l_t) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$:
       From (MV-L2) and inner induction
4. $\tau = (\tau_1, \tau_2)$:
   In this case we are given that $(T, \ell, ({}^{t}v_1, {}^{t}v_2)) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$

   This means from Definition 33 we have
   $H(\ell) = ({}^{s}v_1, {}^{s}v_2) \wedge (T, {}^{s}v_1, {}^{t}v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, {}^{s}v_2, {}^{t}v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$ (MV-P0)

   and we need to prove $(T', \ell, ({}^{t}v_1, {}^{t}v_2)) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$
   Similarly from Definition 33 it suffices to prove that
   $H(\ell) = ({}^{s}v_1, {}^{s}v_2) \wedge (T', {}^{s}v_1, {}^{t}v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T', {}^{s}v_2, {}^{t}v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$
   We get this directly from (MV-P0), IH1 and IH2

□

Lemma 37 (Monotonicity for functions). $\forall^{s} v, {}^{t}v, T, \tau, H.$
$(T, f(x) = e_s, \mathsf{fix} f.\lambda u.\lambda x.e_t) \in \lfloor \tau_1 \overset{q/q'}{\to} \tau_2 \rfloor^{H} \implies \forall\, T' \leq T\ .(T', f(x) = e_s, \mathsf{fix} f.\lambda u.\lambda x.e_t) \in \lfloor \tau_1 \overset{q/q'}{\to} \tau_2 \rfloor^{H}$

Proof. We need to prove that $(T', f(x) = e_s, \mathsf{fix} f.\lambda u.\lambda x.e_t) \in \lfloor \tau_1 \overset{q/q'}{\to} \tau_2 \rfloor^{H}$

This means from Definition 33 it suffices to prove that

$\forall^s v', {}^t v', T''<T' .(T'', {}^s v', {}^t v') \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \implies (T'', e_s, e_t[()/u][{}^t v'/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor \tau_2 \rfloor_{\mathcal{E}}^{\{x \mapsto {}^s v'\},H}$

This means given some ${}^s v', {}^t v', T''<T'$ s.t $(T'', {}^s v', {}^t v') \in \lfloor \tau_1 \rfloor_{\mathcal{V}}$ it suffices to prove that
$(T'', e_s, e_t[()/u][{}^t v'/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor \tau_2 \rfloor_{\mathcal{E}}^{\{x \mapsto {}^s v'\},H}$ (MF0)

Since we are given that $(T, f(x) = e_s, \mathsf{fix} f.\lambda u.\lambda x.e_t) \in \lfloor \tau_1 \xrightarrow{q/q'} \tau_2 \rfloor^H$ therefore from Definition 33 we have
$\forall^s v_1', {}^t v_1', T_1'<T .(T_1', {}^s v_1', {}^t v_1') \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \implies (T_1', e_s, e_t[()/u][{}^t v_1'/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor \tau_2 \rfloor_{\mathcal{E}}^{\{x \mapsto {}^s v_1'\},H}$

Instantiating with the given ${}^s v', {}^t v', T''$ we get the desired

□

Lemma 38 (Monotonicity for expressions)**.** $\forall e_s, e_t, T, \tau, H.$
$(T, e_s, e_t) \in \lfloor \tau \rfloor_{\mathcal{E}}^H \implies \forall\, T' \leq T .(T', e_s, e_t) \in \lfloor \tau \rfloor_{\mathcal{E}}^H$

Proof. To prove: $(T', e_s, e_t) \in \lfloor \tau \rfloor_{\mathcal{E}}^H$
This means from Definition 33 it suffices to prove that
$\forall H', {}^s v, p, p', t < T' .\ V, H \vdash_{p'}^{p} e_s \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.e_t \Downarrow_- {}^t v_t \Downarrow_-^J {}^t v_f \wedge (T' - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

This means given some $H', {}^s v, p, p', t < T'$ s.t $V, H \vdash_{p'}^{p} e_s \Downarrow_t {}^s v, H'$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J.e_t \Downarrow_- {}^t v_t \Downarrow_-^J {}^t v_f \wedge (T' - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$ (ME0)

Since we are given that $(T, e_s, e_t) \in \lfloor \tau \rfloor_{\mathcal{E}}^H$ therefore again from Definition 33 we know that
$\forall H', {}^s v, p, p', t < T .\ V, H \vdash_{p'}^{p} e_s \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.e_t \Downarrow_- {}^t v_t \Downarrow_-^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

Instantiating with the given $H', {}^s v, p, p', t$ and using Lemma 36 we get the desired

□

Lemma 39 (Monotonicity for Γ)**.** $\forall^s v, {}^t v, T, \tau, H.$
$(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^H \implies \forall\, T' \leq T .(T', V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^H$

Proof. To prove: $(T', V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^H$
From Definition 34 it suffices to prove that
$\forall x : \tau \in dom(\Gamma).(T', V(x), \delta_t(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^H$

This means given some $x : \tau \in dom(\Gamma)$ it suffices to prove that
$(T', V(x), \delta_t(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^H$

Since we are given that $(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^H$
therefore from Definition 34 we have
$\forall x : \tau \in dom(\Gamma).(T, V(x), \delta_t(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^H$
Instantiating it with the given x and using Lemma 36 we get the desired □

Lemma 40 (Monotonicity for Σ)**.** $\forall^s v, {}^t v, T, \tau, H.$
$(T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor_{\mathcal{V}}^H \implies \forall\, T' \leq T .(T', \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor_{\mathcal{V}}^H$

Proof. To prove: $(T', \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor_{\mathcal{V}}^H$
From Definition 35 it suffices to prove that
$(\forall f : (\tau_1 \xrightarrow{q/q'} \tau_2) \in dom(\Sigma).(T', \delta_{sf}(f)\ \delta_{sf}, \delta_{tf}(f)\ \delta_{tf}) \in \lfloor (\tau_1 \xrightarrow{q/q'} \tau_2) \rfloor^H)$

This means given some $f : (\tau_1 \xrightarrow{q/q'} \tau_2) \in dom(\Sigma)$ it suffices to prove that
$(T', \delta_{sf}(f)\ \delta_{sf}, \delta_{tf}(f)\ \delta_{tf}) \in \lfloor (\tau_1 \xrightarrow{q/q'} \tau_2) \rfloor^H$

Since we are given that $(T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor_{\mathcal{V}}^H$
therefore from Definition 34 we have
$(\forall f : (\tau_1 \xrightarrow{q/q'} \tau_2) \in dom(\Sigma).(T, \delta_{sf}(f)\ \delta_{sf}, \delta_{tf}(f)\ \delta_{tf}) \in \lfloor (\tau_1 \xrightarrow{q/q'} \tau_2) \rfloor^H)$
Instantiating it with the given f and using Lemma 37 we get the desired □

Theorem 41 (Fundamental theorem)**.** $\forall \Sigma, \Gamma, q, q', \tau, e_s, e_t, I, V, H, \delta_t, \delta_{sf}, \delta_{tf}, T.$
$\Sigma; \Gamma \vdash_{q'}^{q} e_s : \tau \rightsquigarrow e_t \wedge$
$(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^H \wedge (T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor_{\mathcal{V}}^H$
$\implies$
$(T, e_s \delta_{sf}, e_t\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

Proof. Proof by induction on $\Sigma; \Gamma \vdash_{q'}^{q} e_s : \tau \rightsquigarrow e_t$

1. unit:

$$\frac{}{\Sigma; . \vdash_q^{q+K^{unit}} () : unit \rightsquigarrow E_t} \text{unit}$$

where
$E_t = \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{unit}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{ret}(a)$
$E_t' = \mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{unit}} \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{ret}(a)$
To prove: $(T, x\delta_{sf}, E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t$ s.t $V, H \vdash_{r'}^{r} () \Downarrow_t (), H$. From (E:Unit) we know that $t = 1$
Therefore it suffices to prove that
 (a) $\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow_{-} {}^t v_t \Downarrow_{-}^{J} {}^t v_f \wedge (T - 1, (), {}^t v_f) \in \lfloor unit \rfloor_{\mathcal{V}}$:
 We choose ${}^t v_t, {}^t v_f, J$ as $E_t', (), K^{unit}$ respectively
 Since from E-app we know that $E_t \Downarrow E_t'$, also since $E_t' \Downarrow^{K^{unit}} ()$ (from E-release, E-bind, E-store, E-return)
 Therefore we get the desired from Definition 34
 (b) $r - r' \leq J$:
 From (E:Unit) we know that $\exists p.r = p + K^{unit}$, $r' = p$ and since we know that $J = K^{unit}$, therefore we are done
2. base:

$$\frac{}{\Sigma; . \vdash_q^{q+K^{base}} c : \mathsf{b} \rightsquigarrow E_t} \text{unit}$$

where
$E_t = \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{base}} \text{ in } \mathsf{bind}\, a = \mathsf{store}(!c) \text{ in } \mathsf{ret}(a)$
$E_t' = \mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{base}} \text{ in } \mathsf{bind}\, a = \mathsf{store}(!c) \text{ in } \mathsf{ret}(a)$
To prove: $(T, x\delta_{sf}, E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor \mathsf{b} \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t$ s.t $V, H \vdash_{r'}^{r} c \Downarrow_t c, H$. From (E:base) we know that $t = 1$
Therefore it suffices to prove that
 (a) $\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow_{-} {}^t v_t \Downarrow_{-}^{J} {}^t v_f \wedge (T - 1, (), {}^t v_f) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}$:
 We choose ${}^t v_t, {}^t v_f, J$ as $E_t', !c, K^{base}$ respectively
 Since from E-app we know that $E_t \Downarrow E_t'$, also since $E_t' \Downarrow^{K^{base}} !c$ (from E-release, E-bind, E-store, E-return)
 Therefore we get the desired from Definition 34
 (b) $r - r' \leq J$:
 From (E:base) we know that $\exists p.r = p + K^{base}$, $r' = p$ and since we know that $J = K^{base}$, therefore we are done
3. var:

$$\frac{}{\Sigma; x : \tau \vdash_q^{q+K^{var}} x : \tau \rightsquigarrow E_t} \text{var}$$

where
$E_t = \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{var}} \text{ in } \mathsf{bind}\, a = \mathsf{store}\, x \text{ in } \mathsf{ret}(a)$
$E_t' = \mathsf{release} - = () \text{ in } \mathsf{bind} - = \uparrow^{K^{var}} \text{ in } \mathsf{bind}\, a = \mathsf{store}\, x \text{ in } \mathsf{ret}(a)$
To prove: $(T, x\delta_{sf}, E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t$ s.t $V, H \vdash_{r'}^{r} x \Downarrow_t V(x), H$. From (E:Var) we know that $t = 1$
Therefore it suffices to prove that
 (a) $\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow_{-} {}^t v_t \Downarrow_{-}^{J} {}^t v_f \wedge (T - 1, V(x), {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}$:
 We choose ${}^t v_t, {}^t v_f, J$ as $E_t', \delta_t(x)$ respectively
 Since from E-app we know that $E_t \Downarrow E_t'$, also since $E_t \Downarrow^{K^{var}} \delta_t(x)$ (from E-release, E-bind, E-store, E-return)
 Therefore we get the desired from Definition 34 and Lemma 40
 (b) $r - r' \leq J$:
 From (E:VAR) we know that $\exists p.r = p + K^{var}$, $r' = p$ and $J = K^{var}$, so we are done
4. app:

$$\frac{\tau_1 \overset{q/q'}{\to} \tau_2 \in \Sigma(f)}{\Sigma; x : \tau_1 \vdash^{q+K^{app}_1}_{q'-K^{app}_2} f\ x : \tau_2 \rightsquigarrow E_t} \text{app}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u \mathsf{\ in\ bind} - = \uparrow^{K^{app}_1} \mathsf{\ in\ bind\ } P = \mathsf{store}() \mathsf{\ in\ } E_1$
$E_1 = \mathsf{bind\ } f_1 = (f\ P\ x) \mathsf{\ in\ release\ } f_2 = f_1 \mathsf{\ in\ bind} - = \uparrow^{K^{app}_2} \mathsf{\ in\ bind\ } f_3 = \mathsf{store\ } f_2 \mathsf{\ in\ ret\ } f_3$
To prove: $(T, f\ x, E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor\tau_2\rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t <T$ s.t $V, H \vdash^{r}_{r'} f\ x\delta_{sf} \Downarrow_t {}^s v, H'$
and it suffices to prove that
$\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor\tau_2\rfloor^{H'}_{\mathcal{V}} \wedge r - r' \leq J$ (F-A0)

Since we are given that $(T, \delta_{sf}, \delta_{tf}) \in \lfloor\Sigma\rfloor^{H}_{\mathcal{V}}$ therefore from Definition 35 we know that
$(T, \delta_{sf}(f)\ \delta_{sf}, \delta_{tf}(f)\ \delta_{tf}) \in \lfloor(\tau_1 \overset{q/q'}{\to} \tau_2)\rfloor^{H}$

From Definition 33 we know that $\delta_{sf}(f) = (f(x) = e_s)$ and $\delta_{tf}(f) = \mathsf{fix} f.\lambda u.\lambda x.e_t$ and we have
$\forall^s v', {}^t v', T'<T\ .(T', {}^s v', {}^t v') \in \lfloor\tau_1\rfloor^{H}_{\mathcal{V}} \implies (T', e_s, e_t[()/u][{}^t v'/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor\tau_2\rfloor^{\{x\mapsto {}^s v'\},H}_{\mathcal{E}}$
(F-A1)

Since we are given that $(T, V, \delta_t) \in \lfloor\Gamma\rfloor^{H}_{\mathcal{V}}$ therefore we have
$(T, V(x), \delta_t(x)) \in \lfloor\tau_1\rfloor^{H}_{\mathcal{V}}$

This means from Lemma 36 we also have $(T - 1, V(x), \delta_t(x)) \in \lfloor\tau_1\rfloor^{H}_{\mathcal{V}}$

Instantiating (F-A1) with $T - 1, V(x), \delta_t(x)$ we get
$(T - 1, e_s, e_t[()/u][\delta_t(x)/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f]) \in \lfloor\tau_2\rfloor^{\{x\mapsto V(x)\},H}_{\mathcal{E}}$

This means from Definition 33 we have
$\forall H'_1, {}^s v_1, r_1, r'_1, t' <T - 1.\ V, H \vdash^{r_1}_{r'_1} e_s \Downarrow_{t'} {}^s v_1, H'_1 \implies \exists^t v_t, {}^t v_f, J_1.e_t[()/u][\delta_t(x)/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f] \Downarrow {}^t v_t \Downarrow^{J_1} {}^t v_f \wedge (T - 1 - t', {}^s v_1, {}^t v_f) \in \lfloor\tau_2\rfloor^{H'_1}_{\mathcal{V}} \wedge r_1 - r'_1 \leq J_1$ (F-A2)

Since we know that $V, H \vdash^{r}_{r'} f\ x\delta_{sf} \Downarrow_t {}^s v, H'$ where $t <T$ therefore from (E:FunApp) we know that $V, H \vdash^{r-K^{app}_1}_{r'+K^{app}_2} e_s \Downarrow_{t-1} {}^s v, H'$ therefore instantiating (F-A2) with $H', {}^s v, r - K^{app}_1, r' + K^{app}_2, t - 1$ we get $\exists^t v_t, {}^t v_f, J_1.e_t[()/u][\delta_t(x)/x][\mathsf{fix} f.\lambda u.\lambda x.e_t/f] \Downarrow {}^t v_t \Downarrow^{J}\ {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor\tau_2\rfloor^{H'}_{\mathcal{V}} \wedge (r - K^{app}_1) - (r' + K^{app}_2) \leq J_1$ (F-A3)

From E-release, E-bind, E-store we know that $J = J_1 + K^{app}_1 + K^{app}_2$ therefore we get the desired from (F-A3)

5. nil:

$$\frac{}{\Sigma; \emptyset \vdash^{q+K^{nil}}_{q} nil : L^{\vec{p}}\tau \rightsquigarrow E_t} \text{nil}$$

where
$E_t = \lambda u.\mathsf{release} - = u \mathsf{\ in\ bind} - = \uparrow^{K^{nil}} \mathsf{\ in\ bind\ } a = \mathsf{store}() \mathsf{\ in\ bind\ } b = \mathsf{store}\langle\langle a, nil\rangle\rangle \mathsf{\ in\ ret}(b)$
To prove: $(T, nil, E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor L^{\vec{p}}\tau\rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, t <T$ s.t $\emptyset, \emptyset \vdash^{p}_{p'} nil \Downarrow_t {}^s v, H'$
From (E:NIL) we know that ${}^s v = NULL$, $H' = H$ and $t = 1$ and it suffices to prove that

(a) $\exists^t v_t, {}^t v_f, J.e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - 1, nil, {}^t v_f) \in \lfloor L^{\vec{p}}\tau\rfloor_{\mathcal{V}}$:
From E-bind, E-release, E-return we know that ${}^t v = \langle\langle(), nil\rangle\rangle$ therefore from Definition 33 we get the desired

(b) $p - p' \leq J$:
Here $p = q + K^{nil}$, $p' = q$ and $J = K^{nil}$, so we are done

6. cons:

$$\frac{\vec{p} = (p_1, \ldots, p_k)}{\Sigma; x_h : \tau, x_t : L^{(\lhd\ \vec{p})}\tau \vdash^{q+p_1+K^{cons}}_{q} cons(x_h, x_t) : L^{p}\tau \rightsquigarrow E_t} \text{cons}$$

where

$E_t = \lambda u.\mathsf{release} - = u \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0$
$E_0 = x_t; x.\,\mathsf{let}\langle\langle x_1, x_2\rangle\rangle = x \text{ in } E_1$
$E_1 = \mathsf{release} - = x_1 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{store}\langle\langle a, x_h :: x_2\rangle\rangle$
$E'_t = \mathsf{release} - = () \text{ in } \mathsf{bind} - = \uparrow^{K^{cons}} \text{ in } E_0$

To prove: $(T, cons(x_h, x_t), E_t\ ()\ \delta_t\delta_{tf}) \in \lfloor L^{\vec{p}}\tau\rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, p, p', t < T$ s.t $\emptyset, \emptyset \vdash^{p}_{p'} cons(x_h, x_t)\delta_{sf} \Downarrow_t {}^s v, H'$
and it suffices to prove that

(a) $\exists^t v_t, {}^t v_f, J. E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, H'(\ell), {}^t v_f) \in \lfloor L^{\vec{p}}\tau\rfloor^{H}_{\mathcal{V}}$:
From (E-app) of λ-amor we know that $E_t\ () \Downarrow E'_t$
Also from E-release, E-bind, E-store we know that ${}^t v_f = \langle\langle (), \delta_t(x_h) :: \delta_t(x_t) \downarrow_2\rangle\rangle$

Therefore it suffices to prove that $(T - t, \ell, \langle\langle (), \delta_t(x_h) :: \delta_t(x_t) \downarrow_2\rangle\rangle) \in \lfloor L^{\vec{p}}\tau\rfloor^{H'}_{\mathcal{V}}$

From Definition 33 it further suffices to prove that
$(T - t, \ell, \delta_t(x_h) :: \delta_t(x_t) \downarrow_2) \in \lfloor L\ \tau\rfloor^{H'}_{\mathcal{V}}$

Since from (E:CONS) rule of univariate RAML we know that $H' = H[\ell \mapsto v]$ where $v = (V(x_h), V(x_t))$

Therefore it further suffices to prove that
$(T - t, V(x_h), \delta_t(x_h)) \in \lfloor\tau\rfloor^{H'}_{\mathcal{V}}$ and $(T - t, V(x_t), \delta_t(x_t) \downarrow_2) \in \lfloor L\ \tau\rfloor^{H'}_{\mathcal{V}}$

Since we are given that $(T, V, \delta_{tf}) \in \lfloor\Sigma\rfloor^{V,H}$ therefore from Definition 34 and Lemma 36 it means we have
$(T - t, V(x_h), \delta_t(x_h)) \in \lfloor\tau\rfloor^{H}_{\mathcal{V}}$ (F-C1)
and
$(T - t, V(x_t), \delta_t(x_t)) \in \lfloor L^{\lhd\vec{p}}\tau\rfloor^{H}_{\mathcal{V}}$
This means we also have $(T - t, V(x_t), \delta_t(x_t) \downarrow_2) \in \lfloor L\ \tau\rfloor^{H}_{\mathcal{V}}$ (F-C2)

Since $H' = H[\ell \mapsto v]$ where $v = (V(x_h), V(x_t))$ therefore we also have
We get the desired from (F-C1), (F-C2) and Definition 33

(b) $p - p' \leq J$:
From (E:CONS) we know that $p = q' + K^{cons}$ and $p' = q'$ for some q'. Also we know that $J = K^{cons}$.
Therefore we are done.

7. match:

$$\frac{\vec{p} = (p_1, \ldots, p_k) \qquad \begin{array}{c}\Sigma; \Gamma \vdash^{q - K_1^{matN}}_{q' + K_2^{matN}} e_1 : \tau' \rightsquigarrow e_{a1} \\ \Sigma; \Gamma, h : \tau, t : L^{(\lhd\ \vec{p})}\tau \vdash^{q + p_1 - K_1^{matC}}_{q' + K_2^{matN}} e_2 : \tau' \rightsquigarrow e_{a2}\end{array}}{\Sigma; \Gamma; x : L^p\tau \vdash^q_{q'} \mathsf{match}\ x \text{ with } |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau' \rightsquigarrow \qquad \lambda u.E_0} \text{ match}$$

where
$E_0 = \mathsf{release} - = u \text{ in } E_{0.1}$
$E_{0.1} = x; a.\,\mathsf{let}\langle\langle x_1, x_2\rangle\rangle = a \text{ in } E_1$
$E_1 = \mathsf{match}\ x_2 \text{ with } |nil \mapsto E_2\ |h :: l_t \mapsto E_3$
$E_2 = \mathsf{bind} - = \uparrow^{K_1^{matN}} \text{ in } E_{2.1}$
$E_{2.1} = \mathsf{bind}\, b = \mathsf{store}() \text{ in } E'_2$
$E'_2 = \mathsf{bind}\, c = (e_{a1}\ b) \text{ in } E'_{2.1}$
$E'_{2.1} = \mathsf{release}\, d = c \text{ in } E'_{2.2}$
$E'_{2.2} = \mathsf{bind} - = \uparrow^{K_2^{matN}} \text{ in } E'_{2.3}$
$E'_{2.3} = \mathsf{release} - = x_1 \text{ in } \mathsf{store}\, d$
$E_3 = \mathsf{bind} - = \uparrow^{K_1^{matC}} \text{ in } E_{3.1}$
$E_{3.1} = \mathsf{release} - = x_1 \text{ in } E_{3.2}$
$E_{3.2} = \mathsf{bind}\, b = \mathsf{store}() \text{ in } E_{3.3}$
$E_{3.3} = \mathsf{bind}\, t = \mathsf{ret}\langle\langle b, l_t\rangle\rangle \text{ in } E_{3.4}$
$E_{3.4} = \mathsf{bind}\, d = \mathsf{store}() \text{ in } E_{3.5}$
$E_{3.5} = \mathsf{bind}\, f = e_{a2}\ d \text{ in } E_{3.6}$
$E_{3.6} = \mathsf{release}\, g = f \text{ in } E_{3.7}$
$E_{3.7} = \mathsf{bind} - = \uparrow^{K_2^{matC}} \text{ in } \mathsf{store}\, g$
To prove: $(T, \mathsf{match}\ x \text{ with } |nil \mapsto e_1\ |h :: t \mapsto e_2, \lambda u.E_0\ ()\ \delta_t\delta_{tf}) \in \lfloor\tau'\rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, p, p', t < T$ s.t $V, H \vdash^{p}_{p'} (\mathsf{match}\ x \text{ with } |nil \mapsto e_1\ |h :: t \mapsto e_2)\delta_{sf} \Downarrow_t {}^s v, H'$
2 cases arise:

(a) $V(x) = NULL$:
Since $(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^{V,H}$ therefore from Definition 34 and Definition 33 we have
$\delta_t(x) = \langle\langle (), nil \rangle\rangle$

<u>IH</u>: $(T-1, e_1 \delta_{sf}, e_{a1}\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we have
$\forall H'_1, {}^s v_1, p_1, p'_1, t_1.\ V, H \vdash^{p_1}_{p'_1} e_1 \Downarrow_{t_1} {}^s v_1, H'_1 \implies \exists^t v_{t1}, {}^t v_{f1}, J_1.e_{a1} \Downarrow {}^t v_{t1} \Downarrow^{J_1} {}^t v_{f1} \wedge (T-1-t_1, {}^s v_1, {}^t v_{f1}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'_1} \wedge p_1 - p'_1 \le J_1$ (F-RUA-M0)

Since we are given that $V, H \vdash^p_{p'} (\mathsf{match}\ x\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\delta_{sf} \Downarrow_t {}^s v, H'$ therefore from (E:MatvhN) we know that $V, H \vdash^{p-K_1^{matN}}_{p'+K_2^{matN}} e_1 \Downarrow_{t-1} {}^s v, H'$ therefore instantiating (F-RUA-M0) with $H', {}^s v, p - K_1^{matN}, p' + K_2^{matN}$ we get
$\exists^t v_{t1}, {}^t v_{f1}, J_1.e_{a1} \Downarrow {}^t v_{t1} \Downarrow^{J_1} {}^t v_{f1} \wedge (T-t, {}^s v, {}^t v_{f1}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - K_1^{matN} - p' - K_2^{matN} \le J_1$
(F-RUA-M1)

It suffices to prove that
$\exists^t v_t, {}^t v_f, J.\lambda u.E_0\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T-t, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \le J$

We choose ${}^t v_t$ as ${}^t v_{t1}$, ${}^t v_f$ as ${}^t v_{f1}$ and J as $J_1 + K_1^{matN} + K_2^{matN}$ and we get the desired from E-bind, E-release, E-store and (F-RUA-M1)

(b) $V(x) = \ell_s$:
Since $(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^{V,H}$ therefore from Definition 34 and Definition 33 we have
$\delta_t(x) = \langle\langle (), {}^t v_h :: l_t \rangle\rangle$ s.t
$H(\ell_s) = ({}^s v_h, \ell_{ts})$, $({}^s v, {}^t v) \in \lfloor \tau' \rfloor_{\mathcal{V}}$ and $(\ell_s, l_t) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$ and

Let $V' = V \cup \{h \mapsto {}^s v_h\} \cup \{t \mapsto \ell_{ts}\}$ and $\delta'_t = \delta_t \cup \{h \mapsto {}^t v_h\} \cup \{t \mapsto l_{ts}\}$
From Definition 34 and Lemma 36 we have $(T-1, V', \delta'_t) \in \lfloor \Gamma, h : \tau, t : L^{\triangleleft \vec{p}} \tau \rfloor_{\mathcal{V}}^{V',H}$
Therefore from IH we have
$(T-1, e_2 \delta_{sf}, e_{a2}\ ()\ \delta'_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V',H}$

This means from Definition 33 we have
$\forall H'_2, {}^s v_2, p_2, p'_2, t_1.\ V, H \vdash^{p_1}_{p'_1} e_2 \Downarrow_{t_1} {}^s v_2, H'_2 \implies \exists^t v_{t2}, {}^t v_{f2}, J_2.e_{a2} \Downarrow {}^t v_{t2} \Downarrow^{J_2} {}^t v_{f2} \wedge (T-1-t_1, {}^s v_2, {}^t v_{f2}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'_2} \wedge p_2 - p'_2 \le J_2$ (F-RUA-M0.0)

Since we are given that $V, H \vdash^p_{p'} (\mathsf{match}\ x\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\delta_{sf} \Downarrow_t {}^s v, H'$ therefore from (E:MatvhC) we know that $V, H \vdash^{p-K_1^{matC}}_{p'+K_2^{matC}} e_2 \Downarrow_{t-1} {}^s v, H'$ therefore instantiating (F-RUA-M0.0) with $H', {}^s v, p - K_1^{matC}, p' + K_2^{matC}, t-1$ we get
$\exists^t v_{t2}, {}^t v_{f2}, J_2.e_{a2} \Downarrow {}^t v_{t2} \Downarrow^{J_2} {}^t v_{f2} \wedge (T-t, {}^s v_2, {}^t v_{f2}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'_2} \wedge p_2 - p'_2 \le J_2$ (F-RUA-M2)

It suffices to prove that
$\exists^t v_t, {}^t v_f, J.\lambda u.E_0\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T-t, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \le J$

We choose ${}^t v_t$ as ${}^t v_{t2}$, ${}^t v_f$ as ${}^t v_{f2}$ and J as $J_2 + K_1^{matC} + K_2^{matC}$ and we get the desired from E-bind, E-release, E-store and (F-RUA-M2)

8. Share:

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash^q_{q'} e : \tau' \leadsto e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = \tau_1 = \tau_2 = \mathbf{1}}{\Sigma; \Gamma, z : \tau \vdash^q_{q'} e[z/x, z/y] : \tau' \leadsto E_0} \text{ Share-unit}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\ a = coerce_{\mathbf{1},\mathbf{1},\mathbf{1}}\ z\ \mathsf{in}\ \mathsf{let} \langle\langle x, y \rangle\rangle = a\ \mathsf{in}\ e_a\ u$

$coerce_{\mathbf{1},\mathbf{1},\mathbf{1}} \triangleq \lambda u.\ \mathsf{ret} \langle\langle !(), !() \rangle\rangle$

To prove: $(T, e[z/x, z/y], E_0\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, p, p', t$ s.t $V, H \vdash^p_{p'} e[z/x, z/y]\delta_{sf} \Downarrow_t {}^s v, H'$
And we need to prove
$\exists^t v_t, {}^t v_f, J.E_0\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T-t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \le J$

Let
$V' = V \cup \{x \mapsto V(z)\} \cup \{y \mapsto V(z)\}$
$\delta'_t = \delta_t \cup \{x \mapsto \delta_t(z)\} \cup \{y \mapsto \delta_t(z)\}$

Since we are given that $(T, V, \delta_t) \in \lfloor \Gamma, z : \mathbf{1} \rfloor_{\mathcal{V}}^{V,H}$ therefore from Definition 34 we also have
$(T, V', \delta'_t) \in \lfloor \Gamma, x : \mathbf{1}, y : \mathbf{1} \rfloor_{\mathcal{V}}^{V',H}$
<u>IH</u>
$(T, e, e_a\ ()\ \delta'_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V',H}$
This means from Definition 33 we have
$\forall H'_1, {}^s v_1, p_1, p'_1, t_1.\ V', H \vdash_{p'_1}^{p_1} e \Downarrow_{t_1} {}^s v_1, H'_1 \implies \exists^t v_t, {}^t v_f, J.e_a() \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v_1, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge$
$p_1 - p'_1 \leq J$
Instantiating it with the given $H', {}^s v, p, p', t$ we get the desired

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = \tau_1 = \tau_2 = \mathsf{b}}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-base}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\mathsf{b,b,b}}\ z \mathsf{\ in\ let} \langle\langle x, y \rangle\rangle = a \mathsf{\ in\ } e_a\ u$

$coerce_{\mathsf{b,b,b}} \triangleq \lambda u.\, \mathsf{let}\,!\, u' = u \mathsf{\ in\ ret} \langle\langle u', u' \rangle\rangle$

Similar reasonign as in the unit case above

$$\frac{\begin{array}{c}\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \\ \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = L^{\vec{p}} \tau'' \qquad \tau_1 = L^{\vec{p_1}} \tau''_1 \qquad \tau_2 = L^{\vec{p_2}} \tau''_2 \qquad \tau'' = \tau''_1 \oplus \tau''_2 \qquad \vec{p} = \vec{p_1} \oplus \vec{p_2}\end{array}}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{ Share-list}$$

$E_0 = \lambda u.E_1$
$E_1 = \mathsf{bind}\, a = coerce_{\tau, \tau_1, \tau_2}\ z \mathsf{\ in\ let} \langle\langle x, y \rangle\rangle = a \mathsf{\ in\ } e_a\ u$
$coerce_{L^{\vec{p}} \tau, L^{\vec{p_1}} \tau_1, L^{\vec{p_2}} \tau_2} \triangleq \mathsf{fix} f. \lambda\_g. \lambda e.\, \mathsf{let}\,!\, g' = g \mathsf{\ in\ } e; x.\, \mathsf{let} \langle\langle p, l \rangle\rangle = x \mathsf{\ in\ } E_0$
where
$E_0 \triangleq \mathsf{release} - = p \mathsf{\ in\ } E_1$
$E_1 \triangleq \mathsf{match}\ l \mathsf{\ with\ } |nil \mapsto E_{2.1}\ |h :: t \mapsto E_3$
$E_{2.1} \triangleq \mathsf{bind}\, z_1 = \mathsf{store}() \mathsf{\ in\ } E_{2.2}$
$E_{2.2} \triangleq \mathsf{bind}\, z_2 = \mathsf{store}() \mathsf{\ in\ } E_{2.3}$
$E_{2.3} \triangleq \mathsf{ret} \langle\langle \langle\langle z_1, nil \rangle\rangle, \langle\langle z_2, nil \rangle\rangle \rangle\rangle$
$E_3 \triangleq \mathsf{bind}\, H = g'\ h \mathsf{\ in\ } E_{3.1}$
$E_{3.1} \triangleq \mathsf{bind}\, o_t = () \mathsf{\ in\ } E_{3.2}$
$E_{3.2} \triangleq \mathsf{bind}\, T = f\ g\ \langle\langle o_t, t \rangle\rangle \mathsf{\ in\ } E_4$
$E_4 \triangleq \mathsf{let} \langle\langle H_1, H_2 \rangle\rangle = H \mathsf{\ in\ } E_5$
$E_5 \triangleq \mathsf{let} \langle\langle T_1, T_2 \rangle\rangle = T \mathsf{\ in\ } E_6$
$E_6 \triangleq T_1; tp_1.\, \mathsf{let} \langle\langle p'_1, l'_1 \rangle\rangle = tp_1 \mathsf{\ in\ } E_{7.1}$
$E_{7.1} \triangleq T_2; tp_2.\, \mathsf{let} \langle\langle p'_2, l'_2 \rangle\rangle = tp_2 \mathsf{\ in\ } E_{7.2}$
$E_{7.2} \triangleq \mathsf{release} - = p'_1 \mathsf{\ in\ } E_{7.3}$
$E_{7.3} \triangleq \mathsf{release} - = p'_2 \mathsf{\ in\ } E_{7.4}$
$E_{7.4} \triangleq \mathsf{bind}\, o_1 = \mathsf{store}() \mathsf{\ in\ } E_{7.5}$
$E_{7.5} \triangleq \mathsf{bind}\, o_2 = \mathsf{store}() \mathsf{\ in\ } E_8$
$E_8 \triangleq \mathsf{ret} \langle\langle \langle\langle o_1, H_1 :: T_1 \rangle\rangle, \langle\langle o_2, H_2 :: T_2 \rangle\rangle \rangle\rangle$

To prove: $(T, e[z/x, z/y], E_0\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, p, p', t < T$ s.t $V, H \vdash_{p'}^{p} e[z/x, z/y]\delta_{sf} \Downarrow_t {}^s v, H'$
And we need to prove
$\exists^t v_t, {}^t v_f, J.E_0\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

Let
$V' = V \cup \{x \mapsto V(z)\} \cup \{y \mapsto V(z)\}$
$\delta'_t = \delta_t \cup \{x \mapsto \delta_t(z)\} \cup \{y \mapsto \delta_t(z)\}$
Since we are given that $(T, V, \delta_t) \in \lfloor \Gamma, z : \tau \rfloor_{\mathcal{V}}^{V,H}$ therefore from Definition 34 we also have
$(T, V', \delta'_t) \in \lfloor \Gamma, x : \tau_1, y : \tau_2 \rfloor_{\mathcal{V}}^{V',H}$
<u>IH</u>

$(T, e, e_a\ ()\ \delta'_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V',H}$
This means from Definition 33 we have
$\forall H'_1, {}^s v_1, p_1, p'_1, t_1.\ V', H \vdash_{p'_1}^{p_1} e \Downarrow_{t_1} {}^s v_1, H'_1 \implies \exists^t v_t, {}^t v_f, J. e_a() \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v_1, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p_1 - p'_1 \leq J$
Instantiating it with the given $H', {}^s v, p, p', t$ we get the desired

$$\frac{\Sigma; \Gamma, x : \tau_1, y : \tau_2 \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a \qquad \tau = \tau_1 \curlyvee \tau_2 \qquad \tau = (\tau_a, \tau_b) \qquad \tau_1 = (\tau'_a, \tau'_b) \qquad \tau_2 = (\tau''_a, \tau''_b) \qquad \tau = \tau_1 \oplus \tau_2}{\Sigma; \Gamma, z : \tau \vdash_{q'}^{q} e[z/x, z/y] : \tau' \rightsquigarrow E_0} \text{Share-pair}$$

$E_0 = \lambda u. E_1$
$E_1 = \mathsf{bind}\, a = coerce_{(\tau_a, \tau_b), (\tau'_a, \tau'_b), (\tau''_a, \tau''_b)}\ z \ \mathsf{in}\ \mathsf{let} \langle\langle x, y \rangle\rangle = a \ \mathsf{in}\ e_a\ u$

$coerce_{(\tau_a, \tau_b), (\tau'_a, \tau'_b), (\tau''_a, \tau''_b)} \triangleq \lambda\_g_1. \lambda\_g_2. \lambda p.\ \mathsf{let}\,!\langle\langle p_1, p_2 \rangle\rangle = p \ \mathsf{in}\ E_0$
where
$E_0 \triangleq \mathsf{let}\,!\, g'_1 = g_1 \ \mathsf{in}\ E_1$
$E_1 \triangleq \mathsf{let}\,!\, g'_2 = g_2 \ \mathsf{in}\ E_2$
$E_2 \triangleq \mathsf{bind}\, P'_1 = g'_1 p_1 \ \mathsf{in}\ E_3$
$E_3 \triangleq \mathsf{bind}\, P'_2 = g'_2 p_2 \ \mathsf{in}\ E_4$
$E_4 \triangleq \mathsf{let}\,!\langle\langle p'_{11}, p'_{12} \rangle\rangle = P'_1 \ \mathsf{in}\ E_5$
$E_5 \triangleq \mathsf{let}\,!\langle\langle p'_{21}, p'_{22} \rangle\rangle = P'_2 \ \mathsf{in}\ E_6$
$E_6 \triangleq \mathsf{ret} \langle\langle p'_{11}, p'_{21} \rangle\rangle, \langle\langle p'_{12}, p'_{22} \rangle\rangle$
Same reasoning as in the list subcase above

9. Sub:

$$\frac{\Sigma; \Gamma \vdash_{q'}^{q} e : \tau \rightsquigarrow e_a \qquad \tau <: \tau'}{\Sigma; \Gamma \vdash_{q'}^{q} e : \tau' \rightsquigarrow e_a}$$

To prove: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V,H}$
<u>IH</u>: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

We get the desired from IH and Lemma 43

10. Relax:

$$\frac{\Sigma; \Gamma \vdash_{p'}^{p} e : \tau \rightsquigarrow e_a \qquad q \geq p \qquad q - p \geq q' - p'}{\Sigma; \Gamma \vdash_{q'}^{q} e : \tau \rightsquigarrow E_t}$$

where
$E_t = \lambda o. E_0$
$E_0 = \mathsf{release} - = o \ \mathsf{in}\ E_1$
$E_1 = \mathsf{bind}\, a = \mathsf{store}() \ \mathsf{in}\ E_2$
$E_2 = \mathsf{bind}\, b = e_a\ a \ \mathsf{in}\ E_3$
$E_3 = \mathsf{release}\, c = b \ \mathsf{in}\ \mathsf{store}\, c$

To prove: $(T, e, E_t\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t < T$ s.t $\emptyset, \emptyset \vdash_{r'}^{r} e \Downarrow_t {}^s v, H'$
And it suffices to prove that
$\exists^t v_t, {}^t v_f, J. E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}} \wedge r - r' \leq J$ (F-R0)

<u>IH</u>: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$
This means from Definition 33 we have
$\forall^s v_1, H'_1, r_1, r'_1, t_1 < T\ .V, H \vdash_{r'_1}^{r_1} e \Downarrow_{t_1} {}^s v_1, H' \implies \exists^t v_t, {}^t v_f, J. e_a\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}} \wedge r - r' \leq J$

Instantiating it with the given ${}^s v, H', r, r', t$ we get
$\exists^t v'_t, {}^t v'_f, J'. e_a\ () \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f \wedge (T - t, {}^s v, {}^t v'_f) \in \lfloor \tau \rfloor_{\mathcal{V}} \wedge r - r' \leq J'$ (F-R1)

In order to prove (F-R0) we choose ${}^t v_t, {}^t v_f, J$ as ${}^t v'_t, {}^t v'_f, J'$ and we get the desired from E-app, E-release, E-bind, E-store and (F-R1)

11. Super:

$$\frac{\Sigma; \Gamma, x : \tau_1 \vdash^q_{q'} e : \tau \leadsto e_a \qquad \tau_1' <: \tau_1}{\Sigma; \Gamma, x : \tau_1' \vdash^q_{q'} e : \tau \leadsto e_a} \text{Super}$$

Given: $(T, V, \delta_t) \in \lfloor \Gamma, x : \tau_1' \rfloor^H_{\mathcal{V}}$
To prove: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 it suffices to prove that
$\forall H', {}^s v, p, p', t < T\ .\ V, H \vdash^p_{p'} e_s \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J. e_a\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p - p' \leq J$

This means given some $H', {}^s v, p, p', t < T$ s.t $V, H \vdash^p_{p'} e_s \Downarrow_t {}^s v, H'$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J. e_a\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p - p' \leq J$ (F-Su0)

Since we are given that $(T, V, \delta_t) \in \lfloor \Gamma, x : \tau_1' \rfloor^H_{\mathcal{V}}$ therefore from Definition 34 we know that $(T, V(x), \delta_t(x)) \in \lfloor \tau_1' \rfloor^H_{\mathcal{V}}$
Therefore from Lemma 42 we know that $(T, V(x), \delta_t(x)) \in \lfloor \tau_1 \rfloor^H_{\mathcal{V}}$

<u>IH</u>: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we have
$\forall H_i', {}^s v_i, p_i, p_i', t_1.\ V, H \vdash^{p_i}_{p_i'} e \Downarrow_{t_1} {}^s v, H' \implies \exists^t v_t, {}^t v_f, J. e_a\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v_i, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p_i - p_i' \leq J$

Instantiating it with the given $H', {}^s v, p, p', t$ we get the desired

12. Let:

$$\frac{\Sigma; \Gamma_1 \vdash^{q - K_1^{let}}_{p} e_1 : \tau_1 \leadsto e_{a1} \qquad \Sigma; \Gamma_2, x : \tau_1 \vdash^{p - K_2^{let}}_{q' + K_3^{let}} e_2 : \tau_1 \leadsto e_{a2}}{\Sigma; \Gamma_1, \Gamma_2 \vdash^q_{q'} \mathsf{let}\ x = e_1\ in\ e_2 : \tau \leadsto E_t} \text{Let}$$

where
$E_t = \lambda u. E_0$
$E_0 = \mathsf{release} - = u\ \mathsf{in}\ E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K_1^{let}}\ \mathsf{in}\ E_2$
$E_2 = \mathsf{bind}\ a = \mathsf{store}()\ \mathsf{in}\ E_3$
$E_3 = \mathsf{bind}\ b = e_{a1}\ a\ \mathsf{in}\ E_4$
$E_4 = \mathsf{release}\ x = b\ \mathsf{in}\ E_5$
$E_5 = \mathsf{bind} - = \uparrow^{K_2^{let}}\ \mathsf{in}\ E_6$
$E_6 = \mathsf{bind}\ c = \mathsf{store}()\ \mathsf{in}\ E_7$
$E_7 = \mathsf{bind}\ d = e_{a2}\ c\ \mathsf{in}\ E_8$
$E_8 = \mathsf{release}\ f = d\ \mathsf{in}\ E_9$
$E_9 = \mathsf{bind} - = \uparrow^{K_3^{let}}\ \mathsf{in}\ E_{10}$
$E_{10} = \mathsf{bind}\ g = \mathsf{store}\ f\ \mathsf{in}\ \mathsf{ret}\ g$

To prove: $(T, \mathsf{let}\ x = e_1\ in\ e_2, E_t\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we are given some
${}^s v, H', {}^s v, r, r', t < T$ s.t $V, H \vdash^r_{r'} (\mathsf{let}\ x = e_1\ in\ e_2)\delta_{sf} \Downarrow_t {}^s v, H'$
it suffices to prove that
$\exists^t v_t, {}^t v_f, J. e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge r - r' \leq J$ (F-L0)

Since we are given that $(T, V, \delta_t)\ in \lfloor \Gamma_1, \Gamma_2 \rfloor^H_{\mathcal{V}}$ therefore we know that
$\exists V_1, V_2, \delta_t^1, \delta_t^2$ s.t $V = V_1, V_2,\ \delta_t = \delta_t^1, \delta_t^2$ and
$(T, V_1, \delta_t^1) \in \lfloor \Gamma_1 \rfloor^H_{\mathcal{V}}$ and $(T, V_2, \delta_t^2) \in \lfloor \Gamma_2 \rfloor^H_{\mathcal{V}}$

<u>IH1</u>
$(T, e_1, e_{a1}\ ()\ \delta_t^1 \delta_{tf}) \in \lfloor \tau_1 \rfloor^{V_1,H}_{\mathcal{E}}$

This means from Definition 33 we have
$\forall H_1', {}^s v_1, p_1, p_1', t_1.\ V, H \vdash^{p_1}_{p_1'} e_1 \Downarrow_{t_1} {}^s v_1, H' \implies \exists^t v_{t1}, {}^t v_{f1}, J_1. e_{a1}\ () \Downarrow {}^t v_{t1} \Downarrow^{J_1} {}^t v_{f1} \wedge (T - t_1, {}^s v_1, {}^t v_{f1}) \in \lfloor \tau_1 \rfloor^{H_1'}_{\mathcal{V}} \wedge p_1 - p_1' \leq J_1$ (F-L1)

Since we know that $V, H \vdash^r_{r'} (\mathsf{let}\ x = e_1\ in\ e_2)\delta_{sf} \Downarrow_t {}^s v, H'$ therefore from (E:Let) we know that $\exists H_1', {}^s v_1, r_1, t_1$ s.t $V, H \vdash^{r - K_1^{let}}_{r_1} e_1 \delta_{sf} \Downarrow_{t_1} {}^s v_1, H_1'$

Instantiating (F-L1) with $H_1', {}^s v_1, r - K_1^{let}, r_1, t_1$ we get
$\exists^t v_{t1}, {}^t v_{f1}, J_1.e_{a1}\ () \Downarrow {}^t v_{t1} \Downarrow^{J_1} {}^t v_{f1} \wedge (T\ {-t_1}, {}^s v_1, {}^t v_{f1}) \in \lfloor \tau_1 \rfloor_{\mathcal{V}}^{H_1'} \wedge r - K_1^{let} - r_1 \leq J_1$ (F-L1.1)

<u>IH2</u>
$(T\ {-t_1}, e_2, e_{a2}\ ()\ \delta_t^2 \cup \{x \mapsto {}^t v_{f1}\}\delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V_2 \cup \{x \mapsto {}^s v_1\}, H_1'}$

This means from Definition 33 we have
$\forall H_2', {}^s v_2, p_2, p_2', t_2 < T\ {-t_1}.\ V, H \vdash_{p_2'}^{p_2} e_2 \Downarrow_{t_2} {}^s v_2, H' \implies \exists^t v_{t2}, {}^t v_{f2}, J_2.e_{a2}\ () \Downarrow {}^t v_{t2} \Downarrow^{J_2} {}^t v_{f2} \wedge (T\ {-t_1} - t_2, {}^s v_2, {}^t v_{f2}) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H_2'} \wedge p_2 - p_2' \leq J_2$ (F-L2)

Since we know that $V, H \vdash_{r'}^{r} (\mathsf{let}\ x = e_1\ in\ e_2)\delta_{sf} \Downarrow_t {}^s v, H'$ therefore from (E:Let) we know that $\exists H_2', {}^s v_2, t_2 < t - t_1$ s.t $V, H \vdash_{r' + K_3^{let}}^{r_1 - K_2^{let}} e_2 \delta_{sf} \Downarrow_{t_2} {}^s v, H_2'$

Instantiating (F-L2) with $H_2', {}^s v, r_1 - K_2^{let}, r' + K_3^{let}, t_2$ we get
$\exists^t v_{t2}, {}^t v_{f2}, J_2.e_{a2}\ () \Downarrow {}^t v_{t2} \Downarrow^{J_2} {}^t v_{f2} \wedge (T\ {-t_1} - t_2, {}^s v, {}^t v_{f2}) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H_2'} \wedge r_1 - K_2^{let} - (r' + K_3^{let}) \leq J_2$ (F-L2.1)

In order to prove (F-L0) we choose ${}^t v_t$ as ${}^t v_{t2}$, ${}^t v_f$ as ${}^t v_{f2}$, J as $J_1 + J_2 + K_1^{let} + K_2^{let} + K_3^{let}$, t as $t_1 + t_2 + 1$ and we get the desired from (F-L1.1) and (F-L2.1) and Lemma 36

13. Pair:

$$\frac{}{\Sigma; x_1 : \tau_1, x_2 : \tau_2 \vdash_q^{q + K^{pair}} (x_1, x_2) : (\tau_1, \tau_2) \rightsquigarrow E_t}\ \text{pair}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u\ \mathsf{in}\ E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K^{pair}}\ \mathsf{in}\ E_2$
$E_2 = \mathsf{bind}\ a = \mathsf{store}(x_1, x_2)\ \mathsf{in}\ \mathsf{ret}\ a$

Given: $(T, V, \delta_t) \in \lfloor x_1 : \tau_1, x_2 : \tau_2 \rfloor_{\mathcal{V}}^H$
To prove: $(T, (x_1, x_2), E_t\ ()\ \delta_t \delta_{tf}) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 it suffices to prove that
$\forall H', {}^s v, r, r', t < T\ .\ V, H \vdash_{r'}^{r} (x_1, x_2) \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T\ {-t}, {}^s v, {}^t v_f) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H'} \wedge r - r' \leq J$

This means given some $H', {}^s v, r, r', t < T$ s.t $V, H \vdash_{r'}^{r} (x_1, x_2) \Downarrow_t {}^s v, H'$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T\ {-t}, {}^s v, {}^t v_f) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H'} \wedge r - r' \leq J$ (F-P0)

This means we need to prove that $\exists^t v_t, {}^t v_f, J$

- $E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f$:
  From E-app, E-release,E-bind,E-tick, E-store and E-return we know that ${}^t v_t = E_0, {}^t v_f = (\delta_t(x_1), \delta_t(x_2))$ and $J = K^{pair}$
- $(T\ {-t}, {}^s v, {}^t v_f) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H'}$:
  Since we are given that $V, H \vdash_{r'}^{r} (x_1, x_2) \Downarrow_t {}^s v, H'$, therefore from (E:Pair) we know that ${}^s v = \ell$ where $\ell \notin dom(H)$ and $H' = H[\ell \mapsto (V(x_1), V(x_2))]$

  Since we are given that $(T, V, \delta_t) \in \lfloor x_1 : \tau_1, x_2 : \tau_2 \rfloor_{\mathcal{V}}^H$ therefore from Definition 34, Definition 33 and Lemma 36 we get the desired.
- $r - r' \leq J$:
  From (E:Pair) we know that $\exists p.r = p + K^{pair}$ and $r' = p$. Since we know that $J = K^{pair}$, therefore we are done.

14. MatP:

$$\frac{\tau = (\tau_1, \tau_2) \qquad \Sigma, \Gamma, x_1 : \tau_1, x_2 : \tau_2 \vdash_{q' + K_2^{matP}}^{q - K_1^{matP}} e : \tau' \rightsquigarrow e_t}{\Sigma; \Gamma, x : \tau \vdash_q^q match\ x\ with\ (x_1, x_2) \to e : \tau' \rightsquigarrow E_t}\ \text{matP}$$

where
$E_t = \lambda u.E_0$
$E_0 = \mathsf{release} - = u\ \mathsf{in}\ E_1$
$E_1 = \mathsf{bind} - = \uparrow^{K_1^{matP}}\ \mathsf{in}\ E_2$
$E_2 = \mathsf{let}\langle\langle x_1, x_2 \rangle\rangle = x\ \mathsf{in}\ E_3$
$E_3 = \mathsf{bind}\ a = \mathsf{store}()\ \mathsf{in}\ E_4$

$E_4 = \mathsf{bind}\, b = e_t\ a\ \mathsf{in}\ E_5$
$E_5 = \mathsf{release}\, c = b\ \mathsf{in}\ E_6$
$E_6 = \mathsf{bind}\, - = \uparrow^{K_2^{matP}}\ \mathsf{in}\ E_7$
$E_7 = \mathsf{bind}\, d = \mathsf{store}\, c\ \mathsf{in}\ \mathsf{ret}\, d$

Given: $(T, V, \delta_t) \in \lfloor \Gamma, x : \tau \rfloor_{\mathcal{V}}^{H}$
To prove: $(T, (match\ x\ with\ (x_1, x_2) \to e), E_t\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 it suffices to prove that
$\forall H', {}^s v, p, p', t < T\ .\ V, H \vdash^{p}_{p'} (match\ x\ with\ (x_1, x_2) \to e) \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

This means given some $H', {}^s v, p, p', t < T$ s.t $V, H \vdash^{p}_{p'} (match\ x\ with\ (x_1, x_2) \to e) \Downarrow_t {}^s v, H'$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J.E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$ (F-MP0)

Since we are given that $(T, V, \delta_t) \in \lfloor \Gamma, x : \tau \rfloor_{\mathcal{V}}^{H}$ therefore from Definition 34 and since $\tau = (\tau_1, \tau_2)$ therefore we know that $(T, V(x), \delta_t(x)) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$

This means from Definition 33 that $\exists \ell$ s.t $H(\ell) = ({}^s v_1, {}^s v_2) \wedge (T, {}^s v_1, {}^t v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, {}^s v_2, {}^t v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$

<u>IH</u>: $(T, e, e_t\ ()\ \delta_t \cup \{x_1 \mapsto {}^t v_1\} \cup \{x_2 \mapsto {}^t v_2\} \delta_{tf}) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V \cup \{x_1 \mapsto {}^s v_1\} \cup \{x_2 \mapsto {}^s v_2\}, H}$

This means from Definition 33 we have
$\forall H'_i, {}^s v_i, p_i, p'_i, t_1 < T - 1.\ V, H \vdash^{p_i}_{p'_i} e \Downarrow_{t_1} {}^s v, H'_i \implies \exists^t v_{t1}, {}^t v_{f1}, J_1.e\ () \Downarrow {}^t v_{t1} \Downarrow^J {}^t v_{f1} \wedge (T - t_1, {}^s v_i, {}^t v_{f1}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'_i} \wedge p_i - p'_i \leq J_1$

Since we are given that $V, H \vdash^{p}_{p'} (match\ x\ with\ (x_1, x_2) \to e) \Downarrow {}^s v, H'$ therefore from (E:MatP) we know that
$V \cup \{x_1 \mapsto {}^s v_1\} \cup \{x_2 \mapsto {}^s v_2\}, H \vdash^{p - K_1^{matP}}_{p' + K_2^{matP}} e \Downarrow_{t-1} {}^s v, H'$
Instantiating it with the given $H', {}^s v, p - K_1^{matP}, p' + K_2^{matP}, t - 1$ we get
$\exists^t v_{t1}, {}^t v_{f1}, J_1.e\ () \Downarrow {}^t v_{t1} \Downarrow^J {}^t v_{f1} \wedge (T - t, {}^s v, {}^t v_{f1}) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - K_1^{matP} - (p' + K_2^{matP}) \leq J_1$ (F-MP1)

In order to prove (F-MP0) we choose ${}^t v_t$ as ${}^t v_{t1}$, ${}^t v_f$ as ${}^t v_{f1}$, J as $J_1 + K_1^{matP} + K_2^{matP}$ and t_1 as $t - 1$ and it suffices to prove that

- $E_t\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f$:
  We get the desired from E-app, E-bind, E-release, E-store, E-tick, E-return and (F-MP1)
- $(T - t, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'}$:
  From (F-MP1)
- $p - p' \leq J$:
  We get this directly from (F-MP1)

15. Augment:

$$\frac{\Sigma; \Gamma \vdash^{q}_{q'} e : \tau \rightsquigarrow e_a}{\Sigma; \Gamma, x : \tau' \vdash^{q}_{q'} e : \tau \rightsquigarrow e_a}\ \text{Augment}$$

Given: $(T, V \cup \{x \mapsto {}^s v_x\}, \delta_t \cup \{x \mapsto {}^t v_x\}) \in \lfloor \Gamma, x : \tau' \rfloor_{\mathcal{V}}^{H}$
To prove: $(T, e, e_a\ ()\ \delta_t \cup \{x \mapsto {}^t v_x\} \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V \cup \{x \mapsto {}^s v_x\}, H}$

This means from Definition 33 it suffices to prove that
$\forall H', {}^s v, p, p', t < T\ .\ V \cup \{x \mapsto {}^s v_x\}, H \vdash^{p}_{p'} e_s \Downarrow_t {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.e_a\ () \delta_t \cup \{x \mapsto {}^t v_x\} \delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

This means given some $H', {}^s v, p, p', t < T$ s.t $V \cup \{x \mapsto {}^s v_x\}, H \vdash^{p}_{p'} e_s \Downarrow_t {}^s v, H'$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J.e_a\ () \delta_t \cup \{x \mapsto {}^t v_x\} \delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$ (F-Ag0)

Since we are given that $(T, V \cup \{x \mapsto {}^s v_x\}, \delta_t \cup \{x \mapsto {}^t v_x\}) \in \lfloor \Gamma, x : \tau' \rfloor_{\mathcal{V}}^{H}$
therefore from Definition 34 we know that
$(T, V, \delta_t) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^{H}$

<u>IH</u>: $(T, e, e_a\ ()\ \delta_t \delta_{tf}) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H}$

This means from Definition 33 we have
$\forall H'_i, {}^s v_i, p_i, p'_i, t_1 < T\ .\ V, H \vdash^{p_i}_{p'_i} e \Downarrow_{t_1} {}^s v, H' \implies \exists^t v_t, {}^t v_f, J.e_a\ () \delta_t \delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v_i, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p_i - p'_i \leq J$ (F-Ag1)

Since we are given $V \cup \{x \mapsto {}^s v_x\}, H \vdash^p_{p'} e_s \Downarrow_t {}^s v, H'$ and since $x \notin free(e)$ therefore we also have $V, H \vdash^p_{p'} e_s \Downarrow_t {}^s v, H'$
Instantiating (F-Ag1) with the given $H', {}^s v, p, p', t$ we get
$\exists^t v_t, {}^t v_f, J.e_a\ ()\delta_t \delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v_i, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p_i - p'_i \leq J$

Also since $x \notin free(e)$ therefore we get
$\exists^t v_t, {}^t v_f, J.e_a\ ()\delta_t \cup \{x \mapsto {}^t v_x\}\delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v_i, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p_i - p'_i \leq J$

□

Lemma 42 (Value subtyping lemma)**.** $\forall \tau, \tau', H, {}^s v, {}^t v, T.$
$\tau <: \tau' \wedge (T, {}^s v, {}^t v) \in \lfloor \tau \rfloor^H_{\mathcal{V}} \implies (T, {}^s v, {}^t v) \in \lfloor \tau' \rfloor^H_{\mathcal{V}}$

Proof. Proof by induction on the subtyping relation of Univariate RAML

1. Unit:

$$\frac{}{unit <: unit}$$

Given: $(T, {}^s v, {}^t v) \in \lfloor unit \rfloor^H_{\mathcal{V}}$
To prove: $(T, {}^s v, {}^t v) \in \lfloor unit \rfloor^H_{\mathcal{V}}$
Trivial

2. Base:

$$\frac{}{\mathsf{b} <: \mathsf{b}}$$

Given: $(T, {}^s v, {}^t v) \in \lfloor \mathsf{b} \rfloor^H_{\mathcal{V}}$
To prove: $(T, {}^s v, {}^t v) \in \lfloor \mathsf{b} \rfloor^H_{\mathcal{V}}$
Trivial

3. Pair:

$$\frac{\tau_1 <: \tau'_1 \qquad \tau_2 <: \tau'_2}{(\tau_1, \tau_2) <: (\tau'_1, \tau'_2)}$$

Given: $(T, {}^s v, {}^t v) \in \lfloor (\tau_1, \tau_2) \rfloor^H_{\mathcal{V}}$
To prove: $(T, {}^s v, {}^t v) \in \lfloor (\tau'_1, \tau'_2) \rfloor^H_{\mathcal{V}}$
From Definition 33 we know that ${}^s v = \ell$ s.t
$H(\ell) = ({}^s v_1, {}^s v_2) \wedge (T, {}^s v_1, {}^t v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, {}^s v_2, {}^t v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$ (S-P0)

<u>IH1</u> $(T, {}^s v_1, {}^t v_1) \in \lfloor \tau'_1 \rfloor^H_{\mathcal{V}}$
<u>IH2</u> $(T, {}^s v_2, {}^t v_2) \in \lfloor \tau'_2 \rfloor^H_{\mathcal{V}}$

Again from Definition 33 it suffices to prove that
$H(\ell) = ({}^s v_1, {}^s v_2) \wedge (T, {}^s v_1, {}^t v_1) \in \lfloor \tau'_1 \rfloor_{\mathcal{V}} \wedge (T, {}^s v_2, {}^t v_2) \in \lfloor \tau'_2 \rfloor_{\mathcal{V}}$

We get this directly from (S-P0), IH1 and IH2

4. List:

$$\frac{\tau_1 <: \tau_2 \qquad \vec{p} \geq \vec{q}}{L^{\vec{p}} \tau_1 <: L^{\vec{q}} \tau_2}$$

Given: $(T, {}^s v, {}^t v) \in \lfloor L^{\vec{p}} \tau_1 \rfloor^H_{\mathcal{V}}$
To prove: $(T, {}^s v, {}^t v) \in \lfloor L^{\vec{q}} \tau_2 \rfloor^H_{\mathcal{V}}$

From Definition 33 we know that ${}^s v = l_s$ and ${}^t v = \langle\langle (), l_t \rangle\rangle$ s.t $(T, l_s, l_t) \in \lfloor L\ \tau_1 \rfloor_{\mathcal{V}}$
Similarly from Definition 33 it suffices to show that
$(T, l_s, l_t) \in \lfloor L\ \tau_2 \rfloor_{\mathcal{V}}$

We induct on $(T, l_s, l_t) \in \lfloor L\ \tau_1 \rfloor_{\mathcal{V}}$

- Base case:
  In this case $l_s = NULL$ and $l_t = nil$:
  It suffices to prove that $(T, NULL, nil) \in \lfloor L\ \tau_2 \rfloor_{\mathcal{V}}$
  This holds trivially from Definition 33
- Inductive case
  In this case we have $l_s = \ell$ and $l_t = {}^t v_h :: l_{tt}$:
  It suffices to prove that $(T, \ell, {}^t v_h :: l_{tt}) \in \lfloor L\ \tau_2 \rfloor_{\mathcal{V}}$
  Again from Definition 33 it suffices to show that

$\exists {}^s v_{h1}, \ell_{s1}. H(\ell) = ({}^s v_{h1}, \ell_{s1}) \wedge (T, {}^s v_{h1}, {}^t v_h) \in \lfloor \tau_2 \rfloor_{\mathcal{V}} \wedge (T, \ell_{s1}, l_{tt}) \in \lfloor L\ \tau_2 \rfloor_{\mathcal{V}}$

Since we are given that $(T, \ell, {}^t v_h :: l_{tt}) \in \lfloor L\ \tau_1 \rfloor_{\mathcal{V}}$ therefore from Definition 33 we have
$\exists {}^s v_h, \ell_s. H(\ell) = ({}^s v_h, \ell_s) \wedge (T, {}^s v_h, {}^t v_h) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, \ell_s, l_{tt}) \in \lfloor L\ \tau_1 \rfloor_{\mathcal{V}}$ (S-L1)

We choose ${}^s v_{h1}$ as ${}^s v_h$ and ℓ_{s1} as ℓ_s

- $H(\ell) = ({}^s v_h, \ell_s)$:
  Directly from (S-L1)
- $(T, {}^s v_h, {}^t v_h) \in \lfloor \tau_1 \rfloor_{\mathcal{V}}$:
  From IH of outer induction
- $(T, \ell_s, l_{tt}) \in \lfloor L\ \tau_2 \rfloor_{\mathcal{V}}$:
  From IH of inner induction

□

Lemma 43 (Expression subtyping lemma). $\forall \tau, \tau', V, H, e_s, e_t.$
$\tau <: \tau' \wedge (T, e_s, e_t) \in \lfloor \tau \rfloor_{\mathcal{E}}^{V,H} \implies (T, e_s, e_t) \in \lfloor \tau' \rfloor_{\mathcal{E}}^{V,H}$

Proof. From Definition 33 we are given that
$\forall H', {}^s v, p, p', t < T\ .\ V, H \vdash^{p}_{p'} e_s \Downarrow_t {}^s v, H' \implies \exists {}^t v_t, {}^t v_f, J. e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$ (SE0)

Also from Definition 33 it suffices to prove that
$\forall H', {}^s v, p, p', t_1 < T\ .\ V, H \vdash^{p}_{p'} e_s \Downarrow_{t_1} {}^s v, H' \implies \exists {}^t v_t, {}^t v_f, J. e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

This means given some $H', {}^s v, p, p', t_1 < T$ s.t $V, H \vdash^{p}_{p'} e_s \Downarrow_{t_1} {}^s v, H'$ it suffices to prove that
$\exists {}^t v_t, {}^t v_f, J. e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v, {}^t v_f) \in \lfloor \tau' \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$

We instantiate (SE0) with $H', {}^s v, p, p', t_1$ and we get
$\exists {}^t v_t, {}^t v_f, J. e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t_1, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'} \wedge p - p' \leq J$ (SE1)

We get the desired from (SE1) and Lemma 42

□

A.5.3 Re-deriving Univariate RAML's soundness

Definition 44 (Translation of Univariate RAML stack). $\overline{(V : \Gamma)_H} \triangleq \forall x \in dom(\Gamma). \overline{(V(x))_{H,\Gamma(x)}}$

Definition 45 (Translation of Univariate RAML values).

$$\overline{({}^s v)_{H,\tau}} \triangleq \begin{cases} {}^s v & \tau = unit \\ !{}^s v & \tau = \mathsf{b} \\ \langle\langle (), \overline{({}^s v)_{H,L\ \tau'}} \rangle\rangle & \tau = L^{-}\tau' \\ nil & \tau = L\ \tau' \wedge {}^s v = NULL \\ \overline{(H(\ell) \downarrow_1)_{H,\tau'}} :: \overline{(H(\ell) \downarrow_2)_{H,L\tau'}} & \tau = L\ \tau' \wedge {}^s v = \ell \\ \langle\langle \overline{(H(\ell) \downarrow_1)_{H,\tau_1}}, \overline{(H(\ell) \downarrow_2)_{H,\tau_2}} \rangle\rangle & \tau = (\tau_1, \tau_2) \wedge {}^s v = \ell \end{cases}$$

Lemma 46 (Irrelevance of T for translated value). $\forall {}^s v, \tau, H.$
$H \models {}^s v \in [\![\tau]\!]$ *in RAML* $\implies \forall\, T\, .(\Phi_H({}^s v : \tau), T, \overline{({}^s v)_{H,\tau}}) \in [\![(\!|\tau|\!)]\!]$ *in λ-amor*

Proof. By induction on τ

1. $\tau = unit$:
   To prove: $\forall\, T\, .(\Phi_H({}^s v : \tau), T, \overline{({}^s v)_{H,\tau}}) \in [\![(\!|unit|\!)]\!]$

   This means given some T it suffices to prove that
   $(\Phi_H({}^s v : unit), T, \overline{({}^s v)_{H,unit}}) \in [\![\mathbf{1}]\!]$

   We know that $\Phi_H({}^s v : unit) = 0$ therefore it suffices to prove that
   $(0, T, {}^s v) \in [\![\mathbf{1}]\!]$

   Since we know that ${}^s v \in [\![unit]\!]$ therefore we know that ${}^s v = ()$
   Therefore we get the desired directly from Definition 15
2. $\tau = \mathsf{b}$:
   To prove: $\forall\, T\, .(\Phi_H({}^s v : \tau), T, \overline{({}^s v)_{H,\tau}}) \in [\![(\!|\mathsf{b}|\!)]\!]$

   This means given some T it suffices to prove that
   $(\Phi_H({}^s v : \mathsf{b}), T, \overline{({}^s v)_{H,\tau}}) \in [\![!\mathsf{b}]\!]$

We know that $\Phi_H({}^s v : \mathsf{b}) = 0$ therefore it suffices to prove that
$(0, T, !{}^s v) \in [\![!\mathsf{b}]\!]$
From Definition 15 it suffices to prove that
$(0, T, {}^s v) \in [\![\mathsf{b}]\!]$

Since we know that ${}^s v \in [\![\mathsf{b}]\!]$
Therefore we get the desired directly from Definition 15

3. $\tau = L^{\vec{q}}\tau'$:
By induction on ${}^s v$
   - ${}^s v = NULL = []$:
     To prove: $\forall\, T\, .(\Phi_H({}^s v : \tau), T, \overline{({}^s v)_{H, L^{\vec{q}}\tau'}}) \in [\![(\!|L^{\vec{q}}\tau'|\!)]\!]$

     This means given some T it suffices to prove that
     $(\Phi_H([] : L^{\vec{q}}\tau'), T, \langle\!\langle (), nil\rangle\!\rangle) \in [\![\exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau|\!))]\!]$

     We know that $\Phi_H([] : L^{\vec{q}}\tau') = 0$ therefore it suffices to prove that
     $(0, T, \langle\!\langle (), nil\rangle\!\rangle) \in [\![\exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau|\!))]\!]$
     From Definition 15 it suffices to prove that
     $\exists s'.(0, T, \langle\!\langle (), nil\rangle\!\rangle) \in [\![([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau|\!))[s'/s]]\!]$

     We choose s' as 0 and it suffices to prove that
     $(0, T, \langle\!\langle (), nil\rangle\!\rangle) \in [\![([\phi(\vec{q}, 0)]\, \mathbf{1} \otimes L[0](\!|\tau|\!))]\!]$

     From Definition 15 it further suffices to prove that
     $\exists p_1, p_2 . p_1 + p_2 \leq 0 \wedge (p_1, T, ()) \in [\![([\phi(\vec{q}, 0)]\, \mathbf{1}]\!] \wedge (p_1, T, nil) \in [\![L[0](\!|\tau|\!))]\!]$
     We choose p_1 and p_2 as 0 and we get the desired directly from Definition 15
   - ${}^s v = \ell = [{}^s v_1, \ldots, {}^s v_n]$:
     To prove: $\forall\, T\, .(\Phi_H([{}^s v_1 \ldots {}^s v_n] : L^{\vec{q}}\tau'), T, \overline{({}^s v)_{H,\tau}}) \in [\![\exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))]\!]$

     This means given some T it suffices to prove that
     $(\Phi_H([{}^s v_1 \ldots {}^s v_n] : L^{\vec{q}}\tau'), T, \overline{({}^s v)_{H,\tau}}) \in [\![\exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))]\!]$

     We know that $\Phi_H([{}^s v_1 \ldots {}^s v_n] : L^{\vec{q}}\tau') = (\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau'))$ therefore it suffices to prove that
     $((\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{({}^s v)_{H,\tau}}) \in [\![\exists s.([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))]\!]$
     From Definition 15 it suffices to prove that
     $\exists s'.((\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{({}^s v)_{H,\tau}}) \in [\![([\phi(\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))[s'/s]]\!]$

     We choose s' as n and it suffices to prove that
     $((\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{({}^s v)_{H,\tau}}) \in [\![([\phi(\vec{q}, n)]\, \mathbf{1} \otimes L^n(\!|\tau'|\!))]\!]$

     From Definition 45 we know that $\overline{({}^s v)_{H,\tau}} = \langle\!\langle (), \overline{(H(\ell)\downarrow_1)_{H,\tau'}} :: \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}\rangle\!\rangle$
     From Definition 15 it further suffices to prove that
     $\exists p_1, p_2 . p_1 + p_2 \leq (\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau')) \wedge (p_1, T, ()) \in [\![[\phi(\vec{q}, n)]\, \mathbf{1}]\!] \wedge (p_2, T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}} :: \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^n(\!|\tau'|\!))]\!]$ (L0)

     <u>IH</u>
     $(\Phi_H([{}^s v_2 \ldots {}^s v_n] : L^{\triangleleft\vec{q}}\tau'), T, \overline{(H(\ell)\downarrow_2)_{H,L^{\triangleleft\vec{q}}\tau'}}) \in [\![\exists s.([\phi(\triangleleft\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))]\!]$

     We know that $\Phi_H([{}^s v_2 \ldots {}^s v_n] : L^{\triangleleft\vec{q}}\tau') = (\Phi(n-1, \triangleleft\vec{q}) + \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau'))$ this means we have
     $((\Phi(n-1, \triangleleft\vec{q}) + \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{(H(\ell)\downarrow_2)_{H,L^{\triangleleft\vec{q}}\tau'}}) \in [\![\exists s.([\phi(\triangleleft\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))]\!]$
     From Definition 15 this means we have
     $\exists s'.((\Phi(n-1, \triangleleft\vec{q}) + \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{(H(\ell)\downarrow_2)_{H,L^{\triangleleft\vec{q}}\tau'}}) \in [\![([\phi(\triangleleft\vec{q}, s)]\, \mathbf{1} \otimes L^s(\!|\tau'|\!))[s'/s]]\!]$

     We know that s' as $n-1$ and we have
     $((\Phi(n-1, \triangleleft\vec{q}) + \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau')), T, \overline{(H(\ell)\downarrow_2)_{H,L^{\triangleleft\vec{q}}\tau'}}) \in [\![([\phi(\triangleleft\vec{q}, n-1)]\, \mathbf{1} \otimes L^{n-1}(\!|\tau'|\!))]\!]$

     From Definition 45 we know that $\overline{(H(\ell)\downarrow_2)_{H,L^{\triangleleft\vec{q}}\tau'}} = \langle\!\langle (), l_t\rangle\!\rangle$

     This means from Definition 15 we have
     $\exists p'_1, p'_2 . p'_1 + p'_2 \leq (\Phi(n-1, \triangleleft\vec{q}) + \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau')) \wedge (p'_1, T, ()) \in [\![[\phi(\triangleleft\vec{q}, n)]\, \mathbf{1}]\!] \wedge (p'_2, T, l_t) \in [\![L^{n-1}(\!|\tau'|\!))]\!]$ (L1)

     Inorder to prove (L0) we choose p_1 as $p'_1 + q_1$ and p_2 as $p'_2 + \Phi_H({}^s v_1 : \tau')$
     - $p_1 + p_2 \leq (\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau'))$:
       It suffices to prove that

$p_1' + q_1 + p_2' + \Phi_H({}^s v_1 : \tau') \leq (\Phi(n, \vec{q}) + \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau'))$

Since from (L1) we know that $p_1' \leq \Phi(n-1, \lhd\vec{q})$ therefore we also know that
$p_1' + q_1 \leq \Phi(n, \vec{q}) \qquad$ (L2)

Similarly since from (L1) we know that $p_2' \leq \sum_{2 \leq i \leq n} \Phi_H({}^s v_i : \tau')$
Therefore we also have
$p_2' + \Phi_H({}^s v_1 : \tau') \leq \sum_{1 \leq i \leq n} \Phi_H({}^s v_i : \tau') \qquad$ (L3)

Combining (L2) and (L3) we get the desired

– $(p_1, T, ()) \in [\![[\phi(\vec{q}, n)]\, \mathbf{1}]\!]$:
It suffices to prove that $(p_1' + q_1, T, ()) \in [\![[\phi(\vec{q}, n)]\, \mathbf{1}]\!]$

Since from (L1) we are given that
$(p_1', T, ()) \in [\![[\phi(\lhd\vec{q}, n)]\, \mathbf{1}]\!]$
Therefore we also have
$(p_1' + q_1, T, ()) \in [\![[\phi(\vec{q}, n)]\, \mathbf{1}]\!]$

– $(p_2, T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}} :: \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^n (\!|\tau'|\!)]\!]$:
It suffices to prove that
$(p_2' + \Phi_H({}^s v_1 : \tau'), T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}} :: \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^n (\!|\tau'|\!)]\!]$

From Definition 15 it suffices to show that
$\exists p_1'', p_2''. p_1'' + p_2'' \leq \Phi_H({}^s v_1 : \tau') + p_2' \wedge (p_1'', T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}}) \in [\![\tau']\!] \wedge (p_2'', T, \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^{n-1}\tau']\!]$

We choose p_1'' as $\Phi_H({}^s v_1 : \tau')$ and p_2'' as p_2' and it suffices to prove that

* $(p_1'', T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}}) \in [\![\tau']\!]$:
This means we need to prove that
$(\Phi_H({}^s v_1 : \tau'), T, \overline{(H(\ell)\downarrow_1)_{H,\tau'}}) \in [\![\tau']\!]$

We get this from IH of outer induction

* $(p_2'', T, \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^{n-1}\tau']\!]$:
This means we need to prove that
$(p_2', T, \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in [\![L^{n-1}\tau']\!]$

Since we know that $\overline{(H(\ell)\downarrow_2)_{H,L\tau'}} = l_t$ therefore we get the desired from (L1)

4. $\tau = (\tau_1, \tau_2)$:
To prove: $\forall T . (\Phi_H(({}^s v_1, {}^s v_2) : (\tau_1, \tau_2)), T, \overline{({}^s v_1, {}^s v_2)_{H,(\tau_1,\tau_2)}}) \in [\![(\!|(\tau_1, \tau_2)|\!)]\!]$

This means given some T it suffices to prove that
$(\Phi_H(({}^s v_1, {}^s v_2) : (\tau_1, \tau_2)), T, \overline{({}^s v_1, {}^s v_2)_{H,(\tau_1,\tau_2)}}) \in [\![(\!|\tau_1|\!) \otimes (\!|\tau_2|\!)]\!]$

We know that $\Phi_H(({}^s v_1, {}^s v_2) : (\tau_1, \tau_2)) = \Phi_H({}^s v_1 : \tau_1) + \Phi_H({}^s v_2 : \tau_2)$ therefore it suffices to prove that
$(\Phi_H({}^s v_1 : \tau_1) + \Phi_H({}^s v_2 : \tau_2), T, (\overline{(H(\ell)\downarrow_1)_{H,\tau_1}}, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}})) \in [\![(\!|\tau_1|\!) \otimes (\!|\tau_2|\!)]\!]$
From Definition 15 it suffices to prove that
$\exists p_1, p_2. p_1 + p_2 \leq (\Phi_H({}^s v_1 : \tau_1) + \Phi_H({}^s v_2 : \tau_2)) \wedge (p_1, T, \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}) \in [\![(\!|\tau_1|\!)]\!] \wedge (p_2, T, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}}) \in [\![(\!|\tau_2|\!)]\!]$

Choosing p_1 as $\Phi_H({}^s v_1 : \tau_1)$ and p_2 as $\Phi_H({}^s v_2 : \tau_2)$ and it suffices to prove that
$(\Phi_H({}^s v_1 : \tau_1), T, \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}) \in [\![(\!|\tau_1|\!)]\!] \wedge (\Phi_H({}^s v_2 : \tau_2), T, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}}) \in [\![(\!|\tau_2|\!)]\!]$
We get this directly from IH1 and IH2

□

Lemma 47 (Irrelevance of T for translated Γ)**.** $\forall {}^s v, \tau, H.$
$H \models V : \Gamma$ *in RAML* $\implies \forall T . (\Phi_{V,H}(\Gamma), T, \overline{(V : \Gamma)_H}) \in [\![(\!|\Gamma|\!)]\!]$ *in λ-amor*

Proof. To prove: $\forall T . (\Phi_{V,H}(\Gamma), T, \overline{(V : \Gamma)_H}) \in [\![(\!|\Gamma|\!)]\!]$

This means given soem T it suffices to prove that
$(\Phi_{V,H}(\Gamma), T, \overline{(V : \Gamma)_H}) \in [\![(\!|\Gamma|\!)]\!]$

From Definition 16 it suffices to prove that
$\exists f : \mathit{Vars} \to \mathit{Pots}. (\forall x \in \mathit{dom}((\!|\Gamma|\!)). (f(x), T, \overline{(V : \Gamma)_H}(x)) \in [\![(\!|\Gamma|\!)(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x \in \mathit{dom}((\!|\Gamma|\!))} f(x) \leq \Phi_{V,H}(\Gamma))$

We choose $f(x)$ as $\Phi_H(V(x) : \Gamma(x))$ for every $x \in \mathit{dom}(\Gamma)$ and it suffices to prove that

- $(\forall x \in \mathit{dom}((\!|\Gamma|\!)). (\Phi_H(V(x) : \Gamma(x)), T, \overline{(V : \Gamma)_H}(x)) \in [\![(\!|\Gamma|\!)(x)]\!]_{\mathcal{E}})$:
This means given some $x \in \mathit{dom}((\!|\Gamma|\!))$ it suffices to prove that
$(\Phi_H(V(x) : \Gamma(x)), T, \overline{(V : \Gamma_H)}(x)) \in [\![(\!|\Gamma(x)|\!)]\!]_{\mathcal{E}}$

From Definition 44 it suffices to prove that
$(\Phi_H(V(x) : \Gamma(x)), T, \overline{(V(x))_{H,\Gamma(x)}}) \in [\![(\!|\Gamma(x)|\!)]\!]_{\mathcal{E}}$

From Lemma 46 we know that
$(\Phi_H(V(x) : \Gamma(x)), T, \overline{(V(x))_{H,\Gamma(x)}}) \in [\![(\!|\Gamma(x)|\!)]\!]$

And finally from Definition 15 we have
$(\Phi_H(V(x) : \Gamma(x)), T, \overline{(V(x))_{H,\Gamma(x)}}) \in [\![(\!|\Gamma(x)|\!)]\!]_{\mathcal{E}}$

- $(\sum_{x \in dom((\!|\Gamma|\!))} f(x) \leq \Phi_{V,H}(\Gamma))$:
  Since we know that $\Phi_{V,H}(\Gamma) = \sum_{x \in dom(\Gamma)} \Phi_H(V(x) : \Gamma(x))$ therefore we are done

□

Lemma 48 (RAML's stack and its translation are in the cross-lang relation)**.** $\forall H, V, \Gamma.$
$H \models V : \Gamma \implies \forall\, T\,.(T, V, \overline{(V : \Gamma)_H}) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^{H}$

Proof. Given some T, it suffices to prove that $(T, V, \overline{(V : \Gamma)_H}) \in \lfloor \Gamma \rfloor_{\mathcal{V}}^{H}$

From Definition 34 it suffices to prove that
$\forall x : \tau \in dom(\Gamma).(T, V(x), \overline{(V : \Gamma)_H}(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$

This means given some $x : \tau \in dom(\Gamma)$ and we need to prove that
$(T, V(x), \overline{(V : \Gamma)_H}(x)) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$

Since we are given that $H \models V : \Gamma$, it means we have $\forall x \in dom(\Gamma).H \models V(x) \in [\![\Gamma(x)]\!]$

Therefore we get the desired from Lemma 49

□

Lemma 49 (RAML's value and its translation are in the cross-lang relation)**.** $\forall H, {}^s v, \tau.$
$H \models {}^s v \in [\![\tau]\!] \implies \forall\, T\,.(T, {}^s v, \overline{({}^s v)_{H,\tau}}) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H}$

Proof. By induction on τ

1. $\tau = unit$:
   To prove: $\forall\, T\,.(T, {}^s v, \overline{({}^s v)_{H,\tau}}) \in \lfloor unit \rfloor_{\mathcal{V}}^{H}$
   This means given some T, from Definition 45 it suffices to prove that
   $(T, {}^s v, {}^s v) \in \lfloor unit \rfloor_{\mathcal{V}}^{H}$

   We get this directly from Definition 33
2. $\tau = \mathsf{b}$:
   To prove: $\forall\, T\,.(T, {}^s v, \overline{({}^s v)_{H,\tau}}) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
   This means given some T, from Definition 45 it suffices to prove that
   $(T, {}^s v, !{}^s v) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$

   We get this directly from Definition 33
3. $\tau = L^{\vec{q}}\tau'$:
   By induction on ${}^s v$
   - ${}^s v = NULL$:
     To prove: $\forall\, T\,.(T, NULL, \overline{({}^s v)_{H,\tau}}) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
     Given some T, from Definition 45 it suffices to prove that
     $(T, NULL, \langle\!\langle (), nil \rangle\!\rangle) \in \lfloor L^{\vec{q}}\tau' \rfloor_{\mathcal{V}}^{H}$
     We get this directly from Definition 33
   - ${}^s v = \ell = [{}^s v_1 \ldots {}^s v_n]$:
     To prove: $\forall\, T\,.(T, \ell, \overline{({}^s v)_{H,\tau}}) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
     Given some T, from Definition 45 it suffices to prove that
     $(T, \ell, \langle\!\langle (), \overline{(H(\ell)\downarrow_1)_{H,\tau'}} :: \overline{(H(\ell)\downarrow_2)_{H,L\tau'}} \rangle\!\rangle) \in \lfloor L^{\vec{q}}\tau' \rfloor_{\mathcal{V}}^{H}$
     From Definition 33 it further suffices to prove that
     $(T, H(\ell)\downarrow_1, \overline{(H(\ell)\downarrow_1)_{H,\tau'}}) \in \lfloor \tau' \rfloor_{\mathcal{V}} \wedge (T, H(\ell)\downarrow_2, \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$

     We get $(T, H(\ell)\downarrow_1, \overline{(H(\ell)\downarrow_1)_{H,\tau'}}) \in \lfloor \tau' \rfloor_{\mathcal{V}}$ from IH of outer induction
     and $(T, H(\ell)\downarrow_2, \overline{(H(\ell)\downarrow_2)_{H,L\tau'}}) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$ from IH of inner induction
4. $\tau = (\tau_1, \tau_2)$:
   To prove: $\forall\, T\,.(T, \ell, \overline{(\ell)_{H,(\tau_1,\tau_2)}}) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$
   Given some T, from Definition 45 it suffices to prove that
   $(T, \ell, \langle\!\langle \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}} \rangle\!\rangle) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$

   From Definition 33 it suffices to prove that
   $(T, H(\ell)\downarrow_1, \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, H(\ell)\downarrow_2, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}}) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$

   We get this directly from IH

$\square$

Lemma 50. $\forall {}^s v, {}^t v, \tau, H, T.$
$(T, {}^s v, {}^t v) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H} \implies {}^t v = \overline{({}^s v)_{H,\tau}}$

Proof. Proof by induction on the $\lfloor . \rfloor_{\mathcal{V}}$ relation

1. $\lfloor unit \rfloor_{\mathcal{V}}^{H}$:
   Given: $(T, {}^s v, {}^s v) \in \lfloor unit \rfloor_{\mathcal{V}}^{H}$
   To prove: ${}^s v = \overline{({}^s v)_{H,unit}}$
   Directly from Definition 45
2. $\lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$:
   Given: $(T, {}^s v, !{}^s v) \in \lfloor \mathsf{b} \rfloor_{\mathcal{V}}^{H}$
   To prove: $!{}^s v = \overline{({}^s v)_{H,\tau}}$
   Directly from Definition 45
3. $\lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$:
   Given: $(T, \ell, \langle\langle {}^t v_1, {}^t v_2 \rangle\rangle) \in \lfloor (\tau_1, \tau_2) \rfloor_{\mathcal{V}}^{H}$
   This means from Definition 33 we have
   $H(\ell) = ({}^s v_1, {}^s v_2) \wedge (T, {}^s v_1, {}^t v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}} \wedge (T, {}^s v_2, {}^t v_2) \in \lfloor \tau_2 \rfloor_{\mathcal{V}}$ (R0)

   To prove: $\langle\langle {}^t v_1, {}^t v_2 \rangle\rangle = \overline{(\ell)_{H,(\tau_1,\tau_2)}}$
   From Definition 45 we know that
   $\overline{(\ell)_{H,(\tau_1,\tau_2)}} = \langle\langle \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}} \rangle\rangle$
   From (R0) we know that $H(\ell)\downarrow_1 = {}^s v_1$ and $H(\ell)\downarrow_2 = {}^s v_2$ therefore we have
   $\overline{(\ell)_{H,(\tau_1,\tau_2)}} = \langle\langle \overline{(H(\ell)\downarrow_1)_{H,\tau_1}}, \overline{(H(\ell)\downarrow_2)_{H,\tau_2}} \rangle\rangle = \langle\langle \overline{{}^s v_1}, \overline{{}^s v_2} \rangle\rangle$ (R1)

   Since from (R0) we know that $(T, {}^s v_1, {}^t v_1) \in \lfloor \tau_1 \rfloor_{\mathcal{V}}$ therefore we have
   ${}^t v_1 = \overline{{}^s v_1}$ (IH1)

   Similarly we also have
   ${}^t v_2 = \overline{{}^s v_2}$ (IH2)

   We get the desired from IH1, IH2 and (R1)
4. $\lfloor L^{\vec{q}} \tau' \rfloor_{\mathcal{V}}^{H}$:
   Given: $(T, \ell_s, \langle\langle (), l_t \rangle\rangle) \in \lfloor L^{\vec{q}} \tau' \rfloor_{\mathcal{V}}^{H}$ where $(T, \ell_s, l_t) \in \lfloor L \tau' \rfloor_{\mathcal{V}}^{H}$
   To prove: $\langle\langle (), l_t \rangle\rangle = \overline{(\ell_s)_{H,\tau}}$

   From Definition 45 we know that
   $\overline{(\ell_s)_{H,L^{-}\tau'}} = \langle\langle (), \overline{(\ell_s)_{H,L\tau'}} \rangle\rangle$

   Therefore it suffices to prove that $l_t = \overline{(\ell_s)_{H,L\tau'}}$
   We induct on $(T, \ell_s, l_t) \in \lfloor L \tau' \rfloor_{\mathcal{V}}^{H}$
   (a) $\ell_s = NULL$:
   In this case we know that $l_t = nil$
   From Definition 45 we get the desired
   (b) $\ell_s = \ell \neq NULL$:
   In this case we know that $l_t = {}^t v_h :: l'_t$ s.t
   $H(\ell) = ({}^s v', \ell'_s) \wedge (T, {}^s v', {}^t v_h) \in \lfloor \tau' \rfloor_{\mathcal{V}} \wedge (T, \ell'_s, l'_t) \in \lfloor L\ \tau' \rfloor_{\mathcal{V}}$
   We get the desired from Definition 45, IH of outer induction and IH of inner induction

$\square$

Definition 51 (Top level RAML program translation)**.** *Given a top-level RAML program*
$P \triangleq F, e_{main}$ *where* $F \triangleq f1(x) = e_{f1}, \ldots, fn(x) = e_{fn}$ *s.t*
$\Sigma, x : \tau_{f1} \vdash_{q'_1}^{q_1} e_{f1} : \tau'_{f1}$
$\ldots$
$\Sigma, x : \tau_{fn} \vdash_{q'_n}^{q_n} e_{fn} : \tau'_{fn}$
$\Sigma, \Gamma \vdash_{q'}^{q} e_{main} : \tau$
where $\Sigma = f1 : \tau_{f1} \overset{q_1/q'_1}{\to} \tau'_{f1}, \ldots, fn : \tau_{fn} \overset{q_n/q'_n}{\to} \tau'_{fn}$

Translation of P *denoted by* $\overline{P}$ *is defined as* $\overline{F}, e_t$ *where*
$\overline{F} = \mathsf{fix} f_1.\lambda u.\lambda x.e_{t1}, \ldots, \mathsf{fix} f_n.\lambda u.\lambda x.e_{tn}$ *s.t*
$\Sigma, x : \tau_{f1} \vdash_{q'_1}^{q_1} e_{f1} : \tau'_{f1} \rightsquigarrow e_{t1}$
$\ldots$
$\Sigma, x : \tau_{fn} \vdash_{q'_n}^{q_n} e_{fn} : \tau'_{fn} \rightsquigarrow e_{tn}$
and
$\Sigma, \Gamma \vdash_{q'}^{q} e_{main} : \tau \rightsquigarrow e_t$

Theorem 52 (RAML univariate soundness). $\forall H, H', V, \Gamma, \Sigma, e, \tau, {}^s v, p, p', q, q', t.$
$P = F, e$ and $\overline{P}$ be a RAML top-level program and its translation respectively (as defined in Definition 51)
$H \models V : \Gamma \wedge \Sigma, \Gamma \vdash^q_{q'} e : \tau \wedge V, H \vdash^p_{p'} e \Downarrow_t {}^s v, H'$
$\implies$
$p - p' \leq (\Phi_{H,V}(\Gamma) + q) - (q' + \Phi_H({}^s v : \tau))$

Proof. From Definition 51 we are given that
$F \triangleq f_1(x) = e_{f_1}, \ldots, f_n(x) = e_{f_n}$ s.t
$\Sigma, x : \tau_{f_1} \vdash^{q_1}_{q'_1} e_{f_1} : \tau'_{f_1} \rightsquigarrow e_{t_1}$
$\ldots$
$\Sigma, x : \tau_{f_n} \vdash^{q_n}_{q'_n} e_{f_n} : \tau'_{f_n} \rightsquigarrow e_{t_n}$

Let $\forall i \in [1 \ldots n].\delta_{sf}(f_i) = (f_i(x) = e_{f_i})$ and $\forall i \in [1 \ldots n].\delta_{tf}(f_i) = (\mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i})$

Claim: $\forall\, T\, .(T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor^H$
Proof.
This means given some T, it suffices to prove that
$(T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor^H$

We induct on T
Base case: Trivial
Inductive case:
IH: $\forall\, T''{<}T\, .(T'', \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor^H$

From Definition 35 it suffices to prove that
$\forall f_i \in dom(\Sigma).(T, f_i(x) = e_{f_i}\ \delta_{sf}, \mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i}\ \delta_{tf}) \in \lfloor \tau_{f_i} \overset{q_i/q'_i}{\to} \tau'_{f_i} \rfloor^H$
Given some $f_i \in dom(\Sigma)$ it suffices to prove that
$(T, f_i(x) = e_{f_i}\ \delta_{sf}, \mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i}\ \delta_{tf}) \in \lfloor \tau_{f_i} \overset{q_i/q'_i}{\to} \tau'_{f_i} \rfloor^H$
From Definition 33 it suffices to prove that
$\forall^s v', {}^t v', T'{<}T\, .(T', {}^s v', {}^t v') \in \lfloor \tau_{f_i} \rfloor^H_{\mathcal{V}} \implies (T', e_{f_i}\ \delta_{sf}, e_{t_i}\ \delta_{tf}[()/u][{}^t v'/x][\mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i}\ \delta_{tf}/f_i]) \in \lfloor \tau'_{f_1} \rfloor^{\{x \mapsto {}^s v'\},H}_{\mathcal{E}}$
This means given some ${}^s v', {}^t v', T'{<}T$ s.t $(T', {}^s v', {}^t v') \in \lfloor \tau_{f_i} \rfloor^H_{\mathcal{V}}$ it suffices to prove that
$(T', e_{f_i}\ \delta_{sf}, e_{t_i}\ \delta_{tf}[()/u][{}^t v'/x][\mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i}\ \delta_{tf}/f_i]) \in \lfloor \tau'_{f_i} \rfloor^{\{x \mapsto {}^s v'\},H}_{\mathcal{E}}$
Since $\delta_{tf} = \delta_{tf} \cup \{f_i \mapsto \mathsf{fix} f_i.\lambda u.\lambda x.e_{t_i}\ \delta_{tf}\}$, therefore it suffices to prove that
$(T', e_{f_i}\ \delta_{sf}, e_{t_i}\ \delta_{tf}[()/u][{}^t v'/x]) \in \lfloor \tau'_{f_i} \rfloor^{\{x \mapsto {}^s v'\},H}_{\mathcal{E}}$ (C0)
Also since are given $(T', {}^s v', {}^t v') \in \lfloor \tau_{f_i} \rfloor^H_{\mathcal{V}}$ therefore we have
$(T', \{x \mapsto {}^s v'\}, \{x \mapsto {}^t v'\}) \in \lfloor x : \tau_{f_i} \rfloor^H_{\mathcal{V}}$

Also from IH we have $(T', \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor^{V,H}$
We can apply Theorem 41 to get
$(T', e_{f_i}\delta_{sf}, e_{t_i}\ ()\ \{x \mapsto {}^t v'\}\delta_{tf}) \in \lfloor \tau'_{f_i} \rfloor^{\{x \mapsto {}^s v'\},H}_{\mathcal{E}}$
And this prove (C0)

□

From Theorem 31 we know that $\exists e_t$ s.t
$\Sigma, \Gamma \vdash^q_{q'} e : \tau \rightsquigarrow e_t$ and $.; .; (\!|\Sigma|\!); (\!|\Gamma|\!) \vdash e_t : [q]\,\mathbf{1} \multimap \mathbb{M}\, 0\, [q']\, (\!|\tau|\!)$

From Lemma 48 we know that $\forall\, T\, .(T, V, \overline{(V : \Gamma)_H}) \in \lfloor \Gamma \rfloor^H_{\mathcal{V}}$

Also from the Claim proved above we know that $\forall\, T\, .(T, \delta_{sf}, \delta_{tf}) \in \lfloor \Sigma \rfloor^H$
Therefore from Theorem 41 we know that $\forall\, T\, .(T, e\delta_{sf}, e_t\ ()\ \overline{(V : \Gamma)_H}\delta_{tf}) \in \lfloor \tau \rfloor^{V,H}_{\mathcal{E}}$

This means from Definition 33 we have
$\forall\, T\, .\forall H'_1, {}^s v_1, p_1, p'_1, t' < T\, .\ V, H \vdash^{p_1}_{p'_1} e\delta_{sf} \Downarrow_{t'} {}^s v_1, H'_1 \implies \exists^t v_t, {}^t v_f, J.e_t\ ()\ \overline{(V : \Gamma)_H}\delta_{tf} \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge (T - t', {}^s v, {}^t v_f) \in \lfloor \tau \rfloor^{H'_1}_{\mathcal{V}} \wedge p_1 - p'_1 \leq J$ (RD-0.0)

We are given that $V, H \vdash^p_{p'} e \Downarrow_t {}^s v, H'$
Therefore instantiating (RD-0.0) with $t + 1, H', {}^s v, p, p', t$ we get
$\exists^t v_t, {}^t v_f, J.e_t\ ()\ \overline{(V : \Gamma)_H}\delta_{tf} \Downarrow_- {}^t v_t \Downarrow^J_- {}^t v_f \wedge (1, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor^{H'}_{\mathcal{V}} \wedge p - p' \leq J$ (RD-0)

From reduction rules we know that $\exists t_1, t_2$ s.t $e_t\ ()\ \overline{(V : \Gamma)_H}\delta_{tf} \Downarrow_{t_1} {}^t v_t \Downarrow^J_{t_2} {}^t v_f$

Since from Lemma 47 we know that $\forall\, T\, .(\Phi_{V,H}(\Gamma), T, \overline{(V:\Gamma)_H}) \in [\![(\!|\Gamma|\!)]\!]$
Therefore we also have $(\Phi_{V,H}(\Gamma), t_1 + t_2 + 1, \overline{(V:\Gamma)_H}) \in [\![(\!|\Gamma|\!)]\!]$

Therefore from Theorem 29 we get
$\exists p_v.(p_v, 1, {}^t v_f) \in [\![(\!|\tau|\!)]\!] \wedge J \leq (q + \Phi_{V,H}(\Gamma)) - (q' + p_v) \qquad$ (RD-1)

Since we have $(1, {}^s v, {}^t v_f) \in \lfloor \tau \rfloor_{\mathcal{V}}^{H'}$ therefore from Lemma 50 we know that ${}^t v_f = \overline{({}^s v)_{H',\tau}}$

From Lemma 46 we know that $\forall\, T\, .(\Phi_H({}^s v : \tau), T, \overline{({}^s v)_{H',\tau}}) \in [\![(\!|\tau|\!)]\!]$

Therefore we have $(\Phi_H({}^s v : \tau), 1, \overline{({}^s v)_{H',\tau}}) \in [\![(\!|\tau|\!)]\!] \qquad$ (RD-2)

From (RD-1), (RD-2) and Lemma 61 we know that $p_v \geq \Phi_H({}^s v : \tau)$

Since from (RD-1) we know that $J \leq (q + \Phi_{V,H}(\Gamma)) - (q' + p_v)$ therefore we also have
$J \leq (q + \Phi_{V,H}(\Gamma)) - (q' + \Phi_H({}^s v : \tau)) \qquad$ (RD-3)

Finally from (RD-0) and (RD-3) we get the desired.

$\square$

B Development of λ-amor (full)

B.1 Syntax

Expressions	e	$::=$	$v \mid e_1\, e_2 \mid \langle\!\langle e_1, e_2 \rangle\!\rangle \mid \mathsf{let}\langle\!\langle x, y\rangle\!\rangle = e_1 \;\mathsf{in}\; e_2 \mid$ $\langle e, e\rangle \mid \mathsf{fst}(e) \mid \mathsf{snd}(e) \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid \mathsf{case}\; e, x.e, y.e \mid$ $\mathsf{let}\,!\, x = e_1 \;\mathsf{in}\; e_2 \mid e\, [] \mid e :: e \mid e; x.e$
Values	v	$::=$	$x \mid c \mid \lambda x.e \mid \langle\!\langle v_1, v_2\rangle\!\rangle \mid \langle v, v\rangle \mid \mathsf{inl}(e) \mid \mathsf{inr}(e) \mid nil \mid$ $!e \mid \Lambda.e \mid \mathsf{ret}\, e \mid \mathsf{bind}\, x = e_1 \;\mathsf{in}\; e_2 \mid \uparrow^I \mid \mathsf{release}\, x = e_1 \;\mathsf{in}\; e_2 \mid \mathsf{store}\, e$ (No value forms for $[I]\,\tau$)
Index	I	$::=$	$N \mid i \mid I + I \mid I - I \mid \sum_{a<I} I \mid \bigcirc\!\!\!\!\!\bigtriangleup_a^{I,I} I \mid \lambda_s i.I \mid I\; I$
Sort	S	$::=$	$\mathbb{N} \mid \mathbb{R}^+ \mid \mathsf{S} \to \mathsf{S}$
Kind	K	$::=$	$\mathit{Type} \mid \mathsf{S} \to \mathsf{K}$
Types	τ	$::=$	$\mathbf{1} \mid \mathsf{b} \mid \tau_1 \multimap \tau_2 \mid \tau_1 \otimes \tau_2 \mid \tau_1 \,\&\, \tau_2 \mid \tau_1 \oplus \tau_2 \mid !_{a<I}\tau \mid [I]\,\tau \mid \mathbb{M}\, I\, \tau \mid$ $\alpha \mid \forall \alpha : \mathsf{K}.\tau \mid \forall i : \mathsf{S}.\tau \mid \lambda_t i.\tau \mid \tau\; I \mid L^I \tau \mid \exists i : \mathsf{S}.\tau \mid c \Rightarrow \tau \mid c \& \tau$
Constraints	c	$::=$	$I = I \mid I < I \mid c \wedge c$
Lin. context for term variables	Γ	$::=$	$. \mid \Gamma, x : \tau$
Bounded Lin. context for term variables	Ω	$::=$	$. \mid \Omega, x :_{a<I} \tau$
Unres. context for sort variables	Θ	$::=$	$. \mid \Theta, i : \mathsf{S}$
Unres. context for type variables	Ψ	$::=$	$. \mid \Psi, \alpha : \mathsf{K}$

Definition 53 (Bounded sum of context for dlPCF)**.** $\sum_{a<I} . = .$
$\sum_{a<I} \Gamma, x : [b < J]\tau = (\sum_{a<I} \Gamma), x : [c < \sum_{a<I} J]\sigma$
where
$\tau = \sigma[(\sum_{d<a} J[d/a] + b)/c]$

Definition 54 (Bounded sum of multiplicity context)**.** $\sum_{a<I} . = .$
$\sum_{a<I} \Omega, x :_{b<J} \tau = (\sum_{a<I} \Omega), x :_{c<\sum_{a<I} J} \sigma$
where
$\tau = \sigma[(\sum_{d<a} J[d/a] + b)/c]$

Definition 55 (Binary sum of context for dlPCF)**.**

$$\Gamma_1 \oplus \Gamma_2 \triangleq \begin{cases} \Gamma_2 & \Gamma_1 = . \\ (\Gamma_1' \oplus \Gamma_2/x), x : [c < I + J]\tau & \Gamma_1 = \Gamma_1', x : [a < I]\tau[a/c] \wedge (x : [b < J]\tau[I + b/c]) \in \Gamma_2 \\ (\Gamma_1' \oplus \Gamma_2), x :_{a<I} \tau & \Gamma_1 = \Gamma_1', x : [a < I]\tau \wedge (x : [-]-) \notin \Gamma_2 \end{cases}$$

Definition 56 (Binary sum of multiplicity context).

$$\Omega_1 \oplus \Omega_2 \triangleq \begin{cases} \Omega_2 & \Omega_1 = . \\ (\Omega_1' \oplus \Omega_2/x), x :_{c<I+J} \tau & \Omega_1 = \Omega_1', x :_{a<I} \tau[a/c] \wedge (x :_{b<J} \tau[I+b/c]) \in \Omega_2 \\ (\Omega_1' \oplus \Omega_2), x :_{a<I} \tau & \Omega_1 = \Omega_1', x :_{a<I} \tau \wedge (x :_{-} -) \notin \Omega_2 \end{cases}$$

Definition 57 (Binary sum of affine context).

$$\Gamma_1 \oplus \Gamma_2 \triangleq \begin{cases} \Gamma_2 & \Gamma_1 = . \\ (\Gamma_1' \oplus \Gamma_2), x : \tau & \Gamma_1 = \Gamma_1', x : \tau \wedge (x : -) \notin \Gamma_2 \end{cases}$$

B.2 Typesystem

$\boxed{\text{Typing } \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}$

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau \vdash x : \tau}\text{ T-var1} \qquad \frac{\Theta, \Delta \models I \geq 1}{\Psi; \Theta; \Delta; \Omega, x :_{a<I} \tau; \Gamma \vdash x : \tau[0/a]}\text{ T-var2} \qquad \frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash () : \mathbf{1}}\text{ T-unit}$$

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash c : \mathsf{b}}\text{ T-base} \qquad \frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash nil : L^0\ \tau}\text{ T-nil}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : L^n\tau \qquad \Theta; \Delta \vdash n : \mathbb{N}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e_1 :: e_2 : L^{n+1}\tau}\text{ T-cons}$$

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : L^n\ \tau \qquad \Psi; \Theta; \Delta, n = 0; \Omega_2; \Gamma_2 \vdash e_1 : \tau' \\ \Psi; \Theta, I; \Delta, n = I + 1; \Omega_2; \Gamma_2, h : \tau, t : L^I\tau \vdash e_2 : \tau' \qquad \Theta; \Delta \vdash n : \mathbb{N} \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}\end{array}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau'}\text{ T-match}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau[n/s] \qquad \Theta; \Delta \vdash n : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \exists s : \mathsf{S}.\tau}\text{ T-existI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : \exists s : \mathsf{S}.\tau \qquad \Psi; \Theta, s : \mathsf{S}; \Delta; \Omega; \Gamma_2, x : \tau \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash e; x.e' : \tau'}\text{ T-existE}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \lambda x.e : (\tau_1 \multimap \tau_2)}\text{ T-lam} \qquad \frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : (\tau_1 \multimap \tau_2) \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e_1\ e_2 : \tau_2}\text{ T-app}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau'}\text{ T-sub}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \langle\langle e_1, e_2\rangle\rangle : (\tau_1 \otimes \tau_2)}\text{ T-tensorI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (\tau_1 \otimes \tau_2) \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1, y : \tau_2 \vdash e' : \tau}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\langle\langle x, y\rangle\rangle = e\ \mathsf{in}\ e' : \tau}\text{ T-tensorE}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_2 : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \langle e_1, e_2\rangle : (\tau_1\ \&\ \tau_2)}\text{ T-withI} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1\ \&\ \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{fst}(e) : \tau_1}\text{ T-fst}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1\ \&\ \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{snd}(e) : \tau_2}\text{ T-snd}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inl}(e) : \tau_1 \oplus \tau_2} \text{ T-inl} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inr}(e) : \tau_1 \oplus \tau_2} \text{ T-inr}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (\tau_1 \oplus \tau_2) \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, y : \tau_2 \vdash e_2 : \tau}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{case}\ e, x.e_1, y.e_2 : \tau} \text{ T-case}$$

$$\frac{\Psi; \Theta, a; \Delta, a < I; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \sum_{a<I} \Omega; . \vdash !e : !_{a<I}\tau} \text{ T-subExpI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (!_{a<I}\tau) \qquad \Psi; \Theta; \Delta; \Omega_2, x :_{a<I} \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\,!\,x = e\ \mathsf{in}\ e' : \tau'} \text{ T-subExpE}$$

$$\frac{\Psi, \alpha : \mathsf{K}; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall \alpha : \mathsf{K}.\tau)} \text{ T-tabs} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall \alpha : \mathsf{K}.\tau) \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [] : (\tau[\tau'/\alpha])} \text{ T-tapp}$$

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall i : \mathsf{S}.\tau)} \text{ T-iabs} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall i : \mathsf{S}.\tau) \qquad \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [] : (\tau[I\ /i])} \text{ T-iapp}$$

$$\frac{\Psi; \Theta, b; \Delta, b < L; \Omega, x :_{a<I} \tau[(b + 1 + \bigotimes_b^{b+1,a} I)/b]; . \vdash e : \tau \qquad L \geq \bigotimes_b^{0,1} I}{\Psi; \Theta; \Delta; \sum_{b<L} \Omega; . \vdash \mathsf{fix} x.e : \tau[0/b]} \text{ T-fix}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi; \Theta; \Delta \models \Gamma' \sqsubseteq \Gamma \qquad \Psi; \Theta; \Delta \models \Omega' \sqsubseteq \Omega}{\Psi; \Theta; \Delta; \Omega'; \Gamma' \vdash e : \tau} \text{ T-weaken}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{ret}\, e : \mathbb{M}\, 0\, \tau} \text{ T-ret}$$

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \mathbb{M}\ I_1\, \tau_1 \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}\ I_2\, \tau_2 \qquad \Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2 : \mathbb{M}(I_1 + I_2)\, \tau_2} \text{ T-bind}$$

$$\frac{\Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \uparrow^I : \mathbb{M}\ I\, \mathbf{1}} \text{ T-tick}$$

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : [I_1]\, \tau_1 \\ \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}(I_1 + I_2)\, \tau_2 \qquad \Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+\end{array}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{release}\, x = e_1\ \mathsf{in}\ \ e_2 : \mathbb{M}\ I_2\, \tau_2} \text{ T-release}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{store}\, e : \mathbb{M}\ I\, ([I]\, \tau)} \text{ T-store} \qquad \frac{\Psi; \Theta; \Delta, c; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.\ e : (c \Rightarrow \tau)} \text{ T-CI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \Rightarrow \tau) \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e[] : \tau} \text{ T-CE} \qquad \frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \& \tau)} \text{ T-CAndI}$$

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \& \tau) \qquad \Psi; \Theta; \Delta, c; \Omega; \Gamma, x : \tau \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{clet}\, x = e\ \mathsf{in}\ e' : \tau'} \text{ T-CAndE}$$

Figure 27: Typing rules for λ-amor

$$\frac{}{\Psi; \Theta; \Delta \vdash \tau <: \tau}\ \text{sub-refl} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1' <: \tau_1 \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \multimap \tau_2 <: \tau_1' \multimap \tau_2'}\ \text{sub-arrow}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \otimes \tau_2 <: \tau_1' \otimes \tau_2'}\ \text{sub-tensor} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \,\&\, \tau_2 <: \tau_1' \,\&\, \tau_2'}\ \text{sub-with}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \oplus \tau_2 <: \tau_1' \oplus \tau_2'}\ \text{sub-sum} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Theta; \Delta \models n' \leq n}{\Psi; \Theta; \Delta \vdash [n]\,\tau <: [n']\,\tau'}\ \text{sub-potential}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Theta; \Delta \models n \leq n'}{\Psi; \Theta; \Delta \vdash \mathbb{M}\,n\,\tau <: \mathbb{M}\,n'\,\tau'}\ \text{sub-monad} \qquad \frac{\Psi; \Theta, a; \Delta, a < J \vdash \tau <: \tau' \qquad \Theta, a; \Delta \models J \leq I}{\Psi; \Theta; \Delta \vdash !_{a<I}\tau <: !_{a<J}\tau'}\ \text{sub-subExp}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash L^n\ \tau <: L^n\ \tau'}\ \text{sub-list} \qquad \frac{\Psi; \Theta, s; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash \exists s.\tau <: \exists s.\tau'}\ \text{sub-exist}$$

$$\frac{\Psi, \alpha : \mathsf{K}; \Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall \alpha : \mathsf{K}.\tau_1 <: \forall \alpha.\tau_2}\ \text{sub-typePoly} \qquad \frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall i : \mathsf{S}.\tau_1 <: \forall i.\tau_2}\ \text{sub-indexPoly}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_2 \implies c_1}{\Psi; \Theta; \Delta \vdash c_1 \Rightarrow \tau_1 <: c_2 \Rightarrow \tau_2}\ \text{sub-constraint} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_1 \implies c_2}{\Psi; \Theta; \Delta \vdash c_1 \& \tau_1 <: c_2 \& \tau_2}\ \text{sub-CAnd}$$

$$\frac{\Theta; \Delta \vdash k : \mathbb{R}^+ \qquad \Theta; \Delta \vdash k' : \mathbb{R}^+}{\Psi; \Theta; \Delta \vdash [k](\tau_1 \multimap \tau_2) <: ([k']\,\tau_1 \multimap [k'+k]\,\tau_2)}\ \text{sub-potArrow} \qquad \frac{}{\Psi; \Theta; \Delta \vdash \tau <: [0]\,\tau}\ \text{sub-potZero}$$

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash \lambda_s i : \mathsf{S}.\tau <: \lambda_t i : \mathsf{S}.\tau'}\ \text{sub-familyAbs} \qquad \frac{\Theta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta \vdash (\lambda_t i : \mathsf{S}.\tau)\ I <: \tau[I/i]}\ \text{sub-familyApp1}$$

$$\frac{\Theta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta \vdash \tau[I/i] <: (\lambda_t i : \mathsf{S}.\tau)\ I}\ \text{sub-familyApp2} \qquad \frac{}{\Psi; \Theta; \Delta \vdash [\sum_{a<I} K]\, !_{a<I}\tau <: !_{a<I}\,[K]\,\tau}\ \text{sub-bSum}$$

Figure 28: Subtyping

$$\frac{}{\Psi; \Theta; \Delta \vdash \Gamma \sqsubseteq .}\ \text{dlpcf-subBase}$$

$$\frac{x : [a < J]\tau' \in \Gamma_1 \qquad \Psi; \Theta, a; \Delta, a < I \vdash \tau' \sqsubseteq \tau \qquad \Psi; \Theta; \Delta \vdash I \leq J \qquad \Psi; \Theta; \Delta \vdash \Gamma_1/x \sqsubseteq \Gamma_2}{\Theta; \Delta \vdash \Gamma_1 \sqsubseteq \Gamma_2, x : [a < I]\tau}\ \text{dlpcf-subInd}$$

Figure 29: Γ Subtyping for dlPCF

$$\frac{}{\Psi; \Theta; \Delta \vdash \Omega \sqsubseteq .}\ \text{sub-mBase}$$

$$\frac{x :_{a<J} \tau' \in \Omega_1 \qquad \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau \qquad \Theta; \Delta \vdash I \leq J \qquad \Psi; \Theta; \Delta \vdash \Omega_1/x \sqsubseteq \Omega_2}{\Psi; \Theta; \Delta \vdash \Omega_1 \sqsubseteq \Omega_2, x :_{a<I} \tau}\ \text{sub-mInd}$$

Figure 30: Ω Subtyping

$$\frac{}{\Psi; \Theta; \Delta \vdash \Gamma \sqsubseteq .}\ \text{sub-lBase} \qquad \frac{x : \tau' \in \Gamma_1 \qquad \Psi; \Theta; \Delta \vdash \tau' <: \tau \qquad \Psi; \Theta; \Delta \vdash \Gamma_1/x \sqsubseteq \Gamma_2}{\Psi; \Theta; \Delta \vdash \Gamma_1 \sqsubseteq \Gamma_2, x : \tau}\ \text{sub-lBase}$$

Figure 31: Γ Subtyping

$$\frac{}{\Theta, i : \mathsf{S}; \Delta \vdash i : \mathsf{S}}\ \text{S-var} \qquad \frac{}{\Theta; \Delta \vdash N : \mathbb{N}}\ \text{S-nat} \qquad \frac{}{\Theta; \Delta \vdash R : \mathbb{R}^+}\ \text{S-real} \qquad \frac{\Theta; \Delta \vdash i : \mathbb{N}}{\Theta \vdash i : \mathbb{R}^+}\ \text{S-real1}$$

$$\frac{\Theta; \Delta \vdash I_1 : \mathbb{N} \qquad \Theta; \Delta \vdash I_2 : \mathbb{N}}{\Theta; \Delta \vdash I_1 + I_2 : \mathbb{N}}\ \text{S-add-Nat} \qquad \frac{\Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+}{\Theta; \Delta \vdash I_1 + I_2 : \mathbb{R}^+}\ \text{S-add-Real}$$

$$\frac{\Theta; \Delta \vdash I_1 : \mathbb{R}^+ \qquad \Theta; \Delta \vdash I_2 : \mathbb{R}^+ \qquad \Theta; \Delta \models I_1 \geq I_2}{\Theta; \Delta \vdash I_1 - I_2 : \mathbb{R}^+}\ \text{S-minus-Real} \qquad \frac{\Theta; \Delta \vdash I_1 : \mathbb{N} \qquad \Theta; \Delta \vdash I_2 : \mathbb{N}}{\Theta; \Delta \vdash \sum_{a<I_1} I_2 : \mathbb{N}}\ \text{S-bSum}$$

$$\frac{\Theta; \Delta \vdash I_1 : \mathbb{N} \qquad \Theta; \Delta \vdash I_2 : \mathbb{N} \qquad \Theta; \Delta \vdash I_3 : \mathbb{N}}{\Theta \vdash \bigtriangleup_a^{I_1, I_2} I_3 : \mathbb{N}}\ \text{S-forest} \qquad \frac{\Theta, i : \mathsf{S}; \Delta \vdash I : \mathsf{S}'}{\Theta; \Delta \vdash \lambda_s i.\, I : \mathsf{S} \to \mathsf{S}'}\ \text{S-family}$$

Figure 32: Typing rules for sorts

$$\frac{}{\Psi; \Theta; \Delta \vdash \mathbf{1} : \mathit{Type}}\ \text{K-unit} \qquad \frac{}{\Psi; \Theta; \Delta \vdash \mathsf{b} : \mathit{Type}}\ \text{K-base} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \multimap \tau_2 : \mathsf{K}}\ \text{K-arrow}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \otimes \tau_2 : \mathsf{K}}\ \text{K-tensor} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \,\&\, \tau_2 : \mathsf{K}}\ \text{K-with}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 : \mathsf{K} \qquad \Psi; \Theta; \Delta \vdash \tau_2 : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \tau_1 \oplus \tau_2 : \mathsf{K}}\ \text{K-or} \qquad \frac{\Psi; \Theta, a : \mathsf{S}; \Delta, a < I \vdash \tau : \mathsf{K} \qquad \Theta \vdash I : \mathbb{N}}{\Psi; \Theta; \Delta \vdash !_{a<I} \tau : \mathsf{K}}\ \text{K-subExp}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta \vdash [I]\, \tau : \mathsf{K}}\ \text{K-lab} \qquad \frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathbb{R}^+}{\Psi; \Theta; \Delta \vdash \mathbb{M}\, I\, \tau : \mathsf{K}}\ \text{K-monad}$$

$$\frac{\Psi, \alpha : \mathsf{K}'; \Theta; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \forall \alpha.\tau : \mathsf{K}}\ \text{K-tabs} \qquad \frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \forall i.\tau : \mathsf{K}}\ \text{K-iabs} \qquad \frac{\Psi; \Theta; \Delta, c \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash c \Rightarrow \tau : \mathsf{K}}\ \text{K-constraint}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{K} \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta \vdash c \& \tau : \mathsf{K}}\ \text{K-consAnd} \qquad \frac{\Psi; \Theta, i : \mathsf{S}; \Delta \vdash \tau : \mathsf{K}}{\Psi; \Theta; \Delta \vdash \lambda_t i.\tau : \mathsf{S} \to \mathsf{K}}\ \text{K-family}$$

$$\frac{\Psi; \Theta; \Delta \vdash \tau : \mathsf{S} \to \mathsf{K} \qquad \Theta; \Delta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta \vdash \tau \;\; I : \mathsf{K}}\ \text{K-iapp}$$

Figure 33: Kind rules for types

B.3 Semantics

$\boxed{\text{Pure reduction, } e \Downarrow_t v}$ $\boxed{\text{Forcing reduction, } e \Downarrow_t^c v}$

$$\frac{e_1 \Downarrow_{t_1} v \qquad e_2 \Downarrow_{t_2} l}{e_1 :: e_2 \Downarrow_{t_1+t_2+1} v :: l} \text{ E-cons} \qquad \frac{e_1 \Downarrow_{t_1} nil \qquad e_2 \Downarrow_{t_2} v}{\mathsf{match}\ e_1\ \mathsf{with}\ |nil \mapsto e_2\ |h :: t \mapsto e_3 \Downarrow_{t_1+t_2+1} v} \text{ E-matchNil}$$

$$\frac{e_1 \Downarrow_{t_1} v_h :: l \qquad e_3[v_h/h][l/t] \Downarrow_{t_2} v}{\mathsf{match}\ e_1\ \mathsf{with}\ |nil \mapsto e_2\ |h :: t \mapsto e_3 \Downarrow_{t_1+t_2+1} v} \text{ E-matchCons} \qquad \frac{e_1 \Downarrow_{t_1} v \qquad e_2[v/x] \Downarrow_{t_2} v'}{e_1; x.e_2 \Downarrow_{t_1+t_2+1} v'} \text{ E-exist}$$

$$\frac{e_1 \Downarrow_{t_1} \lambda x.e' \qquad e'[e_2/x] \Downarrow_{t_2} v'}{e_1\ e_2 \Downarrow_{t_1+t_2+1} v'} \text{ E-app} \qquad \frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2 \Downarrow_{t_2} v_2}{\langle\langle e_1, e_2 \rangle\rangle \Downarrow_{t_1+t_2+1} \langle\langle v_1, v_2 \rangle\rangle} \text{ E-TI}$$

$$\frac{e \Downarrow_{t_1} \langle\langle v_1, v_2 \rangle\rangle \qquad e'[v_1/x][v_2/y] \Downarrow_{t_2} v}{\mathsf{let}\langle\langle x, y \rangle\rangle = e\ \mathsf{in}\ e' \Downarrow_{t_1+t_2+1} v} \text{ E-TE} \qquad \frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2 \Downarrow_{t_2} v_2}{\langle e_1, e_2 \rangle \Downarrow_{t_1+t_2+1} \langle v_1, v_2 \rangle} \text{ E-WI} \qquad \frac{e \Downarrow_t \langle v_1, v_2 \rangle}{\mathsf{fst}(e) \Downarrow_{t+1} v_1} \text{ E-fst}$$

$$\frac{e \Downarrow_t \langle v_1, v_2 \rangle}{\mathsf{fst}(e) \Downarrow_{t+1} v_2} \text{ E-snd} \qquad \frac{e \Downarrow_t v}{\mathsf{inl}(e) \Downarrow_{t+1} \mathsf{inl}(v)} \text{ E-inl} \qquad \frac{e \Downarrow_t v}{\mathsf{inr}(e) \Downarrow_{t+1} \mathsf{inr}(v)} \text{ E-inr}$$

$$\frac{e \Downarrow_{t_1} \mathsf{inl}(v) \qquad e'[v/x] \Downarrow_{t_2} v'}{\mathsf{case}\ e, x.e', y.e'' \Downarrow_{t_1+t_2+1} \mathsf{inl}(v')} \text{ E-case1} \qquad \frac{e \Downarrow_{t_1} \mathsf{inr}(v) \qquad e''[v/y] \Downarrow_{t_2} v''}{\mathsf{case}\ e, x.e', y.e'' \Downarrow_{t_1+t_2+1} \mathsf{inl}(v'')} \text{ E-case2} \qquad \frac{}{!e \Downarrow_0 !e} \text{ E-expI}$$

$$\frac{e \Downarrow_{t_1} !e'' \qquad e'[e''/x] \Downarrow_{t_2} v}{\mathsf{let}\,!\,x = e\ \mathsf{in}\ e' \Downarrow_{t_1+t_2+1} v} \text{ E-expE} \qquad \frac{e[\mathsf{fix}x.e/x] \Downarrow_t v}{\mathsf{fix}x.e \Downarrow_{t+1} v} \text{ E-fix}$$

$$\frac{v \in \{(), x, nil, \lambda y.e, \Lambda.e, \mathsf{ret}\,e, \mathsf{bind}\,x = e_1\ \mathsf{in}\ e_2, \uparrow^\kappa, \mathsf{release}\,x = e_1\ \mathsf{in}\ e_2, \mathsf{store}\,e\}}{v \Downarrow_0 v} \text{ E-val}$$

$$\frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_2} v}{e\,[] \Downarrow_{t_1+t_2+1} v} \text{ E-tapp} \qquad \frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_2} v}{e\,[] \Downarrow_{t_1+t_2+1} v} \text{ E-iapp} \qquad \frac{e \Downarrow_{t_1} \Lambda.e' \qquad e' \Downarrow_{t_1} v}{e\,[] \Downarrow_{t_1+t_2+1} v} \text{ E-CE}$$

$$\frac{e_1 \Downarrow_{t_1} v \qquad e_2[v/x] \Downarrow_{t_2} v'}{\mathsf{clet}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+1} v'} \text{ E-CandE} \qquad \frac{e \Downarrow_t v}{\mathsf{ret}\,e \Downarrow_{t+1}^0 v} \text{ E-return}$$

$$\frac{e_1 \Downarrow_{t_1} v_1 \qquad v_1 \Downarrow_{t_2}^{c_1} v_1' \qquad e_2[v_1'/x] \Downarrow_{t_3} v_2 \qquad v_2 \Downarrow_{t_4}^{c_2} v_2'}{\mathsf{bind}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+t_3+t_4+1}^{c_1+c_2} v_2'} \text{ E-bind} \qquad \frac{}{\uparrow^\kappa \Downarrow_1^\kappa ()} \text{ E-tick}$$

$$\frac{e_1 \Downarrow_{t_1} v_1 \qquad e_2[v_1/x] \Downarrow_{t_2} v_2 \qquad v_2 \Downarrow_{t_3}^c v_2'}{\mathsf{release}\,x = e_1\ \mathsf{in}\ e_2 \Downarrow_{t_1+t_2+t_3+1}^c v_2'} \text{ E-release} \qquad \frac{e \Downarrow_t v}{\mathsf{store}\,e \Downarrow_{t+1}^0 v} \text{ E-store}$$

Figure 34: Evaluation rules: pure and forcing

B.4 Model

Definition 58 (Value and expression relation).

$$
\begin{array}{lll}
[\![\mathbf{1}]\!] & \triangleq & \{(p, T, ())\} \\
[\![\mathsf{b}]\!] & \triangleq & \{(p, T, v) \mid v \in [\![\mathsf{b}]\!]\} \\
[\![L^0\tau]\!] & \triangleq & \{(p, T, nil)\} \\
[\![L^{s+1}\tau]\!] & \triangleq & \{(p, T, v :: l) | \exists p_1, p_2. p_1 + p_2 \leq p \wedge (p_1, T, v) \in [\![\tau]\!] \wedge (p_2, T, l) \in [\![L^s\tau]\!]\} \\
[\![\tau_1 \otimes \tau_2]\!] & \triangleq & \{(p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \mid \exists p_1, p_2. p_1 + p_2 \leq p \wedge (p_1, T, v_1) \in [\![\tau_1]\!] \wedge (p_2, T, v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \,\&\, \tau_2]\!] & \triangleq & \{(p, T, \langle v_1, v_2\rangle) \mid (p, T, v_1) \in [\![\tau_1]\!] \wedge (p, T, v_2) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \oplus \tau_2]\!] & \triangleq & \{(p, T, \mathsf{inl}(v)) \mid (p, T, v) \in [\![\tau_1]\!]\} \cup \{(p, T, \mathsf{inr}(v)) \mid (p, T, v) \in [\![\tau_2]\!]\} \\
[\![\tau_1 \multimap \tau_2]\!] & \triangleq & \{(p, T, \lambda x.e) \mid \forall p', e', T'{<}T .(p', T', e') \in [\![\tau_1]\!]_{\mathcal{E}} \implies (p + p', T', e[e'/x]) \in [\![\tau_2]\!]_{\mathcal{E}}\} \\
[\![!_{a<I}\tau]\!] & \triangleq & \{(p, T, !e) \mid \exists p_0, \ldots, p_{I-1}. p_0 + \ldots + p_{I-1} \leq p \wedge \forall 0 \leq i < I.(p_i, T, e) \in [\![\tau[i/a]]\!]_{\mathcal{E}}\} \\
[\![[n]\,\tau]\!] & \triangleq & \{(p, T, v) \mid \exists p'. p' + n \leq p \wedge (p', T, v) \in [\![\tau]\!]\} \\
[\![\mathbb{M}\, n\, \tau]\!] & \triangleq & \{(p, T, v) \mid \forall n', T'{<}T, v'. v \Downarrow_{T'}^{n'} v' \implies \exists p'. n' + p' \leq p + n \wedge (p', T - T', v') \in [\![\tau]\!]\} \\
[\![\forall\alpha.\tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid \forall\tau', T'{<}T .(p, T', e) \in [\![\tau[\tau'/\alpha]]\!]_{\mathcal{E}}\} \\
[\![\forall i.\tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid \forall I\; T'{<}T .(p, T', e) \in [\![\tau[I/i]]\!]_{\mathcal{E}}\} \\
[\![c \Rightarrow \tau]\!] & \triangleq & \{(p, T, \Lambda.e) \mid \forall\, T'{<}T \,.\models c \implies (p, T', e) \in [\![\tau]\!]_{\mathcal{E}}\} \\
[\![c \& \tau]\!] & \triangleq & \{(p, T, v) \mid\; \models c \wedge (p, T, v) \in [\![\tau]\!]\} \\
[\![\exists s.\tau]\!] & \triangleq & \{(p, T, v) \mid \exists s'.(p, T, v) \in [\![\tau[s'/s]]\!]\} \\
[\![\lambda_t i.\tau]\!] & \triangleq & f \;\; where\; \forall I.\, f\; I = [\![\tau[I/i]]\!] \\
[\![\tau\;\, I]\!] & \triangleq & [\![\tau]\!]\; I \\
 & & \\
[\![\tau]\!]_{\mathcal{E}} & \triangleq & \{(p, T, e) \mid \forall v, T'{<}T .e \Downarrow_{T'} v \implies (p, T - T', v) \in [\![\tau]\!]\}
\end{array}
$$

Definition 59 (Interpretation of typing contexts).

$$
\begin{array}{lll}
[\![\Gamma]\!]_{\mathcal{E}} & = & \{(p, T, \gamma) \mid \exists f : \mathcal{V}ars \to \mathcal{P}ots. \\
 & & (\forall x \in dom(\Gamma).\, (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x \in dom(\Gamma)} f(x) \leq p)\} \\
[\![\Omega]\!]_{\mathcal{E}} & = & \{(p, T, \delta) \mid \exists f : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots. \\
 & & (\forall (x :_{a<I} \tau) \in \Omega. \forall 0 \leq i < I.\, (f\;\, x\;\, i, T, \delta(x)) \in [\![\tau[i/a]]\!]_{\mathcal{E}}) \wedge \\
 & & (\sum_{x :_{a<I} \tau \in \Omega} \sum_{0 \leq i < I} f\;\, x\;\, i) \leq p\}
\end{array}
$$

Definition 60 (Type and index substitutions). $\sigma : TypeVar \to Type$, $\iota : IndexVar \to Index$

Lemma 61 (Value monotonicity lemma). $\forall p, p', v, \tau, T', T.$
$(p, T, v) \in [\![\tau]\!] \wedge p \leq p' \wedge T' {\leq} T \implies (p', T', v) \in [\![\tau]\!]$

Proof. Proof by induction on τ ☐

Lemma 62 (Expression monotonicity lemma). $\forall p, p', v, \tau, T', T.$
$(p, T, e) \in [\![\tau]\!]_{\mathcal{E}} \wedge p \leq p' \wedge T' {\leq} T \implies (p', T', e) \in [\![\tau]\!]_{\mathcal{E}}$

Proof. From Definition 58 and Lemma 61 ☐

Lemma 63 (Lemma for substitution). $\forall p, \delta, I, \Omega.$
$(p, \delta) \in [\![\sum_{a<I} \Omega]\!] \implies \exists p_0, \ldots, p_{I-1}.$
$p_0 + \ldots + p_{I-1} \leq p \wedge \forall 0 \leq i < I.(p_i, \delta) \in [\![\Omega[i/a]]\!]$

Proof. Given: $(p, \delta) \in [\![\sum_{a<I} \Omega]\!]$

<u>When $\Omega = .$</u>

The proof is trivial simply choose p_i as 0 and we are done

<u>When $\Omega(a) = x_0 :_{b<J_0(a)} \tau_0(a), \ldots x_n :_{b<J_n(a)} \tau_n(a)$</u>

Therefore from Definition 54 and Definition 59 we have
$\exists f : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots.$
$(\forall (x_j :_{c<\sum_{a<I} J_j} \sigma) \in (\sum_{a<I} \Omega). \forall 0 \leq i < \sum_{a<I} J_j.\, (f\;\, x\;\, i, \delta(x_j)) \in [\![\sigma[i/c]]\!]) \wedge$
$(\sum_{x_j :_{c<\sum_{a<I} J_j} \sigma \in (\sum_{a<I} \Omega)} \sum_{0 \leq i < \sum_{a<I} J_j} f\;\, x_j\;\, i) \leq p$ (SM0)

To prove the desired, for each $i \in [0, I-1]$ we choose
p_i as $\sum_{x_j :_{b<J_j(i)} \tau_j(i) \in (\Omega(i))} \sum_{0 \leq k < J_j(i)} f\;\, x_j\;\, (k + \sum_{d<i} J_j[d/i])$
and we need to prove

1. $p_0 + \ldots + p_{I-1} \leq p$:
   It suffices to prove that
   $\sum_{0\leq i<I} \sum_{x_j :_{b<J_j(i)} \tau_j(i) \in dom(\Omega(i))} \sum_{0\leq k<J_j(i)} f\ x_j\ (k + \sum_{d<i} J_j(i)[d/i]) \leq p$
   We know that $dom(\sum_{a<I} \Omega) = dom(\Omega)$ and from (SM0) we get the desired
2. $\forall 0 \leq i < I.(p_i, \delta) \in [\![\Omega[i/a]]\!]$:
   This means given some $0 \leq i < I$, from Definition 59 it suffices to prove that
   $\exists f' : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots.$
   $(\forall (x_j :_{b<J_j(i)} \tau_j(i)) \in \Omega[i/a].\forall 0 \leq k < J_j(i).\,(f'\ x_j\ k, \delta(x_j)) \in [\![\tau_j(i)[k/b]]\!]) \wedge$
   $(\sum_{x_j :_{b<J_j(i)} \in \Omega[i/a]} \sum_{0\leq k<J_j(i)} f'\ x\ k) \leq p_i$

   We choose f' s.t
   $\forall x_j :_{b<J_j(i)} \tau_j(i) \in (\Omega[i/a]).\forall 0 \leq k < J_j(i).f'\ x_j\ k = f\ x_j\ (k + \sum_{d<i} J_j[d/i]),$

   And we need to prove:
   (a) $\forall (x_j :_{b<J_j(i)} \tau_j(i)) \in \Omega[i/a].\forall 0 \leq k < J_j(i).\,(f'\ x_j\ k, \delta(x_j)) \in [\![\tau_j(i)[k/b]]\!]$:
   This means given some $(x_j :_{b<J_j(i)} \tau_j(i)) \in \Omega[i/a]$ and some $0 \leq k < J_j(i)$ and it suffices to prove that
   $(f'\ x_j\ k, \delta(x_j)) \in [\![\tau_j(i)[k/b]]\!]$

   This means we need to prove that
   $(f\ x_j\ (k + \sum_{d<i} J_j[d/i]), \delta(x_j)) \in [\![\tau_j(i)[k/b]]\!]$ (SM1)

   Instantiating (SM0) with the given x_j and $(k + \sum_{d<i} J_j[d/i])$ we get
   $(f\ x_j\ (k + \sum_{d<i} J_j[d/i]), \delta(x_j)) \in [\![\sigma[(k + \sum_{d<i} J_j[d/i])/c]]\!]$
   And from Definition 54 we get the desired
   (b) $(\sum_{x_j :_{b<J_j(i)} \tau_j(i) \in \Omega[i/a]} \sum_{0\leq k<J_j(i)} f'\ x\ k) \leq p_i$:
   It suffices to prove that
   $(\sum_{x_j :_{b<J_j(i)} \tau_j(i) \in \Omega[i/a]} \sum_{0\leq k<J_j(i)} f\ x\ (k + \sum_{d<i} J_j[d/i])) \leq p_i$
   Since we know that p_i is $\sum_{x_j :_{b<J_j(i)} \tau_j(i) \in (\Omega(i))} \sum_{0\leq k<J_j(i)} f\ x_j\ (k + \sum_{d<i} J_j[d/i])$ therefore we are done

□

Theorem 64 (Fundamental theorem)**.** $\forall \Psi, \Theta, \Delta, \Omega, \Gamma, e, \tau \in Type.$
$\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \wedge (p_l, T, \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}} \wedge (p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}} \wedge . \models \Delta\ \iota \implies$
$(p_l + p_m, T, e\ \gamma\delta) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}.$

Proof. Proof by induction on the typing judgment
1. T-var1:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau \vdash x : \tau}\ \text{T-var1}$$

   Given: $(p_l, T, \gamma) \in [\![\Gamma, x : \tau\ \sigma\iota]\!]_{\mathcal{E}}$ and $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
   To prove: $(p_l + p_m, T, x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

   Since we are given that $(p_l, T, \gamma) \in [\![\Gamma, x : \tau\ \sigma\iota]\!]_{\mathcal{E}}$ therefore from Definition 59 we know that
   $\exists f.(f(x), T, \gamma(x)) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ where $f(x) \leq p_l$

   Therefore from Lemma 62 we get $(p_l + p_m, T, x\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
2. T-var2:

$$\frac{\Theta, \Delta \models I \geq 1}{\Psi; \Theta; \Delta; \Omega, x :_{a<I} \tau; \Gamma \vdash x : \tau[0/a]}\ \text{T-var2}$$

   Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$ and $(p_m, T, \delta) \in [\![(\Omega, x :_{a<I} \tau)\ \sigma\iota]\!]_{\mathcal{E}}$
   To prove: $(p_l + p_m, x\ \delta\gamma) \in [\![\tau[0/a]\ \sigma\iota]\!]_{\mathcal{E}}$

   Since we are given that $(p_m, T, \delta) \in [\![(\Omega, x :_{a<I} \tau)\ \sigma\iota]\!]_{\mathcal{E}}$ therefore from Definition 59 we know that
   $\exists f : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots.$
   $((f\ x\ 0, T, \delta(x)) \in [\![\tau[0/a]\ \sigma\iota]\!]_{\mathcal{E}})$ where $(f\ x\ 0) \leq p_m$

   Therefore from Lemma 62 we get $(p_l + p_m, T, x\ \delta\gamma) \in [\![\tau[0/a]\ \sigma\iota]\!]_{\mathcal{E}}$
3. T-unit:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash () : \mathbf{1}}\ \text{T-unit}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, ()\ \delta\gamma) \in [\![\mathbf{1}\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall\, T' {<} T\ .()\Downarrow_{T'} () \implies (p_m + p_l, T - T', ()) \in [\![\mathbf{1}]\!]$

This means given $() \Downarrow_0 ()$ it suffices to prove that
$(p_l + p_m, T, ()) \in [\![\mathbf{1}]\!]$

We get this directly from Definition 58

4. T-base:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash c : \mathsf{b}}\ \text{T-base}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, c) \in [\![\mathsf{b}]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v, T' {<} T\ .c \Downarrow_{T'} v \implies (p_m + p_l, T - T', c) \in [\![\mathsf{b}]\!]$

This means given some $v, T' {<} T$ s.t $c \Downarrow_{T'} v$. Also from E-val we know that $T'{=}\ 0$ therefore it suffices to prove that
$(p_l + p_m, T, v) \in [\![\mathsf{b}]\!]$

From (E-val) we know that $v = c$ therefore it suffices to prove that
$(p_l + p_m, T, c) \in [\![\mathsf{b}]\!]$
We get this directly from Definition 58

5. T-nil:

$$\frac{}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathit{nil} : L^0\ \tau}\ \text{T-nil}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, \mathit{nil}\ \delta\gamma) \in [\![L^0\ \tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall\, T' {<} T, v'.\mathit{nil} \Downarrow_{T'} v' \implies (p_l + p_m, T - T', v') \in [\![L^0\ \tau\ \sigma\iota]\!]$

This means given some $T' {<} T, v'$ s.t $\mathit{nil} \Downarrow_{T'} v'$ it suffices to prove that
$(p_l + p_m, T - T', v') \in [\![L^0\ \tau\ \sigma\iota]\!]$

From (E-val) we know that $T'{=}\ 0$ and $v' = \mathit{nil}$, therefore it suffices to prove that
$(p_l + p_m, T, \mathit{nil}) \in [\![L^0\ \tau\ \sigma\iota]\!]$
We get this directly from Definition 15

6. T-cons:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : L^n\tau \qquad \Theta \vdash n : \mathbb{N}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e_1 :: e_2 : L^{n+1}\tau}\ \text{T-cons}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (e_1 :: e_2)\ \delta\gamma) \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v', t {<} T\ .(e_1 :: e_2)\ \delta\gamma \Downarrow_t v' \implies (p_l + p_m, T - t, v') \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]$

This means given some $v', t {<} T$ s.t $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v'$, it suffices to prove that
$(p_l + p_m, T - t, v') \in [\![L^{n+1}\ \tau\ \sigma\iota]\!]$

From (E-cons) we know that $\exists v_f, l.v' = v_f :: l$

Therefore from Definition 58 it suffices to prove that
$$\exists p_1, p_2.p_1 + p_2 \leq p_l + p_m \wedge (p_1, T - t, v_f) \in [\![\tau\ \sigma\iota]\!] \wedge (p_2, T - t, l) \in [\![L^n\tau\ \sigma\iota]\!] \qquad \text{(F-C0)}$$

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$
Similarly from Definition 59 and Definition 56 we also know that
$\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>:

$(p_{l1} + p_{m1}, T, e_1\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 58 we have
$\forall t1 < T.e_1\ \delta\gamma \Downarrow v_f \implies (p_{l1} + p_{m1}, T - t1, v_f) \in [\![\tau]\!]$

Since we are given that $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v_f :: l$ therefore fom E-cons we also know that $\exists t1 < t.\ e_1\ \delta\gamma \Downarrow_{t1} v_f$
Therefore we have $(p_{l1} + p_{m1}, T - t1, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-C1)

IH2:
$(p_{l2} + p_{m2}, T, e_2\ \delta\gamma) \in [\![L^n\tau\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 58 we have
$\forall t2 < T .e_2\ \delta\gamma \Downarrow_{t2} l \implies (p_{l2} + p_{m2}, T - t2, l) \in [\![L^n\tau\ \sigma\iota]\!]$

Since we are given that $(e_1 :: e_2)\ \delta\gamma \Downarrow_t v_f :: l$ therefore fom E-cons we also know that $\exists t2 < t - t1.$ $e_2\ \delta\gamma \Downarrow l$
Since $t2 < t - t1 < t < T$, therefore we have
$(p_{l2} + p_{m2}, T - t2, l) \in [\![L^n\tau\ \sigma\iota]\!]$ (F-C2)

In order to prove (F-C0) we choose p_1 as $p_{l1} + p_{m1}$ and p_2 as $p_{l2} + p_{m2}$, we get the desired from (F-C1), (F-C2) and Lemma 61

7. T-match:

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega_2; \Gamma_1 \vdash e : L^n\ \tau \qquad \Psi; \Theta; \Delta, n = 0; \Omega_2; \Gamma_2 \vdash e_1 : \tau' \\ \Psi; \Theta, I; \Delta, n = I + 1; \Omega_2; \Gamma_2, h : \tau, t : L^I\tau \vdash e_2 : \tau' \qquad \Theta \vdash n : \mathbb{N} \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}\end{array}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2 : \tau'} \text{T-match}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that $(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-M0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$
Similarly from Definition 59 and Definition 56 we also know that
$\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

IH1
$(p_{l1} + p_{m1}, T, e\ \delta\gamma) \in [\![L^n\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t' < T .e\ \delta\gamma \Downarrow_{t'} v_1 \implies (p_{l1} + p_{m1}, T - t', v_1) \in [\![L^n\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t' < t, v_1.e\ \delta\gamma \Downarrow_{t'} v_1$.
Since $t' < t < T$, therefore we have $(p_{l1} + p_{m1}, T - t', v_1) \in [\![L^n\tau\ \sigma\iota]\!]$
2 cases arise:

(a) $v_1 = nil$:
In this case we know that $n = 0$ therefore
IH2
$(p_{l2} + p_{m2}, T, e_1\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T .e_1\ \delta\gamma \Downarrow_{t_1} v_f \implies (p_{l2} + p_{m2}, T - t_1, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t_1 < t.\ e_1\ \delta\gamma \Downarrow_{t_1} v_f$.
Since $t_1 < t < T$ therefore we have
$(p_{l2} + p_{m2}, T - t_1, v_f) \in [\![\tau'\ \sigma\iota]\!]$

And from Lemma 61 we get
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

And finally since $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

(b) $v_1 = v :: l$:
In this case we know that $n > 0$ therefore
IH2
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T, e_2\ \delta\gamma) \in [\![\tau'\ \sigma\iota']\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{h \mapsto v\} \cup \{t \mapsto l\}$ and
$\iota' = \iota \cup \{I \mapsto n-1\}$

This means from Definition 58 we have
$\forall t_2 < T\ .e_2\ \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since we know that $(\mathsf{match}\ e\ \mathsf{with}\ |nil \mapsto e_1\ |h :: t \mapsto e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-match we know that $\exists t_2 < t.\ e_2\ \delta\gamma' \Downarrow v_f$.
Since $t_2 < t < T$ therefore we have
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

From Lemma 61 we get
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t, v_f) \in [\![\tau'\ \sigma\iota']\!]$

And finally since $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$ therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota']\!]_{\mathcal{E}}$

And finally since we have $\Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}$ therefore we also have
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

8. T-existI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau[n/s] \qquad \Theta \vdash n : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \exists s : \mathsf{S}.\tau}\ \text{T-existI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, e\ \delta\gamma) \in [\![\exists s.\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.e\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f\ \delta\gamma) \in [\![\exists s.\tau\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $e\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![\exists s.\tau\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\exists s'.(p_l + p_m, T - t, v_f) \in [\![\tau[s'/s]\ \sigma\iota]\!]$ (F-E0)

IH: $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau[n/s]\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t' < T\ .e\ \delta\gamma \Downarrow_{t'} v_f \implies (p_l + p_m, T - t', v_f) \in [\![\tau[n/s]\ \sigma\iota]\!]$

Since we are given that $e\ \delta\gamma \Downarrow_t v_f$ therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau[n/s]\ \sigma\iota]\!]$ (F-E1)

To prove (F-E0) we choose s' as n and we get the desired from (F-E1)

9. T-existsE:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : \exists s.\tau \qquad \Psi; \Theta, s; \Delta; \Omega_2; \Gamma_2, x : \tau \vdash e' : \tau' \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e; x.e' : \tau'}\ \text{T-existE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (e; x.e')\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(e; x.e')\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-EE0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$
Similarly from Definition 59 and Definition 56 we also know that
$\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

IH1
$(p_{l1}+p_{m1}, T, e\ \delta\gamma) \in [\![\exists s.\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T\ .e\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{l1}, T - t_1, v_1) \in [\![\exists s.\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we know that $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-existE we know that $\exists t_1 < t, v_1.e\ \delta\gamma \Downarrow v_1$. Therefore we have
$(p_{l1}+p_{m1}, T - t_1, v_1) \in [\![\exists s.\tau\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\exists s'.(p_{l1}+p_{m1}, T - t_1, v_1) \in [\![\tau[s'/s]\ \sigma\iota]\!]$ (F-EE1)

IH2
$(p_{l2}+p_{m2}+p_{l1}+p_{m1}, T, e'\ \delta'\gamma) \in [\![\tau'\ \sigma\iota']\!]_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto e_1\}$ and $\iota' = \iota \cup \{s \mapsto s'\}$

This means from Definition 58 we have
$\forall t_2 < T\ .e'\ \delta'\gamma \Downarrow_{t_2} v_f \implies (p_{l2}+p_{m2}+p_{l1}+p_{m1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since we know that $(e; x.e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-existE we know that $\exists t_2 < t.\ e'\ \delta'\gamma \Downarrow v_f$.
Since $t_2 < t < T$ therefore we have
$(p_{l2}+p_{m2}+p_{l1}+p_{m1}, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since $p_l = p_{l1}+p_{l2}$ and $p_m = p_{m1}+p_{m2}$ therefore we get
$(p_l+p_m, T - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

And finally from Lemma 61 and since we have $\Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}$ therefore we also have
$(p_l+p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$
And we are done.

10. T-lam:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \lambda x.e : (\tau_1 \multimap \tau_2)}\ \text{T-lam}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l+p_m, T, (\lambda x.e)\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(\lambda x.e)\ \delta\gamma \Downarrow_t v_f \implies (p_l+p_m, T - t, v_f) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\lambda x.e)\ \delta\gamma \Downarrow_t v_f$. From E-val we know that $t = 0$ and $v_f = (\lambda x.e)\ \delta\gamma$. Therefore we have
$(p_l+p_m, T, (\lambda x.e)\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall p', e', T' < T\ .(p', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p_l+p_m+p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $p', e', T' < T$ s.t $(p', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that
$(p_l+p_m+p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-L1)

From IH we know that
$(p_l+p'+p_m, T, e\ \delta\gamma') \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{x \mapsto e'\}$

Therefore from Lemma 62 we get the desired

11. T-app:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : (\tau_1 \multimap \tau_2) \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e_1\ e_2 : \tau_2}\ \text{T-app}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, \delta) \in [\![(\Omega_1 \oplus \Omega_2)\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l+p_m, T, e_1\ e_2\ \delta\gamma) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_m+p_l, T - t, v_f) \in [\![\tau_2\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that

$(p_m + p_l, T - t, v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-A0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e_1\ \delta\gamma) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T\ .e_1 \Downarrow_{t_1} \lambda x.e \implies (p_{l1} + p_{m1}, T - t_1, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$
Since we know that $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-app we know that $\exists t_1 < t.e_1 \Downarrow_{t_1} \lambda x.e$, therefore we have
$(p_{l1} + p_{m1}, T - t_1, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\forall p', e_1, T_1 < T - t_1.(p', T_1, e_1') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p_{l1} + p_{m1} + p', T_1, e[e_1'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-A1)

<u>IH2</u>
$(p_{l2} + p_{m2}, T - t_1 - 1, e_2\ \delta\gamma) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ (F-A2)

Instantiating (F-A1) with $p_{l2} + p_{m2}$ and $e_2\ \delta\gamma$ we get
$(p_{l1} + p_{m1} + p_{l2} + p_{m2}, T - t_1 - 1, e[e_2\ \delta\gamma/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_2 < T - t_1 - 1.e[e_2\ \delta\gamma/x] \Downarrow_{t_2} v_f \implies (p_l + p_m, T - t_1 - 1 - t_2, v_f) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(e_1\ e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-app we know that $\exists t_2.e[e_2\ \delta\gamma/x] \Downarrow_{t_2} v_f$, where $t_2 = t - t_1 - 1$, therefore we have
$(p_l + p_m, T - t_1 - t_2 - 1, v_f) \in [\![\tau_2\ \sigma\iota]\!]$
Since from E-app we know that $t = t_1 + t_2 + 1$, this proves (F-A0)

12. T-sub:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau'} \text{ T-sub}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma)\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
<u>IH</u> $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH and Lemma 22

13. T-weaken:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Psi; \Theta \models \Gamma' <: \Gamma \qquad \Psi; \Theta \models \Omega' <: \Omega}{\Psi; \Theta; \Delta; \Omega'; \Gamma' \vdash e : \tau} \text{ T-weaken}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma')\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega')\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $(p_l, T, \gamma) \in [\![(\Gamma')\sigma\iota]\!]_{\mathcal{E}}$ therefore from Lemma 67 we also have $(p_l, T, \gamma) \in [\![(\Gamma)\sigma\iota]\!]_{\mathcal{E}}$

Similarly since we are given that $(p_m, T, \delta) \in [\![(\Omega')\sigma\iota]\!]_{\mathcal{E}}$ therefore from Lemma 68 we also have $(p_m, T, \delta) \in [\![(\Omega)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH</u>:
$(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH

14. T-tensorI:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \langle\!\langle e_1, e_2 \rangle\!\rangle : (\tau_1 \otimes \tau_2)} \text{ T-tensorI}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega_1 \oplus \Omega_2)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, \langle\!\langle e_1, e_2 \rangle\!\rangle\ \delta\gamma) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T . \langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle \implies (p_l + p_m, T - t, \langle\langle v_{f1}, v_{f2} \rangle\rangle \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$

This means given some $t < T$ s.t $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ it suffices to prove that
$(p_l + p_m, T - t, \langle\langle v_{f1}, v_{f2} \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$ (F-TI0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}. p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

IH1:
$(p_{l1} + p_{m1}, T, e_1\ \delta\gamma) \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 58 we have
$\forall t_1 < T . e_1\ \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_{l1} + p_{m1}, T - t_1, v_{f1}) \in [\![\tau_1\ \sigma\iota]\!]$

Since we are given that $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ therefore fom E-TI we know that $\exists t_1 < t. e_1\ \delta\gamma \Downarrow_{t_1} v_{f1}$
Hence we have $(p_{l1} + p_{m1}, T - t_1, v_{f1}) \in [\![\tau_1\ \sigma\iota]\!]$ (F-TI1)

IH2:
$(p_{l2} + p_{m2}, T, e_2\ \delta\gamma) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$
Therefore from Definition 58 we have
$\forall t_2 < T . e_2\ \delta\gamma \Downarrow_{t_2} v_{f2} \implies (p_{l2} + p_{m2}, T - t_2, v_{f2} \in [\![\tau_2\ \sigma\iota]\!]$

Since we are given that $\langle\langle e_1, e_2 \rangle\rangle\ \delta\gamma \Downarrow_t \langle\langle v_{f1}, v_{f2} \rangle\rangle$ therefore fom E-TI we also know that $\exists t_2 < t. e_2\ \delta\gamma \Downarrow_{t_2} v_{f2}$
Since $t_2 < t < T$ therefore we have
$(p_{l2} + p_{m2}, T - t_2, v_{f2}) \in [\![\tau_2\ \sigma\iota]\!]$ (F-TI2)

Applying Lemma 61 on (F-TI1) and (F-TI2) and by using Definition 15 we get the desired.

15. T-tensorE:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (\tau_1 \otimes \tau_2) \qquad \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1, y : \tau_2 \vdash e' : \tau}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e' : \tau} \text{ T-tensorE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-TE0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}. p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

IH1
$(p_{l1} + p_{m1}, T, e\ \delta\gamma) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e\ \delta\gamma \Downarrow_{t_1} \langle\langle v_1, v_2 \rangle\rangle\ \delta\gamma \implies (p_{l1} + p_{m1}, T - t_1, \langle\langle v_1, v_2 \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$

Since we know that $(\mathsf{let}\langle\langle x, y \rangle\rangle = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-subExpE we know that $\exists t_1 < t, v_1, v_2. e\ \delta\gamma \Downarrow_{t_1} \langle\langle v_1, v_2 \rangle\rangle$. Therefore we have
$(p_{l1} + p_{m1}, T - t_1, \langle\langle v_1, v_2 \rangle\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]_{\mathcal{E}}$
From Definition 15 we know that
$\exists p_1, p_2. p_1 + p_2 \leq p_{l1} + p_{m1} \wedge (p_1, T, v_1) \in [\![\tau_1\ \sigma\iota]\!] \wedge (p_2, T, v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-TE1)

IH2
$(p_{l2} + p_{m2} + p_1 + p_2, T, e'\ \delta\gamma') \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{x \mapsto v_1\} \cup \{y \mapsto v_2\}$

This means from Definition 15 we have
$\forall t_2 < T . e' \ \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_{m2} + p_1 + p_2, T - t_2, v_f) \in \llbracket \tau \ \sigma\iota \rrbracket$

Since we know that $(\mathsf{let}\langle\!\langle x, y \rangle\!\rangle = e \ \mathsf{in} \ e') \ \delta\gamma \Downarrow_t v_f$ therefore from E-TE we know that $\exists t_2 < t . e' \ \delta\gamma' \Downarrow_{t_2} v_f$.
Therefore we have
$(p_{l2} + p_{m2} + p_1 + p_2, T - t_2, v_f) \in \llbracket \tau \ \sigma\iota \rrbracket$

From Lemma 61 we get
$(p_l + p_m, T - t, v_f) \in \llbracket \tau \ \sigma\iota \rrbracket_{\mathcal{E}}$
And we are done

16. T-withI:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_1 : \tau_1 \qquad \Psi; \Theta; \Delta; \Omega; \Gamma \vdash e_2 : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \langle e_1, e_2 \rangle : (\tau_1 \ \& \ \tau_2)} \text{ T-withI}$$

Given: $(p_l, T, \gamma) \in \llbracket \Gamma \sigma\iota \rrbracket_{\mathcal{E}}$, $(p_m, T, \delta) \in \llbracket \Omega \ \sigma\iota \rrbracket_{\mathcal{E}}$
To prove: $(p_l + p_m, T, \langle e_1, e_2 \rangle \ \delta\gamma) \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T . \langle e_1, e_2 \rangle \ \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2} \rangle \implies (p_l + p_m, T - t, \langle v_{f1}, v_{f2} \rangle \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket$

This means given $\langle e_1, e_2 \rangle \ \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2} \rangle$ it suffices to prove that
$(p_l + p_m, T - t, \langle v_{f1}, v_{f2} \rangle) \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket$ (F-WI0)

<u>IH1</u>:
$(p_l + p_m, T, e_1 \ \delta\gamma) \in \llbracket \tau_1 \ \sigma\iota \rrbracket_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_1 < T . e_1 \ \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_l + p_m, T - t_1, v_{f1}) \in \llbracket \tau_1 \ \sigma\iota \rrbracket$

Since we are given that $\langle e_1, e_2 \rangle \ \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2} \rangle$ therefore fom E-WI we know that $\exists t_1 < t . e_1 \ \delta\gamma \Downarrow_{t_1} v_{f1}$
Since $t_1 < t < T$, therefore we have
$(p_l + p_m, T - t_1, v_{f1}) \in \llbracket \tau_1 \ \sigma\iota \rrbracket$ (F-WI1)

<u>IH2</u>:
$(p_l + p_m, T, e_2 \ \delta\gamma) \in \llbracket \tau_2 \ \sigma\iota \rrbracket_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_2 < T . e_2 \ \delta\gamma \Downarrow_{t_2} v_{f2} \implies (p_l + p_m, T - t_2, v_{f2} \in \llbracket \tau_2 \ \sigma\iota \rrbracket$

Since we are given that $\langle e_1, e_2 \rangle \ \delta\gamma \Downarrow_t \langle v_{f1}, v_{f2} \rangle$ therefore fom E-WI we also know that $\exists t_2 < t . e_2 \ \delta\gamma \Downarrow_{t_2} v_{f2}$
Since $t_2 < t < T$, therefore we have
$(p_l + p_m, T - t_2, v_{f2}) \in \llbracket \tau_2 \ \sigma\iota \rrbracket$ (F-WI2)

Applying Lemma 61 on (F-W1) and (F-W2) we get the desired.

17. T-fst:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\tau_1 \ \& \ \tau_2)}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{fst}(e) : \tau_1} \text{ T-fst}$$

Given: $(p_l, T, \gamma) \in \llbracket (\Gamma) \ \sigma\iota \rrbracket_{\mathcal{E}}$, $(0, T, \delta) \in \llbracket \Omega \ \sigma\iota \rrbracket_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (\mathsf{fst}(e)) \ \delta\gamma) \in \llbracket \tau_1 \ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{fst}(e)) \ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in \llbracket \tau_1 \ \sigma\iota \rrbracket$

This means given some $t < T, v_f$ s.t $(\mathsf{fst}(e)) \ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in \llbracket \tau_1 \ \sigma\iota \rrbracket$ (F-F0)

<u>IH</u>
$(p_l + p_m, T, e \ \delta\gamma) \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t_1 < T . e \ \delta\gamma \Downarrow_{t_1} \langle v_1, v_2 \rangle \ \delta\gamma \implies (p_l + p_m, T - t_1, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket$

Since we know that $(\mathsf{fst}(e)) \ \delta\gamma \Downarrow_t v_f$ therefore from E-fst we know that $\exists t_1 < t . v_1, v_2 . e \ \delta\gamma \Downarrow_{t_1} \langle v_1, v_2 \rangle$.
Since $t_1 < t < T$, therefore we have
$(p_l + p_m, T - t_1, \langle v_1, v_2 \rangle) \in \llbracket (\tau_1 \ \& \ \tau_2) \ \sigma\iota \rrbracket$

From Definition 15 we know that

$(p_l + p_m, T - t_1, v_1) \in \llbracket \tau_1\ \sigma\iota \rrbracket$

Finally using Lemma 61 we also have
$(p_l + p_m, T - t, v_1) \in \llbracket \tau_1\ \sigma\iota \rrbracket$
Since from E-fst we know that $v_f = v_1$, therefore we are done.

18. T-snd:
Similar reasoning as in T-fst case above.
19. T-inl:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau_1}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{inl}(e) : \tau_1 \oplus \tau_2} \text{ T-inl}$$

Given: $(p_l, T, \gamma) \in \llbracket \Gamma\sigma\iota \rrbracket_{\mathcal{E}}$, $(0, T, \delta) \in \llbracket \Omega\ \sigma\iota \rrbracket_{\mathcal{E}}$
To prove: $(p_l + p_m, T, \mathsf{inl}(e)\ \delta\gamma) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T . \mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v) \implies (p_l + p_m, T - t, \mathsf{inl}(v) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket$

This means given some $t < T$ s.t $\mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v)$ it suffices to prove that
$(p_l + p_m, T - t, \mathsf{inl}(v)) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket$ (F-IL0)

<u>IH</u>:
$(p_l + p_m, T, e_1\ \delta\gamma) \in \llbracket \tau_1\ \sigma\iota \rrbracket_{\mathcal{E}}$
Therefore from Definition 15 we have
$\forall t_1 < T . e_1\ \delta\gamma \Downarrow_{t_1} v_{f1} \implies (p_l + p_m, T - t_1, v_{f1}) \in \llbracket \tau_1\ \sigma\iota \rrbracket$

Since we are given that $\mathsf{inl}(e)\ \delta\gamma \Downarrow_t \mathsf{inl}(v)$ therefore fom E-inl we know that $\exists t_1 < t . e\ \delta\gamma \Downarrow_{t_1} v$
Hence we have $(p_l + p_m, T - t_1, v) \in \llbracket \tau_1\ \sigma\iota \rrbracket$
From Lemma 61 we get $(p_l + p_m, T - t, v) \in \llbracket \tau_1\ \sigma\iota \rrbracket$

And finally from Definition 15 we get (F-IL0)

20. T-inr:
Similar reasoning as in T-inr case above.
21. T-case:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \vdash e : (\tau_1 \oplus \tau_2) \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, x : \tau_1 \vdash e_1 : \tau \qquad \Psi; \Theta; \Delta; \Omega; \Gamma_2, y : \tau_2 \vdash e_2 : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{case}\ e, x.e_1, y.e_2 : \tau} \text{ T-case}$$

Given: $(p_l, T, \gamma) \in \llbracket (\Gamma_1 \oplus \Gamma_2)\ \sigma\iota \rrbracket_{\mathcal{E}}$, $(0, T, \delta) \in \llbracket \Omega\ \sigma\iota \rrbracket_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma) \in \llbracket \tau\ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 15 it suffices to prove that
$\forall t < T, v_f . (\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in \llbracket \tau\ \sigma\iota \rrbracket$

This means given some $t < T, v_f$ s.t $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in \llbracket \tau\ \sigma\iota \rrbracket$ (F-C0)

From Definition 16 and Definition 14 we know that $\exists p_{l1}, p_{l2} . p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in \llbracket (\Gamma_1)\sigma\iota \rrbracket_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in \llbracket (\Gamma_2)\sigma\iota \rrbracket_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2} . p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in \llbracket (\Omega_1)\sigma\iota \rrbracket_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in \llbracket (\Omega_2)\sigma\iota \rrbracket_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e\ \delta\gamma) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket_{\mathcal{E}}$

This means from Definition 15 we have
$\forall t' < T . e\ \delta\gamma \Downarrow_{t'} v_1\ \delta\gamma \implies (p_{l1} + p_{m1}, T - t', v_1) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket$

Since we know that $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-case we know that $\exists t' < t, v_1 . e\ \delta\gamma \Downarrow_{t'} v_1$.
Since $t' < t < T$, therefore we have
$(p_{l1} + p_{m1}, T - t', v_1) \in \llbracket (\tau_1 \oplus \tau_2)\ \sigma\iota \rrbracket$
2 cases arise:

(a) $v_1 = \mathsf{inl}(v)$:
<u>IH2</u>
$(p_{l2} + p_{m2} p_{l1} + p_{m1}, T - t', e_1\ \delta\gamma') \in \llbracket \tau\ \sigma\iota \rrbracket_{\mathcal{E}}$
where

$\gamma' = \gamma \cup \{x \mapsto v\}$

This means from Definition 15 we have
$\forall t_1 <T -t'.e_1\ \delta\gamma' \Downarrow_{t_1} v_f \implies (p_{l2} + p_{m2} + p_{l1} + p_{m1}, T -t' - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we know that $(\mathsf{case}\ e, x.e_1, y.e_2)\ \delta\gamma \Downarrow_t v_f$ therefore from E-case we know that $\exists t_1.e_1\ \delta\gamma' \Downarrow v_f$ where $t_1 = t - t' - 1$.
Since $t_1 = t - t' - 1 <T -t'$ therefore we have
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T -t' - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]$

From Lemma 61 we get
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T -t, v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$
And we are done

(b) $v_1 = \mathsf{inr}(v)$:
Similar reasoning as in the inl case above.

22. T-subExpI:

$$\frac{\Psi; \Theta, a; \Delta, a < I; \Omega; . \vdash e : \tau}{\Psi; \Theta; \Delta; \sum_{a<I} \Omega; . \vdash !e :!_{a<I}\tau} \text{ T-subExpI}$$

Given: $(p_l, \gamma) \in [\![.]\!]_{\mathcal{E}}$, $(p_m, \delta) \in [\![(\sum_{a<I} \Omega)\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, !e\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t <T .(!e)\ \delta\gamma \Downarrow_t (!e)\ \delta\gamma \implies (p_m + p_l, T -t, (!e)\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]$

This means given some $t <T$. s.t $(!e)\ \delta\gamma \Downarrow_t (!e)\ \delta\gamma$ it suffices to prove that
$(p_m + p_l, T -t, (!e)\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\exists p_0, \ldots, p_{I-1}.p_0 + \ldots + p_{I-1} \leq (p_m + p_l) \wedge \forall 0 \leq i < I.(p_i, T, e\ \delta\gamma) \in [\![\tau[i/a]]\!]_{\mathcal{E}}$ (F-SI0)

Since we know that $(p_m, T, \delta) \in [\![(\sum_{a<I} \Omega)\ \sigma\iota]\!]_{\mathcal{E}}$ therefore from Lemma 63 we know that
$\exists p'_0, \ldots, p'_{I-1}.\ p'_0 + \ldots + p'_{I-1} \leq p_m \wedge \forall 0 \leq i < I.(p_i, T, \delta) \in [\![\Omega[i/a]]\!]_{\mathcal{E}}$ (F-SI1)

Instantiating IH with each $p'_0 \ldots p'_{I-1}$ we get
$(p'_0, T, e\ \delta\gamma) \in [\![\tau[0/a]\ \sigma\iota]\!]_{\mathcal{E}}$ and
...
$(p'_{I-1}, T, e\ \delta\gamma) \in [\![\tau[I-1/a]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SI2)

Therefore we get (F-SI0) from (F-SI1) and (F-SI2)

23. T-subExpE:

$$\frac{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e : (!_{a<I}\tau) \qquad \Psi; \Theta; \Delta; \Omega_2, x :_{a<I} \tau; \Gamma_2 \vdash e' : \tau'}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{let}\,!\,x = e\ \mathsf{in}\ e' : \tau'} \text{ T-subExpE}$$

Given: $(p_l, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, \delta) \in [\![(\Omega_1 \oplus \Omega_2)\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, (\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t <T, v_f.(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f \implies (p_m + p_l, T -t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $t <T$ s.t. $(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_m + p_l, T -t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-SE0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 <T .e\ \delta\gamma \Downarrow_{t_1} !e_1\ \delta\gamma \implies (p_{l1} + p_{m1}, T -t_1, !e_1\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]$

Sice we know that $(\mathsf{let}\,!\,x = e\ \mathsf{in}\ e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-subExpE we know that $\exists t_1 < t, e_1.e\ \delta\gamma \Downarrow_{t_1} !e_1\ \delta\gamma$. Therefore we have

$(p_{l1} + p_{m1}, T - t_1, !e_1\ \delta\gamma) \in [\![!_{a<I}\tau\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\exists p_0, \ldots, p_{I-1}.p_0 + \ldots + p_{I-1} \leq (p_{l1} + p_{m1}) \wedge \forall 0 \leq i < I.(p_i, T - t_1, e_1\ \delta\gamma) \in [\![\tau[i/a]]\!]_{\mathcal{E}}$ (F-SE1)

IH2
$(p_{l2} + p_{m2} + p_0 + \ldots + p_{I-1}, T - t_1, e'\ \delta'\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto e_1\}$

This means from Definition 58 we have
$\forall t_2 < T - t_1.e'\ \delta'\gamma \Downarrow_{t_2} v_f \implies (p_{l2} + p_{m2} + p_0 + \ldots + p_{I-1}, T - t_1 - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since we know that $(\mathsf{let}\,!\,x = e \mathsf{\ in\ } e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-subExpE we know that $\exists t_2.e'\ \delta'\gamma \Downarrow v_f$ s.t. $t_2 = t - t_1 - 1$. Therefore we have
$(p_{l2} + p_{m2} + p_0 + \ldots + p_{I-1}, T - t_1 - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since from (F-SE1) we know that $p_0 + \ldots + p_{I-1} \leq p_{l1} + p_{m1}$ therefore from Lemma 62 we get
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

And finally since $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$ therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$
And we are done

24. T-tabs:

$$\frac{\Psi, \alpha : \mathsf{K}; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall \alpha : \mathsf{K}.\tau)}\ \text{T-tabs}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(\forall \alpha : \mathsf{K}.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v.\Lambda.e\ \delta\gamma \Downarrow_t v \implies (p_m + p_l, T - t, v) \in [\![(\forall \alpha : \mathsf{K}.\tau)\ \sigma\iota]\!]$

This means given some v s.t $\Lambda.e\ \delta\gamma \Downarrow v$ and from (E-val) we know that $v = \Lambda.e\ \delta\gamma$ and $t = 0$ therefore it suffices to prove that
$(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(\forall \alpha : \mathsf{K}.\tau)\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall \tau', T' < T\ .(p_l + p_m, T', e\ \delta\gamma) \in [\![\tau[\tau'/\alpha]\sigma\iota]\!]_{\mathcal{E}}$

This means given some $\tau', T' < T$ it suffices to prove that
$(p_l + p_m, T', e\ \delta\gamma) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-TAb0)

IH $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma'\iota]\!]_{\mathcal{E}}$
where
$\sigma' = \sigma \cup \{\alpha \mapsto \tau'\}$

We get the desired directly from IH

25. T-tapp:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall \alpha : \mathsf{K}.\tau) \qquad \Psi; \Theta; \Delta \vdash \tau' : \mathsf{K}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [] : (\tau[\tau'/\alpha])}\ \text{T-tapp}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, e\ []\ \delta\gamma) \in [\![(\tau[\tau'/\alpha])\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(e\ [])\ \delta\gamma \Downarrow_t v_f \implies (p_m + p_l, T - t, v_f) \in [\![(\tau[\tau'/\alpha])\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e\ [])\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_m + p_l, T - t, v_f) \in [\![(\tau[\tau'/\alpha])\ \sigma\iota]\!]$ (F-Tap0)

IH
$(p_l + p_m, T, e\ \delta\gamma) \in [\![(\forall \alpha.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T, v'.e\ \delta\gamma \Downarrow_{t_1} v' \implies (p_l + p_m, T - t_1, v') \in [\![(\forall \alpha\tau)\ \sigma\iota]\!]$
Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-tapp we know that $\exists t_1 < t.e\delta\gamma \Downarrow_{t_1} \Lambda.e$, therefore we have

$(p_l + p_m, T - t_1, \Lambda.e) \in [\![(\forall\alpha.\tau)\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\forall\tau'', T_1 < T - t_1.(p_l + p_m, T - t_1 - T_1, e\ \delta\gamma) \in [\![\tau[\tau''/\alpha]\ \sigma\iota]\!]_{\mathcal{E}}$

Instantiating it with the given τ' and $T - t_1 - 1$ we get
$(p_l + p_m, T - t_1 - 1, e\ \delta\gamma) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 we know that
$\forall t_2 < T - t_1 - 1, v''.e\ \delta\gamma \Downarrow_{t_2} v'' \implies (p_l + p_m, T - t_1 - 1 - t_2, v'') \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]$

Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-tapp we know that $\exists t_2.e \Downarrow_{t_2} v_f$ where $t_2 = t - t_1 - 1$
Since $t_2 = t - t_1 - 1 < T - t_1 - 1$, therefore we have
$(p_l + p_m, T - t, v_f) \in [\![\tau[\tau'/\alpha]\ \sigma\iota]\!]$
And we are done.

26. T-iabs:

$$\frac{\Psi; \Theta, i : \mathsf{S}; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (\forall i : \mathsf{S}.\tau)}\ \text{T-iabs}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma, \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v.\Lambda.e\ \delta\gamma \Downarrow_t v \implies (p_m + p_l, T - t, v) \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]$

This means given some $t < T, v$ s.t $\Lambda.e\ \delta\gamma \Downarrow_t v$ and from (E-val) we know that $v = \Lambda.e\ \delta\gamma$ and $t = 0$ therefore it suffices to prove that
$(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall I.(p_l + p_m, T, e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some I it suffices to prove that
$(p_l + p_m, T, e) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$ (F-TAb0)

<u>IH</u> $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota']\!]_{\mathcal{E}}$
where
$\iota' = \iota \cup \{i \mapsto I\}$

We get the desired directly from IH

27. T-iapp:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (\forall i : \mathsf{S}.\tau) \qquad \Theta \vdash I : \mathsf{S}}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e\ [] : (\tau[I\ /i])}\ \text{T-iapp}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, e\ []\ \delta\gamma) \in [\![(\tau[I\ /i])\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t < T, v_f.(e\ [])\ \delta\gamma \Downarrow_t v_f \implies (p_m + p_l, T - t, v_f) \in [\![(\tau[I\ /i])\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t $(e\ [])\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_m + p_l, T - t, v_f) \in [\![(\tau[I\ /i])\ \sigma\iota]\!]$ (F-Iap0)

<u>IH</u>
$(p_l + p_m, T, e\ \delta\gamma) \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T, v'.e\ \delta\gamma \Downarrow_{t_1} v' \implies (p_l + p_m, T - t_1, v') \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]$
Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from (E-iapp) we know that $\exists t_1 < t.e\delta\gamma \Downarrow_{t_1} \Lambda.e$, therefore we have
$(p_l + p_m, T - t_1, \Lambda.e) \in [\![(\forall i : \mathsf{S}.\tau)\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\forall I'', T_1 < T - t_1.(p_l + p_m, T - t_1 - T_1, e\ \delta\gamma) \in [\![\tau[I''/i]\ \sigma\iota]\!]_{\mathcal{E}}$

Instantiating it with the given I and $T - t_1 - 1$ we get
$(p_l + p_m, T - t_1 - 1, e\ \delta\gamma) \in [\![\tau[I/i]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 we know that
$\forall v'', t_2 < T - t_1 - 1. e\ \delta\gamma \Downarrow_{t_2} v'' \implies (p_l + p_m, T - t_1 - 1 - t_2, v'') \in [\![\tau[I/i]\ \sigma\iota]\!]$

Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-iapp we know that $\exists t_2. e \Downarrow_{t_2} v_f$ where $t_2 = t - t_1 - 1$
Since $t_2 = t - t_1 - 1 < T - t_1 - 1$, therefore we have
$(p_l + p_m, v_f) \in [\![\tau[I/i]\ \sigma\iota]\!]$
And we are done.

28. T-CI:

$$\frac{\Psi; \Theta; \Delta, c; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \Lambda.e : (c \Rightarrow \tau)}\ \text{T-CI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v, t < T . \Lambda.e\ \delta\gamma \Downarrow_t v \implies (p_m + p_l, T - t, v) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

This means given some $v, t < T$ s.t $\Lambda.e\ \delta\gamma \Downarrow_t v$ and from (E-val) we know that $v = \Lambda.e\ \delta\gamma$ and $t = 0$ therefore it suffices to prove that
$(p_l + p_m, T, \Lambda.e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall\ T' < T\ . \models c\ \iota \implies (p_l + p_m, T', e\ \delta\gamma) \in [\![\tau\sigma\iota]\!]_{\mathcal{E}}$

This means given some $T' < T$ s.t. $. \models c\ \iota$ it suffices to prove that
$(p_l + p_m, T', e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

<u>IH</u> $(p_l + p_m, T', e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

We get the desired directly from IH

29. T-CE:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : (c \Rightarrow \tau) \qquad \Theta; \Delta \models c}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e[] : \tau}\ \text{T-CE}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, e\ []\ \delta\gamma) \in [\![(\tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v_f, t < T . (e\ [])\ \delta\gamma \Downarrow_t v_f \implies (p_m + p_l, T - t, v_f) \in [\![(\tau)\ \sigma\iota]\!]$

This means given some $v_f, t < T$ s.t. $(e\ [])\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_m + p_l, T - t, v_f) \in [\![(\tau)\ \sigma\iota]\!]$ (F-Tap0)

<u>IH</u>
$(p_l + p_m, T, e\ \delta\gamma) \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall v', t' < T . e\ \delta\gamma \Downarrow_{t'} v' \implies (p_l + p_m, T - t', v') \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$
Since we know that $(e\ [])\ \delta\gamma \Downarrow_t v_f$ therefore from E-CE we know that $\exists t' < t. e\delta\gamma \Downarrow_{t'} \Lambda.e'$, therefore we have
$(p_l + p_m, T - t', \Lambda.e') \in [\![(c \Rightarrow \tau)\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$\forall t'' < T - t'. \models c\ \iota \implies (p_l + p_m, T - t' - t'', e'\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we are given $\Theta; \Delta \models c$ and $. \models \Delta\ \iota$. Therefore instantiating it with $T - t' - 1$ and since we know that $. \models c\ \iota$. Hence we get
$(p_l + p_m, T - t' - 1, e'\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall v'_f, t'' < T - t' - 1.(e')\ \delta\gamma \Downarrow v'_f \implies (p_m + p_l, v'_f) \in [\![(\tau)\ \sigma\iota]\!]$

Since from E-CE we know that $e'\delta\gamma \Downarrow_t v_f$ therefore we know that $\exists t''. e'\ \delta\gamma \Downarrow_{t''} v_f$ s.t $t = t' + t'' + 1$
Therefore instantiating (F-CE1) with the given v_f and t'' we get
$(p_m + p_l, T - t, v_f) \in [\![(\tau)\ \sigma\iota]\!]$
and we are done.

30. T-CAndI:

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e:\tau \qquad \Theta;\Delta \models c}{\Psi;\Theta;\Delta;\Omega;\Gamma \vdash e:(c\&\tau)} \text{ T-CAndI}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, e\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v_f, t < T . e\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]$

This means given some $v_f, t < T$ s.t $e\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![c\&\tau\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$. \models c\iota \wedge (p_l + p_m, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we are given that $. \models \Delta\iota$ and $\Theta;\Delta \models c$ therefore it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-CAI0)

<u>IH</u>: $(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t' < T . e\ \delta\gamma \Downarrow_{t'} v_f \implies (p_l + p_m, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]$

Since we are given that $e\ \delta\gamma \Downarrow_t v_f$ therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau\ \sigma\iota]\!]$ (F-CAI1)

We get the desired from (F-CAI1)

31. T-CAndE:

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma_1 \vdash e:(c\&\tau) \qquad \Psi;\Theta;\Delta,c;\Omega;\Gamma_2, x:\tau \vdash e':\tau'}{\Psi;\Theta;\Delta;\Omega;\Gamma_1 \oplus \Gamma_2 \vdash \mathsf{clet}\, x = e \,\mathsf{in}\, e' : \tau'} \text{ T-CAndE}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\ \sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega)\ \sigma\iota]\!]_{\mathcal{E}}$
To prove: $(p_l + p_m, T, (\mathsf{clet}\, x = e \,\mathsf{in}\, e')\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall v_f, t < T .(\mathsf{clet}\, x = e \,\mathsf{in}\, e')\ \delta\gamma \Downarrow_t v_f \implies (p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$

This means given some $v_f, t < T$ s.t. $(\mathsf{clet}\, x = e \,\mathsf{in}\, e')\ \delta\gamma \Downarrow_t v_f$ it suffices to prove that
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota]\!]$ (F-CAE0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}. p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$
Similarly from Definition 59 and Definition 56 we also know that
$\exists p_{m1}, p_{m2}. p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e\ \delta\gamma) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T . e\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{l1}, T - t_1 v_1) \in [\![c\&\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we know that $(\mathsf{clet}\, x = e \,\mathsf{in}\, e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-CAndE we know that $\exists t_1 < t, v_1 . e\ \delta\gamma \Downarrow_{t_1} v_1$.
Therefore we have
$(p_{l1} + p_{m1}, T - t_1, v_1) \in [\![c\&\tau\ \sigma\iota]\!]$

Therefore from Definition 58 we have
$. \models c\iota \wedge (p_{l1} + p_{m1}, T - t_1, v_1) \in [\![\tau\ \sigma\iota]\!]$ (F-CAE1)

<u>IH2</u>
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t_1, e'\ \delta\gamma') \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\gamma' = \gamma \cup \{x \mapsto v_1\}$

This means from Definition 58 we have
$\forall t_2 < T . e'\ \delta\gamma' \Downarrow_{t_2} v_f \implies (p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t_1 - t_2, v_f) \in [\![\tau'\ \sigma\iota]\!]$

Since we know that $(\mathsf{clet}\, x = e \mathsf{~in~} e')\ \delta\gamma \Downarrow_t v_f$ therefore from E-CAndE we know that $\exists t_2.e'\ \delta'\gamma \Downarrow_{t_2} v_f$ s.t $t_2 = t - t_1 - 1$
Therefore we have
$(p_{l2} + p_{m2} + p_{l1} + p_{m1}, T - t_1 - t_2, v_f) \in [\![\tau'\ \sigma\iota']\!]$

Since $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$ therefore we get
$(p_l + p_m, T - t, v_f) \in [\![\tau'\ \sigma\iota']\!]$

And we are done.

32. T-fix:

$$\frac{\Psi; \Theta, b; \Delta, b < L; \Omega, x :_{a<I} \tau[(b + 1 + \bigcirc\!\!\!\!\triangle_b^{b+1,a} I)/b]; . \vdash e : \tau \qquad L \geq \bigcirc\!\!\!\!\triangle_b^{0,1} I}{\Psi; \Theta; \Delta; \sum_{b<L} \Omega; . \vdash \mathsf{fix} x.e : \tau[0/b]} \text{ T-fix}$$

Given: $(p_l, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\sum_{b<L} \Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, (\mathsf{fix} x.e)\ \delta\gamma) \in [\![\tau[0/b]\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall\, T' < T, v_f.(\mathsf{fix} x.e)\ \delta\gamma \Downarrow_{T'} v_f \implies (p_m + p_l, T - T', v_f) \in [\![\tau[0/b]\ \sigma\iota]\!]$

This means given some $t < T, v_f$ s.t. $\mathsf{fix} x.e\ \delta\gamma \Downarrow_{T'} v_f$ therefore it suffices to prove that
$(p_l + p_m, T - T', v_f) \in [\![\tau[0/b]\ \sigma\iota]\!]$ (F-FX0)

Also from Lemma 63 we know that
$\exists p'_0, \ldots, p'_{(I-1)}.\ p'_0 + \ldots + p'_{(L-1)} \leq p_m \wedge \forall 0 \leq i < L.(p_i, \delta) \in [\![\Omega[i/a]]\!]_{\mathcal{E}}$

We define

$$\begin{array}{lcl} p_N(leaf) & \triangleq & p'_{leaf} \\ p_N(t) & \triangleq & p'_t + (\sum_{a<I(t)} p_N((t + 1 + \bigcirc\!\!\!\!\triangle_b^{t+1,a} I(b)))) \end{array}$$

<u>Claim</u>
$\forall 0 \leq t < L.\ (p_N(t), T, e\ \delta'\gamma) \in [\![\tau[t/b]\ \sigma\iota]\!]_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto \mathsf{fix} x.e\delta\}$

This means given some t it suffices to prove
$(p_N(t), T, e\ \delta'\gamma) \in [\![\tau[t/b]\ \sigma\iota]\!]_{\mathcal{E}}$

We prove this by induction on t
<u>Base case</u>: when t is a leaf node (say l)
It suffices to prove that $(p'_l, T, e\ \delta'\gamma) \in [\![\tau[l/b]\ \sigma\iota]\!]_{\mathcal{E}}$

We know that $I(l) = 0$ therefore from IH (of the outer induction) we get the desired

<u>Inductive case</u>: when t is some arbitrary non-leaf node
From IH we know that
$\forall a < I(t).(p_N(t'), T, e\ \delta'\gamma) \in [\![\tau[t'/b]\ \sigma\iota]\!]_{\mathcal{E}}$ where $t' = (t + 1 + \bigcirc\!\!\!\!\triangle_b^{t+1,a} I(b))$

<u>Claim</u>
$\forall \tau'.(p_N(t'), T, e\ \delta'\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$ where $\delta' = \delta \cup \{x \mapsto \mathsf{fix} x.e\delta\} \implies$
$(p_N(t'), T, \mathsf{fix} x.e\ \delta\gamma) \in [\![\tau'\ \sigma\iota]\!]_{\mathcal{E}}$

Proof is trivial

$\square$

Therefore we have
$\forall a < I(t).(p_N(t'), T, \mathsf{fix} x.e\ \delta\gamma) \in [\![\tau[t'/b]\ \sigma\iota]\!]_{\mathcal{E}}$ where $t' = (t + 1 + \bigcirc\!\!\!\!\triangle_b^{t+1,a} I(b))$

Now from the IH of the outer induction we get
$(p'_t + \sum_{a<I} p_N(t'), T, e\ \delta'\gamma) \in [\![\tau[t/b]\ \sigma\iota]\!]_{\mathcal{E}}$

Which means we get the desired i.e
$(p_N(t), T, e\ \delta'\gamma) \in [\![\tau[t/b]\ \sigma\iota]\!]_{\mathcal{E}}$

$\square$

Since we have proved
$\forall 0 \leq t < L.\ (p_N(t), T, e\ \delta'\gamma) \in \llbracket \tau[t/b]\ \sigma\iota \rrbracket_{\mathcal{E}}$
where
$\delta' = \delta \cup \{x \mapsto \mathsf{fix}x.e\}$

Therefore from Definition 58 we have
$\forall 0 \leq t < L.\ \forall\, T''{<}T\ .e\ \delta'\gamma \Downarrow_{T''} v_f \implies (p_N(t), T - T'', v_f) \in \llbracket \tau[t/b]\ \sigma\iota \rrbracket_{\mathcal{E}}$

Instantiating with t with 0 and since we know that $\mathsf{fix}x.e\ \delta\gamma \Downarrow_{T'} v_f$ therefore knwo that $\exists\, T''{<}T'\ .e\ \delta'\gamma \Downarrow_{T''} v_f$ where $T''{=}T'\ -1$
$(p_N(0), T - T'', v_f) \in \llbracket \tau[0/b]\ \sigma\iota \rrbracket_{\mathcal{E}}$
Since $p_N(0) \leq p_m$ therefore $p_N(0) \leq p_l + p_m$
And we get the (F-FX0) from Lemma 61

33. T-ret:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{ret}\, e : \mathbb{M}\, 0\, \tau}\ \text{T-ret}$$

Given: $(p_l, T, \gamma) \in \llbracket \Gamma\sigma\iota \rrbracket_{\mathcal{E}}$, $(p_m, T, \delta) \in \llbracket \Omega\ \sigma\iota \rrbracket_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \mathsf{ret}\, e\ \delta\gamma) \in \llbracket \mathbb{M}\, 0\, \tau\ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t\ {<}T\ .(\mathsf{ret}\, e)\ \delta\gamma \Downarrow (\mathsf{ret}\, e)\ \delta\gamma \implies (p_m + p_l, T\ {-}t, (\mathsf{ret}\, e)\ \delta\gamma) \in \llbracket \mathbb{M}\, 0\, \tau\ \sigma\iota \rrbracket$

Since from E-val we know that $t = 0$ therefore it suffices to prove that
$(p_m + p_l, T, (\mathsf{ret}\, e)\ \delta\gamma) \in \llbracket \mathbb{M}\, 0\, \tau\ \sigma\iota \rrbracket$

From Definition 58 it suffices to prove that
$\forall n', t'\ {<}T, v_f.(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f \implies \exists p'.n' + p' \leq p_l + p_m \wedge (p', T\ {-}t', v_f) \in \llbracket \tau \rrbracket$

This means given some $n', t'\ {<}T, v_f$ s.t. $(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_{t'}^{n'} v_f$ it suffices to prove that
$\exists p'.n' + p' \leq p_l + p_m \wedge (p', T\ {-}t', v_f) \in \llbracket \tau \rrbracket$

From (E-ret) we know that $n' = 0$ therefore we choose p' as $p_l + p_m$ and it suffices to prove that
$(p_l + p_m, T\ {-}t', v_f) \in \llbracket \tau\ \sigma\iota \rrbracket$ (F-R0)

<u>IH</u>
$(p_l + p_m, T, e\ \delta\gamma) \in \llbracket \tau\ \sigma\iota \rrbracket_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1\ {<}T\ .(e)\ \delta\gamma \Downarrow_{t_1} v_f \implies (p_m + p_l, T\ {-}t_1, v_f) \in \llbracket \tau\ \sigma\iota \rrbracket$

Since we know that $(\mathsf{ret}\, e)\ \delta\gamma \Downarrow_{t'}^{0} v_f$ therefore from (E-ret) we know that $\exists t_1 < t.e\ \delta\gamma \Downarrow_{t''} v_f$ s.t $t_1 + 1 = t'$

Therefore we have $(p_m + p_l, T\ {-}t_1, v_f) \in \llbracket \tau\ \sigma\iota \rrbracket$ and from Lemma 61 we are done

34. T-bind:

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \mathbb{M}\, n_1\, \tau_1 \\ \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}\, n_2\, \tau_2 \qquad \Theta \vdash n_1 : \mathbb{R}^+ \qquad \Theta \vdash n_2 : \mathbb{R}^+\end{array}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2 : \mathbb{M}(n_1 + n_2)\, \tau_2}\ \text{T-bind}$$

Given: $(p_l, T, \gamma) \in \llbracket (\Gamma_1 \oplus \Gamma_2)\sigma\iota \rrbracket_{\mathcal{E}}$, $(p_m, T, \delta) \in \llbracket (\Omega_1 \oplus \Omega_2)\ \sigma\iota \rrbracket_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \in \llbracket \mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota \rrbracket_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$\forall t\ {<}T, v.(\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_t v \implies (p_m + p_l, T\ {-}t, (\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma) \in \llbracket \mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota \rrbracket$

This means given some $t\ {<}T, v$ s.t. $(\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_t v$ and from E-val we know that $v = (\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma$ and $t = 0$. It suffices to prove that
$(p_m + p_l, T, (\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma) \in \llbracket \mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota \rrbracket$

This means from Definition 58 it suffices to prove that
$\forall s', t'\ {<}T, v_f.(\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f \implies \exists p'.s' + p' \leq p_l + p_m + n \wedge (p', T\ {-}t', v_f) \in \llbracket \tau_2\ \sigma\iota \rrbracket$

This means given some $s', t'\ {<}T, v_f$ s.t $(\mathsf{bind}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f$ and we need to prove that
$\exists p'.s' + p' \leq p_l + p_m + n \wedge (p', T\ {-}t', v_f) \in \llbracket \tau_2\ \sigma\iota \rrbracket$ (F-B0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in \llbracket (\Gamma_1)\sigma\iota \rrbracket_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in \llbracket (\Gamma_2)\sigma\iota \rrbracket_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t $(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e_1\ \delta\gamma) \in [\![\mathbb{M}(n_1)\,\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it means we have
$\forall t_1 < T\ .(e_1)\ \delta\gamma \Downarrow_{t_1} (e_1)\ \delta\gamma \implies (p_{m1} + p_{l1}, T - t_1, (e_1)\ \delta\gamma) \in [\![\mathbb{M}(n_1)\,\tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_{t'}^{s'} v_f$ therefore from E-bind we know that $\exists t_1 < t', v_{m1}.(e_1)\ \delta\gamma \Downarrow (e_1)\ \delta\gamma$.
Since $t_1 < t' < T$, therefore we have
$(p_{m1} + p_{l1}, T - t_1, (e_1)\ \delta\gamma) \in [\![\mathbb{M}(n_1)\,\tau_1\ \sigma\iota]\!]$

This means from Definition 58 we are given that
$\forall t_1' < T - t_1.(e_1\ \delta\gamma) \Downarrow_{t_1'}^{s_1} v_1 \implies \exists p_1'.s_1 + p_1' \leq p_{l1} + p_{m1} + n_1 \wedge (p_1', T - t_1 - t_1', v_1) \in [\![\tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_{t'} v_f$ therefore from E-bind we know that $\exists t_1' < t' - t_1.(e_1)\ \delta\gamma \Downarrow_{t_1'}^{s_1} v_1$.
This means we have
$\exists p_1'.s_1 + p_1' \leq p_{l1} + p_{m1} + n_1 \wedge (p_1', T - t_1 - t_1', v_1) \in [\![\tau_1\ \sigma\iota]\!]$ (F-B1)

<u>IH2</u>
$(p_{l2} + p_{m2} + p_1', T - t_1 - t_1', e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_2)\,\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it means we have
$\forall t_2 < T - t_1 - t_1'.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} (e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \implies (p_{m2} + p_{l2} + p_1' + n_2, T - t_1 - t_1' - t_2, (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_2)\,\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow - \Downarrow_t^- v_f$ therefore from E-bind we know that $\exists t_2 < t' - t_1 - t_1'.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}$.
Since $t_2 < t' - t_1 - t_1' < T - t_1 - t_1'$ therefore we have
$(p_{m2} + p_{l2} + p_1' + n_2, T - t_1 - t_1' - t_2, (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_2)\,\tau_2\ \sigma\iota]\!]$

This means from Definition 58 we are given that
$\forall t_2' < T - t_1 - t_1' - t_2.(e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \Downarrow_{t_2'}^{s_2} v_2 \implies \exists p_2'.s_2 + p_2' \leq p_{l2} + p_{m2} + p_1' + n_2 \wedge (p_2', T - t_1 - t_1' - t_2 - t_2', v_2) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{bind}\, x = e_1 \ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow - \Downarrow_{t'}^- v_f$ therefore from E-bind we know that $\exists t_2' < t' - t_1 - t_1' - t_2, s_2, v_2.v_{m2} \Downarrow_{t_2'}^{s_2} v_2$.
This means we have
$\exists p_2'.s_2 + p_2' \leq p_{l2} + p_{m2} + p_1' + n_2 \wedge (p_2', T - t_1 - t_1' - t_2 - t_2', v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-B2)

In order to prove (F-B0) we choose p' as p_2' and it suffices to prove
(a) $s' + p_2' \leq p_l + p_m + n$:
Since from (F-B2) we know that
$s_2 + p_2' \leq p_{l2} + p_{m2} + p_1' + n_2$

Adding s_1 on both sides we get
$s_1 + s_2 + p_2' \leq p_{l2} + p_{m2} + s_1 + p_1' + n_2$

Since from (F-B1) we know that
$s_1 + p_1' \leq p_{l1} + p_{m1} + n_1$

therefore we also have
$s_1 + s_2 + p_2' \leq p_{l2} + p_{m2} + p_{l1} + p_{m1} + n_1 + n_2$

And finally since we know that $n = n_1 + n_2$, $s' = s_1 + s_2$, $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$ therefore we get the desired
(b) $(p_2', T - t_1 - t_1' - t_2 - t_2', v_f) \in [\![\tau_2\ \sigma\iota]\!]$:
From E-bind we know that $v_f = v_2$ therefore we get the desired from (F-B2)

35. T-tick:

$$\frac{\Theta \vdash n : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \uparrow^n : \mathbb{M}\, n\, \mathbf{1}}\ \text{T-tick}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$

To prove: $(p_l + p_m, T, \uparrow^n\ \delta\gamma) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$(\uparrow^n)\ \delta\gamma \Downarrow_0 (\uparrow^n)\ \delta\gamma \implies (p_m + p_l, T, (\uparrow^n)\ \delta\gamma) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]$

It suffices to prove that
$(p_m + p_l, T, (\uparrow^n)\ \delta\gamma) \in [\![\mathbb{M}\, n\, \mathbf{1}\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall t' < T, n'.(\uparrow^n)\ \delta\gamma \Downarrow_{t'}^{n'} () \implies \exists p'.n' + p' \leq p_l + p_m + n \wedge (p', T - t', ()) \in [\![\mathbf{1}]\!]$

This means given some $t' < T, n'$ s.t. $(\uparrow^n)\ \delta\gamma \Downarrow_{t'}^{n'} ()$ it suffices to prove that
$\exists p'.n' + p' \leq p_l + p_m + n \wedge (p', T - t', ()) \in [\![\mathbf{1}]\!]$

From (E-tick) we know that $n' = n$ therefore we choose p' as $p_l + p_m$ and it suffices to prove that
$(p_l + p_m, T - t', ()) \in [\![\mathbf{1}]\!]$

We get this directly from Definition 58

36. T-release:

$$\frac{\begin{array}{c}\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : [n_1]\, \tau_1 \\ \Psi; \Theta; \Delta; \Omega_2; \Gamma_2, x : \tau_1 \vdash e_2 : \mathbb{M}(n_1 + n_2)\, \tau_2 \qquad \Theta \vdash n_1 : \mathbb{R}^+ \qquad \Theta \vdash n_2 : \mathbb{R}^+\end{array}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash \mathsf{release}\, x = e_1\ \mathsf{in}\ e_2 : \mathbb{M}\, n_2\, \tau_2}\ \text{T-release}$$

Given: $(p_l, T, \gamma) \in [\![(\Gamma_1 \oplus \Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![(\Omega_1 \oplus \Omega_2)\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \mathsf{release}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_0 (\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \implies (p_m + p_l, (\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

This means given $(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow_0 (\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma$ it suffices to prove that
$(p_m + p_l, (\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma) \in [\![\mathbb{M}(n_2)\, \tau_2\ \sigma\iota]\!]$

This means from Definition 58 it suffices to prove that
$\forall t' < T, v_f, s'.(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f \implies \exists p'.s' + p' \leq p_l + p_m + n_2 \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$

This means given some $t' < T, v_f, s'$ s.t. $(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2\ \delta\gamma) \Downarrow_{t'}^{s'} v_f$ and we need to prove that
$\exists p'.s' + p' \leq p_l + p_m + n_2 \wedge (p', T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-R0)

From Definition 59 and Definition 57 we know that $\exists p_{l1}, p_{l2}.p_{l1} + p_{l2} = p_l$ s.t
$(p_{l1}, T, \gamma) \in [\![(\Gamma_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{l2}, T, \gamma) \in [\![(\Gamma_2)\sigma\iota]\!]_{\mathcal{E}}$

Similarly from Definition 59 and Definition 56 we also know that $\exists p_{m1}, p_{m2}.p_{m1} + p_{m2} = p_m$ s.t
$(p_{m1}, T, \delta) \in [\![(\Omega_1)\sigma\iota]\!]_{\mathcal{E}}$ and $(p_{m2}, T, \delta) \in [\![(\Omega_2)\sigma\iota]\!]_{\mathcal{E}}$

<u>IH1</u>
$(p_{l1} + p_{m1}, T, e_1\ \delta\gamma) \in [\![[n_1]\, \tau_1\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it means we have
$\forall t_1 < T\ .(e_1)\ \delta\gamma \Downarrow_{t_1} v_1 \implies (p_{m1} + p_{l1}, T - t_1, v_1) \in [\![[n_1]\, \tau_1\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow - \Downarrow_{t'}^{-} v_f$ therefore from E-rel we know that $\exists t_1 < t'.(e_1)\ \delta\gamma \Downarrow_{t_1} v_1$. This means we have
$(p_{m1} + p_{l1}, T - t_1, v_1) \in [\![[n_1]\, \tau_1\ \sigma\iota]\!]$

This means from Definition 58 we have
$\exists p'_1.p'_1 + n_1 \leq p_{l1} + p_{m1} \wedge (p'_1, T - t_1, v_1) \in [\![\tau_1]\!]$ (F-R1)

<u>IH2</u>
$(p_{l2} + p_{m2} + p'_1, T - t_1, e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it means we have
$\forall t_2 < T - t_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} (e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \implies (p_{m2} + p_{l2} + p'_1 + n_2, T - t_1 - t_2, (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1\ \mathsf{in}\ e_2)\ \delta\gamma \Downarrow - \Downarrow_{t'}^{-} v_f$ therefore from E-rel we know that
$\exists t_2 < t - t_1.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow_{t_2} (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}$. This means we have
$(p_{m2} + p_{l2} + p'_1 + n_2, T - t_1 - t_2, (e_2)\ \delta\gamma \cup \{x \mapsto v_1\}) \in [\![\mathbb{M}(n_1 + n_2)\, \tau_2\ \sigma\iota]\!]$

This means from Definition 58 we are given that

$\forall t'_2 < T - t_1 - t_2.(e_2\ \delta\gamma \cup \{x \mapsto v_1\}) \Downarrow^{s_2}_{t'_2} v_2 \implies \exists p'_2.s_2 + p'_2 \leq p_{l2} + p_{m2} + p'_1 + n_1 + n_2 \wedge (p'_2, T - t_1 - t_2 - t'_2, v_2) \in [\![\tau_2\ \sigma\iota]\!]$

Since we know that $(\mathsf{release}\, x = e_1 \mathsf{\ in\ } e_2)\ \delta\gamma \Downarrow - \Downarrow^{-}_{t'} v_f$ therefore from E-rel we know that $\exists t'_2.(e_2)\ \delta\gamma \cup \{x \mapsto v_1\} \Downarrow^{s_2} v_2$ s.t. $t'_2 = t' - t_1 - t_2 - 1$
Since $t'_2 = t' - t_1 - t_2 < T - t_1 - t_2 - 1 < T - t_1 - t_2$, therefore we have
$\exists p'_2.s_2 + p'_2 \leq p_{l2} + p_{m2} + p'_1 + n_1 + n_2 \wedge (p'_2, T - t_1 - t_2 - t'_2, v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-R2)

In order to prove (F-R0) we choose p' as p'_2 and it suffices to prove

(a) $s' + p'_2 \leq p_l + p_m + n_2$:
Since from (F-R2) we know that
$s_2 + p'_2 \leq p_{l2} + p_{m2} + p'_1 + n_1 + n_2$

Since from (F-R1) we know that
$p'_1 + n_1 \leq p_{l1} + p_{m1}$

therefore we also have
$s_2 + p'_2 \leq p_{l2} + p_{m2} + p_{l1} + p_{m1} + n_2$

And finally since we know that $s' = s_2$, $p_l = p_{l1} + p_{l2}$ and $p_m = p_{m1} + p_{m2}$ therefore we get the desired

(b) $(p'_2, T - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]$:
From E-rel we know that $v_f = v_2$ therefore we get the desired from (F-R2) and Lemma 61

37. T-store:

$$\frac{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash e : \tau \qquad \Theta \vdash n : \mathbb{R}^+}{\Psi; \Theta; \Delta; \Omega; \Gamma \vdash \mathsf{store}\, e : \mathbb{M}\, n\, ([n]\, \tau)} \text{T-store}$$

Given: $(p_l, T, \gamma) \in [\![\Gamma\sigma\iota]\!]_{\mathcal{E}}$, $(p_m, T, \delta) \in [\![\Omega\ \sigma\iota]\!]_{\mathcal{E}}$ and $\models \Delta\ \iota$
To prove: $(p_l + p_m, T, \mathsf{store}\, e\ \delta\gamma) \in [\![\mathbb{M}\, n\, ([n]\, \tau)\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 it suffices to prove that
$(\mathsf{store}\, e)\ \delta\gamma \Downarrow (\mathsf{store}\, e)\ \delta\gamma \implies (p_m + p_l, T, (\mathsf{store}\, e)\ \delta\gamma) \in [\![\mathbb{M}\, n\, ([n]\, \tau)\ \sigma\iota]\!]$

It suffices to prove that
$(p_m + p_l, T, (\mathsf{store}\, e)\ \delta\gamma) \in [\![\mathbb{M}\, n\, ([n]\, \tau)\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\forall t' < T, v_f, n'.(\mathsf{store}\, e)\ \delta\gamma \Downarrow^{n'}_{t'} v_f \implies \exists p'.n' + p' \leq p_l + p_m + n \wedge (p', T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]$

This means given some $t' < T, v_f, n'$ s.t. $(\mathsf{store}\, e)\ \delta\gamma \Downarrow^{n'}_{t'} v_f$ it suffices to prove that
$\exists p'.n' + p' \leq p_l + p_m + n \wedge (p', T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]$

From (E-store) we know that $n' = 0$ therefore we choose p' as $p_l + p_m + n$ and it suffices to prove that
$(p_l + p_m + n, T - t', v_f) \in [\![[n]\, \tau\ \sigma\iota]\!]_{\mathcal{E}}$

This further means that from Definition 58 we have
$\exists p''.p'' + n \leq p_l + p_m + n \wedge (p'', T - t', v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}\}$

We choose p'' as $p_l + p_m$ and it suffices to prove that
$(p_l + p_m, T - t', v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}\}$ (F-S0)

<u>IH</u>
$(p_l + p_m, T, e\ \delta\gamma) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we have
$\forall t_1 < T\ .(e)\ \delta\gamma \Downarrow_{t_1} v_f \implies (p_m + p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$

Since we know that $(\mathsf{store}\, e)\ \delta\gamma \Downarrow - \Downarrow^{0}_{t'} v_f$ therefore from (E-store) we know that $\exists t_1 < t'.e\ \delta\gamma \Downarrow_{t_1} v_f$ where $t_1 + 1 = t'$

Therefore from Lemma 61 we get $(p_m + p_l, T - t_1, v_f) \in [\![\tau\ \sigma\iota]\!]_{\mathcal{E}}$ and we are done

□

Lemma 65 (Γ Subtyping: domain containment)**.** $\forall p, \gamma, \Gamma_1, \Gamma_2.$
$\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2 \implies \forall x : \tau \in \Gamma_2.\ x : \tau' \in \Gamma_1 \wedge \Psi; \Theta; \Delta \vdash \tau' <: \tau$

Proof. Proof by induction on $\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta; \Delta \vdash \Gamma_1 <: .}\text{ sub-lBase}$$

To prove: $\forall x : \tau' \in (.).x : \tau \in \Gamma_1 \land \Psi; \Theta; \Delta \vdash \tau' <: \tau$
Trivial

2. sub-lInd:

$$\frac{x : \tau' \in \Gamma_1 \qquad \Psi; \Theta; \Delta \vdash \tau' <: \tau \qquad \Psi; \Theta; \Delta \vdash \Gamma_1/x <: \Gamma_2}{\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2, x : \tau}\text{ sub-lBase}$$

To prove: $\forall y : \tau \in \Gamma_2.y : \tau \in \Gamma_1 \land \Psi; \Theta; \Delta \vdash \tau' <: \tau$

This means given some $y : \tau \in (\Gamma_2, x : \tau)$ it suffices to prove that
$y : \tau \in \Gamma_1 \land \Psi; \Theta; \Delta \vdash \tau' <: \tau$
The follwing cases arise:
- $y = x$:
In this case we are given that $x : \tau' \in \Gamma_1 \land \Psi; \Theta; \Delta \vdash \tau' <: \tau$
Therefore we are done
- $y \neq x$:
Since we are given that $\Psi; \Theta; \Delta \vdash \Gamma_1/x <: \Gamma_2$ therefore we get the desired from IH

□

Lemma 66 (Ω Subtyping: domain containment)**.** $\forall p, \gamma, \Omega_1, \Omega_2.$
$\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2 \implies$
$\forall x :_{a<I} \tau \in \Omega_2.\ x :_{a<J} \tau' \in \Omega_1 \land \Psi; \Theta; \Delta \vdash I \leq J \land \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau$

Proof. Proof by induction on $\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta; \Delta \vdash \Omega <: .}\text{ sub-mBase}$$

To prove: $\forall x :_{a<I} \tau \in (.).x :_{a<J} \tau' \in \Omega_1 \land \Psi; \Theta; \Delta \vdash I \leq J \land \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau$
Trivial

2. sub-lInd:

$$\frac{x :_{a<J} \tau' \in \Omega_1 \qquad \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau \qquad \Theta; \Delta \vdash I \leq J \qquad \Psi; \Theta; \Delta \vdash \Omega_1/x <: \Omega_2}{\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2, x :_{a<I} \tau}\text{ sub-mInd}$$

To prove: $\forall y :_{a<I} \tau \in \Omega_2.y :_{a<J} \tau' \in \Omega_1 \land \Psi; \Theta; \Delta \vdash I \leq J \land \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau$

This means given some $y :_{a<I} \tau \in (\Omega_2, x :_{a<I} \tau)$ it suffices to prove that
$y :_{a<J} \tau' \in \Omega_1 \land \Psi; \Theta; \Delta \vdash I \leq J \land \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau$
The follwing cases arise:
- $y = x$:
In this case we are given that
$x :_{a<J} \tau' \in \Omega_1 \land \Psi; \Theta; \Delta \vdash I \leq J \land \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau$
Therefore we are done
- $y \neq x$:
Since we are given that $\Psi; \Theta; \Delta \vdash \Omega_1/x <: \Omega_2$ therefore we get the desired from IH

□

Lemma 67 (Γ subtyping lemma)**.** $\forall p, \gamma, \Gamma_1, \Gamma_2, \sigma, \iota.$
$\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2 \implies [\![\Gamma_1 \sigma \iota]\!] \subseteq [\![\Gamma_2 \sigma \iota]\!]$

Proof. Proof by induction on $\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta; \Delta \vdash \Gamma <: .}\text{ sub-lBase}$$

To prove: $\forall (p, T, \gamma) \in [\![\Gamma_1 \sigma \iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Gamma_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$

From Definition 59 it suffices to prove that
$\exists f : \mathcal{V}ars \to \mathcal{P}ots. (\forall x \in dom(.). (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}}) \wedge (\sum_{x \in dom(.)} f(x) \leq p)$
We choose f as a constant function $f' - = 0$ and we get the desired

2. sub-lInd:

$$\frac{x : \tau' \in \Gamma_1 \qquad \Psi; \Theta; \Delta \vdash \tau' <: \tau \qquad \Psi; \Theta; \Delta \vdash \Gamma_1/x <: \Gamma_2}{\Psi; \Theta; \Delta \vdash \Gamma_1 <: \Gamma_2, x : \tau} \text{ sub-lBase}$$

To prove: $\forall (p, T, \gamma) \in [\![\Gamma_1 \sigma \iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![\Gamma_2, x : \tau]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Gamma_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![\Gamma_2, x : \tau]\!]_{\mathcal{E}}$

This means from Definition 59 we are given that
$\exists f : \mathcal{V}ars \to \mathcal{P}ots.$
$(\forall x \in dom(\Gamma_1). (f(x), T, \gamma(x)) \in [\![\Gamma(x)]\!]_{\mathcal{E}})$ (L0)
$(\sum_{x \in dom(\Gamma_1)} f(x) \leq p)$ (L1)

Similarly from Definition 59 it suffices to prove that
$\exists f' : \mathcal{V}ars \to \mathcal{P}ots. (\forall y \in dom(\Gamma_2, x : \tau). (f'(y), T, \gamma(y)) \in [\![\Gamma(y)]\!]_{\mathcal{E}}) \wedge (\sum_{y \in dom(\Gamma_2, x:\tau)} f'(y) \leq p)$

We choose f' as f and it suffices to prove that

(a) $\forall y \in dom(\Gamma_2, x : \tau). (f(y), T, \gamma(y)) \in [\![\Gamma(y)]\!]_{\mathcal{E}}$:
This means given some $y \in dom(\Gamma_2, x : \tau)$ it suffices to prove that
$(f(y), T, \gamma(y)) \in [\![\tau_2]\!]_{\mathcal{E}}$ where say $\Gamma(y) = \tau_2$

From Lemma 65 we know that
$y : \tau_1 \in \Gamma_1 \wedge \Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2$

By instantiating (L0) with the given y
$(f(y), T, \gamma(y)) \in [\![\tau_1]\!]_{\mathcal{E}}$

Finally from Lemma 70 we also get $(f(y), T, \gamma(y)) \in [\![\tau_2]\!]_{\mathcal{E}}$
And we are done

(b) $(\sum_{y \in dom(\Gamma_2, x:\tau)} f(y) \leq p)$:
From (L1) we know that $(\sum_{x \in dom(\Gamma_1)} f(x) \leq p)$ and since from Lemma 65 we know that $dom(\Gamma_2, x : \tau) \subseteq dom(\Gamma_1)$ therefore we also have
$(\sum_{y \in dom(\Gamma_2, x:\tau)} f(y) \leq p)$

□

Lemma 68 (Ω subtyping lemma)**.** $\forall p, \gamma, \Omega_1, \Omega_2, \sigma, \iota.$
$\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2 \implies [\![\Omega_1 \sigma \iota]\!] \subseteq [\![\Omega_2 \sigma \iota]\!]$

Proof. Proof by induction on $\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2$

1. sub-lBase:

$$\frac{}{\Psi; \Theta; \Delta \vdash \Omega <: .} \text{ sub-mBase}$$

To prove: $\forall (p, T, \gamma) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![.]\!]_{\mathcal{E}}$
From Definition 59 it suffices to prove that
$\exists f : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots. (\forall (x :_{a<I} \tau) \in ..\forall 0 \leq i < I. (f\ x\ i, T, \delta(x)) \in [\![\tau[i/a]]\!]_{\mathcal{E}}) \wedge (\sum_{x :_{a<I} \tau \in .} \sum_{0 \leq i < I} f\ x\ i) \leq p$
We choose f as a constant function $f' - = 0$ and we get the desired

2. sub-lInd:

$$\frac{x :_{a<J} \tau' \in \Omega_1 \qquad \Psi; \Theta, a; \Delta, a < I \vdash \tau' <: \tau \qquad \Theta; \Delta \vdash I \leq J \qquad \Psi; \Theta; \Delta \vdash \Omega_1/x <: \Omega_2}{\Psi; \Theta; \Delta \vdash \Omega_1 <: \Omega_2, x :_{a<I} \tau} \text{ sub-mInd}$$

To prove: $\forall (p, T, \gamma) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}.(p, T, \gamma) \in [\![\Omega_2, x : \tau]\!]_{\mathcal{E}}$

This means given some $(p, T, \gamma) \in [\![\Omega_1 \sigma \iota]\!]_{\mathcal{E}}$ it suffices to prove that $(p, T, \gamma) \in [\![\Omega_2, x : \tau]\!]_{\mathcal{E}}$

This means from Definition 59 we are given that
$\exists f : \mathcal{V}ars \to \mathcal{P}ots.$
$(\forall (x :_{a<I} \tau) \in \Omega_1. \forall 0 \leq i < I. (f\ x\ i, T, \delta(x)) \in [\![\tau[i/a]]\!]_{\mathcal{E}})$ (L0)

$(\sum_{x:_{a<I}\tau\in\Omega_1}\sum_{0\leq i<I} f\ x\ i) \leq p$ (L1)

Similarly from Definition 59 it suffices to prove that
$\exists f' : \mathcal{V}ars \to \mathcal{I}ndices \to \mathcal{P}ots.(\forall (y :_{a<I_y} \tau_y) \in \Omega_2, x:\tau.\forall 0 \leq i < I_y.(f\ x\ i, T, \delta(y)) \in [\![\tau_y[i/a]]\!]_{\mathcal{E}}) \wedge (\sum_{y:_{a<I_y}\tau\in\Omega_2,x:\tau}\sum_{0\leq i<I_y} f'\ y\ i) \leq p$

We choose f' as f and it suffices to prove that

(a) $(\forall (y :_{a<I_y} \tau_y) \in \Omega_2, x:\tau.\forall 0 \leq i < I_y.(f\ x\ i, T, \delta(y)) \in [\![\tau_y[i/a]]\!]_{\mathcal{E}})$:
This means given some $(y :_{a<I} \tau_y) \in \Omega_2, x:\tau$ and some $0 \leq i < I_y$ it suffices to prove that $(f\ x\ i, T, \delta(y)) \in [\![\tau_y[i/a]]\!]_{\mathcal{E}})$

From Lemma 65 we know that
$y :_{a<J_y} \tau_1 \in \Omega_1 \wedge \Psi;\Theta;\Delta \vdash I_y \leq J_y \wedge \Psi;\Theta,a;\Delta,a<I_y \vdash \tau_1 <: \tau_y$

Instantiating (L0) with the given y and i we get
$(f\ x\ i, T, \delta(y)) \in [\![\tau_1[i/a]]\!]_{\mathcal{E}}$

Finally using Lemma 70 we also get
$(f\ x\ i, T, \delta(y)) \in [\![\tau_y[i/a]]\!]_{\mathcal{E}}$

(b) $(\sum_{y:_{a<I_y}\tau_y\in\Omega_2,x:\tau}\sum_{0\leq i<I_y} f'\ y\ i) \leq p$:
From Lemma 66 we know that
$\forall y :_{a<I_y} \tau_y \in (\Omega_2, x:\tau).y :_{a<J_y} \tau_1 \in \Omega_1 \wedge \Psi;\Theta;\Delta \vdash I_y \leq J_y \wedge \Psi;\Theta,a;\Delta,a<I_y \vdash \tau_1 <: \tau_y$
And since from (L1) we know that $(\sum_{x:_{a<I}\tau\in\Omega_1}\sum_{0\leq i<I} f\ x\ i) \leq p$ therefore we also have $(\sum_{y:_{a<I_y}\tau_y\in\Omega_2,x:\tau}\sum_{0\leq i<I_y} f'\ y\ i) \leq p$

□

Lemma 69 (Value subtyping lemma)**.** $\forall \Psi,\Theta,\Delta,\tau \in Type, \tau', \sigma, \iota.$
$\Psi;\Theta;\Delta \vdash \tau <: \tau' \wedge . \models \Delta\iota \implies [\![\tau\ \sigma\iota]\!] \subseteq [\![\tau'\ \sigma\iota]\!]$

Proof. Proof by induction on the $\Psi;\Theta;\Delta \vdash \tau <: \tau'$ relation

1. sub-refl:

$$\frac{}{\Psi;\Theta;\Delta \vdash \tau <: \tau}\text{ sub-refl}$$

To prove: $\forall (p,T,v) \in [\![\tau\ \sigma\iota]\!] \implies (p,T,v) \in [\![\tau\ \sigma\iota]\!]$

Trivial

2. sub-arrow:

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1' <: \tau_1 \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \multimap \tau_2 <: \tau_1' \multimap \tau_2'}\text{ sub-arrow}$$

To prove: $\forall (p,T,\lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!] \implies (p,T,\lambda x.e) \in [\![(\tau_1' \multimap \tau_2')\ \sigma\iota]\!]$

This means given some $(p,T,\lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]$ we need to prove $(p,T,\lambda x.e) \in [\![(\tau_1' \multimap \tau_2')\ \sigma\iota]\!]$

From Definition 58 we are given that
$\forall p', e', T'{<}T\ .(p',T',e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p+p', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SL0)

Also from Definition 58 it suffices to prove that
$\forall p', e', T''{<}T\ .(p',T'',e') \in [\![\tau_1'\ \sigma\iota]\!]_{\mathcal{E}} \implies (p+p', T'', e[e'/x]) \in [\![\tau_2'\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some p', e', T'' s.t $(p',T'',e') \in [\![\tau_1'\ \sigma\iota]\!]_{\mathcal{E}}$ we need to prove $(p+p', T'', e[e'/x]) \in [\![\tau_2'\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SL1)

Since $\Psi;\Theta;\Delta \vdash \tau_1' <: \tau_1$ therefore from Lemma 70 we know that given some $(p',T'',e'') \in [\![\tau_1'\ \sigma\iota]\!]$ we also have $(p',T'',e'') \in [\![\tau_1\ \sigma\iota]\!]$

Therefore instantiating (F-SL0) with p', e'', T'' we get
$(p+p', T'', e[e''/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Lemma 70 we get the desired

3. sub-tensor:

$$\frac{\Psi;\Theta;\Delta \vdash \tau_1 <: \tau_1' \qquad \Psi;\Theta;\Delta \vdash \tau_2 <: \tau_2'}{\Psi;\Theta;\Delta \vdash \tau_1 \otimes \tau_2 <: \tau_1' \otimes \tau_2'}\text{ sub-tensor}$$

To prove: $\forall (p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!] \implies (p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \in [\![(\tau_1' \otimes \tau_2')\ \sigma\iota]\!]$

This means given $(p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \in [\![(\tau_1 \otimes \tau_2)\ \sigma\iota]\!]$
It suffices prove that
$(p, T, \langle\!\langle v_1, v_2\rangle\!\rangle) \in [\![(\tau_1' \otimes \tau_2')\ \sigma\iota]\!]$

This means from Definition 58 we are given that
$\exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v_1) \in [\![\tau_1\ \sigma\iota]\!] \wedge (p_2, T, v_2) \in [\![\tau_2\ \sigma\iota]\!]$

Also from Definition 58 it suffices to prove that
$\exists p_1', p_2'.p_1' + p_2' \leq p \wedge (p_1', T, v_1) \in [\![\tau_1'\ \sigma\iota]\!] \wedge (p_2', T, v_2) \in [\![\tau_2'\ \sigma\iota]\!]$

<u>IH1</u> $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_1')\ \sigma\iota]\!]$
<u>IH2</u> $[\![(\tau_2)\ \sigma\iota]\!] \subseteq [\![(\tau_2')\ \sigma\iota]\!]$

Instantiating p_1', p_2' with p_1, p_2 we get the desired from IH1 and IH2

4. sub-with:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \mathbin{\&} \tau_2 <: \tau_1' \mathbin{\&} \tau_2'} \text{ sub-with}$$

To prove: $\forall (p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1 \mathbin{\&} \tau_2)\ \sigma\iota]\!] \implies (p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1' \mathbin{\&} \tau_2')\ \sigma\iota]\!]$

This means given $(p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1 \mathbin{\&} \tau_2)\ \sigma\iota]\!]$
It suffices prove that
$(p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1' \mathbin{\&} \tau_2')\ \sigma\iota]\!]$

This means from Definition 58 we are given that
$(p, T, v_1) \in [\![\tau_1\ \sigma\iota]\!] \wedge (p, T, v_2) \in [\![\tau_2\ \sigma\iota]\!]$ (F-SW0)

Also from Definition 58 it suffices to prove that
$(p, T, v_1) \in [\![\tau_1'\ \sigma\iota]\!] \wedge (p, T, v_2) \in [\![\tau_2'\ \sigma\iota]\!]$

<u>IH1</u> $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_1')\ \sigma\iota]\!]$
<u>IH2</u> $[\![(\tau_2)\ \sigma\iota]\!] \subseteq [\![(\tau_2')\ \sigma\iota]\!]$

We get the desired from (F-SW0), IH1 and IH2

5. sub-sum:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_1' \qquad \Psi; \Theta; \Delta \vdash \tau_2 <: \tau_2'}{\Psi; \Theta; \Delta \vdash \tau_1 \oplus \tau_2 <: \tau_1' \oplus \tau_2'} \text{ sub-sum}$$

To prove: $\forall (p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!] \implies (p, T, \langle v_1, v_2\rangle) \in [\![(\tau_1' \oplus \tau_2')\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![(\tau_1 \oplus \tau_2)\ \sigma\iota]\!]$
It suffices prove that
$(p, T, v) \in [\![(\tau_1' \oplus \tau_2')\ \sigma\iota]\!]$

This means from Definition 58 2 cases arise

(a) $v = \mathsf{inl}(v')$:
This means from Definition 58 we have $(p, T, v') \in [\![\tau_1\ \sigma\iota]\!]$ (F-SS0)

Also from Definition 58 it suffices to prove that
$(p, T, v') \in [\![\tau_1'\ \sigma\iota]\!]$

<u>IH</u> $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_1')\ \sigma\iota]\!]$

We get the desired from (F-SS0), IH

(b) $v = \mathsf{inr}(v')$:
Symmetric reasoning as in the inl case

6. sub-potential:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Psi; \Theta; \Delta \vdash n' \leq n}{\Psi; \Theta; \Delta \vdash [n]\,\tau <: [n']\,\tau'} \text{ sub-potential}$$

To prove: $\forall (p, T, v) \in [\![[n]\,\tau\ \sigma\iota]\!].(p, T, v) \in [\![[n']\,\tau'\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![[n]\,\tau\ \sigma\iota]\!]$ and we need to prove

$(p, T, v) \in [\![[n'] \, \tau' \; \sigma\iota]\!]$

This means from Definition 58 we are given
$\exists p'.p' + n \leq p \wedge (p', T, v) \in [\![\tau \; \sigma\iota]\!]$ (F-SP0)

And we need to prove
$\exists p''.p'' + n' \leq p \wedge (p'', T, v) \in [\![\tau' \; \sigma\iota]\!]$ (F-SP1)

In order to prove (F-SP1) we choose p'' as p'

Since from (F-SP0) we know that $p' + n \leq p$ and we are given that $n' \leq n$ therefore we also have $p' + n' \leq p$

<u>IH</u> $(p', T, v) \in [\![\tau' \; \sigma\iota]\!]$

$(p', T, v) \in [\![\tau' \; \sigma\iota]\!]$ we get directly from IH

7. sub-monad:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau' \qquad \Psi; \Theta; \Delta \vdash n \leq n'}{\Psi; \Theta; \Delta \vdash \mathbb{M}\, n\, \tau <: \mathbb{M}\, n'\, \tau'} \text{ sub-monad}$$

To prove: $\forall (p, T, v) \in [\![\mathbb{M}\, n\, \tau \; \sigma\iota]\!].(p, T, v) \in [\![\mathbb{M}\, n'\, \tau' \; \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\mathbb{M}\, n\, \tau \; \sigma\iota]\!]$ and we need to prove
$(p, T, v) \in [\![\mathbb{M}\, n'\, \tau' \; \sigma\iota]\!]$

This means from Definition 58 we are given
$\forall t' < T, n_1, v'.v \Downarrow_{t'}^{n_1} v' \implies \exists p'.n_1 + p' \leq p + n \wedge (p', T - t', v') \in [\![\tau \; \sigma\iota]\!]$ (F-SM0)

Again from Definition 58 we need to prove that
$\forall t'' < T, n_2, v''.v \Downarrow_{t''}^{n_2} v'' \implies \exists p''.n_1 + p'' \leq p + n' \wedge (p'', T - t'', v') \in [\![\tau' \; \sigma\iota]\!]$

This means given some $t'' < T, n_2, v''$ s.t. $v \Downarrow_{t''}^{n_2} v''$ it suffices to prove that
$\exists p''.n_1 + p'' \leq p + n' \wedge (p'', T - t'', v') \in [\![\tau' \; \sigma\iota]\!]$ (F-SM1)

Instantiating (F-SM0) with t'', n_2, v'' Since $v \Downarrow_{t''}^{n_2} v''$ therefore from (F-SM0) we know that
$\exists p'.n_1 + p' \leq p + n \wedge (p', T - t'', v'') \in [\![\tau \; \sigma\iota]\!]$ (F-SM2)

<u>IH</u> $[\![\tau \; \sigma\iota]\!] \subseteq [\![\tau' \; \sigma\iota]\!]$

In order to prove (F-SM1) we choose p'' as p' and we need to prove

(a) $n_1 + p' \leq p + n'$:
Since we are given that $n \leq n'$ therefore we get the desired from (F-SM2)

(b) $(p', T - t'', v') \in [\![\tau' \; \sigma\iota]\!]$
We get this directly from IH

8. sub-subExp:

$$\frac{\Psi; \Theta, a; \Delta, a < J \vdash \tau <: \tau' \qquad \Psi; \Theta, a; \Delta \vdash J \leq I}{\Psi; \Theta; \Delta \vdash !_{a<I}\tau <: !_{a<J}\tau'} \text{ sub-subExp}$$

To prove: $\forall (p, T, v) \in [\![!_{a<I}\tau \; \sigma\iota]\!].(p, T, v) \in [\![!_{a<J}\tau' \; \sigma\iota]\!]$

This means given $(p, T, !v) \in [\![!_{a<I}\tau \; \sigma\iota]\!]$ and we need to prove
$(p, T, !v) \in [\![!_{a<J}\tau' \; \sigma\iota]\!]$

This means from Definition 58 we are given
$\exists p_0, \ldots, p_{I-1}.p_0 + \ldots + p_{I-1} \leq p \wedge \forall 0 \leq i < I.(p_i, T, v) \in [\![\tau[i/a]]\!]$ (F-SE0)

Again from Definition 58 we need to prove that
$\exists p'_0, \ldots, p'_{J-1}.p'_0 + \ldots + p'_{J-1} \leq p \wedge \forall 0 \leq j < J.(p_j, T, v) \in [\![\tau'[j/a]]\!]$ (F-SE1)

In order to prove (F-SE1) we choose $p'_0 \ldots p'_{J-1}$ as $p_0 \ldots p_{J-1}$ and we need to prove

(a) $p_0 + \ldots + p_{J-1} \leq p$:
Since we are given that $J \leq I$ therefore we get the desired from (F-SE0)

(b) $\forall 0 \leq j < J.(p_j, T, v) \in [\![\tau'[j/a] \; \sigma\iota]\!]$
We get this directly from IH and (F-SE0)

9. sub-list:

$$\frac{\Psi; \Theta; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash L^n\ \tau <: L^n\ \tau'} \text{ sub-list}$$

To prove: $\forall (p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!].(p, T, v) \in [\![L^n\ \tau'\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!]$ and we need to prove $(p, T, v) \in [\![L^n\ \tau'\ \sigma\iota]\!]$

We induct on $(p, T, v) \in [\![L^n\ \tau\ \sigma\iota]\!]$

(a) $(p, T, nil) \in [\![L^0\ \tau\ \sigma\iota]\!]$:
We need to prove $(p, T, nil) \in [\![L^0\ \tau'\ \sigma\iota]\!]$
We get this directly from Definition 58

(b) $(p, T, v' :: l') \in [\![L^{m+1}\ \tau\ \sigma\iota]\!]$:
In this case we are given $(p, T, v' :: l') \in [\![L^{m+1}\ \tau\ \sigma\iota]\!]$
and we need to prove $(p, T, v' :: l') \in [\![L^{m+1}\ \tau'\ \sigma\iota]\!]$

This means from Definition 58 are given
$\exists p_1, p_2.p_1 + p_2 \leq p \wedge (p_1, T, v') \in [\![\tau\ \sigma\iota]\!] \wedge (p_2, T, l') \in [\![L^m \tau\ \sigma\iota]\!]$ (Sub-List0)

Similarly from Definition 58 we need to prove that
$\exists p'_1, p'_2.p'_1 + p'_2 \leq p \wedge (p'_1, T, v') \in [\![\tau'\ \sigma\iota]\!] \wedge (p_2, T, l') \in [\![L^m \tau'\ \sigma\iota]\!]$

We choose p'_1 as p_1 and p'_2 as p_2 and we get the desired from (Sub-List0) IH of outer induction and IH of innner induction

10. sub-exist:

$$\frac{\Psi; \Theta, s; \Delta \vdash \tau <: \tau'}{\Psi; \Theta; \Delta \vdash \exists s.\tau <: \exists s.\tau'} \text{ sub-exist}$$

To prove: $\forall (p, T, v) \in [\![\exists s.\tau\ \sigma\iota]\!].(p, T, v) \in [\![\exists s.\tau'\ \sigma\iota]\!]$

This means given some $(p, T, v) \in [\![\exists s.\tau\ \sigma\iota]\!]$ we need to prove $(p, T, v) \in [\![\exists s.\tau'\ \sigma\iota]\!]$

From Definition 58 we are given that
$\exists s'.(p, T, v) \in [\![\tau\sigma\iota[s'/s]]\!]$ (F-exist0)

<u>IH</u>: $[\![(\tau)\ \sigma\iota \cup \{s \mapsto s'\}]\!] \subseteq [\![(\tau')\ \sigma\iota \cup \{s \mapsto s'\}]\!]$

Also from Definition 58 it suffices to prove that
$\exists s''.(p, T, v) \in [\![\tau'\sigma\iota[s''/s]]\!]$

We choose s'' as s' and we get the desired from IH

11. sub-typePoly:

$$\frac{\Psi, \alpha; \Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall\alpha.\tau_1 <: \forall\alpha.\tau_2} \text{ sub-typePoly}$$

To prove: $\forall (p, T, \Lambda\alpha.e) \in [\![(\forall\alpha.\tau_1)\ \sigma\iota]\!].(p, T, \Lambda\alpha.e) \in [\![(\forall\alpha.\tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda\alpha.e) \in [\![(\forall\alpha.\tau_1)\ \sigma\iota]\!]$ we need to prove $(p, T, \Lambda\alpha.e) \in [\![(\forall\alpha.\tau_2)\ \sigma\iota]\!]$

From Definition 58 we are given that
$\forall \tau', T' {<} T\ .(p, T', e) \in [\![\tau_1[\tau'/\alpha]]\!]_{\mathcal{E}}$ (F-STF0)

Also from Definition 58 it suffices to prove that
$\forall \tau'', T'' {<} T\ .(p, T''\ e) \in [\![\tau_2[\tau''/\alpha]]\!]_{\mathcal{E}}$

This means given some $\tau'', T'' {<} T$ and we need to prove
$(p, T'', e[\tau''/\alpha]) \in [\![\tau_2[\tau''/\alpha]]\!]_{\mathcal{E}}$ (F-STF1)

<u>IH</u>: $[\![(\tau_1)\ \sigma \cup \{\alpha \mapsto \tau''\}\iota]\!] \subseteq [\![(\tau_2)\ \sigma \cup \{\alpha \mapsto \tau''\}\iota]\!]$

Instantiating (F-STF0) with τ'', T'' we get
$(p, T'', e) \in [\![\tau_1[\tau''/\alpha]]\!]_{\mathcal{E}}$

and finally from IH we get the desired

12. sub-indexPoly:

$$\frac{\Psi; \Theta, i; \Delta \vdash \tau_1 <: \tau_2}{\Psi; \Theta; \Delta \vdash \forall i.\tau_1 <: \forall i.\tau_2} \text{ sub-indexPoly}$$

To prove: $\forall (p, T, \Lambda i.e) \in [\![(\forall i.\tau_1)\ \sigma\iota]\!].(p, T, \Lambda i.e) \in [\![(\forall i.\tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda i.e) \in [\![(\forall i.\tau_1)\ \sigma\iota]\!]$ we need to prove $(p, T, \Lambda i.e) \in [\![(\forall i.\tau_2)\ \sigma\iota]\!]$

From Definition 58 we are given that
$\forall I, T'{<}T\ .(p, T', e) \in [\![\tau_1[I/i]]\!]_{\mathcal{E}}$ (F-SIF0)

Also from Definition 58 it suffices to prove that
$\forall I', T''{<}T\ .(p, T'', e) \in [\![\tau_2[I'/i]]\!]_{\mathcal{E}}$

This means given some $I', T''{<}T$ and we need to prove
$(p, T'', e) \in [\![\tau_2[I'/i]]\!]_{\mathcal{E}}$ (F-SIF1)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota \cup \{i \mapsto I'\}]\!] \subseteq [\![(\tau_2)\ \sigma\iota \cup \{i \mapsto I'\}]\!]$

Instantiating (F-SIF0) with I', T'' we get
$(p, T'', e) \in [\![\tau_1[I'/i]]\!]_{\mathcal{E}}$

and finally from IH we get the desired

13. sub-constraint:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_2 \implies c_1}{\Psi; \Theta; \Delta \vdash c_1 \Rightarrow \tau_1 <: c_2 \Rightarrow \tau_2} \text{ sub-constraint}$$

To prove: $\forall (p, T, \Lambda.e) \in [\![(c_1 \Rightarrow \tau_1)\ \sigma\iota]\!].(p, T, \Lambda.e) \in [\![(c_2 \Rightarrow \tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \Lambda.e) \in [\![(c_1 \Rightarrow \tau_1)\ \sigma\iota]\!]$ we need to prove $(p, T, \Lambda.e) \in [\![(c_2 \Rightarrow \tau_2)\ \sigma\iota]\!]$

From Definition 58 we are given that
$\forall\, T'{<}T\ . \models c_1\iota \implies (p, T', e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SC0)

Also from Definition 58 it suffices to prove that
$\forall\, T''{<}T\ . \models c_2\iota \implies (p, T'', e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$

This means given some $T''{<}T$ s.t. $\models c_2\iota$ and we need to prove
$(p, T'', e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$ (F-SC1)

Since we are given that $\Theta; \Delta \models c_2 \implies c_1$ therefore we know that $. \models c_1\iota$

Hence from (F-SC0) we have
$(p, T'', e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SC2)

<u>IH</u>: $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_2)\ \sigma\iota]\!]$

Therefore we ge the desired from IH and (F-SC2)

14. sub-CAnd:

$$\frac{\Psi; \Theta; \Delta \vdash \tau_1 <: \tau_2 \qquad \Theta; \Delta \models c_1 \implies c_2}{\Psi; \Theta; \Delta \vdash c_1 \& \tau_1 <: c_2 \& \tau_2} \text{ sub-CAnd}$$

To prove: $\forall (p, T, v) \in [\![(c_1 \& \tau_1)\ \sigma\iota]\!].(p, T, v) \in [\![(c_2 \& \tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, v) \in [\![(c_1 \& \tau_1)\ \sigma\iota]\!]$ we need to prove $(p, T, v) \in [\![(c_2 \& \tau_2)\ \sigma\iota]\!]$

From Definition 58 we are given that
$. \models c_1\iota \wedge (p, T, e) \in [\![\tau_1\sigma\iota]\!]_{\mathcal{E}}$ (F-SCA0)

Also from Definition 58 it suffices to prove that
$. \models c_2\iota \wedge (p, T, e) \in [\![\tau_2\sigma\iota]\!]_{\mathcal{E}}$

Since we are given that $\Theta; \Delta \models c_2 \implies c_1$ and $. \models c_1\iota$ therefore we also know that $. \models c_2\iota$

Also from (F-SCA0) we have $(p, T, e) \in [\![\tau_1 \sigma\iota]\!]_{\mathcal{E}}$ (F-SCA1)

IH: $[\![(\tau_1)\ \sigma\iota]\!] \subseteq [\![(\tau_2)\ \sigma\iota]\!]$

Therefore we ge the desired from IH and (F-SCA1)

15. sub-potArrow:

$$\frac{\Psi; \Theta; \Delta \vdash k'}{\Psi; \Theta; \Delta \vdash [k](\tau_1 \multimap \tau_2) <: ([k']\,\tau_1 \multimap [k'+k]\,\tau_2)} \text{ sub-potArrow}$$

To prove: $\forall (p, T, \lambda x.e) \in [\![([k](\tau_1 \multimap \tau_2))\ \sigma\iota]\!].(p, T, \lambda x.e) \in [\![([k']\,\tau_1 \multimap [k'+k]\,\tau_2)\ \sigma\iota]\!]$

This means given some $(p, T, \lambda x.e) \in [\![([k](\tau_1 \multimap \tau_2))\ \sigma\iota]\!]$ we need to prove $(p, T, \lambda x.e) \in [\![(([k']\,\tau_1 \multimap [k'+k]\,\tau_2))\ \sigma\iota]\!]$

From Definition 58 we are given that
$\exists p'.p' + k \leq p \wedge (p', T, \lambda x.e) \in [\![(\tau_1 \multimap \tau_2)\ \sigma\iota]\!]\}$ (F-SPA0)

Again from Definition 58 we know that
$\forall p''', e', T'{<}T\ .(p''', T', e') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p' + p''', T', e[e'/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SPA1)

Also from Definition 58 it suffices to prove that
$\forall p'', e'', T''{<}T\ .(p'', T'', e'') \in [\![[k']\,\tau_1\ \sigma\iota]\!]_{\mathcal{E}} \implies (p + p'', T'', e[e''/x]) \in [\![[k+k']\,\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

This means given some $p'', e'', T''{<}T$ s.t $(p'', T'', e'') \in [\![[k']\,\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ we need to prove $(p + p'', T'', e[e''/x]) \in [\![[k+k']\,\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$ (F-SSP2)

Applying Definition 58 on (F-SPA2) we get
$\forall v_f, t' {<} T''\ .e[e''/x] \Downarrow_{t'} v_f \implies (p + p'', T'' - t', v_f) \in [\![[k+k']\,\tau_2\ \sigma\iota]\!]$

This means that given some $v_f, t' {<} T''$ s.t. $e[e''/x] \Downarrow_{t'} v_f$ and we need to prove that $(p + p'', T'' - t', v_f) \in [\![[k+k']\,\tau_2\ \sigma\iota]\!]$

This means From Definition 58 it suffices to prove that
$\exists p''_2.p''_2 + (k + k') \leq (p + p'') \wedge (p''_2, T'' - t', v_f) \in [\![\tau_2\ \sigma\iota]\!]\}$ (F-SPA4)

Also since we are given that $(p'', T'', e'') \in [\![[k']\,\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$ we apply Definition 58 on it to obtain
$\forall t {<} T'', v'.e'' \Downarrow_t v' \implies (p'', T'' - t, v') \in [\![[k']\,\tau_1\ \sigma\iota]\!]$
Also since we are given that $e[e''/x] \Downarrow_{t'} v_f$ therefore we also know that
$\exists t'' < t' {<} T''\ .e'' \Downarrow_{t''} v''$
Instantiating with t'', v'' we get $(p'', T'' - t'', v'') \in [\![[k']\,\tau_1\ \sigma\iota]\!]$

Again using Definition 58 we know that we are given
$\exists p''_1.p''_1 + k' \leq p'' \wedge (p''_1, T'' - t'', v'') \in [\![\tau_1\ \sigma\iota]\!]$ (F-SPA3)

Since $(p''_1, T'' - t'', v'') \in [\![\tau_1\ \sigma\iota]\!]$ therefore from Definition 58 we also have $(p''_1, T'' - t'', v'') \in [\![\tau_1\ \sigma\iota]\!]_{\mathcal{E}}$

Instantiating (F-SPA1) with $p''_1, v'', T'' - t''$ we get
$(p' + p''_1, T'' - t'', e[v''/x]) \in [\![\tau_2\ \sigma\iota]\!]_{\mathcal{E}}$

From Definition 58 this means that
$\forall t''' {<} T'' - t'', v_f.e[v''/x] \Downarrow v_f \implies (p' + p''_1, T'' - t'' - t''', v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-SPA4.1)

Since we know that $e[e''/x] \Downarrow_{t'} v_f$ therefore we also know that $\exists t'''.e[v''/x] \Downarrow_{t'''} v_f$ s.t. $t''' + t'' \leq t'$
Since we already know that $\exists t'' < t' {<} T''\ .e'' \Downarrow_{t''} v''$ therefore we have $t'' + t''' \leq t' {<} T''$.
Instantiating (F-SPA4.1) with t''' we get
$(p' + p''_1, T'' - t'' - t''', v_f) \in [\![\tau_2\ \sigma\iota]\!]$ (F-SPA5)

Since from (F-SPA0) we know that
$p' + k \leq p$

And from (F-SPA3) we know that
$p''_1 + k' \leq p''$

We add the two to get
$p' + p''_1 + k + k' \leq p + p''$ (F-SPA6)

In order to prove (F-SPA4) we choose p''_2 as $p' + p''_1$
and we get the desired from (F-SPA6) and (F-SPA5) and Lemma 61

16. sub-potZero:

$$\frac{}{\Psi; \Theta; \Delta \vdash \tau <: [0]\, \tau} \text{ sub-potZero}$$

To prove: $\forall (p, T, v) \in [\![\tau\ \sigma\iota]\!].(p, T, v) \in [\![[0]\, \tau\ \sigma\iota]\!]$

This means that given $(p, T, v) \in [\![\tau\ \sigma\iota]\!]$

And we need to prove $(p, T, v) \in [\![[0]\, \tau\ \sigma\iota]\!]$

From Definition 58 it suffices to prove that
$\exists p'.p' + 0 \leq p \wedge (p', T, v) \in [\![\tau\ \sigma\iota]\!]$

We choose p' as p and we get the desired

17. sub-familyAbs:

$$\frac{\Psi; \Theta, i : \mathsf{S} \vdash \tau <: \tau'}{\Psi; \Theta \vdash \lambda_t i : \mathsf{S}.\tau <: \lambda_t i : \mathsf{S}.\tau'} \text{ sub-familyAbs}$$

To prove:
$\forall f \in [\![\lambda_t i : \mathsf{S}.\tau\ \sigma\iota]\!].f \in [\![\lambda_t i : \mathsf{S}.\tau'\ \sigma\iota]\!]$

This means given $f \in [\![\lambda_t i : \mathsf{S}.\tau\ \sigma\iota]\!]$ and we need to prove
$f \in [\![\lambda_t i : \mathsf{S}.\tau'\ \sigma\iota]\!]$

This means from Definition 58 we are given
$\forall I.f\ I \in [\![\tau[I/i]\ \sigma\iota]\!]$ (F-SFAbs0)

This means from Definition 58 we need to prove
$\forall I'.f\ I' \in [\![\tau'[I'/i]\ \sigma\iota]\!]$
This further means that given some I' we need to prove
$f\ I' \in [\![\tau'[I'/i]\ \sigma\iota]\!]$ (F-SFAbs1)

Instantiating (F-SFAbs0) with I' we get
$f\ I' \in [\![\tau[I'/i]\ \sigma\iota]\!]$

From IH we know that $[\![\tau\ \sigma\iota \cup \{i \mapsto I'\ \iota\}]\!] \subseteq [\![\tau'\ \sigma\iota \cup \{i \mapsto I'\ \iota\}]\!]$
And this completes the proof.

18. Sub-tfamilyApp1:

$$\frac{}{\Psi; \Theta; \Delta \vdash \lambda_t i : \mathsf{S}.\tau\ I <: \tau[I/i]} \text{ sub-familyApp1}$$

To prove: $\forall (p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!].(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$ and we need to prove
$(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$

This means from Definition 58 we are given
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau]\!]\ I\ \sigma\iota$
This further means that we have
$(p, T, v) \in f\ I\iota$ where $f\ I = [\![\tau\sigma[I\iota/i]]\!]$

This means we have $(p, T, v) \in [\![\tau\sigma[I\iota/i]]\!]$
And this completes the proof.

19. Sub-tfamilyApp2:

$$\frac{}{\Psi; \Theta; \Delta \vdash \tau[I/i] <: \lambda_t i : \mathsf{S}.\tau\ I} \text{ sub-familyApp2}$$

To prove: $\forall (p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!].(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$

This means given $(p, T, v) \in [\![\tau[I/i]\ \sigma\iota]\!]$ (Sub-tF0)
And we need to prove
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau\ I\ \sigma\iota]\!]$

This means from Definition 58 it suffices to prove that
$(p, T, v) \in [\![\lambda_t i : \mathsf{S}.\tau]\!]\ I\ \sigma\iota$
It further suffices to prove that
$(p, T, v) \in f\ I\iota$ where $f\ I\iota = [\![\tau\sigma[I\iota/i]]\!]$

which means we need to show that
$(p, T, v) \in [\![\tau\sigma[I\iota/i]]\!]$
We get this directly from (Sub-tF0)

20. sub-bSum:

$$\frac{}{\Psi; \Theta; \Delta \vdash [\sum_{a<I} K] \, !_{a<I}\tau <: !_{a<I} \, [K] \, \tau} \text{ sub-bSum}$$

To prove: $\forall (p, T, v) \in [\![[\sum_{a<I} K] \, !_{a<I}\tau \; \sigma\iota]\!] \implies (p, T, v) \in [\![!_{a<I} \, [K] \, \tau \; \sigma\iota]\!]$

This means given some (p, T, v) s.t $(p, T, v) \in [\![[\sum_{a<I} K] \, !_{a<I}\tau \; \sigma\iota]\!]$ it suffices to prove that $(p, T, v) \in [\![!_{a<I} \, [K] \, \tau \; \sigma\iota]\!]$

This means from Definition 58 we are given that
$\exists p'.p' + \sum_{a<I} K \leq p \wedge (p', T, v) \in [\![!_{a<I}\tau \; \sigma\iota]\!]\}$ (Sub-BS0)

Since $(p', T, v) \in [\![!_{a<I}\tau \; \sigma\iota]\!]\}$ therefore again from Definition 58 it means that $\exists e'. \; v = !e'$ and $\exists p_0, \ldots, p_{I-1}.p_0 + \ldots + p_{I-1} \leq p' \wedge \forall 0 \leq i < I.(p_i, T, e') \in [\![\tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$ (Sub-BS1)

Since $\forall 0 \leq i < I.(p_i, T, e') \in [\![\tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$ therefore from Definition 58 we have
$\forall 0 \leq i < I.\forall t < T, v''.e' \Downarrow_t v'' \implies (p_i, T - t, v') \in [\![\tau[i/a] \; \sigma\iota]\!]$ (Sub-BS1.1)

Since we know that $v = !e'$ therefore it suffices to prove that $(p, T, !e') \in [\![!_{a<I} \, [K] \, \tau \; \sigma\iota]\!]$

From Definition 58 it further suffices to prove that
$\exists p'_0, \ldots, p'_{I-1}.p'_0 + \ldots + p'_{I-1} \leq p \wedge \forall 0 \leq i < I.(p'_i, T, e') \in [\![[K] \, \tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$

We choose p'_0 as $p_0 + K[0/a] \ldots p'_{I-1}$ as $p_{I-1} + K[(I-1)/a]$ and it suffices to prove that

- $p'_0 + \ldots + p'_{I-1} \leq p$:
  We need to prove that
  $(p_0 + K[0/a]) + \ldots + (p_{I-1} + K[(I-1)/a]) \leq p$
  We get this from (Sub-BS0) and (Sub-BS1)
- $\forall 0 \leq i < I.(p'_i, T, e') \in [\![[K] \, \tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$:
  Given some $0 \leq i < I$ it suffices to prove that
  $(p'_i, T, e') \in [\![[K] \, \tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$

  Since p'_i is $p_i + K[i/a]$ therefore it suffices to prove that
  $(p_i + K[i/a], T, e') \in [\![[K[i/a]] \, \tau[i/a] \; \sigma\iota]\!]_{\mathcal{E}}$

  From Definition 58 we need to prove that
  $\forall v', t'' < T \, .e' \Downarrow_{t''} v' \implies (p_i + K[i/a], T - t'', v') \in [\![[K[i/a]] \, \tau[i/a] \; \sigma\iota]\!]$

  This means given some v' s.t $e' \Downarrow_{t''} v'$ we need to prove that
  $(p_i + K[i/a], T - t'', v') \in [\![[K[i/a]] \, \tau[i/a] \; \sigma\iota]\!]$

  From Definition 58 it suffices to prove that
  $\exists p''.p'' + K[i/a] \leq p_i + K[i/a] \wedge (p'', T - t'', v') \in [\![\tau[i/a] \; \sigma\iota]\!]\}$

  We choose p'' as p_i and we need to prove
  $(p_i, T - t'', v') \in [\![\tau[i/a] \; \sigma\iota]\!]$
  Instantiating (Sub-BS1.1) with the given i and v', t'' we get the desired

□

Lemma 70 (Expression subtyping lemma)**.** $\forall \Psi, \Theta, \Delta, \tau \in Type, \tau'.$
$\Psi; \Theta; \Delta \vdash \tau <: \tau' \implies [\![\tau \; \sigma\iota]\!]_{\mathcal{E}} \subseteq [\![\tau' \; \sigma\iota]\!]_{\mathcal{E}}$

Proof. To prove: $\forall (p, T, e) \in [\![\tau \; \sigma\iota]\!]_{\mathcal{E}} \implies (p, T, e) \in [\![\tau' \; \sigma\iota]\!]_{\mathcal{E}}$
This means given some $(p, T, e) \in [\![\tau \; \sigma\iota]\!]_{\mathcal{E}}$ it suffices to prove that
$(p, T, e) \in [\![\tau' \; \sigma\iota]\!]_{\mathcal{E}}$

This means from Definition 58 we are given
$\forall v, t < T \, .e \Downarrow_t v \implies (p, T - t, v) \in [\![\tau \; \sigma\iota]\!]$ (S-E0)

Similarly from Definition 58 it suffices to prove that
$\forall v', t' < T \, .e \Downarrow_{t'} v' \implies (p, T - t', v') \in [\![\tau' \; \sigma\iota]\!]$

This means given some $v', t' < T$ s.t $e \Downarrow_{t'} v'$ it suffices to prove that
$(p, T - t', v') \in [\![\tau' \; \sigma\iota]\!]$

Instantiating (S-E0) with v', t' we get $(p, T - t', v') \in [\![\tau\ \sigma\iota]\!]$

And finally from Lemma 69 we get the desired.

$\square$

Theorem 71 (Soundness). $\forall e, n, n', \tau \in Type, t.$
$\vdash e : \mathbb{M}\, n\, \tau \wedge e \Downarrow_t^{n'} v \implies n' \leq n$

Proof. From Theorem 64 we know that $(0, t+1, e) \in [\![\mathbb{M}\, n\, \tau]\!]_{\mathcal{E}}$

From Definition 58 this means we have
$\forall t' < t + 1. e \Downarrow_{t'} v' \implies (0, t + 1 - t'v') \in [\![\mathbb{M}\, n\, \tau]\!]$

From the evaluation relation we know that $e \Downarrow_0 e$ therefore we have
$(0, t+1, e) \in [\![\mathbb{M}\, n\, \tau]\!]$

Again from Definition 58 it means we have
$\forall t'' < t + 1. e \Downarrow_{t'}^{n'} v \implies \exists p'. n' + p' \leq 0 + n \wedge (p', t + 1 - t'', v) \in [\![\tau]\!]$

Since we are given that $e \Downarrow_t^{n'} v$ therefore we have
$\exists p'. n' + p' \leq n \wedge (p', 1, v) \in [\![\tau]\!]$

Since $p' \geq 0$ therefore we get $n' \leq n$

$\square$

Theorem 72 (Soundness). $\forall e, n, n', \tau \in Type.$
$\vdash e : [n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau \wedge e\ ()\Downarrow_{t_1} - \Downarrow_{t_2}^{n'} v \implies n' \leq n$

Proof. From Theorem 64 we know that $(0, t_1 + t_2 + 2, e) \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$

Therefore from Definition 58 we know that
$\forall t' < t_1 + t_2 + 2, v. e \Downarrow_{t'} v \implies (0, t_1 + t_2 + 2 - t', v) \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]$ (S0)

Since we know that $e\ () \Downarrow_{t_1} -$ therefore from E-app we know that $\exists e'. e \Downarrow_{t_1} \lambda x. e'$

Instantiating (S0) with $t_1, \lambda x. e'$ we get $(0, t_2 + 2, \lambda x. e') \in [\![[n]\, \mathbf{1} \multimap \mathbb{M}\, 0\, \tau]\!]$

This means from Definition 58 we have
$\forall p', e', t'' < t_2 + 2. (p', t'', e'') \in [\![[n]\, \mathbf{1}]\!]_{\mathcal{E}} \implies (0 + p', t'', e'[e''/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$ (S1)

Claim: $\forall t. (I, t, ()) \in [\![[I]\, \mathbf{1}]\!]_{\mathcal{E}}$
Proof:
From Definition 58 it suffices to prove that
$() \Downarrow_0 v \implies (I, t, v) \in [\![[I]\, \mathbf{1}]\!]$

Since we know that $v = ()$ therefore it suffices to prove that
$(I, t, v) \in [\![[I]\, \mathbf{1}]\!]$

From Definition 58 it suffices to prove that
$\exists p'. p' + I \leq I \wedge (p', t, v) \in [\![\mathbf{1}]\!]\}$

We choose p' as 0 and we get the desired

Instantiating (S1) with $n, (), t_2 + 1$ we get $(n, t_2 + 1, e'[()/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]_{\mathcal{E}}$

This means again from Definition 58 we have
$\forall t' < t_2 + 1. e'[()/x] \Downarrow_{t'} v' \implies (n, t_2 + 1 - t', v') \in [\![\mathbb{M}\, 0\, \tau]\!]$

From E-val we know that $v' = e'[()/x]$ and $t' = 0$ therefore we have
$(n, t_2 + 1, e'[()/x]) \in [\![\mathbb{M}\, 0\, \tau]\!]$

Again from Definition 58 we have
$\forall t' < t_2 + 1. e'[()/x] \Downarrow_{t'}^{n'} v'' \implies \exists p'. n' + p' \leq n + 0 \wedge (p', t_2 + 1 - t', v'') \in [\![\tau]\!]$

Since we are given that $e \Downarrow_{t_1} - \Downarrow_{t_2}^{n'} v$ therefore we get
$\exists p'. n' + p' \leq n \wedge (p', 1, v'') \in [\![\tau]\!]$

Since $p' \geq 0$ therefore we have $n' \leq n$

$\square$

B.5 Embedding dlPCF

Type translation

$$\begin{array}{lcl} (\!|b|\!) & = & b \\ (\!|[a < I]\tau_1 \multimap \tau_2|\!) & = & (!_{a<I}\, \mathbb{M}\, 0 (\!|\tau_1|\!)) \multimap [I]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau_2|\!) \end{array}$$

Judgment translation

$$\boxed{\Theta; \Delta; \Gamma \vdash_K e_d : \tau \quad \rightsquigarrow \quad .; \Theta; \Delta; (\!|\Gamma|\!); . \vdash e_a : [K + \mathit{count}(\Gamma)]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau|\!)}$$

where

$$\begin{array}{lcl} \mathit{count}(.) & = & 0 \\ \mathit{count}(\Gamma, x : [a < I]\tau) & = & \mathit{count}(\Gamma) + I \end{array}$$

Definition 73 (Context translation)**.**

$$\begin{array}{lcl} (\!|.|\!) & = & . \\ (\!|\Gamma, x : [a < I]\tau|\!) & = & (\!|\Gamma|\!), x :_{a<I} \mathbb{M}\, 0\, (\!|\tau|\!) \end{array}$$

Expression translation

$$\frac{\Theta; \Delta \models J \geq 0 \qquad \Theta; \Delta \models I \geq 1 \qquad \Theta; \Delta \vdash \sigma[0/a] <: \tau \qquad \Theta; \Delta \models [a < I]\sigma \Downarrow \qquad \Theta; \Delta \models \Gamma \Downarrow}{\Theta; \Delta; \Gamma, x : [a < I]\tau \vdash_J x : \tau[0/a] \rightsquigarrow \lambda p.\mathsf{release} - = p \;\mathsf{in}\; \mathsf{bind} - = \uparrow^1 \mathsf{in}\; x} \text{ var}$$

$$\frac{\Theta; \Delta; \Gamma, x : [a < I]\tau_1 \vdash_J e : \tau_2 \rightsquigarrow e_t}{\Theta; \Delta; \Gamma \vdash_J \lambda x.e : ([a < I].\tau_1) \multimap \tau_2 \rightsquigarrow} \text{ lam}$$
$$\lambda p_1.\, \mathsf{ret}\; \lambda y.\lambda p_2.\, \mathsf{let}\, !\, x = y \;\mathsf{in}\; \mathsf{release} - = p_1 \;\mathsf{in}\; \mathsf{release} - = p_2 \;\mathsf{in}\; \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; e_t\; a$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash_J e_1 : ([a < I].\tau_1) \multimap \tau_2 \rightsquigarrow e_{t1} \\ \Theta, a; \Delta, a < I; \Delta \vdash_K e_2 : \tau_1 \rightsquigarrow e_{t2} \qquad \Gamma' \sqsupseteq \Gamma \oplus \sum_{a<I} \Delta \qquad H \geq J + I + \sum_{a<I} K\end{array}}{\Theta; \Delta; \Gamma' \vdash_H e_1\; e_2 : \tau_2 \rightsquigarrow} \text{ app}$$
$$\lambda p.\mathsf{release} - = p \;\mathsf{in}\; \mathsf{bind}\, a = \mathsf{store}() \;\mathsf{in}\; \mathsf{bind}\, b = e_{t1}\; a \;\mathsf{in}\; \mathsf{bind}\, c = \mathsf{store!}() \;\mathsf{in}\; \mathsf{bind}\, d = \mathsf{store}() \;\mathsf{in}\; b\; (\mathit{coerce}1\; !e_{t2}\; c)\; d$$

$$\frac{\begin{array}{c}\Theta, b; \Delta, b < L; \Gamma, x : [a < I]\sigma \vdash_K e : \tau \rightsquigarrow e_t \\ \tau[0/a] <: \mu \qquad \Theta, a, b; \Delta, a < I, b < L; \Gamma \vdash \tau[(b + 1 + \bigcirc\!\!\!\!\triangle_b^{b+1,a} I)/b] <: \sigma \\ \Gamma' \sqsubseteq \sum_{b<L} \Gamma \qquad L, M \geq \bigcirc\!\!\!\!\triangle_b^{0,1} I \qquad N \geq M - 1 + \sum_{b<L} K\end{array}}{\Theta; \Delta; \Gamma' \vdash_N \mathsf{fix} x.e : \mu \rightsquigarrow E_0} \text{ T-fix}$$

$E_0 = \mathsf{fix} Y.E_1$
$E_1 = \lambda p.E_2$
$E_2 = \mathsf{release} - = p \;\mathsf{in}\; E_3$
$E_3 = \mathsf{bind}\, A = \mathsf{store}() \;\mathsf{in}\; E_4$
$E_4 = \mathsf{let}\, !\, x = (E_{4.1}\; E_{4.2}) \;\mathsf{in}\; E_5$
$E_{4.1} = \mathit{coerce}1\; !Y$
$E_{4.2} = (\lambda u.!())\; A$
$E_5 = \mathsf{bind}\, C = \mathsf{store}() \;\mathsf{in}\; E_6$
$E_6 = e_t\; C$

B.5.1 Type preservation

Theorem 74 (Type preservation: dlPCF to λ-amor)**.** *If $\Theta; \Delta; \Gamma \vdash_I e : \tau$ in dlPCF then there exists e' such that $\Theta; \Delta; \Gamma \vdash_I e : \tau \rightsquigarrow e'$ such that there is a derivation of $.; \Theta; \Delta; (\!|\Gamma|\!); . \vdash e' : [I + \mathit{count}(\Gamma)]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau|\!)$ in λ-amor.*

Proof. Proof by induction on the $\Theta; \Delta; \Gamma \vdash_I e : \tau$

- var:

$$\frac{\Theta;\Delta \models J \geq 0 \qquad \Theta;\Delta \models I \geq 1 \qquad \Theta;\Delta \vdash \sigma[0/a] <: \tau \qquad \Theta;\Delta \models [a<I]\sigma \Downarrow \qquad \Theta;\Delta \models \Gamma \Downarrow}{\Theta;\Delta;\Gamma, x:[a<I]\sigma \vdash_J x:\tau[0/a] \rightsquigarrow \lambda p.\mathsf{release} - = p \mathsf{~in~} \mathsf{bind} - = \uparrow^1 \mathsf{~in~} x} \text{var}$$

D2:

$$\frac{\Theta;\Delta \vdash \sigma[0/a] <: \tau}{\Theta;\Delta \vdash (\!|\sigma[0/a]|\!) <: (\!|\tau|\!)} \text{Lemma 79}$$

D1:

$$\frac{\dfrac{}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), \vdash x:\mathbb{M}\,0\,(\!|\sigma|\!)[0/a]} \text{T-var2}}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), \vdash x:\mathbb{M}\,0\,(\!|\sigma[0/a]|\!)} \text{Lemma 80}$$

D0:

$$\frac{\dfrac{\dfrac{}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), \vdash \uparrow^1 : \mathbb{M}\,1\,\mathbf{1}}}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), \vdash \uparrow^1 : \mathbb{M}(I+J+count(\Gamma))\,\mathbf{1}} \qquad D1}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), \vdash \mathsf{bind} - = \uparrow^1 \mathsf{~in~} x : \mathbb{M}(I+J+count(\Gamma))\,(\!|\sigma[0/a]|\!)} \text{bind}$$

Main derivation:

$$\frac{\dfrac{\dfrac{}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!), p:([I+J+count(\Gamma)]\,\mathbf{1}) \vdash \quad p:([I+J+count(\Gamma)]\,\mathbf{1})} \qquad D0}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!); p:([I+J+count(\Gamma)]\,\mathbf{1}) \vdash \quad \mathsf{release} - = p \mathsf{~in~} \mathsf{bind} - = \uparrow^1 \mathsf{~in~} x : \mathbb{M}\,0\,(\!|\tau|\!)} \text{T-release}}{\begin{array}{c}\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\sigma|\!); \cdot \vdash \\ \lambda p.\mathsf{release} - = p \mathsf{~in~} \mathsf{bind} - = \uparrow^1 \mathsf{~in~} x : (([I+J+count(\Gamma)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\!|\tau|\!))\end{array}} \text{T-lam}$$

- lam:

$$\frac{\Theta;\Delta;\Gamma, x:[a<I]\tau_1 \vdash_J e:\tau_2 \rightsquigarrow e_t}{\begin{array}{c}\Theta;\Delta;\Gamma \vdash_J \lambda x.e : ([a<I].\tau_1) \multimap \tau_2 \rightsquigarrow \\ \lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a\end{array}}$$

$E_0 = \lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_1 = \mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_2 = \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_3 = \lambda p_2.\,\mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_4 = \mathsf{let}\,!\,x = y \mathsf{~in~release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_{4.1} = \mathsf{release} - = p_1 \mathsf{~in~release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_{4.2} = \mathsf{release} - = p_2 \mathsf{~in~bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$
$E_{4.3} = \mathsf{bind}\,a = \mathsf{store}() \mathsf{~in~} e_t\ a$

$T_0 = [J+count(\Gamma)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|([a<I]\tau_1) \multimap \tau_2|\!)$
$T_{0.1} = [J+count(\Gamma)]\,\mathbf{1} \multimap \mathbb{M}\,0((!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!))$
$T_{0.2} = [J+count(\Gamma)]\,\mathbf{1}$
$T_1 = \mathbb{M}\,0((!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!))$
$T_2 = ((!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!))$
$T_{2.1} = !_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)$
$T_3 = [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{3.1} = [I]\,\mathbf{1}$
$T_4 = \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{4.1} = \mathbb{M}(J+I+count(\Gamma))\,\mathbf{1}$
$T_{4.2} = \mathbb{M}(J+I+count(\Gamma))\,(\!|\tau_2|\!)$
$T_{4.3} = \mathbb{M}(J+count(\Gamma))\,(\!|\tau_2|\!)$
$T_5 = [(J+I+count(\Gamma))]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$

D6:

$$\frac{}{\cdot;\Theta;\Delta;\cdot;a:[J+I+count(\Gamma)]\,\mathbf{1} \vdash a:[J+I+count(\Gamma)]\,\mathbf{1}} \text{var}$$

D5:

$$\frac{}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\tau_1|\!); \cdot \vdash e_t : T_5} \text{IH}$$

D4:

$$\frac{D5 \qquad D6}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x:_{a<I} \mathbb{M}\,0\,(\!|\tau_1|\!); a:[J+I+count(\Gamma)]\,\mathbf{1} \vdash e_t\ a : T_4} \text{app}$$

D3:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash \mathsf{store}() : T_{4.1}}\,\text{store} \qquad D4}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x :_{a<I} \mathbb{M}\,0\,(\!|\tau_1|\!);\cdot \vdash E_{4.3} : T_{4.2}}\,\text{bind}$$

D2:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;p_2 : T_{3.1} \vdash p_2 : T_{3.1}} \qquad D3}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x :_{a<I} \mathbb{M}\,0\,(\!|\tau_1|\!);p_2 : T_{3.1} \vdash E_{4.2} : T_{4.3}}\,\text{bind}$$

D1:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;p_1 : T_{0.2} \vdash p_1 : T_{0.2}} \qquad D2}{\cdot;\Theta;\Delta;(\!|\Gamma|\!), x :_{a<I} \mathbb{M}\,0\,(\!|\tau_1|\!);p_1 : T_{0.2}, p_2 : T_{3.1} \vdash E_{4.1} : T_4}\,\text{release}$$

D0:

$$\dfrac{\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;y : T_{2.1} \vdash y : T_{2.1}} \qquad D1}{\cdot;\Theta;\Delta;(\!|\Gamma|\!);p_1 : T_{0.2}, y : T_{2.1}, p_2 : T_{3.1} \vdash E_4 : T_4}\,\text{T-subExpE}}{\cdot;\Theta;\Delta;(\!|\Gamma|\!);p_1 : T_{0.2}, y : T_{2.1} \vdash E_3 : T_3}\,\text{lam}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{D0}{\cdot;\Theta;\Delta;(\!|\Gamma|\!);p_1 : T_{0.2} \vdash E_2 : T_2}\,\text{lam}}{\cdot;\Theta;\Delta;(\!|\Gamma|\!);p_1 : T_{0.2} \vdash E_1 : T_1}\,\text{ret}}{\cdot;\Theta;\Delta;(\!|\Gamma|\!);\cdot \vdash E_0 : T_{0.1}}\,\text{lam}$$

- app:

$$\dfrac{\begin{array}{c}\Theta;\Delta;\Gamma_1 \vdash_J e_1 : ([a<I]\tau_1) \multimap \tau_2 \rightsquigarrow e_{t1} \\ \Theta, a;\Delta, a<I;\Gamma_2 \vdash_K e_2 : \tau_1 \rightsquigarrow e_{t2} \qquad \Gamma' \sqsupseteq \Gamma_1 \oplus \sum\limits_{a<I} \Gamma_2 \qquad H \geq J + I + \sum\limits_{a<I} K\end{array}}{\begin{array}{c}\Theta;\Delta;\Gamma' \vdash_H e_1\ e_2 : \tau_2 \rightsquigarrow \\ \lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d\end{array}}\,\text{app}$$

$E_0 = \lambda p.E_1$
$E_1 = \mathsf{release} - = p \text{ in } E_2$
$E_2 = \mathsf{bind}\, a = \mathsf{store}() \text{ in } E_3$
$E_3 = \mathsf{bind}\, b = e_{t1}\ a \text{ in } E_4$
$E_4 = \mathsf{bind}\, c = \mathsf{store!}() \text{ in } E_5$
$E_5 = \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ e_{t2}\ c)\ d$

$T_0 = [H + count(\Gamma')]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{0.11} = [J + I + \sum_{a<I} K + count(\Gamma_1) + count(\sum_{a<I}\Gamma_2)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{0.1} = [J + I + \sum_{a<I} K + count(\Gamma_1) + count(\sum_{a<I}\Gamma_2)]\,\mathbf{1}$
$T_{0.2} = \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{0.3} = \mathbb{M}(J + I + \sum_{a<I} K + count(\Gamma_1) + count(\sum_{a<I}\Gamma_2))\,(\!|\tau_2|\!)$
$T_1 = [(J + count(\Gamma))]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|([a<I]\tau_1) \multimap \tau_2|\!)$
$T_{1.1} = [(J + count(\Gamma))]\,\mathbf{1}$
$T_{1.11} = \mathbb{M}(J + count(\Gamma))\,[(J + count(\Gamma))]\,\mathbf{1}$
$T_{1.12} = \mathbb{M}(I + \sum_{a<I} K + count(\sum_{a<I}\Gamma_2))\,(\!|\tau_2|\!)$
$T_{1.13} = \mathbb{M}(\sum_{a<I} K + count(\sum_{a<I}\Gamma_2))\,T_{1.14}$
$T_{1.131} = \mathbb{M}(\sum_{a<I} K + count(\sum_{a<I}\Gamma_2))\,T_{1.15}$
$T_{1.14} = [(\sum_{a<I} K + count(\sum_{a<I}\Gamma_2))]\,!_{a<I}\mathbf{1} = [\sum_{a<I}(K + count(\Gamma_2))]\,!_{a<I}\mathbf{1}$
$T_{1.15} = !_{a<I}\,[(K + count(\Gamma_2))]\,\mathbf{1}$
$T_{1.2} = \mathbb{M}\,0\,(\!|([a<I]\tau_1) \multimap \tau_2|\!)$
$T_2 = [(J + count(\Gamma))]\,\mathbf{1} \multimap \mathbb{M}\,0\,(!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{2.1} = [(J + count(\Gamma))]\,\mathbf{1}$
$T_{2.2} = \mathbb{M}\,0\,((!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!))$
$T_{2.21} = (!_{a<I}\,\mathbb{M}\,0\,(\!|\tau_1|\!)) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{2.22} = [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_3 = \mathbb{M}\,0\,(\!|\tau_2|\!)$
$T_{3.1} = \mathbb{M}\,I\,(\!|\tau_2|\!)$
$T_4 = \mathbb{M}\,0\,(\!|\tau_1|\!)$

$T_{4.1} = !_{a<I}\, \mathbb{M}\, 0\, (\!|\tau_1|\!)$
$T_5 = [(K + count(\Gamma_2))]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau_1|\!)$
$T_{5.0} = !_{a<I}([(K + count(\Gamma_2))]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau_1|\!))$
$T_{5.1} = !_{a<I}\, [(K + count(\Gamma_2))]\, \mathbf{1} \multimap !_{a<I}\, \mathbb{M}\, 0\, (\!|\tau_1|\!)$

D0.7:

$$\dfrac{}{\cdot;\Theta;\Delta;\cdot;c : T_{1.15} \vdash c : T_{1.15}}\ \text{T-var}$$

D0.6:

$$\dfrac{\dfrac{\dfrac{}{\cdot;\Theta,a;\Delta,a<I;(\!|\Gamma_2|\!);\cdot \vdash e_{t2} : T_5}\ \text{IH}}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);\cdot \vdash !e_{t2} : T_{5.0}}\ \text{subExpI} \qquad D0.7}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);c : T_{1.15} \vdash coerce1\ !e_{t2}\ c : T_{4.1}}\ \text{Lemma 84}$$

D0.5:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);b : T_{2.21} \vdash b : T_{2.21}} \qquad D0.6}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);b : T_{2.21}, c : T_{1.15} \vdash b\ (coerce1\ !e_{t2}\ c) : T_{2.22}}\ \text{T-app}$$

D0.4:

$$\dfrac{D0.5 \qquad \dfrac{}{\cdot;\Theta;\Delta;\cdot;d : [I]\,\mathbf{1} \vdash d : [I]\,\mathbf{1}}}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);b : T_{2.21}, c : T_{1.15}, d : [I]\,\mathbf{1} \vdash b\ (coerce1\ !e_{t2}\ c)\ d : T_3}$$

D0.3:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash \mathsf{store}() : \mathbb{M}\, I\, [I]\, \mathbf{1}} \qquad D0.4}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);b : T_{2.21}, c : T_{1.15} \vdash E_5 : T_{3.1}}\ \text{bind}$$

D0.21:

$$\dfrac{}{\cdot;\Theta;\Delta \vdash T_{1.14} <: T_{1.15}}\ \text{sub-bSum}$$

D0.2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash !() : !_{a<I}\mathbf{1}}}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash \mathsf{store!}() : T_{1.13}} \qquad D0.21}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash \mathsf{store!}() : T_{1.131}}\ \text{T-sub} \qquad D0.3}{\cdot;\Theta;\Delta;\sum_{a<I}(\!|\Gamma_2|\!);b : T_{2.21} \vdash E_4 : T_{1.12}}\ \text{bind}$$

D0.12:

$$\dfrac{}{\cdot;\Theta;\Delta;.;a : T_{2.1} \vdash a : T_{2.1}}\ \text{T-var}$$

D0.11:

$$\dfrac{}{\cdot;\Theta;\Delta;(\!|\Gamma_1|\!);. \vdash e_{t1} : T_1}\ \text{IH1}$$

D0.1:

$$\dfrac{\dfrac{D0.11 \qquad D0.12}{\cdot;\Theta;\Delta;(\!|\Gamma_1|\!);a : T_{2.1} \vdash e_{t1}\ a : T_{2.2}}\ \text{app} \qquad D0.2}{\cdot;\Theta;\Delta;(\!|\Gamma_1|\!) \oplus \sum_{a<I}(\!|\Gamma_2|\!);a : T_{2.1} \vdash E_3 : T_{1.12}}\ \text{bind}$$

D0:

$$\dfrac{\dfrac{}{\cdot;\Theta;\Delta;\cdot;\cdot \vdash \mathsf{store}() : T_{1.11}} \qquad D0.1}{\cdot;\Theta;\Delta;(\!|\Gamma_1|\!) \oplus \sum_{a<I}(\!|\Gamma_2|\!);\cdot \vdash E_2 : T_{0.3}}\ \text{bind}$$

D0.0:

$$\dfrac{\overline{\Theta; \Delta \vdash \Gamma' \sqsubseteq \Gamma_1 \oplus \sum_{a<I} \Gamma_2}\ \text{By inversion}}{\Theta; \Delta \vdash (\!|\Gamma'|\!) <: (\!|\Gamma_1 \oplus \sum_{a<I} \Gamma_2|\!)}\ \text{Lemma 77}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\cdot; \Theta; \Delta; \cdot; p : T_{0.1} \vdash p : T_{0.1}} \qquad D0}{\cdot; \Theta; \Delta; (\!|\Gamma_1|\!) \oplus \sum_{a<I} (\!|\Gamma_2|\!); p : T_{0.1} \vdash E_1 : T_{0.2}}\ \text{release}}{\cdot; \Theta; \Delta; (\!|\Gamma_1|\!) \oplus \sum_{a<I} (\!|\Gamma_2|\!); \cdot \vdash E_0 : T_{0.11}}}{\cdot; \Theta; \Delta; (\!|\Gamma_1|\!) \oplus (\!|\sum_{a<I} \Gamma_2|\!); \cdot \vdash E_0 : T_{0.11}}\ \text{Lemma 76}}{\cdot; \Theta; \Delta; (\!|\Gamma_1 \oplus \sum_{a<I} \Gamma_2|\!); \cdot \vdash E_0 : T_{0.11}}\ \text{Lemma 75} \qquad D0.0}{\cdot; \Theta; \Delta; (\!|\Gamma'|\!); \cdot \vdash E_0 : T_0}\ \text{T-sub,T-weaken}$$

- fix:

$$\dfrac{\begin{array}{c}\Theta, b; \Delta, b < L; \Gamma, x : [a < I]\sigma \vdash_K e : \tau \rightsquigarrow e_t \\ \tau[0/a] <: \mu \qquad \Theta, a, b; \Delta, a < I, b < L \vdash \tau[(b + 1 + \bigcirc_b^{b+1,a} I)/b] <: \sigma \\ \Gamma' \sqsubseteq \sum_{b<L} \Gamma \qquad L, M \geq \bigcirc_b^{0,1} I \qquad N \geq M - 1 + \sum_{b<L} K\end{array}}{\Theta; \Delta; \Gamma' \vdash_N \mathsf{fix} x.e : \mu \rightsquigarrow E_0}\ \text{T-fix}$$

$E_0 = \mathsf{fix} Y.E_1$
$E_1 = \lambda p.E_2$
$E_2 = \mathsf{release} - = p \mathsf{\ in\ } E_3$
$E_3 = \mathsf{bind}\, A = \mathsf{store}() \mathsf{\ in\ } E_4$
$E_4 = \mathsf{let}\, !\, x = (E_{4.1}\ E_{4.2}) \mathsf{\ in\ } E_5$
$E_{4.1} = coerce1\ !Y$
$E_{4.2} = (\lambda u.!())\ A$
$E_5 = \mathsf{bind}\, C = \mathsf{store}() \mathsf{\ in\ } E_6$
$E_6 = e_t\ C$

$cost(b') \triangleq$
$if\ (0 \leq b' < (\bigcirc_b^{0,1} I(b)))\ then$
$K(b') + I(b') + count(\Gamma(b')) + (\sum_{a<I(b')} cost((b' + 1 + \bigcirc_b^{b'+1,a} I(b))))$
$else$
0

$\tau'(b') = [cost(b')]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau(b')|\!)$
$T_{0.0} = \tau'[(b' + 1 + \bigcirc_b^{b'+1,a} I)/b']$
$T_0 = [(N + count(\Gamma'))]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\mu|\!)$
$T_{0.1} = [(M - 1 + \sum_{b'<L} K) + count(\sum_{b'<L} \Gamma)]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau(0)|\!)$
$b'' = (b' + 1 + \bigcirc_b^{b'+1,a} I)$
$T_{1.0} = !_{a<I(b')}([cost(b'')]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau(b'')|\!))$
$T_1 = !_{a<I(b')}\, [cost(b'')]\, \mathbf{1} \multimap !_{a<I(b')}\, \mathbb{M}\, 0\, (\!|\tau(b'')|\!)$
$T_{1.1} = !_{a<I(b')}\, \mathbb{M}\, 0\, (\!|\tau(b'')|\!)$
$T_{1.11} = \mathbb{M}\, 0\, (\!|\tau(b'')|\!)$
$T_{1.12} = \mathbb{M}\, 0\, (\!|\sigma|\!)$
$T_2 = [\sum_{a<I(b')} cost(b'')]\, \mathbf{1}$
$T_{3.0} = \sum_{a<I} cost(b'') !_{a<I} \mathbf{1}$
$T_3 = !_{a<I} [cost(b'')] \mathbf{1}$
$T_4 = \mathbb{M}(K(b') + I(b') + count(\Gamma(b')))\, (\!|\tau(b')|\!)$
$T_{4.1} = \mathbb{M}(K(b') + I(b') + count(\Gamma(b')))\, [(K(b') + I(b') + count(\Gamma(b')))]\, \mathbf{1}$
$T_{4.2} = [(K(b') + I(b') + count(\Gamma(b')))]\, \mathbf{1}$

$T_5 = [(K(b') + I(b') + count(\Gamma(b')))]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\!|\tau(b')|\!)$
$T_{c0} = \mathbf{1} \multimap !_{a<I}\mathbf{1}$
$T_{c0.1} = [0]\,(\mathbf{1} \multimap !_{a<I}\mathbf{1})$
$T_{c1} = [\sum_{a<I} cost(b'')]\,\mathbf{1} \multimap [\sum_{a<I} cost(b'')]\, !_{a<I}\mathbf{1}$

D5.2:

$$\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;\cdot;C:T_{4.2}\vdash C:T_{4.2}}\ \text{var}$$

D5.10:

$$\dfrac{\dfrac{}{.;\Theta,b';\Delta,b'<L\vdash \tau(b'')<:\sigma}\ \text{Given}}{.;\Theta,b';\Delta,b'<L\vdash (\!|\tau(b'')|\!) <: (\!|\sigma|\!)}\ \text{Lemma 79}$$

D5.1:

$$\dfrac{\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),x:_{a<I(b')}T_{1.12};\cdot\vdash e_t:T_5}\ \text{IH} \qquad D5.10}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),x:_{a<I(b')}T_{1.11};\cdot\vdash e_t:T_5}\ \text{T-weaken}$$

D5:

$$\dfrac{D5.1 \qquad D5.2}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),x:_{a<I(b')}T_{1.11};C:T_{4.2}\vdash e_t\ C:\mathbb{M}\,0\,(\!|\tau(b')|\!)}\ \text{app}$$

D4:

$$\dfrac{\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;\cdot;\cdot\vdash \mathsf{store}():T_{4.1}} \qquad D5}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),x:_{a<I(b')}T_{1.1};C:T_{4.2}\vdash E_5:T_4}$$

D3.2:

$$\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;Y:_{a<I}T_{0.0};\cdot\vdash !Y:T_{1.0}}\ \text{Lemma 81}$$

D3.11:

$$\dfrac{D3.2}{\cdot;\Theta,b';\Delta,b'<L;Y:_{a<I}T_{0.0};\cdot\vdash coerce1\ (!Y):T_1}\ \text{Lemma 84}$$

D3.12:

$$\dfrac{}{.;\Theta,b';\Delta,b'<L\vdash T_{3.0}<:T_3}\ \text{sub-bSum}$$

Dc2:

$$\dfrac{}{\cdot;\Theta,b';\Delta,b'<L\vdash T_{c0.1}<:T_{c1}}\ \text{sub-potArrow}$$

Dc1:

$$\dfrac{\dfrac{\dfrac{}{\cdot;\Theta,b';\Delta,b'<L,a<I;\cdot;\cdot\vdash ():\mathbf{1}}\ \text{T-unit}}{\cdot;\Theta,b';\Delta,b'<L;\cdot;u:\mathbf{1}\vdash !():!_{a<I}\mathbf{1}}\ \text{T-subExpI,T-weaken}}{\cdot;\Theta,b';\Delta,b'<L;\cdot;\cdot\vdash \lambda u.!():T_{c0}}\ \text{T-lam}$$

Dc:

$$\dfrac{\dfrac{Dc1 \qquad \dfrac{}{\cdot;\Theta,b';\Delta,b'<L\vdash T_{c0}<:T_{c0.1}}\ \text{sub-potZero}}{\cdot;\Theta,b';\Delta,b'<L;\cdot;\cdot\vdash \lambda u.!():T_{c0.1}}\ \text{T-sub} \qquad Dc2}{\cdot;\Theta,b';\Delta,b'<L;\cdot;\cdot\vdash \lambda u.!():T_{c1}}\ \text{T-sub}$$

D3.1:

$$\dfrac{D3.11 \qquad \dfrac{\dfrac{Dc \qquad \dfrac{}{\cdot;\Theta,b';\Delta,b'<L;\cdot;A:T_2\vdash A:T_2}\ \text{var}}{\cdot;\Theta,b';\Delta,b'<L;\cdot;A:T_2\vdash (\lambda u.!())\ A:T_{3.0}}\ \text{T-app} \qquad D3.12}{\cdot;\Theta,b';\Delta,b'<L;\cdot;A:T_2\vdash (\lambda u.!())\ A:T_3}\ \text{T-sub}}{\cdot;\Theta,b';\Delta,b'<L;Y:_{a<I}T_{0.0};A:T_2\vdash E_{4.1}\ E_{4.2}:T_{1.1}}\ \text{app}$$

D3:

$$\dfrac{D3.1 \qquad D4}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),Y:_{a<I}T_{0.0};A:T_2\vdash E_4:T_4}$$

D2:

$$\dfrac{\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma'|\!);\cdot\vdash \mathsf{store}():\mathbb{M}(\sum\limits_{a<I(b')} cost(b''))T_2} \qquad D3}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma'|\!),Y:_{a<I}T_{0.0};\cdot\vdash E_3:\mathbb{M}(\,cost(b'))(\!|\tau(b')|\!)}$$

D1:

$$\dfrac{\dfrac{}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!);p:[cost(b')]\,\mathbf{1}\vdash p:[cost(b')]\,\mathbf{1}} \quad D2}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),Y:_{a<I}T_{0.0};p:[cost(b')]\,\mathbf{1}\vdash E_2:\mathbb{M}(0)(\!|\tau(b')|\!)}\ \text{release}$$

D0:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D1}{\cdot;\Theta,b';\Delta,b'<L;(\!|\Gamma|\!),Y:_{a<I}T_{0.0};\cdot\vdash E_1:\tau'(b')}\ \text{T-fix}}{\cdot;\Theta;\Delta;\sum_{a<L}(\!|\Gamma|\!);\cdot\vdash E_0:\tau'(0)}\ \text{Claim}}{\cdot;\Theta;\Delta;\sum_{a<L}(\!|\Gamma|\!);\cdot\vdash E_0:T_{0.1}}\ \text{Lemma 76}}{\cdot;\Theta;\Delta;(\!|\sum_{a<L}\Gamma|\!);\cdot\vdash E_0:T_{0.1}}\ \text{Lemma 77,T-weaken}}{\cdot;\Theta;\Delta;(\!|\Gamma'|\!);\cdot\vdash E_0:T_{0.1}}$$

Main derivation:

$$\dfrac{D0}{\cdot;\Theta;\Delta;(\!|\Gamma'|\!);\cdot\vdash E_0:T_0}\ \text{T-sub}$$

Claim:
$\tau'(0)=[(M-1+\sum_{b'<L}K)+count(\sum_{b'<L}\Gamma)]\,\mathbf{1}\multimap\mathbb{M}\,0\,(\!|\tau(0)|\!)$
Proof.
It suffices to prove that
$cost(0)=(M-1+\sum_{b'<L}K)+count(\sum_{b'<L}\Gamma)$

From Definition of *cost* we know that

$cost(0)$	$=$	$(\sum_{b'<L}I(b')+\sum_{b'<L}K(b'))+\sum_{b'<L}count(\Gamma)$	
	$=$	$(M-1+\sum_{b'<L}K(b'))+\sum_{b'<L}count(\Gamma)$	Definition of I and M
	$=$	$(M-1+\sum_{b'<L}K)+count(\sum_{b'<L}\Gamma)$	Lemma 78

□

□

Lemma 75 (Relation b/w dlPCF context and its translation - binary sum)**.** $\forall\Gamma_1,\Gamma_2\in dlPCF.$
$(\!|\Gamma_1\oplus\Gamma_2|\!)=(\!|\Gamma_1|\!)\oplus(\!|\Gamma_2|\!)$

Proof. Proof by induction on Γ_1

$\underline{\Gamma_1=.}$

$(\!|.\oplus\Gamma_2|\!)$	$=$	$(\!|\Gamma_2|\!)$	Definition 55
	$=$	$(\!|.|\!)+(\!|\Gamma_2|\!)$	Definition 56

$\underline{\Gamma_1=\Gamma_1',x:[-]-}$

When $x:[-]-\notin\Gamma_2$

$(\!|\Gamma_1',x:[a<I]\tau\oplus\Gamma_2|\!)$	$=$	$(\!|(\Gamma_1'\oplus\Gamma_2),x:[a<I]\tau|\!)$	Definition 55
	$=$	$(\!|(\Gamma_1'\oplus\Gamma_2)|\!),x:_{a<I}\mathbb{M}\,0\,(\!|\tau|\!)$	Definition 73
	$=$	$((\!|\Gamma_1'|\!)\oplus(\!|\Gamma_2|\!)),x:_{a<I}\mathbb{M}\,0\,(\!|\tau|\!)$	IH
	$=$	$(\!|\Gamma_1'|\!),x:_{a<I}\mathbb{M}\,0\,(\!|\tau|\!)\oplus(\!|\Gamma_2|\!)$	Definition 56
	$=$	$(\!|\Gamma_1',x:[a<I]\tau|\!)\oplus(\!|\Gamma_2|\!)$	Definition 56

When $x:[b<J]\tau[I+b/c]\in\Gamma_2$
Let $(\!|\Gamma_1',x:[a<I]\tau[a/c]\oplus\Gamma_2',x:[b<J]\tau[I+b/c]|\!)=\Gamma_r$

Γ_r	$=$	$(\!|(\Gamma_1'\oplus\Gamma_2'),x:[c<(I+J)]\tau|\!)$	Definition 55
	$=$	$(\!|(\Gamma_1'\oplus\Gamma_2')|\!),x:_{c<(I+J)}\mathbb{M}\,0\,(\!|\tau|\!)$	Definition 73
	$=$	$((\!|\Gamma_1'|\!)\oplus(\!|\Gamma_2'|\!)),x:_{c<(I+J)}\mathbb{M}\,0\,(\!|\tau|\!)$	IH
	$=$	$(\!|\Gamma_1'|\!),x:_{a<I}\mathbb{M}\,0\,(\!|\tau|\!)[a/c]\oplus(\!|\Gamma_2'|\!),x:_{b<J}\mathbb{M}\,0\,(\!|\tau|\!)[I+b/c]$	Definition 56
	$=$	$(\!|\Gamma_1'|\!),x:_{a<I}\mathbb{M}\,0\,(\!|\tau[a/c]|\!)\oplus(\!|\Gamma_2'|\!),x:_{b<J}\mathbb{M}\,0\,(\!|\tau[I+b/c]|\!)$	Lemma 80
	$=$	$(\!|\Gamma_1',x:[a<I]\tau[a/c]|\!)\oplus(\!|\Gamma_2',x:[b<J]\tau[I+b/c]|\!)$	Definition 56

□

Lemma 76 (Relation b/w dlPCF context and its translation - bounded sum)**.** $\forall\Gamma\in dlPCF.$
$(\!|\sum_{a<I}\Gamma|\!)=\sum_{a<I}(\!|\Gamma|\!)$

Proof. Proof by induction on Γ

$\underline{\Gamma = .}$

$$
\begin{array}{llll}
(\!|\sum_{a<I} .|\!) & = & (\!|.|\!) & \text{Definition 53} \\
 & = & . & \text{Definition 73} \\
 & = & \sum_{a<I} (\!|.|\!) & \text{Definition 54}
\end{array}
$$

$\underline{\Gamma = \Gamma', x : [-]-}$

Let $(\!|\sum_{a<I}(\Gamma', x : [b < J]\sigma[\sum_{d<a} J[d/a] + b/c])|\!) = \Gamma_r$

$$
\begin{array}{llll}
\Gamma_r & = & (\!|\sum_{a<I}(\Gamma'), x : [c < \sum_{a<I} J]\sigma|\!) & \text{Definition 53} \\
 & = & (\!|\sum_{a<I}(\Gamma')|\!), x :_{c<\sum_{a<I} J} \mathbb{M}\, 0\, (\!|\sigma|\!) & \text{Definition 73} \\
 & = & \sum_{a<I} (\!|\Gamma'|\!), x :_{c<\sum_{a<I} J} \mathbb{M}\, 0\, (\!|\sigma|\!) & \text{IH} \\
 & = & \sum_{a<I} ((\!|\Gamma'|\!), x :_{b<J} \mathbb{M}\, 0\, (\!|\sigma|\!)[\sum_{d<a} J[d/a] + b/c]) & \text{Definition 54} \\
 & = & \sum_{a<I} ((\!|\Gamma'|\!), x :_{b<J} \mathbb{M}\, 0\, (\!|\sigma[\sum_{d<a} J[d/a] + b/c]|\!)) & \text{Lemma 80} \\
 & = & \sum_{a<I} (\!|\Gamma', x : [b < J]\sigma[\sum_{d<a} J[d/a] + b/c]|\!) & \text{Definition 73}
\end{array}
$$

∎

Lemma 77 (Relation b/w dlPCF context and its translation - subtyping)**.** $\forall \Gamma, \Gamma' \in dlPCF.$
$\Theta; \Delta \models \Gamma_1 \sqsubseteq \Gamma_2 \implies .; \Theta; \Delta \models (\!|\Gamma_1|\!) <: (\!|\Gamma_2|\!)$

Proof. Proof by induction on the $\Theta; \Delta \vdash \Gamma_1 \sqsubseteq \Gamma_2$ relation

1. dlpcf-sub-mBase:

$$
\dfrac{}{.; \Theta; \Delta \vdash (\!|\Gamma_1|\!) <: .}\ \text{sub-mBase}
$$

2. dlpcf-sub-mInd:

D4:

$$
\dfrac{\dfrac{}{.; \Theta; \Delta \vdash \Gamma_1/x <: \Gamma_2}\ \text{By inversion}}{.; \Theta; \Delta \vdash (\!|\Gamma_1|\!)/x <: (\!|\Gamma_2|\!)}\ \text{IH}
$$

D3:

$$
\dfrac{}{\Theta; \Delta \vdash I \leq J}\ \text{By inversion}
$$

D2:

$$
\dfrac{\dfrac{\dfrac{}{.; \Theta, a; \Delta, a < I \vdash \tau' <: \tau}\ \text{By inversion}}{.; \Theta, a; \Delta, a < I \vdash (\!|\tau'|\!) <: (\!|\tau|\!)}\ \text{Lemma 79}}{.; \Theta, a; \Delta, a < I \vdash \mathbb{M}\, 0\, (\!|\tau'|\!) <: \mathbb{M}\, 0\, (\!|\tau|\!)}
$$

D1:

$$
\dfrac{\dfrac{}{x : [a < J]\tau' \in \Gamma_1}\ \text{By inversion}}{x :_{a<J} \mathbb{M}\, 0\, (\!|\tau'|\!) \in (\!|\Gamma_1|\!)}\ \text{Definition 73}
$$

Main derivation:

$$
\dfrac{\dfrac{D1 \qquad D2 \qquad D3 \qquad D4}{.; \Theta; \Delta \vdash (\!|\Gamma_1|\!) <: (\!|\Gamma_2'|\!), x :_{a<I} \mathbb{M}\, 0\, (\!|\tau|\!)}}{.; \Theta; \Delta \vdash (\!|\Gamma_1|\!) <: (\!|\Gamma_2', x : [a < I]\tau|\!)}
$$

∎

Lemma 78. $\forall L, \Gamma.$
$\sum_{a<L} count(\Gamma) = count(\sum_{a<L} \Gamma)$

Proof. By induction on Γ

$\underline{\Gamma = .}$

From Definition of *count* we know that $count(.) = 0$ therefore
$\sum_{a<L} count(.) = 0$

From Definition 54 we know that $\sum_{a<L} . = .$
Therefore again from Definition of *count* we know that $count(.) = 0$
And we are done

$\underline{\Gamma = \Gamma', x :_{b<J} \tau}$

$$
\begin{array}{rcll}
count(\sum_{a<L} \Gamma', x :_{b<J} \tau) & = & count(\sum_{a<L} \Gamma', x :_{c<\sum_{a<L} J} \sigma) & \text{Definition 54} \\
 & & \text{where } \tau = \sigma[(\sum_{d<a} J[d/a] + b)/c] & \\
 & = & count(\sum_{a<L} \Gamma') + \sum_{a<L} J & \text{Definition } count(.) \\
 & = & \sum_{a<L} count(\Gamma') + \sum_{a<L} J & \text{IH} \\
 & = & \sum_{a<L} count(\Gamma', x :_{b<J} \tau) &
\end{array}
$$

□

Lemma 79 (Subtyping is preserved by translation)**.** $\Theta; \Delta \vdash^D \sigma <: \tau \implies \Theta; \Delta \vdash^A (\!|\sigma|\!) <: (\!|\tau|\!)$

Proof. By induction on $\Theta; \Delta \vdash^D \sigma <: \tau$

1. $[a < I]\sigma_1 \multimap \sigma_2 <: [a < J]\tau_1 \multimap \tau_2$:

   D1:

$$
\dfrac{\dfrac{\dfrac{}{\Theta; \Delta \vdash^A I \leq J}\text{By inversion}}{\Theta; \Delta \vdash^A [J]\,\mathbf{1} <: [I]\,\mathbf{1}} \qquad \dfrac{\dfrac{}{\Theta; \Delta \vdash^A (\!|\sigma_2|\!) <: (\!|\tau_2|\!)}\text{IH2}}{\Theta; \Delta \vdash^A \mathbb{M}\,0\,(\!|\sigma_2|\!) <: \mathbb{M}\,0\,(\!|\tau_2|\!)}}{\Theta; \Delta \vdash^A [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\sigma_2|\!) <: [J]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)}
$$

   Main derivation:

$$
\dfrac{\dfrac{\dfrac{}{\Theta, a; \Delta \vdash^A I \leq J}\text{By inversion} \qquad \dfrac{\dfrac{}{\Theta; \Delta \vdash^A (\!|\tau_1|\!) <: (\!|\sigma_1|\!)}\text{IH1}}{\Theta; \Delta \vdash^A \mathbb{M}\,0\,(\!|\tau_1|\!) <: \mathbb{M}\,0\,(\!|\sigma_1|\!)}}{\Theta; \Delta \vdash^A !_{a<J}\,\mathbb{M}\,0\,(\!|\tau_1|\!) <: !_{a<I}\,\mathbb{M}\,0\,(\!|\sigma_1|\!)} \qquad D1}{\Theta; \Delta \vdash^A !_{a<I}\,\mathbb{M}\,0\,(\!|\sigma_1|\!) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\sigma_2|\!) <: !_{a<J}\,\mathbb{M}\,0\,(\!|\tau_1|\!) \multimap [J]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)}
$$

□

Lemma 80 (Index Substitution lemma)**.** $\forall \tau \in dlPCF, J.$
$(\!|\tau|\!)[J/b] = (\!|\tau[J/b]|\!)$

Proof. By induction on τ

1. $\tau = b$:

$$
\begin{array}{ll}
 & (\!|b|\!)[J/b] \\
= & b \\
= & (\!|b[J/b]|\!)
\end{array}
$$

2. $\tau = [a < I]\tau_1 \multimap \tau_2$:

$$
\begin{array}{lll}
 & (\!|[a < I]\tau_1 \multimap \tau_2|\!)[J/b] & \\
= & !_{a<I}\,\mathbb{M}\,0(\!|\tau_1|\!) \multimap [I]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)[J/b] & \\
= & !_{a<I[J/b]}\,\mathbb{M}\,0(\!|\tau_1|\!)[J/b] \multimap [I][J/b]\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2|\!)[J/b] & \\
= & !_{a<I[J/b]}\,\mathbb{M}\,0(\!|\tau_1[J/b]|\!) \multimap [I][J/b]\mathbf{1} \multimap \mathbb{M}\,0\,(\!|\tau_2[J/b]|\!) & \text{(From IH)} \\
= & (\!|[a < I[J/b]]\tau_1[J/b] \multimap \tau_2[J/b]|\!) &
\end{array}
$$

□

Lemma 81. $\Psi; \Theta; \Delta; x :_{a<I} \tau; \cdot \vdash !x :!_{a<I}\tau$

Proof.

$$
\dfrac{\dfrac{\dfrac{}{\Psi; \Theta, a; \Delta, a < I; x :_{b<1} \tau[a + b/a]; \cdot \vdash x : \tau}\text{T-var2}}{\Psi; \Theta; \Delta; \sum\limits_{a<I} x :_{b<1} \tau[a + b/a]; \cdot \vdash !x :!_{a<I}\tau}\text{T-subExpI}}{\Psi; \Theta; \Delta; x :_{a<I} \tau; \cdot \vdash !x :!_{a<I}\tau}\text{Lemma 82}
$$

□

Lemma 82. $\sum_{a<I} x :_{b<1} \tau[a + b/a] = x_{a<I}\tau$

Proof. It suffices to prove that
$\sum_{a<I} x :_{b<1} \tau[a + b/a] = x_{c<I}\tau[c/a]$

From Definition 54 it suffices to prove that
$\sum_{a<I} x :_{b<1} \tau[a + b/a] = x_{c<\sum_{a<I} 1}\tau[c/a]$

Again from Definition 54 it suffices to prove that
$\tau[c/a][(\sum_{d<a} 1[d/a] + b)/c] = \tau[a + b/a]$

$\tau[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$
$\tau[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$

$\tau[c/a][(a+b)/c] =$
$\tau[(a+b)/a]$

So, we a re done $\square$

Definition 83 (Coercion function)**.** $coerce1\ F\ X \triangleq$
$\mathsf{let}\,!\,f = F\ \mathsf{in}\ \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x)$

Lemma 84 (Coerce is well-typed)**.** $\cdot;\cdot;\cdot;\cdot;\cdot \vdash coerce1 :!_{a<I}(\tau_1 \multimap \tau_2) \multimap !_{a<I}\tau_1 \multimap !_{a<I}\tau_2$

Proof. D2.2

$$\frac{}{\cdot; a; a < I; x :_{b<1} \tau_1[a+b/a]; \cdot \vdash x : \tau_1}$$

D2.1:

$$\frac{}{\cdot; a; a < I; f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a]; \cdot \vdash f : \tau_1 \multimap \tau_2}$$

D2:

$$\dfrac{\dfrac{\dfrac{D2.1 \qquad D2.2}{\cdot; a; a < I; f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a], x :_{b<1} \tau_1[a+b/a]; \cdot \vdash (f\ x) : \tau_2}}{\cdot;\cdot;\cdot; \sum_{a<I} f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a], x :_{b<1} \tau_1[a+b/a]; \cdot \vdash !(f\ x) :!_{a<I}\tau_2}\ \text{T-subExpI}}{\cdot;\cdot;\cdot; f :_{a<I} (\tau_1 \multimap \tau_2), x :_{a<I} \tau_1; \cdot \vdash !(f\ x) :!_{a<I}\tau_2}\ \text{Lemma 85}$$

D1:

$$\dfrac{\dfrac{}{\cdot;\cdot;\cdot; f :_{a<I} (\tau_1 \multimap \tau_2); X :!_{a<I}\tau_1 \vdash !(f\ x)} \qquad D2}{\cdot;\cdot;\cdot; f :_{a<I} (\tau_1 \multimap \tau_2); \cdot \vdash \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x)}$$

D0:

$$\dfrac{\dfrac{}{\cdot;\cdot;\cdot;\cdot; F :!_{a<I}(\tau_1 \multimap \tau_2) \vdash F :!_{a<I}(\tau_1 \multimap \tau_2)}\ \text{T-var1} \qquad D1}{\cdot;\cdot;\cdot;\cdot; F :!_{a<I}(\tau_1 \multimap \tau_2) \vdash \mathsf{let}\,!\,f = F\ \mathsf{in}\ \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x)}$$

Main derivation:

$$\dfrac{\dfrac{D0}{\cdot;\cdot;\cdot;\cdot; F :!_{a<I}(\tau_1 \multimap \tau_2) \vdash \lambda X.\ \mathsf{let}\,!\,f = F\ \mathsf{in}\ \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x)}}{\cdot;\cdot;\cdot;\cdot;\cdot \vdash \lambda F.\lambda X.\ \mathsf{let}\,!\,f = F\ \mathsf{in}\ \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x) :!_{a<I}(\tau_1 \multimap \tau_2) \multimap !_{a<I}\tau_1 \multimap !_{a<I}\tau_2}$$

$\square$

Lemma 85. $\sum_{a<I} f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a], x :_{b<1} \tau_1[a+b/a] = f :_{a<I} \tau_1 \multimap \tau_2, x :_{a<I} \tau_1$

Proof. It suffices to prove that
$\sum_{a<I} f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a], x :_{b<1} \tau_1[a+b/a] = f :_{c<I} (\tau_1 \multimap \tau_2)[c/a], x :_{c<I} \tau_1[c/a]$

From Definition 54 it suffices to prove that
$\sum_{a<I} f :_{b<1} (\tau_1 \multimap \tau_2)[a+b/a], x :_{b<1} \tau_1[a+b/a] = f :_{c<\sum_{a<I} 1} (\tau_1 \multimap \tau_2)[c/a], x :_{c<\sum_{a<I} 1} \tau_1[c/a]$

Again from Definition 54 it suffices to prove that

1. $(\tau_1 \multimap \tau_2)[c/a][(\sum_{d<a} 1[d/a] + b)/c] = (\tau_1 \multimap \tau_2)[a+b/a]$:
   $(\tau_1 \multimap \tau_2)[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$
   $(\tau_1 \multimap \tau_2)[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$
   $(\tau_1 \multimap \tau_2)[c/a][(a+b)/c] =$
   $(\tau_1 \multimap \tau_2)[(a+b)/a]$
2. $\tau_1[c/a][(\sum_{d<a} 1[d/a] + b)/c] = \tau_1[a+b/a]$:
   $\tau_1[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$
   $\tau_1[c/a][(\sum_{d<a} 1[d/a] + b)/c] =$
   $\tau_1[c/a][(a+b)/c] =$
   $\tau_1[(a+b)/a]$

So, we are done $\square$

B.5.2 Cross-language model: dlPCF to λ-amor

Definition 86 (Logical relation for dlPCF to λ-Amor)**.**

$$
\begin{array}{lll}
\lfloor \mathsf{b} \rfloor_V & \triangleq & \{({}^s v, {}^t v) \mid {}^s v \in [\![\mathsf{b}]\!] \wedge {}^t v \in [\![\mathsf{b}]\!] \wedge {}^s v = {}^t v\} \\
\lfloor [a < I]\tau_1 \multimap \tau_2 \rfloor_V & \triangleq & \{(\lambda x.e_s, \lambda x.\lambda p.\, \mathsf{let}\,!\, x = y \,\mathsf{in}\, e_t) \mid \forall e'_s, e'_t. \\
& & (e'_s, e'_t) \in \lfloor [a < I]\tau_1 \rfloor_{NE} \implies (e_s[e'_s/x], e_t[e'_t/y][()/p]) \in \lfloor \tau_2 \rfloor_E\} \\
\lfloor \tau \rfloor_E & \triangleq & \{(e_s, e_t) \mid \forall^s v.e_s \Downarrow {}^s v \implies \exists^t v_t, {}^t v_f, J.e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau \rfloor_V\} \\
\lfloor [a < I]\tau \rfloor_{NE} & \triangleq & \{(e_s, e_t) \mid \exists e'_t.e_t = coerce1\ !e'_t\ !() \wedge \forall 0 \le i < I.(e_s, e'_t()) \in \lfloor \tau[i/a] \rfloor_E\}
\end{array}
$$

Definition 87 (Interpretation of typing contexts)**.**

$$
\begin{array}{lll}
\lfloor \Gamma \rfloor_E & = & \{(\delta_s, \delta_t) \mid \\
& & (\forall x : [a < J]\tau \in dom(\Gamma).\forall 0 \le j < J.(\delta_s(x), \delta_t(x)) \in \lfloor \tau[j/a] \rfloor_E)
\end{array}
$$

Theorem 88 (Fundamental theorem)**.** $\forall \Theta, \Delta, \Gamma, \tau, e_s, e_t, I, \delta_s, \delta_t.$
$\Theta; \Delta; \Gamma \vdash_I e_s : \tau \rightsquigarrow e_t \wedge (\delta_s, \delta_t) \in \lfloor \Gamma\ \iota \rfloor_E \wedge .\models \Delta\ \iota$
$\implies$
$(e_s\delta_s, e_t\ ()\ \delta_t) \in \lfloor \tau\ \iota \rfloor_E$

Proof. Proof by induction on the translation relation:

1. var:

$$
\frac{\Theta; \Delta \models J \ge 0 \qquad \Theta; \Delta \models I \ge 1 \qquad \Theta; \Delta \vdash \tau'[0/a] <: \tau \qquad \Theta; \Delta \models [a < I]\tau' \Downarrow \qquad \Theta; \Delta \models \Gamma \Downarrow}{\Theta; \Delta; \Gamma, x : [a < I]\tau' \vdash_J x : \tau[0/a] \rightsquigarrow \lambda p.\mathsf{release} - = p \ \mathsf{in}\ \ \mathsf{bind} - = \uparrow^1 \ \mathsf{in}\ x} \text{var}
$$

$E_1 = \lambda p.\mathsf{release} - = p \ \mathsf{in}\ \ \mathsf{bind} - = \uparrow^1 \ \mathsf{in}\ x$
Given: $(\delta_s, \delta_t) \in \lfloor \Gamma, x \rfloor_E$
To prove: $(x\delta_s, E_1\ ()\delta_t) \in \lfloor \tau[0/a] \rfloor_E$

This means from Definition 86 we need to prove that
$\forall^s v.x\delta_s \Downarrow {}^s v \implies \exists^t v_t, {}^t v_f, J'.E_1\ () \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau[0/a] \rfloor_V$

This means that given some ${}^s v$ s.t $x\delta_s \Downarrow {}^s v$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J'.E_1\ () \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau[0/a] \rfloor_V$ (F-DA-V0)

Since we are given that $(\delta_s, \delta_t) \in \lfloor \Gamma, x \rfloor_E$ therefore from Definition 87 we know that
$\forall y : [a < J]\tau \in dom(\Gamma, x).\ \forall 0 \le i < J.(\delta_s(y), \delta_t(y)) \in \lfloor \tau[i/a] \rfloor_E$

This means we also have $(\delta_s(x), \delta_t(x)) \in \lfloor \tau[0/a] \rfloor_E$. This further means that from Definition 86 we have
$\forall^s v''.\delta_s(x) \Downarrow {}^s v'' \implies \exists J'', {}^t v''_t, {}^t v''_f.\delta_t(x) \Downarrow {}^t v''_t \Downarrow^{J''} {}^t v''_f \wedge ({}^s v'', {}^t v''_f) \in \lfloor \tau[0/a] \rfloor_V$ (F-DA-V1)

We instantiate (F-DA-V1) with ${}^s v$ and in order to prove (F-DA-V0) we choose J' as J'', ${}^t v_t$ as ${}^t v''_t$ and ${}^t v_f$ as ${}^t v''_f$ and we get the desired from (F-DA-V1).

2. lam:

$$
\frac{\Theta; \Delta; \Gamma, x : [a < I]\tau_1 \vdash_J e : \tau_2 \rightsquigarrow e_t}{\begin{array}{c}\Theta; \Delta; \Gamma \vdash_J \lambda x.e : ([a < I].\tau_1) \multimap \tau_2 \rightsquigarrow \\ \lambda p_1.\, \mathsf{ret}\, \lambda y.\lambda p_2.\, \mathsf{let}\,!\, x = y \ \mathsf{in}\ \mathsf{release} - = p_1 \ \mathsf{in}\ \ \mathsf{release} - = p_2 \ \mathsf{in}\ \ \mathsf{bind}\, a = \mathsf{store}() \ \mathsf{in}\ e_t\ a\end{array}} \text{lam}
$$

$E_1 = \lambda p_1.\, \mathsf{ret}\, \lambda y.\lambda p_2.\, \mathsf{let}\,!\, x = y \ \mathsf{in}\ \mathsf{release} - = p_1 \ \mathsf{in}\ \ \mathsf{release} - = p_2 \ \mathsf{in}\ \ \mathsf{bind}\, a = \mathsf{store}() \ \mathsf{in}\ e_t\ a$
$E_2 = \lambda y.\lambda p_2.\, \mathsf{let}\,!\, x = y \ \mathsf{in}\ E_3$
$E_3 = \mathsf{release} - = p_1 \ \mathsf{in}\ \ \mathsf{release} - = p_2 \ \mathsf{in}\ \ \mathsf{bind}\, a = \mathsf{store}() \ \mathsf{in}\ e_t\ a$
Given: $(\delta_s, \delta_t) \in \lfloor \Gamma \rfloor_E$
To prove: $(\lambda x.e\delta_s, E_1()\delta_t) \in \lfloor ([a < I].\tau_1) \multimap \tau_2 \rfloor_E$

This means from Definition 86 we need to prove that
$\forall^s v.\lambda x.e\delta_s \Downarrow {}^s v \implies \exists J', {}^t v_t, {}^t v_f.E_1() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor ([a < I].\tau_1) \multimap \tau_2 \rfloor_V$

This means that given some ${}^s v$ s.t $\lambda x.e\delta_s \Downarrow {}^s v$ it suffices to prove that
$\exists J', {}^t v_t, {}^t v_f.E_1() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor ([a < I].\tau_1) \multimap \tau_2 \rfloor_V$ (F-DA-L0)

We know that ${}^s v = \lambda x.e\delta_s$. Also from E-app, E-ret we know that ${}^t v_f = E_2$ and $J' = 0$

Therefore it suffices to show $(\lambda x.e\ \delta_s, E_2) \in \lfloor(([a < I].\tau_1) \multimap \tau_2)\iota\rfloor_V$

From Definition 86 it further suffices to prove that
$\forall e'_s, e'_t.(e'_s, e'_t) \in \lfloor[a < I]\tau_1\iota\rfloor_{NE} \implies (e_s[e'_s/x], E_3[e'_t/y][()/p_2]) \in \lfloor\tau_2\ \iota\rfloor_E$ (F-DA-L1)

This means given some e'_s, e'_t s.t $(e'_s, e'_t) \in \lfloor[a < I]\tau_1\iota\rfloor_{NE}$. We need to prove that
$(e_s[e'_s/x], E_2[e'_t/x][()/p_2]) \in \lfloor\tau_2\ \iota\rfloor_E$ (F-DA-L1.1)

Since $(e'_s, e'_t) \in \lfloor[a < I]\tau_1\iota\rfloor_{NE}$ therefore from Definition 86 we have
$\exists e''_t.e'_t = coerce1\ !e''_t\ !() \wedge \forall 0 \leq i < I.(e'_s, e''_t()) \in \lfloor\tau_1[i/a]\ \iota\rfloor_E$

Let
$\delta'_s = \delta_s \cup \{x \mapsto e'_s\}$ and
$\delta'_t = \delta_t \cup \{x \mapsto e''_t\ ()\}$

From Definition 87 we know that
$(\delta'_s, \delta'_t) \in \lfloor\Gamma, x : [a < I]\tau_1\ \iota\rfloor_E$

Therefore from IH we have
$(e_s\ \delta'_s, e_t()\ \delta'_t) \in \lfloor\tau_2\ \iota\rfloor_E$ (F-DA-L2)

This means from Definition 86 we have
$\forall^s v_b.e_s\ \delta'_s \Downarrow {}^s v_b \implies \exists J_b, {}^t v_{t1}, {}^t v_b.e_t()\ \delta'_t \Downarrow {}^t v_{t1} \Downarrow^{J_b} {}^t v_b \wedge ({}^s v_b, {}^t v_b) \in \lfloor\tau_2\ \iota\rfloor_V$ (F-DA-L3)

Applying Definition 86 on (F-DA-L1.1) we need to prove
$\forall^s v_f.e_s[e'_s/x]\delta_s \Downarrow {}^s v_f \implies \exists J_1, {}^t v_t, {}^t v_f.E_2[e'_t/x][()/p_2]\delta_t \Downarrow {}^t v_t \Downarrow^{J_1} {}^t v_f \wedge ({}^s v_f, {}^t v_f) \in \lfloor\tau_2\iota\rfloor_V$

This means given some ${}^s v_f$ s.t $e_s[e'_s/x]\delta_s \Downarrow {}^s v_f$ it suffices to prove
$\exists J_1, {}^t v_t, {}^t v_f.E_2[e'_t/x][()/p_2]\delta_t \Downarrow {}^t v_t \Downarrow^{J_1} {}^t v_f \wedge ({}^s v_f, {}^t v_f) \in \lfloor\tau_2\iota\rfloor_V$ (F-DA-L4)

Therefore instantiating (F-DA-L3) with ${}^s v_f$ and we get the desired

3. app:

$$\frac{\Theta; \Delta; \Gamma \vdash_J e_1 : ([a < I].\tau_1) \multimap \tau_2 \rightsquigarrow e_{t1} \qquad \Theta, a; \Delta, a < I; \Delta \vdash_K e_2 : \tau_1 \rightsquigarrow e_{t2} \qquad \Gamma' \sqsubseteq \Gamma \oplus \sum_{a<I} \Delta \qquad H \geq J + I + \sum_{a<I} K}{\Theta; \Delta; \Gamma' \vdash_H e_1\ e_2 : \tau_2 \rightsquigarrow} \text{ app}$$

$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ e_{t2}\ c)\ d$

$E_1 = \lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d$
Given: $(\delta_s, \delta_t) \in \lfloor\Gamma'\ \iota\rfloor_E$
To prove: $(e_1\ e_2\delta_s, E_1()\delta_t) \in \lfloor\tau_2\ \iota\rfloor_E$

This means from Definition 86 we need to prove that
$\forall^s v_f.(e_1\ e_2)\delta_s \Downarrow {}^s v_f \implies \exists J', {}^t v_t, {}^t v_f.E_1() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v_f, {}^t v_f) \in \lfloor\tau_2\ \iota\rfloor_V$

This means that given some ${}^s v_f$ s.t $(e_1\ e_2)\delta_s \Downarrow {}^s v_f$ it suffices to prove that
$\exists J', {}^t v_t, {}^t v_f.E_1() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v_f, {}^t v_f) \in \lfloor\tau_2\ \iota\rfloor_V$ (F-DA-A0)

<u>IH1</u>
$(e_1\delta_s, e_{t1}()\delta_t) \in \lfloor([a < I]\tau_1 \multimap \tau_2)\ \iota\rfloor_E$

This means from Definition 86 we have
$\forall^s v_1.e_1\delta_s \Downarrow {}^s v_1 \implies \exists J_1, {}^t v'_1, {}^t v_1.e_{t1}()\delta_t \Downarrow {}^t v'_1 \Downarrow^{J_1} {}^t v_1 \wedge ({}^s v_1, {}^t v_1) \in \lfloor([a < I]\tau_1 \multimap \tau_2)\ \iota\rfloor_V$

Since we know that $(e_1\ e_2)\delta_s \Downarrow^n {}^s v_f$ therefore we know that $\exists^s v_1$ s.t $e_1\delta_s \Downarrow {}^s v_1$. Therefore we have
$\exists J_1, {}^t v'_1, {}^t v_1.e_{t1}()\delta_t \Downarrow {}^t v'_1 \Downarrow^{J_1} {}^t v_1 \wedge ({}^s v_1, {}^t v_1) \in \lfloor([a < I]\tau_1 \multimap \tau_2)\ \iota\rfloor_V$ (F-DA-A1)

Since we know that $({}^s v_1, {}^t v_1) \in \lfloor([a < I]\tau_1 \multimap \tau_2)\ \iota\rfloor_V$

Let ${}^s v_1 = \lambda x.e_{bs}$ and ${}^t v_1 = \lambda x.\lambda p.\mathsf{let}\,!\,x = y \text{ in } e_{bt}$
Therefore from Definition 86 we have
$\forall e'_s, e'_t.(e'_s, e'_t) \in \lfloor[a < I]\tau_1\ \iota\rfloor_{NE} \implies (e_{bs}[e'_s/x], e_{bt}[e'_t/x][()/p]) \in \lfloor\tau_2\ \iota\rfloor_E$ (F-DA-A2)

<u>IH2</u>
$(e_2\delta_s, e_{t2}()\delta_t) \in \lfloor\tau_1\ \iota \cup \{a \mapsto 0\}\rfloor_E$
$(e_2\delta_s, e_{t2}()\delta_t) \in \lfloor\tau_1\ \iota \cup \{a \mapsto 1\}\rfloor_E$
...

$(e_2\delta_s, e_{t2}()\delta_t) \in \lfloor \tau_1\ \iota \cup \{a \mapsto I-1\} \rfloor_E$ (F-DA-A3)

We claim that
$(e_2\delta_s, coerce\ !e_{t2}\ !()\delta_t) \in \lfloor [a < I]\tau_1\ \iota \rfloor_{NE}$

From Definition 83 we know that
$coerce\ F\ X \triangleq$
$\mathsf{let}\,!\,f = F\ \mathsf{in}\ \mathsf{let}\,!\,x = X\ \mathsf{in}!(f\ x)$

therefore the desired holds from Definition 86 and (F-DA-A3)

Instantiating (F-DA-A2) with $e_2\delta_s, coerce\ !e_{t2}\ !()\delta_t$ we get
$(e_{bs}[e_2\delta_s/x], e_{bt}[coerce\ !e_{t2}\ !()\delta_t/x]()) \in \lfloor \tau_2\ \iota \rfloor_E$ (F-DA-A4)

This further means that from Definition 86 we have
$\forall^s v_{bf}.e_{bs}[e_2\delta_s/x] \Downarrow {}^s v_{bf} \implies \exists J_2, {}^t v_{tb}, {}^t v_{bf}.e_{bt}[coerce\ !e_{t2}\ !()\delta_t/x]() \Downarrow {}^t v_{tb} \Downarrow^{J_2} {}^t v_{bf} \wedge ({}^s v_{bf}, {}^t v_{bf}) \in \lfloor \tau_2\ \iota \rfloor_V$

Since we know that $(e_1\ e_2)\delta_s \Downarrow^n {}^s v_f$ therefore we know that $\exists^s v_{bf}, n_2$ s.t $e_{bs}[e_2\delta_s/x] \Downarrow^{n_2} {}^s v_{bf}$. Therefore we have
$\exists J_2, {}^t v_{tb}, {}^t v_{bf}.e_{bt}[coerce\ !e_{t2}\ !()\delta_t/x]() \Downarrow {}^t v_{tb} \Downarrow^{J_2} {}^t v_{bf} \wedge ({}^s v_{bf}, {}^t v_{bf}) \in \lfloor \tau_2\ \iota \rfloor_V$ (F-DA-A5)

In order to prove (F-DA-A0) we choose J' as $J_1 + J_2$, ${}^t v_t$ as ${}^t v_{tb}$ and ${}^t v_f$ as ${}^t v_{bf}$, we get the desired from (F-DA-A1) and (F-DA-A5)

4. fix:

$$\frac{\begin{array}{c}\Theta, b; \Delta, b < L; \Gamma, x : [a < I]\sigma \vdash_K e : \tau \rightsquigarrow e_t \\ \tau[0/a] <: \mu \qquad \Theta, a, b; \Delta, a < I, b < L; \Gamma \vdash \tau[(b + 1 + \bigcirc\!\!\!\!\!\triangle_b^{b+1,a} I)/b] <: \sigma \\ \Gamma' \sqsubseteq \sum_{b<L} \Gamma \qquad L, M \geq \bigcirc\!\!\!\!\!\triangle_b^{0,1} I \qquad N \geq M - 1 + \sum_{b<L} K\end{array}}{\Theta; \Delta; \Gamma' \vdash_N \mathsf{fix}x.e : \mu \rightsquigarrow E_0} \text{ T-fix}$$

$E_0 = \mathsf{fix}Y.E_1$
$E_1 = \lambda p.E_2$
$E_2 = \mathsf{release} - = p\ \mathsf{in}\ E_3$
$E_3 = \mathsf{bind}\ A = \mathsf{store}()\ \mathsf{in}\ E_4$
$E_4 = \mathsf{let}\,!\,x = (E_{4.1}\ E_{4.2})\ \mathsf{in}\ E_5$
$E_{4.1} = coerce1\ !Y$
$E_{4.2} = (\lambda u.!())\ A$
$E_5 = \mathsf{bind}\ C = \mathsf{store}()\ \mathsf{in}\ E_6$
$E_6 = e_t\ C$

Given: $(\delta_s, \delta_t) \in \lfloor \Gamma \rfloor_E$
To prove: $(\mathsf{fix}x.e\delta_s, (\mathsf{fix}Y.E_1)()\delta_t) \in \lfloor \mu\ \iota \rfloor_E$

This means from Definition 86 we need to prove that
$\forall^s v.\mathsf{fix}x.e\delta_s \Downarrow {}^s v \implies \exists J', {}^t v_t, {}^t v_f.E_0() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \mu\ \iota \rfloor_V$

This means that given some ${}^s v$ s.t $\mathsf{fix}x.e\delta_s \Downarrow {}^s v$ it suffices to prove that
$\exists J', {}^t v_t, {}^t v_f.E_0() \Downarrow {}^t v_t \Downarrow^{J'} {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \mu\ \iota \rfloor_V$ (F-DA-F0)

<u>Claim 1</u>
$\forall 0 \leq t < L.\ (e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$
where $\delta'_s = \delta_s \cup \{x \mapsto (\mathsf{fix}x.e)\delta_s\}$ and $\delta'_t = \delta_t \cup \{x \mapsto (\mathsf{fix}x.E_1)\delta_t\}$

We prove this by induction on the recursion tree

<u>Base case</u>: when t is a leaf node
Since for a leaf node $I(t) = 0$ and $x \notin free(e)$ therefore from IH (outer induction) we get
$(e\ \delta_s, e_t\ ()\ \delta_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$

This means from Definition 86 we have
$\forall^s v'.e_s\ \delta_s \Downarrow {}^s v \implies \exists^t v'_t, {}^t v'_f, J'.e_t\ ()\delta_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f \wedge ({}^s v', {}^t v'_f) \in \lfloor \tau[t/b]\ \iota \rfloor_V$ (BC0)

Since we have to prove $(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$
Therefore from Definition 86 it suffices to prove that
$\forall^s v.e_s\ \delta'_s \Downarrow {}^s v \implies \exists^t v_t, {}^t v_f, J.E_1\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau[t/b]\ \iota \rfloor_V$

This means given some ${}^s v$ s.t $e_s\ \delta'_s \Downarrow {}^s v$ it suffices to prove that
$\exists^t v_t, {}^t v_f, J.E_1\ () \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau[t/b]\ \iota \rfloor_V$ (BC1)

Instantiating (BC0) with ${}^s v$ we get
$\exists^t v'_t, {}^t v'_f, J'.e_t\ ()\delta'_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f \wedge ({}^s v', {}^t v'_f) \in \lfloor \tau[t/b]\ \iota \rfloor_V$ (BC2)

From E-release, E-bind, E-subExpE we also know that if
$e_t\ ()\delta_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f$ then $E_1\ ()\delta'_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f$

Therefore we get we choose ${}^t v_t, {}^t v_f, J$ as ${}^t v'_t, {}^t v'_f, J'$ in (BC1) and we get the desired from (BC2)

Inductive case: when t is a some internal node
From IH we know that
$\forall 0 \leq a < I(t).(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[t'/b]\ \iota \rfloor_E$ where $t' = (t + 1 + \bigcirc\!\!\!\!\triangle_b^{t+1,a} I(t))$

Since $\Theta, a, b; \Delta, a < I, b < L; . \vdash \tau[(b + 1 + \bigcirc\!\!\!\!\triangle_b^{b+1,a} I)/b] <: \sigma$ therefore from Lemma 90 we know that
$\forall 0 \leq a < I(t).(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \sigma\ \iota \rfloor_E$ (F-DA-F0.1)

Claim 2
$(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \sigma\ \iota \rfloor_E \implies ((\mathsf{fix} x.e)\ \delta_s, ((\mathsf{fix} x.(\lambda p.E_2))\ ())\ \delta_t) \in \lfloor \sigma\ \iota \rfloor_E$

Proof is trivial

□

Since from (F-DA-F0.1) we know that
$\forall 0 \leq a < I(t).(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \sigma\ \iota \rfloor_E$
Therefore from Claim2 we also get
$\forall 0 \leq a < I.(\mathsf{fix} x.e\ \delta_s, \mathsf{fix} x.E_1\ ()\ \delta_t) \in \lfloor \sigma\ \iota \rfloor_E$

Let
$\delta''_s = \delta_s \cup \{x \mapsto \mathsf{fix} x.e\delta_s\}$
$\delta''_t = \delta_t \cup \{x \mapsto ((\mathsf{fix} x.E_1)\delta_t\ ())\}$

From Definition 87 it can been that $(\delta''_s, \delta''_t) \in \lfloor \Gamma, x :_{a<I} \sigma \rfloor_E$

Therefore from IH (outer induction) we get
$(e\ \delta''_s, e_t\ ()\ \delta''_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$

This means from Definition 86 we have
$\forall^s v_0.e_s\ \delta''_s \Downarrow {}^s v_0 \implies \exists J_0, {}^t v_t, {}^t v_f.e_t\ ()\ \ \delta''_t \Downarrow {}^t v_t \Downarrow^{J_0} {}^t v_f \wedge ({}^s v_0, {}^t v_f) \in \lfloor \tau[t/b]\ \iota \rfloor_V$ (F-DA-F1)

In order to prove $(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$ from Definition 86 it suffices to prove
$\forall^s v_s.e\ \delta'_s \Downarrow {}^s v_s \implies \exists J_1, {}^t v'_t, {}^t v_t.E_2[()/p]\delta'_t \Downarrow {}^t v'_t \Downarrow^{J_1} {}^t v_t \wedge ({}^s v_s, {}^t v_t) \in \lfloor \tau[t/b]\ \iota \rfloor_V$

This means given some ${}^s v_s$ s.t $e\ \delta'_s \Downarrow {}^s v_s$ and we need to prove that
$\exists J_1, {}^t v'_t, {}^t v_t.E_2[()/p]\delta'_t \Downarrow {}^t v'_t \Downarrow^{J_1} {}^t v_t \wedge ({}^s v_s, {}^t v_t) \in \lfloor \tau[t/b]\ \iota \rfloor_V$ (F-DA-F2)

From E-release, E-bind, E-subExpE we also know that $E_2[()/p]\delta'_t \xrightarrow{*} e_t[(\mathsf{fix} Y.E_1)\ ()/x]\ ()$
therefore from (F-DA-F1) we get the desired.
This proves Claim1

□

Since from Claim1 we know that $\forall 0 \leq t < L.\ (e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[t/b]\ \iota \rfloor_E$. Therefore instantiating it with 0 we get
$(e\ \delta'_s, E_1\ ()\ \delta'_t) \in \lfloor \tau[0/b]\ \iota \rfloor_E$

This means from Definition 86 we have
$\forall^s v'.e\delta'_s \Downarrow {}^s v' \implies \exists^t v'_t, {}^t v'_f, J'.E_1\ ()\delta'_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f \wedge ({}^s v', {}^t v'_f) \in \lfloor \tau[0/b]\iota \rfloor_V$

Instantiating it with the given ${}^s v$ and since know that $\mathsf{fix} x.e\delta_s \Downarrow {}^s v$ therefore from E-fix we also know that $e[\mathsf{fix} x.e/x]\delta_s \Downarrow {}^s v$. Hence we have
$\exists^t v'_t, {}^t v'_f, J'.E_1\ ()\delta'_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f \wedge ({}^s v', {}^t v'_f) \in \lfloor \tau[0/b]\iota \rfloor_V$ (F-DA-F3)

Since $E_1\ ()\delta'_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f$ therefore from E-fix we also know that $\mathsf{fix} x.E_1\ ()\delta_t \Downarrow {}^t v'_t \Downarrow^{J'} {}^t v'_f$. Also since $\tau[0/b] <: \mu$ therefore from (F-DA-F3) and Lemma 89 we get the desired.

□

Lemma 89. $\forall \Theta, \Delta, \tau, \tau', e_s, e_t, \iota.$
(a) $\Theta; \Delta \vdash \tau <: \tau' \wedge \models \Delta\iota \implies \lfloor \tau\ \iota \rfloor_V \subseteq \lfloor \tau'\ \iota \rfloor_V$

(b) $\Theta; \Delta \vdash [a < I]\tau <: [a < J]\tau' \wedge \models \Delta\iota \implies \lfloor [a < I]\tau \ \iota \rfloor_{NE} \subseteq \lfloor [a < J]\tau' \ \iota \rfloor_{NE}$

Proof. Proof by simultaneous induction on $\Theta; \Delta \vdash \tau <: \tau'$ and $\Theta; \Delta \vdash [a < I]\tau <: [a < J]\tau'$

Proof of statement (a)

We case analyze the different cases:

1. $\multimap$:

$$\frac{\Theta; \Delta \vdash B <: A \qquad \Theta; \Delta \vdash \tau <: \tau'}{\Theta; \Delta \vdash A \multimap \tau <: B \multimap \tau'}$$

To prove: $\lfloor (A \multimap \tau) \ \iota \rfloor_V \subseteq \lfloor (B \multimap \tau') \ \iota \rfloor_V$

This means we need to prove that
$\forall (\lambda x.e, \lambda x.\lambda p.e_t) \in \lfloor A \multimap \tau \ \iota \rfloor_V.(\lambda x.e, \lambda x.\lambda p.e_t) \in \lfloor B \multimap \tau' \ \iota \rfloor_E$

This means given $(\lambda x.e_s, \lambda y.\lambda p.\,\mathsf{let}\,!\,x = y \ \mathsf{in} \ e_t) \in \lfloor A \multimap \tau \ \iota \rfloor_V$ and we need to prove
$(\lambda x.e_s, \lambda y.\lambda p.\,\mathsf{let}\,!\,x = y \ \mathsf{in} \ e_t) \in \lfloor B \multimap \tau' \ \iota \rfloor_V$

This means from Definition 86 we are given that
$\forall e'_s, e'_t.(e'_s, e'_t) \in \lfloor A \ \iota \rfloor_{NE} \implies (e_s[e'_s/x], e_t[e'_t/y][()/p]) \in \lfloor \tau \ \iota \rfloor_E\}$ (SV-A0)

And we need to prove that
$\forall e''_s, e''_t.(e''_s, e'_t) \in \lfloor B \ \iota \rfloor_{NE} \implies (e_s[e''_s/x], e_t[e''_t/y][()/p]) \in \lfloor \tau' \ \iota \rfloor_E\}$

This means given $(e''_s, e'_t) \in \lfloor B \ \iota \rfloor_{NE}$ we need to prove that
$(e_s[e''_s/x], e_t[e''_t/y][()/y]) \in \lfloor \tau' \ \iota \rfloor_E$ (SV-A1)

Since we are given that $(e''_s, e'_t) \in \lfloor B \ \iota \rfloor_{NE}$ therefore from IH (Statement (b)) we have
$(e''_s, e'_t) \in \lfloor A \ \iota \rfloor_{NE}$

In order to prove (SV-A1) we instantiate (SV-A0) with e''_s, e''_t and we get
$(e_s[e''_s/x], e_t[e''_t/y][()/p]) \in \lfloor \tau \ \iota \rfloor_E$
Finally from Lemma 90 we get
$(e_s[e''_s/x], e_t[e''_t/y][()/p]) \in \lfloor \tau' \ \iota \rfloor_E$

Proof of statement (b)

$$\frac{\Theta; \Delta \vdash J \leq I \qquad \Theta; \Delta \vdash \tau <: \tau'}{\Theta; \Delta \vdash [a < I]\tau <: [a < J]\tau'}$$

To prove: $\lfloor [a < I]\tau \ \iota \rfloor_{NE} \subseteq \lfloor [a < J]\tau' \ \iota \rfloor_{NE}$

This means we need to prove that
$\forall (e_s, e_t) \in \lfloor [a < I]\tau \ \iota \rfloor_{NE}.(e_s, e_t) \in \lfloor [a < J]\tau' \ \iota \rfloor_{NE}$

This means given $(e_s, e_t) \in \lfloor [a < I]\tau \ \iota \rfloor_{NE}$ and we need to prove
$(e_s, e_t) \in \lfloor [a < J]\tau' \ \iota \rfloor_{NE}$

This means from Definition 86 we are given
$\exists e'_t.e_t = \mathit{coerce}1 \ !e'_t \ !() \wedge \forall 0 \leq i < I.(e_s, e'_t) \in \lfloor \tau[i/a] \ \iota \rfloor_E$ (SNE0)
and we need to prove
$\exists e''_t.e_t = \mathit{coerce}1 \ !e''_t \ !() \wedge \forall 0 \leq j < J.(e_s, e''_t) \in \lfloor \tau'[j/a] \ \iota \rfloor_E$ (SNE1)

In order to prove (SNE1) we choose e''_t as e'_t from (SNE0) and we need to prove
$\forall 0 \leq j < J.(e_s, e'_t) \in \lfloor \tau'[j/a] \ \iota \rfloor_E$

This means given some $0 \leq j < J$ and we need to prove that
$(e_s, e'_t) \in \lfloor \tau'[j/a] \ \iota \rfloor_E$

From (SNE0) we get
$(e_s, e'_t) \in \lfloor \tau[j/a] \ \iota \rfloor_E$

And finally from Lemma 90 we get
$(e_s, e''_t) \in \lfloor \tau'[j/a] \ \iota \rfloor_E$ ∎

Lemma 90. $\forall \Theta, \Delta, \tau, \tau', e_s, e_t, \iota.$
$\Theta; \Delta \vdash \tau <: \tau' \wedge \models \Delta\iota \implies \lfloor \tau \ \iota \rfloor_E \subseteq \lfloor \tau' \ \iota \rfloor_E$

Proof. Given: $\Theta; \Delta \vdash \tau <: \tau'$
To prove: $\lfloor \tau\ \iota \rfloor_E \subseteq \lfloor \tau'\ \iota \rfloor_E$

It suffices to prove that
$\forall (e_s, e_t) \in \lfloor \tau\ \iota \rfloor_E . (e_s, e_t) \in \lfloor \tau'\ \iota \rfloor_E$

This means given $(e_s, e_t) \in \lfloor \tau\ \iota \rfloor_E$ it suffices to prove that
$(e_s, e_t) \in \lfloor \tau'\ \iota \rfloor_E$

This means from Definition 86 we are given that
$\forall^s v_0 . e_s \Downarrow {}^s v_0 \implies \exists J_0, {}^t v_0', {}^t v_0 . e_t \Downarrow {}^t v_0' \Downarrow^{J_0} {}^t v_0 \wedge ({}^s v_0, {}^t v_0) \in \lfloor \tau\ \iota \rfloor_V$ (S0)

And it suffices to prove that
$\forall^s v . e_s \Downarrow {}^s v \implies \exists J, {}^t v_t, {}^t v_f . e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau'\ \iota \rfloor_V$

This means given some ${}^s v$ s.t $e_s \Downarrow {}^s v$ and we need to prove
$\exists J, {}^t v_t, {}^t v_f . e_t \Downarrow {}^t v_t \Downarrow^J {}^t v_f \wedge ({}^s v, {}^t v_f) \in \lfloor \tau'\ \iota \rfloor_V$ (S1)

We get the desired from (S0) and Lemma 89

□

B.5.3 Re-deriving dlPCF's soundness

Definition 91 (Closure translation)**.**

$$\begin{array}{lcl} (\!|(e, [])|\!) & \triangleq & e \\ (\!|(e, \mathtt{C}_1, \ldots, \mathtt{C}_n)|\!) & \triangleq & \lambda x_1 \ldots x_n . e\ (\!|\mathtt{C}_1|\!)\ \ldots (\!|\mathtt{C}_n|\!) \end{array}$$

Definition 92 (Krivine triple translation)**.**

$$\begin{array}{lcl} (\!|(e, \rho, \epsilon)|\!) & \triangleq & (\!|(e, \rho)|\!) \\ (\!|(e, \rho, c.\theta)|\!) & \triangleq & (\!|(\!|(e, \rho)|\!)\ (\!|c|\!), ., \theta|\!) \end{array}$$

Lemma 93 (Type preservation for Closure translation)**.** $\forall \Theta, \Delta, e, \rho, \tau.$
$\Theta; \Delta \vdash_J (e, \rho) : \sigma \implies \Theta; \Delta; . \vdash_J (\!|(e, \rho)|\!) : \sigma$

Proof.

$$\frac{\begin{array}{c}\Theta; \Delta; x_1 : [a < I_1]\tau_1 \ldots x_n : [a < I_n]\tau_n \vdash_K e : \sigma \\ \Theta, a; \Delta, a < I_i \vdash_{H_i} \mathtt{C}_i : \tau_i \qquad J \geq K + I_1 + \ldots + I_n + \sum\limits_{a < I_1} H_1 + \ldots + \sum\limits_{a < I_n} H_n\end{array}}{\Theta; \Delta \vdash_J (e, (\mathtt{C}_1 \ldots \mathtt{C}_n)) : \sigma}$$

$J' = K + I_1 + \ldots + I_n + \sum_{a < I_1} H_1 + \ldots + \sum_{a < I_n} H_n$
D1:

$$\frac{}{\Theta, a; \Delta; .a < I_i \vdash_{H_i} (\!|\mathtt{C}_i|\!) : \tau_i} \text{ IH}$$

D0:

$$\frac{\dfrac{}{\Theta; \Delta; x_1 : [a_1 < I_1]\tau_1, \ldots, x_n : [a_n < I_n]\tau_n \multimap \vdash_K e : \sigma} \text{ Given}}{\Theta; \Delta; . \vdash_K \lambda x_1 \ldots x_n . e : [a_1 < I_1]\tau_1 \multimap [a_2 < I_2]\tau_1 \multimap \ldots [a_n < I_n]\tau_n \multimap \sigma} \text{ D-lam}$$

Main derivation:

$$\frac{\dfrac{\dfrac{D0 \qquad D1}{\Theta; \Delta; . \vdash_{J'} \lambda x_1 \ldots x_n . e\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!) : \sigma} \text{ D-app}}{\Theta; \Delta; . \vdash_J \lambda x_1 \ldots x_n . e\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!) : \sigma} \text{ Lemma 3.5 of [9]}}{\Theta; \Delta; . \vdash_J (\!|(e, \mathtt{C}_1 \ldots \mathtt{C}_n)|\!) : \sigma} \text{ Definition 91}$$

□

Theorem 94 (Type preservation for Krivine triple translation)**.** $\forall \Theta, \Delta, e, \rho, \theta, \tau.$
$\Theta; \Delta \vdash_I (e, \rho, \theta) : \tau \implies \Theta; \Delta; . \vdash_I (\!|(e, \rho, \theta)|\!) : \tau$

Proof.

$$\frac{\Theta; \Delta \vdash_K (e, \rho) : \sigma \qquad \Theta; \Delta \vdash_J \theta : (\sigma, \tau) \qquad I \geq K + J}{\Theta; \Delta \vdash_I (e, \rho, \theta) : \tau}$$

Let $I' = K + J$
Proof by induction on θ

1. Case ϵ:
   Given: $\Theta; \Delta \vdash_I (e, \rho, \epsilon) : \tau$
   To prove: $\Theta; \Delta; . \vdash_I \llparenthesis (e, \rho, \epsilon) \rrparenthesis : \tau$
   D0:

$$\frac{}{\Theta; \Delta; . \vdash_K \llparenthesis (e, \rho) \rrparenthesis : \sigma}\ \text{Lemma 93}$$

   Main derivation:

$$\frac{\dfrac{\dfrac{D0}{\Theta; \Delta; . \vdash_{I'} \llparenthesis (e, \rho) \rrparenthesis : \tau}\ \text{Lemma 3.5 of [9]}}{\Theta; \Delta; . \vdash_{I'} \llparenthesis (e, \rho, \epsilon) \rrparenthesis : \tau}\ \text{Definition 92}}{\Theta; \Delta; . \vdash_I \llparenthesis (e, \rho, \epsilon) \rrparenthesis : \tau}\ \text{Lemma 3.5 of [9]}$$

2. Case $\mathtt{C}.\theta'$:
   Given: $\Theta; \Delta \vdash_I (e, \rho, \mathtt{C}.\theta') : \tau$
   To prove: $\Theta; \Delta; . \vdash_I \llparenthesis (e, \rho, \mathtt{C}.\theta') \rrparenthesis : \tau$

   Since $\theta = \mathtt{C}.\theta'$ therefore from dlPCF's type rule for $\mathtt{C}.\theta'$ we know that
   $\sigma = [d < L]\gamma \multimap \mu$

   That is we are given that

$$\frac{\Theta, d; \Delta, d < L_g \vdash_{K_g} \mathtt{C} : \gamma \qquad \Theta; \Delta \vdash_{H_g} \theta' : (\mu, \tau) \qquad J \geq H_g + \sum_{d < L_g} K_g + L_g}{\Theta; \Delta \vdash_J \mathtt{C}.\theta' : ([d < L_g]\gamma \multimap \mu, \tau)}$$

   D2:

$$\frac{\dfrac{}{\Theta; \Delta \vdash_J \mathtt{C}.\theta' : ([d < L_g]\gamma \multimap \mu, \tau)}\ \text{Given}}{\Theta; \Delta \vdash_{H_g} \theta' : (\mu, \tau)}\ \text{By inversion}$$

   D1:

$$\frac{}{\Theta; \Delta; . \vdash_K \llparenthesis (e, \rho) \rrparenthesis : [d < L_g]\gamma \multimap \mu}\ \text{Lemma 93}$$

   D0:

$$\frac{D1 \qquad \dfrac{\dfrac{}{\Theta, d; \Delta, d < L_g \vdash_{K_g} \mathtt{C} : \gamma}\ \text{Given}}{\Theta, d; \Delta, d < L_g \vdash_{K_g} \llparenthesis \mathtt{C} \rrparenthesis : \gamma}\ \text{Lemma 93}}{\Theta; \Delta; . \vdash_{K + L_g + \sum_{L_g} K_g} \llparenthesis (e, \rho) \rrparenthesis\ \llparenthesis \mathtt{C} \rrparenthesis : \mu}\ \text{D-app}$$

   D0.1:

$$\frac{D0}{\Theta; \Delta; . \vdash_{K + L_g + \sum_{L_g} K_g} (\llparenthesis (e, \rho) \rrparenthesis\ \llparenthesis \mathtt{C} \rrparenthesis, .) : \mu}$$

   D0.0:

$$\frac{\dfrac{D0.1 \qquad D2}{\Theta; \Delta \vdash_{K + L_g + \sum_{L_g} K_g + H_g} (\llparenthesis (e, \rho) \rrparenthesis\ \llparenthesis \mathtt{C} \rrparenthesis, ., \theta') : \tau} \qquad J \geq L_g + \sum_{L_g} K_g + H_g}{\Theta; \Delta \vdash_{K + J} (\llparenthesis (e, \rho) \rrparenthesis\ \llparenthesis \mathtt{C} \rrparenthesis, ., \theta') : \tau}\ \text{Lemma 3.5 of [9]}$$

   Main derivation:

$$\frac{\dfrac{\dfrac{D0.0}{\Theta; \Delta; . \vdash_{I'} \llparenthesis (e, \rho)\ \llparenthesis \mathtt{C} \rrparenthesis, ., \theta \rrparenthesis : \tau}\ \text{IH}}{\Theta; \Delta; . \vdash_{I'} \llparenthesis (e, \rho, \mathtt{C}.\theta) \rrparenthesis : \tau}\ \text{Definition 92}}{\Theta; \Delta; . \vdash_I \llparenthesis (e, \rho, \mathtt{C}.\theta) \rrparenthesis : \tau}\ \text{Lemma 3.5 of [9]}$$

$\square$

Definition 95 (Equivalence for λ-amor)**.**

$$v_1 \overset{s}{\approx}_{aV} v_2 \triangleq \begin{cases} True & v_1 = () \wedge v_2 = () \\ \forall e', e'', s' < s.e' \overset{s'}{\approx}_{aE} e'' \implies & v_1 = \lambda x.e_2 \wedge v_2 = \lambda x.e_2 \\ e_1[e'/x] \overset{s'}{\approx}_{aE} e_2[e''/x] & \\ e_1 \overset{s}{\approx}_{aE} e_2 & v_1 = !e_1 \wedge v_2 = !e_2 \\ & v_1 = \Lambda.e_1 \wedge v_2 = \Lambda.e_2 \\ \forall i < s.v_1 \Downarrow_i^k v_a \implies & v_1 = \mathsf{ret} - \wedge v_2 = \mathsf{ret} - \\ v_2 \Downarrow^k v_b \wedge v_a \overset{s-i}{\approx}_{aE} v_b & v_1 = \mathsf{bind} - = - \mathsf{in} - \wedge v_2 = \mathsf{bind} - = - \mathsf{in} - \\ & v_1 = \uparrow^n \wedge v_2 = \uparrow^n \\ & v_1 = \mathsf{release} - = - \mathsf{in} - \wedge v_2 = \mathsf{release} - = - \mathsf{in} - \\ & v_1 = \mathsf{store} - \wedge v_2 = \mathsf{store} - \\ v_{a1} \overset{s}{\approx}_{aV} v_{b1} \wedge v_{a2} \overset{s}{\approx}_{aV} v_{b2} & v_1 = \langle\langle v_{a1}, v_{a2} \rangle\rangle \wedge v_2 = \langle\langle v_{b1}, v_{b2} \rangle\rangle \\ v_{a1} \overset{s}{\approx}_{aV} v_{b1} \wedge v_{a2} \overset{s}{\approx}_{aV} v_{b2} & v_1 = \langle v_{a1}, v_{a2} \rangle \wedge v_2 = \langle v_{b1}, v_{b2} \rangle \\ v_a \overset{s}{\approx}_{aV} v_b & v_1 = \mathsf{inl}(v_a) \wedge v_2 = \mathsf{inl}(v_b) \\ v_a \overset{s}{\approx}_{aV} v_b & v_1 = \mathsf{inr}(v_a) \wedge v_2 = \mathsf{inr}(v_b) \end{cases}$$

$$e_1 \overset{s}{\approx}_{aE} e_2 \triangleq \forall i < s.e_1 \Downarrow_i v_a \implies e_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$$

$$\delta_1 \overset{s}{\approx}_{aE} \delta_2 \triangleq dom(\delta_1) = dom(\delta_2) \wedge \forall x \in dom(\delta_1).\delta_1(x) \overset{s}{\approx}_{aE} \delta_2(x)$$

Lemma 96 (Monotonicity lemma for value equivalence)**.** $\forall v_1, v_2, s.$
$v_1 \overset{s}{\approx}_{aV} v_2 \implies \forall s' < s.v_1 \overset{s'}{\approx}_{aV} v_2$

Proof. Given: $v_1 \overset{s}{\approx}_{aV} v_2$

To prove: $\forall s' < s.v_1 \overset{s'}{\approx}_{aV} v_2$

This means given some $s' < s$ and it suffices to prove that $v_1 \overset{s'}{\approx}_{aV} v_2$

We induct on v_1

1. $v_1 = ()$:
   Since we are given that $v_1 \overset{s}{\approx}_{aV} v_2$ therefore we get the desired Directly from Definition 95
2. $v_1 = \lambda x.e_1$:
   Since we are given that $v_1 \overset{s}{\approx}_{aV} v_2$ therefore from Definition 95 we are given that
   $\forall e', e'', s'' < s.e' \overset{s''}{\approx}_{aE} e'' \implies e_1[e'/x] \overset{s''}{\approx}_{aE} e_2[e''/x]$ (M-L0)

   and we need to prove that $v_1 \overset{s'}{\approx}_{aV} v_2$ therefore again from Definition 95 we need to prove that
   $\forall e_1', e_1'', s_1'' < s.e_1' \overset{s_1''}{\approx}_{aE} e_1'' \implies e_1[e_1'/x] \overset{s_1''}{\approx}_{aE} e_2[e_1''/x]$

   This means given some $e_1', e_1'', s_1'' < s'$ s.t $e_1' \overset{s_1''}{\approx}_{aE} e_1''$ we need to prove that
   $e_1[e_1'/x] \overset{s_1''}{\approx}_{aE} e_2[e_1''/x]$

   Instantiating (M-L0) with e_1', e_1'', s_1'' we get $e_1[e_1'/x] \overset{s_1''}{\approx}_{aE} e_2[e_1''/x]$
3. $v_1 = !e_1$:
   Since we are given $v_1 \overset{s}{\approx}_{aV} v_2$ therefore from Definition 95 we have
   $e_1 \overset{s}{\approx}_{aE} e_2$ where $v_2 = !e_2$

   Similarly from Definition 95 it suffices to prove that $e_1 \overset{s'}{\approx}_{aE} e_2$
   We get this directly from Lemma 97
4. $v_1 = \Lambda e_1$:
   Similar reasoning as in the $!e_1$ case
5. $v_1 = \mathsf{ret}\, e_1$:
   Since we are given $v_1 \overset{s}{\approx}_{aV} v_2$ therefore from Definition 95 we have
   $\forall i < s.v_1 \Downarrow_i^k v_a \implies v_2 \Downarrow^k v_b \wedge v_a \overset{s-i}{\approx}_{aE} v_b$ where $v_2 = \mathsf{ret}\, e_2$ (MV-R0)

   Similarly from Definition 95 it suffices to prove that
   $\forall j < s'.v_1 \Downarrow_i^k v_a \implies v_2 \Downarrow^k v_b \wedge v_a \overset{s'-j}{\approx}_{aE} v_b$

This means given some $j < s'$ and $v_1 \Downarrow_i^k v_a$ and it suffices to prove that $v_2 \Downarrow^k v_b \wedge v_a \overset{s'-j}{\approx}_{aE} v_b$

Instantiating (MV-R0) with j we get $v_2 \Downarrow^k v_b \wedge v_a \overset{s-j}{\approx}_{aE} v_b$

Since we have $v_a \overset{s-j}{\approx}_{aE} v_b$ therefore from Lemma 97 we also get $v_a \overset{s'-j}{\approx}_{aE} v_b$

6. $v_1 = \mathsf{bind} - = - \mathsf{ in } -, \uparrow^n, \mathsf{release} - = - \mathsf{ in } -, \mathsf{store} -$:
   Similar reasoning as in the $\mathsf{ret} -$ case
7. $v_1 = \langle\langle v_{a1}, v_{a2}\rangle\rangle$:
   From Definition 95 and IH we get the desired
8. $v_1 = \langle v_{a1}, v_{a2}\rangle$:
   From Definition 95 and IH we get the desired
9. $v_1 = \mathsf{inl}(v)$:
   From Definition 95 and IH we get the desired
10. $v_1 = \mathsf{inr}(v)$:
    From Definition 95 and IH we get the desired

□

Lemma 97 (Monotonicity lemma for expression equivalence)**.** $\forall e_1, e_2, s.$
$e_1 \overset{s}{\approx}_{aE} e_2 \implies \forall s' < s. e_1 \overset{s'}{\approx}_{aE} e_2$

Proof. Given: $e_1 \overset{s}{\approx}_{aE} e_2$

To prove: $\forall s' < s. e_1 \overset{s'}{\approx}_{aE} e_2$

This means given some $s' < s$ and we need to prove $e_1 \overset{s'}{\approx}_{aE} e_2$

Since we are given $e_1 \overset{s}{\approx}_{aE} e_2$ therefore from Definition 95 we have
$\forall i < s. e_1 \Downarrow_i v_a \implies e_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (ME0)

Similarly from Definition 95 it suffices to prove that
$\forall j < s'. e_1 \Downarrow_j v_a \implies e_2 \Downarrow v_b \wedge v_a \overset{s'-j}{\approx}_{aV} v_b$

This means given some $j < s'$ s.t $e_1 \Downarrow_j v_a$ and we need to prove
$e_2 \Downarrow v_b \wedge v_a \overset{s'-j}{\approx}_{aV} v_b$

We get the desired from (ME0) and Lemma 96

□

Lemma 98 (Monotonicity lemma for δ equivalence)**.** $\forall \delta_1, \delta_2, s.$
$\delta_1 \overset{s}{\approx}_{aE} \delta_2 \implies \forall s' < s. \delta_1 \overset{s'}{\approx}_{aE} \delta_2$

Proof. From Definition 95 and Lemma 97 □

Theorem 99 (Fundamental theorem for equivalence relation of λ-amor)**.** $\forall \delta_1, \delta_2, e, s.$
$\delta_1 \overset{s}{\approx}_{aE} \delta_2 \implies e\delta_1 \overset{s}{\approx}_{aE} e\delta_2$

Proof. We induct on e

1. $e = x$:
   We need to prove that $x\delta_1 \overset{s}{\approx}_{aE} x\delta_2$

   This means it suffices to prove that $\delta_1(x) \overset{s}{\approx}_{aE} \delta_2(x)$

   We get this directly from Definition 95
2. $e = \lambda y.e'$:
   We need to prove that $\lambda y.e'\delta_1 \overset{s}{\approx}_{aE} \lambda y.e'\delta_2$

   This means from Definition 95 it suffices to prove that
   $\forall i < s. \lambda y.e'\delta_1 \Downarrow_i v_a \implies \lambda y.e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

   This means that given some $i < s$ s.t $\lambda y.e'\delta_1 \Downarrow_i v_a$ it suffices to prove that
   $\lambda y.e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-L0)

   From E-val we know that $v_a = \lambda y.e'\delta_1$

   From (FTE-L0) we need to prove that

(a) $\lambda y.e'\delta_2 \Downarrow v_b$:
From E-val we know that $v_b = \lambda y.e'\delta_2$

(b) $v_a \overset{s-i}{\approx}_{aV} v_b$:
We need to prove that
$\lambda y.e'\delta_1 \overset{s}{\approx}_{aV} \lambda y.e'\delta_2$

This means from Definition 95 it suffices to prove that
$\forall e'_1, e'_2, s' < s.e'_1 \overset{s'}{\approx}_{aE} e'_2 \implies e'\delta_1[e'_1/y] \overset{s'}{\approx}_{aE} e'\delta_2[e'_2/y]$

This further means that given some $e'_1, e'_2, s' < s$ s.t $e'_1 \overset{s'}{\approx}_{aE} e'_2$ it suffices to prove that
$e'\delta_1[e'_1/y] \overset{s'}{\approx}_{aE} e'\delta_2[e'_2/y]$

We get this from IH and Lemma 98

3. $e = \mathsf{fix} y.e'$:
We induct on s
IHi: $\forall s'' < s.\ \delta_1 \overset{s''}{\approx}_{aE} \delta_2 \implies \mathsf{fix} y.e'\delta_1 \overset{s'}{\approx}_{aE} \mathsf{fix} y.e'\delta_2$

To prove: $\delta_1 \overset{s}{\approx}_{aE} \delta_2 \implies \mathsf{fix} y.e'\delta_1 \overset{s}{\approx}_{aE} \mathsf{fix} y.e'\delta_2$
This means we are given $\delta_1 \overset{s}{\approx}_{aE} \delta_2$ and we need to prove
$\mathsf{fix} y.e'\delta_1 \overset{s}{\approx}_{aE} \mathsf{fix} y.e'\delta_2$

From Definition 95 it suffices to prove that
$\forall i < s.\mathsf{fix} y.e'\delta_1 \Downarrow_i v_a \implies \mathsf{fix} y.e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means given some $i < s$ s.t $\mathsf{fix} y.e'\delta_1 \Downarrow_i v_a$ and we need to prove $\mathsf{fix} y.e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

Since we are given that $\mathsf{fix} y.e'\delta_1 \Downarrow_i v_a$ therefore from E-fix we know that
$e'[\mathsf{fix} x.e'\delta_1/y]\delta_1 \Downarrow_{i-1} v_a$

Instantiating IHi with $s-1$ and using Lemma 98 we get
$\mathsf{fix} y.e'\delta_1 \overset{s-1}{\approx}_{aE} \mathsf{fix} y.e'\delta_2$ (F1)

Let
$\delta'_1 = \delta_1 \cup \{y \mapsto \mathsf{fix} y.e'\delta_1\}$
$\delta'_2 = \delta_2 \cup \{y \mapsto \mathsf{fix} y.e'\delta_2\}$

From Lemma 98 and (F1) we know that $\delta'_1 \overset{s-1}{\approx}_{aE} \delta'_2$

Therefore from IH of outer induction we know that we have
$e'\delta'_1 \overset{s-1}{\approx}_{aE} e'\delta'_2$

This means from Definition 95 we know that
$\forall i' < (s-1).e'\delta'_1 \Downarrow_{i'} v_a \implies e'\delta'_2 \Downarrow v_b \wedge v_a \overset{s-1-i'}{\approx}_{aV} v_b$

Instantiating with $i-1$ and since we know that $e'\delta'_1 \Downarrow_{i-1} v_a$ and therefore we get
$e'\delta'_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ which is the desired.

4. $e = e_1\ e_2$:
We need to prove that $e_1\ e_2\delta_1 \overset{s}{\approx}_{aE} e_1\ e_2\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.e_1\ e_2\delta_1 \Downarrow_i v_a \implies e_1\ e_2\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $e_1\ e_2\delta_1 \Downarrow_i v_a$ it suffices to prove that
$e_1\ e_2\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-A0)

<u>IH1</u>: $e_1\delta_1 \overset{s}{\approx}_{aE} e_1\delta_2$

Therefore from Definition 95 we have
$\forall j < s.e_1\delta_1 \Downarrow_j v'_a \implies e_1\delta_2 \Downarrow v'_b \wedge v'_a \overset{s-j}{\approx}_{aV} v'_b$ (FTE-A1)

Since $(e_1\delta_1\ e_2\delta_1) \Downarrow_i v_a$ therefore from E-app we know that $\exists i_1 < i.e_1\delta_1 \Downarrow_{i_1} \lambda y.e'$

Therefore instantiating (FTE-A1) with i_1 we get $e_1\delta_2 \Downarrow v'_b \wedge v'_a \overset{s-i_1}{\approx}_{aV} v'_b$ (FTE-A1.1)

Since $v'_a = \lambda y.e'$ and since $v'_a \overset{s-i_1}{\approx}_{aV} v'_b$ therefore from Definition 95 we know that $v'_b = \lambda y.e''$

Again since $\lambda y.e' \overset{s-i_1}{\approx}_{aV} \lambda y.e''$ therefore from Definition 95 we know that
$\forall e'_1, e'_2, s' < (s - i_1).e'_1 \overset{s'}{\approx}_{aE} e'_2 \implies e'[e'_1/y] \overset{s'}{\approx}_{aE} e''[e'_2/y]$ (FTE-A2)

<u>IH2</u>: $e_2\delta_1 \overset{s-i_1-1}{\approx}_{aE} e_2\delta_2$

Instantiating (FTE-A2) with $e_2\delta_1, e_2\delta_2$ we get
$e'[e_2\delta_1/y] \overset{s-i_1-1}{\approx}_{aE} e''[e_2\delta_1/y]$

Again from Definition 95 we have
$\forall j < (s - i_1 - 1).e'[e_2\delta_1/y] \Downarrow_j v''_a \implies e''[e_2\delta_1/y] \Downarrow v''_b \wedge v''_a \overset{s-i_1-1-j}{\approx}_{aV} v''_b$ (FTE-A2.1)

Since $(e_1\delta_1\ e_2\delta_1) \Downarrow_i v_a$ therefore from E-app we know that $\exists i_2 = i - i_1 - 1.e'[e_2\delta_1/x] \Downarrow_{i_2} v_a$

Instantiating (FTE-A2.1) with i_2 we get $e''[e_2\delta_1/y] \Downarrow v''_b \wedge v_a \overset{s-i_1-1-i_2}{\approx}_{aV} v''_b$

Since $i = i_1 + i_2 + 1$ therefore this proves (FTE-A0) and we are done.

5. $e = \langle\langle e_1, e_2\rangle\rangle$:
We need to prove that $\langle\langle e_1, e_2\rangle\rangle\delta_1 \overset{s}{\approx}_{aE} \langle\langle e_1, e_2\rangle\rangle\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.\langle\langle e_1, e_2\rangle\rangle\delta_1 \Downarrow_i v_a \implies \langle\langle e_1, e_2\rangle\rangle\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\langle\langle e_1, e_2\rangle\rangle\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\langle\langle e_1, e_2\rangle\rangle\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-TI0)

From E-TI we know that $v_a = \langle\langle v_{a1}, v_{a2}\rangle\rangle$ and $e_1\delta_1 \Downarrow_{i_1} v_{a1}$ and $e_2\delta_1 \Downarrow_{i_2} v_{a2}$

<u>IH1</u>: $e_1\delta_1 \overset{s}{\approx}_{aE} e_1\delta_2$
Therefore from Definition 95 we have
$\forall i < s.e_1\delta_1 \Downarrow_i v_{a1} \implies e_1\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i}{\approx}_{aV} v_{b1}$

Since we know that $e_1\delta_1 \Downarrow_{i_1} v_{a1}$ therefore we get
$e_1\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i_1}{\approx}_{aV} v_{b1}$ (FTE-TI1)

<u>IH2</u>: $e_2\delta_1 \overset{s}{\approx}_{aE} e_2\delta_2$
Similarly from Definition 95 we have
$\forall i < s.e_2\delta_1 \Downarrow_i v_{a1} \implies e_2\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i}{\approx}_{aV} v_{b1}$

Since we know that $e_2\delta_1 \Downarrow_{i_2} v_{a2}$ therefore we get
$e_2\delta_2 \Downarrow v_{b2} \wedge v_{a2} \overset{s-i_2}{\approx}_{aV} v_{b2}$ (FTE-TI2)

From (FTE-TI0) we need to prove
 (a) $\langle\langle e_1, e_2\rangle\rangle\delta_2 \Downarrow v_b$:
 We get this from (FTE-TI1), (FTE-TI2) and E-TI
 (b) $v_a \overset{s-i}{\approx}_{aV} v_b$:
 Since $i = i_1 + i_2$, $v_a = \langle\langle v_{a1}, v_{a2}\rangle\rangle$ and $v_b = \langle\langle v_{b1}, v_{b2}\rangle\rangle$ it suffices to prove that
 $\langle\langle v_{a1}, v_{a2}\rangle\rangle \overset{s-i_1-i_2}{\approx}_{aV} \langle\langle v_{b1}, v_{b2}\rangle\rangle$

 From Definition 95 it suffices to prove that
 $v_{a1} \overset{s-i_1-i_2}{\approx}_{aV} v_{b1}$ and $v_{a2} \overset{s-i_1-i_2}{\approx}_{aV} v_{b2}$
 We get this from (FTE-TI1), (FTE-TI2) and Lemma 96

6. $e = \mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2$:
We need to prove that $\mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_1 \overset{s}{\approx}_{aE} \mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s. \mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_1 \Downarrow_i v_a \implies \mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\mathsf{let}\langle\langle x, y\rangle\rangle = e_1 \mathsf{~in~} e_2\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-TE0)

<u>IH1</u>: $e_1\delta_1 \overset{s}{\approx}_{aE} e_1\delta_2$
Therefore from Definition 95 we have

$\forall i < s.e_1\delta_1 \Downarrow_i v_{a1} \implies e_1\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i}{\approx}_{aV} v_{b1}$

Since we know that $\mathsf{let}\langle\langle x, y\rangle\rangle = e_1$ in $e_2\delta_1 \Downarrow_i v_a$ therefore from E-TE we know that $\exists i_1 < s.e_1\delta_1 \Downarrow_{i_1} \langle\langle v'_{a1}, v'_{a2}\rangle\rangle$. Therefore we get

$e_1\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i_1}{\approx}_{aV} v_{b1}$ (FTE-TE1)

Since $v_{a1} \overset{s-i_1}{\approx}_{aV} v_{b1}$ and $v_{a1} = \langle\langle v'_{a1}, v'_{a2}\rangle\rangle$ therefore from Definition 95 we have

$v_{b1} = \langle\langle v'_{b1}, v'_{b2}\rangle\rangle$ (FTE-TE1.1)

Let

$\delta'_1 = \delta_1 \cup \{x \mapsto \langle\langle v'_{a1}, v'_{a2}\rangle\rangle\}$

$\delta'_2 = \delta_2 \cup \{x \mapsto \langle\langle v'_{b1}, v'_{b2}\rangle\rangle\}$

<u>IH2</u>: $e_2\delta'_1 \overset{s-i_1}{\approx}_{aE} e_2\delta'_2$

Therefore from Definition 95 we have

$\forall i < (s - i_1).e_2\delta'_1 \Downarrow_i v_a \implies e_2\delta'_2 \Downarrow v_{b2} \wedge v_a \overset{s-i_1-i}{\approx}_{aV} v_b$

Since we know that $\mathsf{let}\langle\langle x, y\rangle\rangle = e_1$ in $e_2\delta_1 \Downarrow_i v_a$ therefore from E-TE we know that $\exists i_2 = i - i_1.e_2\delta'_1 \Downarrow_{i_2} v_a$. Therefore we get

$e_2\delta'_2 \Downarrow v_{b2} \wedge v_a \overset{s-i_1-i_2}{\approx}_{aV} v_b$ (FTE-TE2)

This proves the desired

7. $e = \langle e_{a1}, e_{a2}\rangle$:
   Similar reasoning as in the $\langle\langle e_{a1}, e_{a2}\rangle\rangle$ case above
8. $e = \mathsf{fst}(e')$:
   We need to prove that $\mathsf{fst}(e')\delta_1 \overset{s}{\approx}_{aE} \mathsf{fst}(e')\delta_2$

   This means from Definition 95 it suffices to prove that

   $\forall i < s.\mathsf{fst}(e')\delta_1 \Downarrow_i v_a \implies \mathsf{fst}(e')\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

   This means that given some $i < s$ s.t $\mathsf{fst}(e')\delta_1 \Downarrow_i v_a$ it suffices to prove that

   $\mathsf{fst}(e')\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-F0)

   Since we know that $\mathsf{fst}(e')\delta_1 \Downarrow_i v_a$ therefore from E-fst we know that $e'\delta_1 \Downarrow_i \langle\langle v_a, -\rangle\rangle$

   <u>IH</u>: $e'\delta_1 \overset{s}{\approx}_{aE} e'\delta_2$

   This means from Definition 95 we have

   $\forall j < s.e'\delta_1 \Downarrow_j v_{a1} \implies e'\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-j}{\approx}_{aV} v_{b1}$

   Instantiating with i we get $e'\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-j}{\approx}_{aV} v_{b1}$

   Since we know that $v_{a1} = \langle\langle v_a, -\rangle\rangle$ therefore from Definition 95 we also know that

   $v_{b1} = \langle\langle v_b, -\rangle\rangle$ s.t $v_a \overset{s}{\approx}_{aV} v_b$

   This proves the desired.
9. $e = \mathsf{snd}(e')$:
   Similar reasoning as in the $\mathsf{fst}(e')$ case
10. $e = \mathsf{inl}(e')$:
    We need to prove that $\mathsf{inl}(e')\delta_1 \overset{s}{\approx}_{aE} \mathsf{inl}(e')\delta_2$

    This means from Definition 95 it suffices to prove that

    $\forall i < s.\mathsf{inl}(e')\delta_1 \Downarrow_i v_a \implies \mathsf{inl}(e')\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

    This means that given some $i < s$ s.t $\mathsf{inl}(e')\delta_1 \Downarrow_i v_a$ it suffices to prove that

    $\mathsf{inl}(e')\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-IL0)

    Since we know that $\mathsf{inl}(e')\delta_1 \Downarrow_i v_a$ therefore from E-inl we know that $v_a = \mathsf{inl}(()v'_a)$ and $e'\delta_1 \Downarrow_i v'_a$

    <u>IH</u>: $e'\delta_1 \overset{s}{\approx}_{aE} e'\delta_2$

    This means from Definition 95 we have

    $\forall j < s.e'\delta_1 \Downarrow_j v_{a1} \implies e'\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-j}{\approx}_{aV} v_{b1}$

    Instantiating with i we get $e'\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i}{\approx}_{aV} v_{b1}$

    Since $e'\delta_2 \Downarrow v_{b1}$ therefore from E-inl we have $\mathsf{inl}(e')\delta_2 \Downarrow \mathsf{inl}(v_{b1})$

    And since we know that $v_{a1} \overset{s-i}{\approx}_{aV} v_{b1}$ therefore from Definition 95 we also know that

    $\mathsf{inl}(v_{a1}) \overset{s-i}{\approx}_{aV} \mathsf{inl}(v_{b1})$

    This proves the desired.

11. $e = \mathsf{inr}(e')$:
Similar reasoning as in the $\mathsf{inl}(e')$ case
12. $e = \mathsf{case}\ e_c, x.e_l, y.e_r$:
We need to prove that $\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \overset{s}{\approx}_{aE} \mathsf{case}\ e_c, x.e_l, y.e_r\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \Downarrow_i v_a \implies \mathsf{case}\ e_c, x.e_l, y.e_r\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\mathsf{case}\ e_c, x.e_l, y.e_r\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-C0)

Since we know that $\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \Downarrow_i v_a$ therefore two cases arise:
2 cases arise:
(a) $e_c\delta_1 \Downarrow \mathsf{inl}(v_{c1})$:
<u>IH1</u> $e_c\delta_1 \overset{s}{\approx}_{aE} e_c\delta_2$
This means from Definition 95 we have
$\forall j < s.e_c\delta_1 \Downarrow_j v_{c1} \implies e_c\delta_2 \Downarrow v_{c2} \wedge v_{c1} \overset{s-j}{\approx}_{aV} v_{c2}$

Since we know that $\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \Downarrow_i v_a$ therefore from E-case1 we know that $\exists i_1$ s.t $e_c\delta_1 \Downarrow_{i_1} \mathsf{inl}(v'_{c1})$

Therefore instantiating with i_1 we get $e_c\delta_2 \Downarrow v_{c2} \wedge v_{c1} \overset{s-i_1}{\approx}_{aV} v_{c2}$

From Definition 95 we know that $\exists v'_{c2}.v_{c2} = \mathsf{inl}(v'_{c2})$ s.t $v'_{c1} \overset{s-i_1}{\approx}_{aV} v'_{c2}$

<u>IH2</u> $e_l\delta_1[v'_{c1}/x] \overset{s-i_1}{\approx}_{aE} e_l\delta_2[v'_{c2}/x]$

This means from Definition 95 we have
$\forall j < (s - i_1).e_l\delta_1[v'_{c1}/x] \Downarrow_j v_{l1} \implies e_l\delta_2[v'_{c2}/x] \Downarrow v_{l2} \wedge v_{l1} \overset{s-i_1-j}{\approx}_{aV} v_b$

Since we know that $\mathsf{case}\ e_c, x.e_l, y.e_r\delta_1 \Downarrow_i v_a$ therefore from E-case1 we know that $\exists i_2$ s.t $e_l\delta_1 \Downarrow_{i_2} v_a$

Therefore instantiating with i_2 we get $e_l\delta_2[v'_{c2}/x] \Downarrow v_{l2} \wedge v_a \overset{s-i_1-j}{\approx}_{aV} v_b$

This proves the desired
(b) $e_c\delta_1 \Downarrow \mathsf{inr}(v_{c1})$:
Similar reasoning as in the previous case
13. $e = !e'$:
We need to prove that $!e'\delta_1 \overset{s}{\approx}_{aE} !e'\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.!e'\delta_1 \Downarrow_i v_a \implies !e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $!e'\delta_1 \Downarrow_i v_a$ it suffices to prove that
$!e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-B0)

From E-val we know that $v_a = !e'\delta_1$ and $i = 0$

<u>IH</u>: $e'\delta_1 \overset{s}{\approx}_{aE} e'\delta_2$
From (FTE-B0) we need to prove that
(a) $!e'\delta_2 \Downarrow v_b$:
From E-val we know that $v_b = !e'\delta_2$
(b) $v_a \overset{s-i}{\approx}_{aV} v_b$:
We need to prove that
$!e'\delta_1 \overset{s}{\approx}_{aV} !e'\delta_2$

This means from Definition 95 it suffices to prove that
$e'\delta_1 \overset{s}{\approx}_{aE} e'\delta_2$

We get this directly from IH
14. $e = \mathsf{let}\,!\,x = e'_1\ \mathsf{in}\ e'_2$:
We need to prove that $\mathsf{let}\,!\,x = e'_1\ \mathsf{in}\ e'_2\delta_1 \overset{s}{\approx}_{aE} \mathsf{let}\,!\,x = e'_1\ \mathsf{in}\ e'_2\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.\,\mathsf{let}\,!\,x = e'_1\ \mathsf{in}\ e'_2\delta_1 \Downarrow_i v_a \implies \mathsf{let}\,!\,x = e'_1\ \mathsf{in}\ e'_2\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\mathsf{let}\,!\,x = e_1'$ in $e_2'\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\mathsf{let}\,!\,x = e_1'$ in $e_2'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-BE0)

IH1: $e_1'\delta_1 \overset{s}{\approx}_{aE} e_1'\delta_2$
This means from Definition 95 we have
$\forall j < s.e_1'\delta_1 \Downarrow_j v_{a11} \implies e_1'\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-j}{\approx}_{aV} v_{b11}$

Since we know that $\mathsf{let}\,!\,x = e_1'$ in $e_2'\delta_1 \Downarrow_i v_a$ therefore from E-subExpE we know that $\exists i_1.e_1'\delta_1 \Downarrow_{i_1} !e_{b1}$

Instantiating with i_1 we get $e_1'\delta_2 \Downarrow v_{b11} \wedge v_{a11} \overset{s-i_1}{\approx}_{aV} v_{b11}$

Since we know that $v_{a11} = !e_{b1}$ therefore from Definition 95 we also know that
$v_{b11} = !e_{b2}$ s.t $e_{b1} \overset{s-i_1}{\approx}_{aE} e_{b2}$

IH2: $e_2'[e_{b1}/x]\delta_1 \overset{s-i_1}{\approx}_{aE} e_2'[e_{b2}/x]\delta_2$
This means from Definition 95 we have
$\forall j < s.e_2'[e_{b1}/x]\delta_1 \Downarrow_j v_a \implies e_2'[e_{b2}/x]\delta_2 \Downarrow v_b \wedge v_a \overset{s-i_1-j}{\approx}_{aV} v_b$

Since we know that $\mathsf{let}\,!\,x = e_1'$ in $e_2'\delta_1 \Downarrow_i v_a$ therefore from E-subExpE we know that $\exists i_2.e_1'[e_{b1}/x]\delta_1 \Downarrow_{i_2} v_a$

Instantiating with i_2 we get $e_2'[e_{b2}/x]\delta_2 \Downarrow v_b \wedge v_a \overset{s-i_1-i_2}{\approx}_{aV} v_b$

This proves the desired

15. $e = \Lambda.e'$:
Similar reasoning as in the $\lambda y.e'$ case
16. $e = e'\ []$:
Similar reasoning as in the app case
17. $e = \mathsf{ret}\, e'$:
We need to prove that $\mathsf{ret}\, e'\delta_1 \overset{s}{\approx}_{aE} \mathsf{ret}\, e'\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.\, \mathsf{ret}\, e'\delta_1 \Downarrow_i v_a \implies \mathsf{ret}\, e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\mathsf{ret}\, e'\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\mathsf{ret}\, e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-R0)

From E-val we know that $v_a = \mathsf{ret}\, e'\delta_1$ and $i = 0$

From (FTE-R0) we need to prove that
(a) $\mathsf{ret}\, e'\delta_2 \Downarrow v_b$:
From E-val we know that $v_b = \mathsf{ret}\, e'\delta_2$
(b) $v_a \overset{s-i}{\approx}_{aV} v_b$:
We need to prove that
$\mathsf{ret}\, e'\delta_1 \overset{s}{\approx}_{aV} \mathsf{ret}\, e'\delta_2$

This means from Definition 95 it suffices to prove that
$\mathsf{ret}\, e'\delta_1 \Downarrow_i^k v_a \implies \mathsf{ret}\, e'\delta_2 \Downarrow^k v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This further means that given some $\mathsf{ret}\, e'\delta_1 \Downarrow_i^k v_a$ it suffices to prove that
$\mathsf{ret}\, e'\delta_2 \Downarrow^k v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-R1)

From E-return we know that $k = 0$ and $e'\delta_1 \Downarrow_i v_a$

IH: $e'\delta_1 \overset{s}{\approx}_{aE} e'\delta_2$
This means from Definition 95 we have
$\forall j < s.e'\delta_1 \Downarrow_j v_a \implies e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-j}{\approx}_{aV} v_b$

Since we are given that $e'\delta_1 \Downarrow_i v_a$ therefore we get
$e'\delta_2 \Downarrow v_b \wedge v_a \overset{s-j}{\approx}_{aV} v_b$

Since $e'\delta_2 \Downarrow v_b$ therefore from E-return we also have
$\mathsf{ret}\, e'\delta_2 \Downarrow^0 v_b$
This proves the desired

18. $e = \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c$:
We need to prove that $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \overset{s}{\approx}_{aE} \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2$

This means from Definition 95 it suffices to prove that
$\forall i < s.\, \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i v_a \implies \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$

This means that given some $i < s$ s.t $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i v_a$ it suffices to prove that
$\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2 \Downarrow v_b \wedge v_a \overset{s-i}{\approx}_{aV} v_b$ (FTE-BI0)
From E-val we know that $v_a = \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1$ and $i = 0$

We need to prove
(a) $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2 \Downarrow v_b$:
From E-val we know that $v_b = \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2$
(b) $v_a \overset{s-i}{\approx}_{aV} v_b$:
We need to prove that $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \overset{s}{\approx}_{aV} \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2$

From Definition 95 it suffices to prove that
$\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i^k v_{t1} \implies \mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2 \Downarrow^k v_{t2} \wedge v_{t1} \overset{s-i}{\approx}_{aV} v_{t2}$

This means that given $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i^k v_{t1}$ it suffices to prove that
$\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_2 \Downarrow^k v_{t2} \wedge v_{t1} \overset{s-i}{\approx}_{aV} v_{t2}$ (F-BI1)

<u>IH1</u>: $e_b\delta_1 \overset{s}{\approx}_{aE} e_b\delta_2$
This means from Definition 95 we have
$\forall j < s. e_b\delta_1 \Downarrow_j v_{a1} \implies e_b\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-j}{\approx}_{aV} v_{b1}$

Since we know that $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i v_a$ therefore from E-bind we know that $\exists i_1. e_b\delta_1 \Downarrow_{i_1} v_{a1}$

Instantiating with i_1 we get $e_b\delta_2 \Downarrow v_{b1} \wedge v_{a1} \overset{s-i_1}{\approx}_{aV} v_{b1}$

Since v_{a1} is a mondic value and $v_{a1} \Downarrow_{i_1'}^{k1} v_{a1}'$

Since $v_{a1} \overset{s-i_1}{\approx}_{aV} v_{b1}$ therefore from Definition 95 we know that
$v_{a1} \Downarrow_{i_1'}^{k1} v_{a1}' \implies v_{b1} \Downarrow^{k1} v_{b1}' \wedge v_{a1}' \overset{s-i_1-i_1'}{\approx}_{aV} v_{b1}'$

Since we are given that $v_{a1} \Downarrow_{i_1'}^{k1} v_{a1}'$ therefore we have
$v_{b1} \Downarrow^{k1} v_{b1}' \wedge v_{a1}' \overset{s-i_1-i_1'}{\approx}_{aV} v_{b1}'$

<u>IH2</u>: $e_c[e_{a1}'/x]\delta_1 \overset{s-i_1-i_1'}{\approx}_{aE} e_c[e_{b1}'/x]\delta_2$
This means from Definition 95 we have
$\forall j < s. e_c[e_{a1}'/x]\delta_1 \Downarrow_j v_{a2} \implies e_c[e_{b1}'/x]\delta_2 \Downarrow v_b \wedge v_a \overset{s-i_1-i_1'-j}{\approx}_{aV} v_{b2}$

Since we know that $\mathsf{bind}\, x = e_b\ \mathsf{in}\ e_c\delta_1 \Downarrow_i v_a$ therefore from E-bind we know that $\exists i_2. e_c[e_{a1}'/x]\delta_1 \Downarrow_{i_2} v_{a2}$

Instantiating with i_2 we get $e_c[e_{b1}'/x]\delta_2 \Downarrow v_b \wedge v_{a2} \overset{s-i_1-i_1'-i_2}{\approx}_{aV} v_{b2}$

From E-bind we know that v_{a2} is a mondic value and $v_{a2} \Downarrow_{i_2'}^{k2} v_{a2}'$

Since $v_{a2} \overset{s-i_1-i_1'-i_2}{\approx}_{aV} v_{b2}$ therefore from Definition 95 we know that
$v_{a2} \Downarrow_{i_2'}^{k2} v_{a2}' \implies v_{b2} \Downarrow^{k2} v_{b2}' \wedge v_{a2}' \overset{s-i_1-i_1'-i_2-i_2'}{\approx}_{aV} v_{b2}'$

Since we are given that $v_{a2} \Downarrow_{i_2'}^{k2} v_{a2}'$ therefore we have
$v_{b2} \Downarrow^{k2} v_{b2}' \wedge v_{a2}' \overset{s-i_1-i_1'-i_2-i_2'}{\approx}_{aV} v_{b2}'$

This proves the desired
19. $e = \uparrow^n$:
Trivial
20. $e = \mathsf{release}\, e_r = x\ \mathsf{in}\ e_c$:
Similar reasoning as in the bind case

21. $e = \mathsf{store}\, e$:
Similar reasoning as in the return case

□

Lemma 100 (Equivalence relation of λ-amor is reflexive for values)**.** $\forall v, s.\ v \stackrel{s}{\approx}_{aV} v$

Proof. Instantiating Theorem 99 with . for δ_1 and δ_2, v for e and with the given s we get $v \stackrel{s}{\approx}_{aE} v$
From Definition 95 this means we have
$\forall i < s.v \Downarrow_i v_a \implies v \Downarrow v_b \wedge v_a \stackrel{s-i}{\approx}_{aV} v_b$
Instantiating it with i as 0 and since we knwo that $v \Downarrow_0 v$ therefore we get the desired □

Lemma 101 (Property of app rule in λ-Amor)**.** $\forall e_1, e_2, e, s.$
$e_1 \stackrel{s}{\approx}_{aE} e_2 \implies e\ e_1 \stackrel{s}{\approx}_{aE} e\ e_2$

Proof. We get the desired from Theorem 99 □

Lemma 102 (Lemma for app1 : empty stack)**.** $\forall t, u, \rho, \theta, v_a, v_1, j.$
$\Theta; \Delta; . \vdash_{-} (\!|(t\ u, \rho, \epsilon)|\!) : - \wedge$
$\underline{\Theta; \Delta; . \vdash_{-}} (\!|(t, \rho, (u, \rho).\epsilon)|\!) : - \wedge$
$\overline{(\!|(t\ u, \rho, \epsilon)|\!)}() \Downarrow v_a \Downarrow^j v_1 \implies$
$\exists v_b, v_2.\ \overline{(\!|(t, \rho, (u, \rho).\epsilon)|\!)}() \Downarrow v_b \Downarrow^j v_2 \wedge \forall s. v_1 \stackrel{s}{\approx}_{aE} v_2$

Proof. From Definition 92 know that
$(\!|(t\ u, \rho, \epsilon)|\!) = (\!|t\ u, \rho|\!) =$
$(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_n|\!) \qquad \text{(A1.0)}$

Similarly from Definition 92 we also have
$(\!|(t, \rho, (u, \rho).\epsilon)|\!) = (\!|(\!|(t, \rho)|\!)\ (\!|(u, \rho)|\!), ., \epsilon|\!) = (\!|(\!|(t, \rho)|\!)\ (\!|(u, \rho)|\!), .|\!) = (\!|(t, \rho)|\!)\ (\!|(u, \rho)|\!) =$
$((\lambda x_1 \ldots x_n.t)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_n|\!))\ \ (\lambda x_1 \ldots x_n.u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_n|\!) \qquad \text{(A1.1)}$

$\underline{\text{Since } \Theta; \Delta}; . \vdash_{-} (\!|(t\ u, \rho, \epsilon)|\!) : -$ therefore from Theorem 74 we know that
$\overline{(\!|(t\ u, \rho, \epsilon)|\!)} =$
$\overline{(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_n|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ } \mathsf{bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1,n}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{t2,n}\ c)\ d$
where
$e_{t1,n} = \overline{(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_{n-1}|\!)}$
$e_{t2,n} = (\!|\mathsf{C}_n|\!)$

$\overline{e_{t1,n}} =$
$\overline{(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_{n-1}|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ } \mathsf{bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1,n-1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
where
$e_{t1,n-1} = \overline{(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)\ \ldots\ (\!|\mathsf{C}_{n-2}|\!)}$
$e_{t2,n-1} = (\!|\mathsf{C}_{n-1}|\!)$

...

$\overline{e_{t1,2}} =$
$\overline{(\lambda x_1 \ldots x_n.t\ u)\ (\!|\mathsf{C}_1|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ } \mathsf{bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1,1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{t2,1}\ c)\ d$
where
$e_{t1,1} = \overline{(\lambda x_1 \ldots x_n.t\ u)}$
$e_{t2,1} = (\!|\mathsf{C}_1|\!)$

$e_{t1,1} =$
$\overline{(\lambda x_1 \ldots x_n.t\ u)} =$
$\lambda p_1.\, \mathsf{ret}\, \lambda y.\lambda p_2.\, \mathsf{let}\,!\, x = y \mathsf{\ in\ release} - = p_1 \mathsf{\ in\ release} - = p_2 \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ } e_{t2}\ a$
where
$e_{t2} = \overline{(\lambda x_2 \ldots x_n.t\ u)}$

...

$e_{tn-1} =$
$\overline{(\lambda x_{n-1} x_n.t\ u)} =$
$\lambda p_1.\, \mathsf{ret}\ \lambda y.\lambda p_2.\, \mathsf{let}\,!\,x = y\ \mathsf{in}\ \mathsf{release} - = p_1\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{tn}\ a$
where
$e_{tn} = \overline{(\lambda x_n.t\ u)}$

$e_{tn} =$
$\overline{(\lambda x_n.t\ u)} =$
$\lambda p_1.\, \mathsf{ret}\ \lambda y.\lambda p_2.\, \mathsf{let}\,!\,x = y\ \mathsf{in}\ \mathsf{release} - = p_1\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e'_t\ a$
where
$e'_t = \overline{(t\ u)}$

$e'_t =$
$\overline{(t\ u)} =$
$\lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_t\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_u\ c)\ d$
where
$e_t = \overline{t}$
$e_u = \overline{u}$

Since we know that $\overline{(\!|(t\ u, \rho, \epsilon)|\!)}() \Downarrow v_a \Downarrow^j v_1$ therefore from the reduction rule we know that
$\exists j_l, L.\overline{(\!|t|\!)}\ () \Downarrow - \Downarrow^{j_l} L$ and $\exists j_a.L\ (coerce1!\overline{(\!|u|\!)}!())\ () \Downarrow - \Downarrow^{j_l} v_1$ s.t $j = j_l + j_a$

Similarly from (A1.1) we know that
$(\!|(t, \rho, (u, \rho).\epsilon)|\!) =$
$((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))\ \ (\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!)$

Since $\Theta; \Delta; . \vdash_{-} (\!|(t, \rho, (u, \rho).\epsilon)|\!) : -$ therefore from Theorem 74 we know that
$\overline{(\!|(t, \rho, (u, \rho).\epsilon)|\!)} =$
$\overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))\ \ ((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))} =$
$\lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t,n}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{u,n}\ c)\ d$
where
$e_{t,n} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))}$
$e_{u,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))}$

$e_{t,n} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))} =$
$\lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n}\ c)\ d$
where
$e_{t1,n} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-1}|\!))}$
$e_{t2,n} = \overline{\mathtt{C}_n}$

$e_{t1,n} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-1}|\!))} =$
$\lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n-1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
where
$e_{t1,n-1} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-2}|\!))}$
$e_{t2,n-1} = \overline{\mathtt{C}_{n-1}}$

...

$e_{t1,2} = \overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!))} =$
$\lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{l1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,1}\ c)\ d$
where
$e_{l1} = \overline{(\lambda x_1 \ldots x_n.t)}$
$e_{t2,1} = \overline{\mathtt{C}_1}$

$e_{l1} = \overline{(\lambda x_1 \ldots x_n.t)} =$
$\lambda p_1.\, \mathsf{ret}\ \lambda y.\lambda p_2.\, \mathsf{let}\,!\,x_1 = y\ \mathsf{in}\ \mathsf{release} - = p_1\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a$
where

$e_{l2} = \overline{(\lambda x_2 \ldots x_n.t)}$

$\ldots$

$e_{ln} = \overline{(\lambda x_n.t)} =$
$\lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_n = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_T\ a$
where
$e_T = \overline{t}$ (A1.2)

Similarly we also have
$e_{u,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))}$

$e_{u,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{u1,n}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u2,n}\ c)\ d$
where
$e_{u1,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-1}|\!))}$
$e_{u2,n} = \overline{\mathtt{C}_n}$

$e_{u1,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-1}|\!))} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{u1,n-1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$
where
$e_{u1,n-1} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-2}|\!))}$
$e_{u2,n-1} = \overline{\mathtt{C}_{n-1}}$

$e_{u1,n-1} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-2}|\!))} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{u1,n-2}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u2,n-2}\ c)\ d$
where
$e_{u1,n-2} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-3}|\!))}$
$e_{u2,n-2} = \overline{\mathtt{C}_{n-2}}$

$\ldots$

$e_{u1,2} = \overline{(\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{u1,1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u2,1}\ c)\ d$
where
$e_{u1,1} = \overline{(\lambda x_1 \ldots x_n.u)}$
$e_{u2,1} = \overline{\mathtt{C}_1}$

$e_{u1,1} = \overline{(\lambda x_1 \ldots x_n.u)} =$
$\lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_1 = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_{U,1}\ a$
where
$e_{U,1} = \overline{(\lambda x_2 \ldots x_n.u)}$

$e_{U,1} = \overline{(\lambda x_2 \ldots x_n.u)} =$
$\lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_2 = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_{U,2}\ a$
where
$e_{U,2} = \overline{(\lambda x_3 \ldots x_n.u)}$
$\ldots$

$e_{U,n-1} = \overline{(\lambda x_n.u)} =$
$\lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_n = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_{U,n}\ a$
where
$e_{U,n} = \overline{u}$ (A1.3)

$E_0 = \lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t,n}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u,n}\ c)\ d$
$v_b = \mathsf{release} - = () \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t,n}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u,n}\ c)\ d$
$E_{0.1} = \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t,n}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{u,n}\ c)\ d$

$E_{0.2} = \mathsf{bind}\, b = e_{t,n}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{u,n}\ c)\ d$
$E_{0.3} = \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{u,n}\ c)\ d$
$E_{0.4} = \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{u,n}\ c)\ d$
$e_{t,n} = \lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n}\ c)\ d$
$E_{t,n,1} = \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n}\ c)\ d$
$E_{t,n,1.1} = \mathsf{bind}\, b = e_{t1,n}\ ()\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n}\ c)\ d$
$e_{t1,n} = \lambda p.$
$\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n-1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
$E_{t1,n,1} = \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{t1,n-1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
$E_{t1,n,2} = \mathsf{bind}\, b = e_{t1,n-1}\ ()\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
$E_{t1,n,3} = \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
$E_{t1,n,4} = \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
$e_{t1,2} = \lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{l1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,1}\ c)\ d$
$E_{t1,2,1} = \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b = e_{l1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,1}\ c)\ d$
$E_{t1,2,2} = \mathsf{bind}\, b = e_{l1}\ a\ \mathsf{in}\ \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,1}\ c)\ d$
$E_{t1,2,3} = \mathsf{bind}\, c = \mathsf{store!}()\ \mathsf{in}\ \mathsf{bind}\, d = \mathsf{store}()\ \mathsf{in}\ b\ (coerce1\ !e_{t2,1}\ c)\ d$
$e_{l1} = \lambda p_1.\,\mathsf{ret}\ \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_1 = y\ \mathsf{in}\ \mathsf{release} - = p_1\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a$
$E_{l1} = \mathsf{ret}\ \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_1 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a$
$E_{l,1,1} = \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_1 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a$
$E_{l,1,2} = \mathsf{let}\,!\,x_1 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a$
$E_{l,1,3} = \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l2}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$
$E_{l2} = \mathsf{ret}\ \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_2 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l3}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$
$E_{l,2,1} = \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_2 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l3}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$
$E_{l,2,2} = \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l3}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2]$
$E_{l3} = \mathsf{ret}\ \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_3 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l4}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2]$
$E_{l,3,1} = \lambda y.\lambda p_2.\,\mathsf{let}\,!\,x_3 = y\ \mathsf{in}\ \mathsf{release} - = ()\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ e_{l4}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2]$

$D_{n-3}2$:

$$\frac{\vdots}{E_{l,n-3,1}\ (coerce1\ !\overline{(\!|\mathsf{C}_{n-3}|\!)}\ !())\ ()\Downarrow^{-} E_{l,n-2,1}}$$

D_12:

$$\frac{}{E_{l,1,1}(coerce1!\overline{(\!|\mathsf{C}_1|\!)}!())()\Downarrow - \Downarrow E_{l,2,1}}$$

D_11:

$$\frac{}{E_{l1}\Downarrow^{0} E_{l,1,1}}$$

$D_22.3$:

$$\frac{}{E_{l3}\Downarrow^{-} E_{l,3,1}}$$

$D_22.2$:

$$\frac{}{e_{l3}\ ()[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2]\Downarrow E_{l3}}$$

$D_22.1$:

$$\frac{}{(coerce1\ !\overline{(\!|\mathsf{C}_2|\!)}\ !())\Downarrow !(\overline{(\!|\mathsf{C}_2|\!)}\ ())}$$

D_22:

$$\frac{\dfrac{D_22.1}{E_{l,2,1}[(coerce1\ !\overline{(\!|\mathsf{C}_2|\!)}\ !())/y][()/p_2]\Downarrow E_{l1,2,2}} \qquad D_22.2 \qquad D_22.3}{E_{l,2,1}\ (coerce1\ !e_{t2,1}\ !())\ ()\Downarrow^{-} E_{l,3,1}}$$

D_21:

$$\frac{\dfrac{}{e_{l1}\ ()\Downarrow E_{l1}} \qquad D_11 \qquad D_12}{E_{t1,2,1}\Downarrow^{0} E_{l,2,1}}$$

$D_3 2$:

$$\frac{\vdots}{E_{l,3,1}\ (coerce1\ !\overline{(\!|\mathtt{C}_3|\!)}\ !())\ ()\ \Downarrow^{-}\ E_{l,4,1}}$$

$D_3 1$:

$$\frac{e_{t1,2}\ ()\Downarrow E_{t1,2,1} \qquad D_2 1 \qquad D_2 2}{E_{t1,3,1}\Downarrow E_{l,3,1}}$$

$D_{n-2}2$:

$$\frac{\vdots}{E_{l,(n-2),1}\ (coerce1\ !\overline{(\!|\mathtt{C}_{n-2}|\!)}\ !())\ ()\ \Downarrow^{0}\ E_{l,(n-1),1}}$$

$D_{n-2}1$:

$$\frac{\dfrac{}{e_{t1,n-3}\ ()\Downarrow E_{t1,n-3,1}} \qquad \dfrac{\dfrac{}{e_{t1,3}()\Downarrow E_{t1,3,1}} \qquad D_3 1 \qquad D_3 2}{\vdots} \qquad D_{n-3}2}{E_{t1,n-2,1}\Downarrow^{-} E_{l,n-2,1}}$$

$D_{n-1}2$:

$$\frac{\vdots}{E_{l,n-1,1}(coerce1!\overline{(\!|\mathtt{C}_{n-1}|\!)}!())()\ \Downarrow^{0}\ E_{l,n,1}}$$

$D_{n-1}1$:

$$\frac{\dfrac{}{e_{t1,n-2}\ ()\Downarrow E_{t1,n-2,1}} \qquad D_{n-2}1 \qquad D_{n-2}2}{E_{t1,n-1,1}\Downarrow E_{l,n-1,1}}$$

$D_n 2$:

$$\frac{\dfrac{\dfrac{}{\overline{t}[!(\overline{(\!|\mathtt{C}_n|\!)}\ ())/x_n]\Downarrow - \Downarrow^{j_l}\ L}\ \text{By inversion}}{E_{l,n,1}[(coerce1!\overline{(\!|\mathtt{C}_n|\!)}!())/x_n][()/p_2]\Downarrow^{j_l}\ L}}{E_{l,n,1}(coerce1!\overline{(\!|\mathtt{C}_n|\!)}!())\ ()\Downarrow^{j_l}\ L}$$

$D_n 1$:

$$\frac{\dfrac{}{e_{t1,n-1}\ ()\Downarrow E_{t1,n-1,1}} \qquad D_{(n-1)}1 \qquad D_{(n-1)}2}{E_{t1,n,1}\Downarrow^{j}\ E_{l,n,1}}$$

D2:

$$\frac{\dfrac{}{v_a\Downarrow^{j}\ v_1}\ \text{Given} \qquad \dfrac{}{v_a \overset{s}{\approx}_{aV} v_b}}{v_b\Downarrow^{j}\ v_2 \qquad v_1 \overset{s}{\approx}_{aV} v_2}\ \text{Definition 95}$$

T1:

$$\frac{\dfrac{\dfrac{}{L\ (coerce1\ !e_{u,n}\ !())\ ()\Downarrow - \Downarrow^{j_a}\ v_b \qquad v_a \overset{s}{\approx}_{aV} v_b}\ \text{Claim, Lemma 101, Definition 95}}{E_{0.4}[L/b][!()/c]\Downarrow^{j_a}\ v_b}}{E_{0.3}[L/b]\Downarrow^{j_a}\ v_b}$$

T0:

$$\frac{\dfrac{}{e_{t1,n}\ ()\Downarrow E_{t1,n,1}} \qquad D_n 1 \qquad D_n 2}{E_{t,n,1}\Downarrow^{j}\ L}\ \text{E-bind}$$

D0.0:

$$\frac{\dfrac{\dfrac{}{\mathsf{store}()\Downarrow^{0}()} \qquad \dfrac{\dfrac{}{e_{t,n}\ ()\Downarrow E_{t,n,1}} \qquad T0 \qquad T1 \qquad D2}{E_{0.2}\Downarrow^{j}\ v_2}\ \text{E-bind}}{E_{0.1}\Downarrow^{j}\ v_2}\ \text{E-bind}}{v_b\Downarrow^{j}\ v_2}\ \text{E-release}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\overline{E_0() \Downarrow v_b} \quad D0.0}{E_0\ () \Downarrow v_b \Downarrow^j v_2}}{\overline{\overline{((\lambda x_1 \ldots x_n.t)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!))\ (\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_n|\!)}\ () \Downarrow v_b \Downarrow^j v_2}}}{(\!|(t, \rho, (u, \rho).\epsilon)|\!)\ () \Downarrow v_b \Downarrow^j v_2}$$

Claim: $\forall s.coerce1\ !\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ !() \overset{s}{\approx}_{aE} coerce1\ !e_{u,n}\ !()$

Proof

From Definition 95 it suffices to prove
$\forall i < s.coerce1\ !\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ !() \Downarrow_i v_1 \implies coerce1\ !e_{u,n}\ !() \Downarrow v_2 \wedge v_1 \overset{s-i}{\approx}_{aV} v_2$

This further means that given some $i < s$ s.t $coerce1\ !\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ !() \Downarrow_i v_1$
and we need to prove
$coerce1\ !e_{u,n}\ !() \Downarrow v_2 \wedge v_1 \overset{s-i}{\approx}_{aV} v_2$ (C0)

Since we are given that $coerce1\ !\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ !() \Downarrow v_1$
This means from Definition 83 we have $v_1 = !(\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ ())$

Similarly again from Definition 83 we know that
$v_2 = !(e_{u,n}\ ())$

In order to prove that $!(\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ ()) \overset{s-i}{\approx}_{aE} !(e_{u,n}\ ())$
from Definition 95 it suffices to prove that
$(\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ ()) \overset{s-i}{\approx}_{aE} (e_{u,n}\ ())$

Using Definition 95 it suffices to prove
$\forall j < (s-i).(\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ ()) \Downarrow_j v_1' \implies (e_{u,n}\ ()) \Downarrow v_2' \wedge v_1' \overset{s-i-j}{\approx}_{aV} v_2'$

This means given some $j < (s-i)$ s.t $(\overline{u}[\overline{(\!|\mathtt{C}_1|\!)}\ ()/x_1] \ldots [\overline{(\!|\mathtt{C}_n|\!)}\ ()/x_n]\ ()) \Downarrow_j v_1'$
it suffices to prove that
$(e_{u,n}\ ()) \Downarrow v_2' \wedge v_1' \overset{s-i-j}{\approx}_{aV} v_2'$

From the embedding of dlPCF into λ-amor we know that v_1' is a value of monadic type

Since we know that
$e_{u,n} = \lambda p.$
$\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{u1,n}\ a \mathsf{\ in\ bind}\, c = \mathsf{store}!() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{u2,n}\ c)\ d$
where
$e_{u1,n} = \overline{((\lambda x_1 \ldots x_n.u)\ (\!|\mathtt{C}_1|\!)\ \ldots\ (\!|\mathtt{C}_{n-1}|\!))}$
$e_{u2,n} = \overline{\mathtt{C}_n}$

$e_{u,n}\ () \Downarrow v_2'$ from E-app where
$v_2' = \mathsf{release} - = () \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{u1,n}\ a \mathsf{\ in\ bind}\, c = \mathsf{store}!() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{u2,n}\ c)\ d$

Now we need to prove that $v_1' \overset{s-i-j}{\approx}_{aV} v_2'$

From Definition 95 it suffices to prove that
$v_1' \Downarrow_l^k v_a' \implies v_2' \Downarrow^k v_b' \wedge v_a' \overset{s-i-j-l}{\approx}_{aV} v_b'$

This means given $v_1' \Downarrow_l^k v_a'$ it suffices to prove
$v_2' \Downarrow^k v_b' \wedge v_a' \overset{s-i-j-l}{\approx}_{aV} v_b'$

$v_2' = \mathsf{release} - = () \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{u1,n}\ a \mathsf{\ in\ bind}\, c = \mathsf{store}!() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{u2,n}\ c)\ d$
$E_{u,n,1} = \mathsf{bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{u1,n}\ a \mathsf{\ in\ bind}\, c = \mathsf{store}!() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{u2,n}\ c)\ d$
$E_{u,n,1.1} = \mathsf{bind}\, b = e_{u1,n}\ () \mathsf{\ in\ bind}\, c = \mathsf{store}!() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{u2,n}\ c)\ d$

$E_{u,n,1.2} = \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n}\ c)\ d$

$e_{u1,n} = \lambda p.$
$\text{release} - = p \text{ in } \text{bind}\, a = \text{store}() \text{ in } \text{bind}\, b = e_{u1,n-1}\ a \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$

$E_{u1,n,1} = \text{release} - = () \text{ in } \text{bind}\, a = \text{store}() \text{ in } \text{bind}\, b = e_{u1,n-1}\ a \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$

$E_{u1,n,2} = \text{bind}\, b = e_{u1,n-1}\ () \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$

$E_{u1,n,3} = \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$

$E_{u1,n,4} = \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,n-1}\ c)\ d$

$e_{u1,2} = \lambda p.\text{release} - = p \text{ in } \text{bind}\, a = \text{store}() \text{ in } \text{bind}\, b = e_{l1}\ a \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,1}\ c)\ d$

$E_{u1,2,1} = \text{release} - = () \text{ in } \text{bind}\, a = \text{store}() \text{ in } \text{bind}\, b = e_{l1}\ a \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,1}\ c)\ d$

$E_{u1,2,2} = \text{bind}\, b = e_{l1}\ a \text{ in } \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,1}\ c)\ d$

$E_{u1,2,3} = \text{bind}\, c = \text{store!}() \text{ in } \text{bind}\, d = \text{store}() \text{ in } b\ (coerce1\ !e_{u2,1}\ c)\ d$

$e_{l1} = \lambda p_1.\, \text{ret}\, \lambda y.\lambda p_2.\, \text{let}\,!\, x_1 = y \text{ in release} - = p_1 \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,2}\ a$

$E_{l1} = \text{ret}\, \lambda y.\lambda p_2.\, \text{let}\,!\, x_1 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,2}\ a$

$E_{l,1,1} = \lambda y.\lambda p_2.\, \text{let}\,!\, x_1 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,2}\ a$

$E_{l,1,2} = \text{let}\,!\, x_1 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,2}\ a$

$E_{l,1,3} = \text{release} - = () \text{ in release} - = () \text{ in bind}\, a = \text{store}() \text{ in } e_{U,2}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$

$E_{l2} = \text{ret}\, \lambda y.\lambda p_2.\, \text{let}\,!\, x_2 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,3}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$

$E_{l,2,1} = \lambda y.\lambda p_2.\, \text{let}\,!\, x_2 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,3}\ a[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1]$

$E_{l,2,2} = (\text{release} - = () \text{ in release} - = () \text{ in bind}\, a = \text{store}() \text{ in } e_{U,3}\ a)\ S_2$

$E_{l3} = (\text{ret}\, \lambda y.\lambda p_2.\, \text{let}\,!\, x_3 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,4}\ a)\ S_2$

$E_{l,3,1} = (\lambda y.\lambda p_2.\, \text{let}\,!\, x_3 = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,4}\ a)\ S_2$

$S_2 = [(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2]$

$E_{l,n,1} = (\lambda y.\lambda p_2.\, \text{let}\,!\, x_n = y \text{ in release} - = () \text{ in release} - = p_2 \text{ in bind}\, a = \text{store}() \text{ in } e_{U,n}\ a)\ S_{n-1}$

$S_{n-1} = [(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1] \ldots [(\overline{(\!|\mathsf{C}_{n-1}|\!)}\ ())/x_{n-1}]$

$D_{n-3}2$:

$$\frac{\vdots}{E_{l,(n-3),1}\ (coerce1\ !\overline{(\!|\mathsf{C}_{n-3}|\!)}\ !())\ ()\Downarrow^0 E_{l,(n-2),1}}$$

$D_1 2$:

$$\frac{}{E_{l,1,1}(coerce1!\overline{(\!|\mathsf{C}_1|\!)}!())() \Downarrow - \Downarrow E_{l,2,1}}$$

$D_1 1$:

$$\frac{}{E_{l1} \Downarrow^0 E_{l,1,1}}$$

$D_2 2.3$:

$$\frac{}{E_{l3} \Downarrow^0 E_{l,3,1}}$$

$D_2 2.2$:

$$\frac{}{e_{l3}\ ()[(\overline{(\!|\mathsf{C}_1|\!)}\ ())/x_1][(\overline{(\!|\mathsf{C}_2|\!)}\ ())/x_2] \Downarrow E_{l3}}$$

$D_2 2.1$:

$$\frac{}{(coerce1\ !\overline{(\!|\mathsf{C}_2|\!)}\ !()) \Downarrow !(\overline{(\!|\mathsf{C}_2|\!)}\ ())}$$

$D_2 2$:

$$\frac{\dfrac{D_2 2.1}{E_{l,2,1}[(coerce1\ !\overline{(\!|\mathsf{C}_2|\!)}\ !())/y][()/p_2] \Downarrow E_{l1,2,2}} \qquad D_2 2.2 \qquad D_2 2.3}{E_{l,2,1}\ (coerce1\ !e_{u2,1}\ !())\ () \Downarrow^0 E_{l,3,1}}$$

$D_2 1$:

$$\frac{\dfrac{}{e_{l1}\ () \Downarrow E_{l1}} \qquad D_1 1 \qquad D_1 2}{E_{u1,2,1} \Downarrow^0 E_{l,2,1}}$$

$D_3 2$:

$$\frac{\vdots}{E_{l,3,1}\ (coerce1\ !\overline{(\!|\mathsf{C}_3|\!)}\ !())\ () \Downarrow^0 E_{l,4,1}}$$

$D_3 1$:

$$\frac{e_{u1,2}\ () \Downarrow E_{u1,2,1} \qquad D_2 1 \qquad D_2 2}{E_{u1,3,1} \Downarrow E_{l,3,1}}$$

$D_{n-2} 2$:

$$\frac{\vdots}{E_{l,(n-2),1}\ (coerce1\ !\overline{(\!|\mathtt{C}_{n-2}|\!)}\ !())\ () \Downarrow^0 E_{l,(n-1),1}}$$

$D_{n-2} 1$:

$$\frac{\dfrac{}{e_{u1,n-3}\ () \Downarrow E_{u1,n-3,1}} \qquad \dfrac{\dfrac{}{e_{u1,3}() \Downarrow E_{u1,3,1}} \quad D_3 1 \quad D_3 2}{\vdots} \qquad D_{n-3} 2}{E_{u1,n-2,1} \Downarrow^0 E_{l,n-2,1}}$$

$D_{n-1} 2$:

$$\frac{\vdots}{E_{l,n-1,1}(coerce1!\overline{(\!|\mathtt{C}_{n-1}|\!)}!())() \Downarrow^0 E_{l,n,1}}$$

$D_{n-1} 1$:

$$\frac{\dfrac{}{e_{u1,n-2}\ () \Downarrow E_{u1,n-2,1}} \qquad D_{n-2} 1 \qquad D_{n-2} 2}{E_{u1,n-1,1} \Downarrow^0 E_{l,n-1,1}}$$

$D_n 2$:

$$\cfrac{\cfrac{\cfrac{}{\overline{u}[!(\overline{(\!|\mathtt{C}_1|\!)}\ ())/x_1] \ldots [!(\overline{(\!|\mathtt{C}_n|\!)}\ ())/x_n] \Downarrow v_1' \Downarrow^k v_a'}\ \text{Given}}{E_{l,n,1}[(coerce1!\overline{(\!|\mathtt{C}_n|\!)}!())/x_n][()/p_2] \Downarrow v_1' \Downarrow^k v_a'}}{E_{l,n,1}(coerce1!\overline{(\!|\mathtt{C}_n|\!)}!())\ () \Downarrow^k v_a'}$$

$D_n 1$:

$$\frac{\dfrac{}{e_{u1,n-1}\ () \Downarrow E_{u1,n-1,1}} \qquad D_{(n-1)} 1 \qquad D_{(n-1)} 2}{E_{u1,n,1} \Downarrow^0 E_{l,n,1}}$$

Main derivation:

$$\cfrac{\cfrac{\dfrac{}{e_{u1,n}\ () \Downarrow E_{u1,n,1}} \qquad D_n 1 \qquad D_n 2}{E_{u,n,1} \Downarrow^k v_a'}\ \text{E-bind}}{v_2' \Downarrow^k v_a'}\ \text{E-release}$$

From Lemma 100 we get $v_a' \overset{s-i-j-l}{\approx}_{aV} v_a'$

$\square$

Lemma 103 (Lemma for app1: non-empty stack). $\forall t, u, \rho, \theta, v_{\epsilon1}', v_{\epsilon1}, v_{\epsilon2}', v_{\epsilon2}, v_{\theta1}, j, j', j''$.
$(t\ u, \rho, \epsilon)$ *and* $(t, \rho, (u, \rho).\epsilon)$ *are well-typed*
$(t\ u, \rho, \theta)$ *and* $(t, \rho, (u, \rho).\theta)$ *are well-typed*
$(t\ u, \rho, \epsilon) \to (t, \rho, (u, \rho).\epsilon) \wedge (t\ u, \rho, \theta) \to (t, \rho, (u, \rho).\theta) \wedge$
$\overline{(\!|(t\ u, \rho, \epsilon)|\!)}\ () \Downarrow v_{\epsilon1}' \Downarrow^j v_{\epsilon1} \wedge \overline{(\!|(t, \rho, (u, \rho).\epsilon)|\!)}\ () \Downarrow v_{\epsilon2}' \Downarrow^{j'} v_{\epsilon2} \wedge \forall s. v_{\epsilon1} \overset{s}{\approx}_{aV} v_{\epsilon2} \wedge$
$\overline{(\!|(t\ u, \rho, \theta)|\!)}\ () \Downarrow v_{\theta1}' \Downarrow^{j''} v_{\theta1}$
$\Longrightarrow$
$\exists v_{\theta2}', v_{\theta2}, j'''.\ \overline{(\!|(t, \rho, (u, \rho).\theta)|\!)}\ () \Downarrow v_{\theta2}' \Downarrow^{j'''} v_{\theta2} \wedge (j - j') = (j'' - j''') \wedge \forall s. v_{\theta1} \overset{s}{\approx}_{aV} v_{\theta2}$

Proof. We prove this by induction on θ

1. Case $\theta = \epsilon$:
   Directly from given
2. Case $\theta = \mathtt{C}'.\theta'$:
   Let $\theta' = \mathtt{C}_1' \ldots \mathtt{C}_n'$ and $\theta'' = \mathtt{C}_1' \ldots \mathtt{C}_{n-1}'$

   Given:
   $(t\ u, \rho, \mathtt{C}'.\theta')$ and $(t, \rho, (u, \rho).\mathtt{C}'.\theta')$ are well-typed $\wedge$
   $(t\ u, \rho, \mathtt{C}'.\theta') \to (t, \rho, (u, \rho).\mathtt{C}'.\theta') \wedge \overline{(\!|(t\ u, \rho, \mathtt{C}'.\theta')|\!)}\ () \Downarrow v_{\theta1}' \Downarrow^{j''} v_{\theta1}$
   We need to prove that

$\exists v'_{\theta 2}, v_{\theta 2}, j'''.$
$\overline{(\!|(t, \rho, (u, \rho).\mathsf{C}'.\theta')|\!)}\ () \Downarrow v'_{\theta 2} \Downarrow^{j'''} v_{\theta 2} \wedge (j - j') = (j'' - j''') \wedge \forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$ (ET-0)

From IH we know
$(t\ u, \rho, \mathsf{C}'.\theta'')$ and $(t, \rho, (u, \rho).\mathsf{C}'.\theta'')$ are well-typed $\wedge$
$(t\ u, \rho, \mathsf{C}'.\theta'') \rightarrow (t, \rho, (u, \rho).\mathsf{C}'.\theta'') \wedge \overline{(\!|(t\ u, \rho, \mathsf{C}'.\theta'')|\!)}\ () \Downarrow v'_{\theta 11} \Downarrow^{j''_1} v_{\theta 11} \implies \exists j'''_1, v'_{\theta 22}, v_{\theta 22}.$
$\overline{(\!|(t, \rho, (u, \rho).\mathsf{C}'.\theta'')|\!)}\ () \Downarrow v'_{\theta 22} \Downarrow^{j'''_1} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1) \wedge \forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22}$ (ET-IH)

From Definition 91 and Definition 92 we know that
$(\!|(t\ u, \rho, \mathsf{C}'.\theta')|\!) = (\!|((\!|t\ u, \rho|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))|\!)$ (ET-1)

Since $(t\ u, \rho, \mathsf{C}'.\theta')$ is well typed therefore we know that
$\overline{(\!|(t\ u, \rho, \mathsf{C}'.\theta')|\!)} = \overline{(\!|((\!|t\ u, \rho|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in}\ b\ (coerce1\ !e_{t2}\ c)\ d$
where
$e_{t1} = \overline{(\!|((\!|t\ u, \rho|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!))|\!)}$
$e_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$ (ET-1.1)

From Krivine reduction (app rule) we also know that $(t\ u, \rho, \mathsf{C}'.\theta'') \rightarrow (t, \rho, (u, \rho).\mathsf{C}'.\theta'')$
Also since we know that $\overline{(\!|(t\ u, \rho, \mathsf{C}'.\theta')|\!)}\ () \Downarrow v'_{\theta 1} \Downarrow^{j''} v_{\theta 1}$ therefore we also know that $\exists j''_1, v'_1, v_1.e_{t1}() \Downarrow v_1 \Downarrow^{j''_1} v'_1$

Also since we know that
$(t\ u, \rho, \mathsf{C}'.\theta')$ and $(t, \rho, (u, \rho).\mathsf{C}'.\theta')$ are well-typed
therefore from Lemma 108 we also know that
$(t\ u, \rho, \mathsf{C}'.\theta'')$ and $(t, \rho, (u, \rho).\mathsf{C}'.\theta'')$ are well-typed

Therefore from (ET-IH) we have
$\exists j'''_1, v'_{\theta 22}, v_{\theta 22}.\ \overline{(\!|(t, \rho, (u, \rho).\mathsf{C}'.\theta'')|\!)}\ () \Downarrow v'_{\theta 22} \Downarrow^{j'''_1} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1)$
$\wedge\ \forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22}$ (ET-2)

From (ET-0) and Definition 91, Definition 92 it suffices to prove that
$\exists j''', v'_{\theta 2}, v_{\theta 2}.\ \overline{(\!|((\!|t, \rho|\!)\ (\!|(u, \rho)|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))|\!)}\ () \Downarrow v'_{\theta 2} \Downarrow^{j'''} v_{\theta 2} \wedge (j - j') = (j'' - j''') \wedge \forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$
(ET-3)

Since $(t, \rho, (u, \rho).\mathsf{C}'.\theta')$ is well typed therefore we know that
$\overline{(\!|((\!|t, \rho|\!)\ (\!|(u, \rho)|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b' = e'_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in}\ b'\ (coerce1\ !e'_{t2}\ c)\ d$
where
$e'_{t1} = \overline{(\!|((\!|t, \rho|\!)\ (\!|(u, \rho)|\!)\ (\!|\mathsf{C}'|\!) \ldots (\!|\mathsf{C}_{n-1}|\!))|\!)}$
$e'_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$

From (ET-2) we know that $e'_{t1}() \Downarrow v'_{\theta 22} \Downarrow^{j'''_1} v_{\theta 22}$

and we need to prove that $v_{\theta 22}\ (coerce1\ !e'_{t2}\ c)\ d \Downarrow v_t \Downarrow^{j''' - j'''_1} v_{\theta 2}$ (ET-p)

Since we are given that $(\!|(t\ u, \rho, \mathsf{C}'.\theta')|\!)\ () \Downarrow v'_{\theta 1} \Downarrow^{j''} v_{\theta 1}$ this means from (ET-1.1) we have
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in}\ b\ (coerce1\ !e_{t2}\ c)\ d \Downarrow v'_{\theta 1} \Downarrow^{j''} v_{\theta 1}$

Also since we are given that $\overline{(\!|(t\ u, \rho, \mathsf{C}'.\theta'')|\!)}\ () \Downarrow v'_{\theta 11} \Downarrow^{j''_1} v_{\theta 11}$ this means we have
$e_{t1}\ () \Downarrow v'_{\theta 11} \Downarrow^{j''_1} v_{\theta 11}$

This means $v_{\theta 11}\ (coerce1\ !e_{t2}\ c)\ d \Downarrow - \Downarrow^{y} v_{\theta 1}$ for some y s.t $y + j''_1 = j''$

Since $\forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22}$ and $e_{t2} = e'_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$ therefore from Definition 95 we get $\forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$. Also from Definition 95 we have
$j'' - j''_1 = j''' - j'''_1 =$
$j'' - j''' = j''_1 - j'''_1 =$
$j'' - j''' = j - j'$ (From ET-IH)

$\square$

Lemma 104 (Cost and size lemma)**.** $\forall e_s, D_s, E_s.$

$(e_s, \epsilon, \epsilon) \xrightarrow{*} D_s \to E_s \wedge$
D_s is well-typed $\wedge$
E_s is well-typed $\wedge$
$e_t = \overline{(\!|D_s|\!)} \wedge e_t\ () \Downarrow v_a \Downarrow^j v_1$
$\implies$
$\exists e'_t.\ e'_t = \overline{(\!|E_s|\!)} \wedge e'_t\ () \Downarrow v_b \Downarrow^{j'} v_2 \wedge \forall s.\ v_1 \stackrel{s}{\approx}_{aE} v_2 \wedge$
1. $j' = j \wedge |D_s| > |E_s|$ *or*
2. $j' = j - 1 \wedge |E_s| < |D_s| + |e_s|$

Proof. We case analyze on the $D_s \to E_s$ reduction

1. App1:
   Given $D_s = (t\ u, \rho, \theta)$ and $E_s = (t, \rho, (u, \rho).\theta)$
   Let $D'_s = (t\ u, \rho, \epsilon)$ and $E'_s = (t, \rho, (u, \rho).\epsilon)$
   Since we are given that D_s is well-typed and E_s is well-typed therefore from Lemma 105 we also have D'_s is well-typed and E'_s is well-typed

   Also since we know that $e_t\ () \Downarrow v_a \Downarrow^j v_1$ therefore from Lemma 106 we also know that
   $\exists j_e.\overline{(\!|D'_s|\!)}\ () \Downarrow v'_d \Downarrow^{j_e} v_d$
   From Lemma 102 we know that $\exists v_e.\ \overline{(\!|E'_s|\!)}\ () \Downarrow v'_e \Downarrow^{j_e} v_e$ s.t $\forall s.v_d \stackrel{s}{\approx}_{aV} v_e$

   And finally from Lemma 103 we know that $\overline{(\!|E_s|\!)}\ () \Downarrow v_b \Downarrow^j v_2$ s.t $\forall s.v_1 \stackrel{s}{\approx}_{aV} v_2$

   $|D_s| > |E_s|$ holds directly from the Definition of $|-|$
2. App2:
   Given: $(\lambda x.t, \rho, c.\theta) \to (t, c.\rho, \theta)$
   We induct on θ
   (a) Case $\theta = \epsilon$:
   Since we are given that D_s i.e $(\lambda x.t, \rho, c.\epsilon)$ is well typed
   Therefore from Theorem 94 $(\!|(\lambda x.t, \rho, c.\epsilon)|\!)$ is well-typed
   From Definition 92 $(\!|((\!|\lambda x.t, \rho|\!)\ (\!|c|\!), ., \epsilon)|\!)$ is well-typed
   Again from Definition 92 $(\!|\lambda x.t, \rho|\!)\ (\!|c|\!)$ is well-typed
   From Definition 91 we have
   $((\lambda x_1 \dots x_n.\lambda x.t)\ (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_n|\!)\ (\!|c|\!))$ is well-typed

   Therefore from Theorem 74 we know that
   $\overline{(\!|D_s|\!)} =$
   $\overline{((\lambda x_1 \dots x_n.\lambda x.t)\ (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_n|\!)\ (\!|\mathsf{C}|\!))} =$
   $\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d$
   where
   $e_{t1} = \overline{((\lambda x_1 \dots x_n.\lambda x.t)\ (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_n|\!))}$
   $e_{t2} = \overline{(\!|\mathsf{C}|\!)}$ (S-A0)

   Since we are given that $\overline{(\!|D_s|\!)}\ () \Downarrow v_a \Downarrow^j v_1$
   therefore from the evaluation rules we know that
   $\overline{(\!|t|\!)}[\overline{(\!|\mathsf{C}|\!)}()/x][\overline{(\!|\mathsf{C}_1|\!)}()/x_1] \dots [\overline{(\!|\mathsf{C}_1|\!)}()/x_1] \Downarrow - \Downarrow^j v_1$ (S-A0.1)

   Similarly since we are given that E_s i.e $(t, c.\rho, \epsilon)$ is well-typed
   Therefore from Theorem 94 $(\!|(t, c.\rho, \epsilon)|\!)$ is well-typed
   From Definition 92 $(\!|t, c.\rho|\!)$ is well-typed
   From Definition 91 we have $((\lambda x, x_1 \dots x_n.t)\ (\!|\mathsf{C}|\!)\ (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_n|\!))$ is well-typed

   Therefore from Theorem 74 we know that
   $\overline{(\!|E_s|\!)} =$
   $\overline{((\lambda x\ x_1 \dots x_n.t)\ (\!|\mathsf{C}|\!)\ (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_n|\!))} =$
   $\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d$
   where
   $e_{t1} = \overline{((\lambda x\ x_1 \dots x_n.t)\ (\!|\mathsf{C}|\!), (\!|\mathsf{C}_1|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!))}$
   $e_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$ (S-A1)

   From (SA-0.1) we know that

$\overline{(\!|E_s|\!)}\ () \Downarrow - \Downarrow^j v_1$
And finally from Theorem 99 we have $\forall s.v_1 \overset{s}{\approx}_{aV} v_1$

(b) Case $\theta = \mathtt{C}'.\theta'$:
Let $\theta' = \mathtt{C}_{\theta_1} \ldots \mathtt{C}_{\theta_n}$ and $\rho = \mathtt{C}_{\rho_1} \ldots \mathtt{C}_{\rho_n}$

Since we are given that D_s i.e $(\lambda x.t, \rho, \mathtt{C}.\mathtt{C}'.\theta')$ is well typed
Therefore from Theorem 94 we know that $(\!|(\lambda x.t, \rho, \mathtt{C}.\mathtt{C}'.\theta')|\!)$ is well-typed
From Definition 92 we also have $(\!|((\!|\lambda x.t, \rho|\!)\ (\!|\mathtt{C}|\!), ., \mathtt{C}'.\theta')|\!)$ is well-typed
which further means that $(\!|((\!|\lambda x.t, \rho|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}'|\!), ., \theta')|\!)$ is well-typed
which further means that $(\!|(\!|\lambda x.t, \rho|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_1}|\!)|\!)$ is well-typed
which further means that $(\lambda x_1 \ldots x_n.\lambda x.t)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)$ is well-typed

From Theorem 74 we have
$\overline{(\!|D_s|\!)} = \overline{(\lambda x_1 \ldots x_n.\lambda x.t)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{~in~} \mathsf{bind}\, a = \mathsf{store}() \mathsf{~in~} \mathsf{bind}\, b = e_{t1}\ a \mathsf{~in~} \mathsf{bind}\, c = \mathsf{store!}() \mathsf{~in~} \mathsf{bind}\, d = \mathsf{store}() \mathsf{~in~} b\ (coerce1\ !e_{t2}\ c)\ d$
where
$e_{t1} = \overline{(\lambda x_1 \ldots x_n.\lambda x.t)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_{m-1}}|\!)}$
$e_{t2} = \overline{(\!|\mathtt{C}_{\theta_m}|\!)}$ (S-A2)

Since we are given that $\overline{(\!|D_s|\!)}\ () \Downarrow v_a \Downarrow^j v_1$
therefore from the evaluation rules we know that
$\exists e', j_1.\overline{(\!|t|\!)}[\overline{(\!|\mathtt{C}|\!)}()/x][\overline{(\!|\mathtt{C}_1|\!)}()/x_1] \ldots [\overline{(\!|\mathtt{C}_1|\!)}()/x_1] \Downarrow - \Downarrow^{j_1} \lambda x' x_1 \ldots x_m.e'$
s.t
$\lambda x' x_1 \ldots x_m.e' \overline{(\!|t|\!)}[\overline{(\!|\mathtt{C}'|\!)}()/x][\overline{(\!|\mathtt{C}_{\theta_1}|\!)}()/x_1] \ldots [\overline{(\!|\mathtt{C}_{\theta_m}|\!)}()/x_m] \Downarrow - \Downarrow^{j_2} v_1$
and $j_1 + j_2 = j$ (S-A2.1)

Similarly since we are given that E_s i.e $(t, \mathtt{C}.\rho, \mathtt{C}'.\theta')$ is well typed
Therefore from Theorem 94 we know that $(\!|(t, \mathtt{C}.\rho, \mathtt{C}'.\theta')|\!)$ is well-typed
From Definition 92 we also have $(\!|((\!|t, \mathtt{C}.\rho|\!)\ (\!|\mathtt{C}'|\!), ., \theta')|\!)$ is well-typed
which further means that $(\!|((\!|t, \mathtt{C}.\rho|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!))|\!)$ is well-typed
which further means that $(\lambda x, x_1 \ldots x_n.t)(\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)$ is well-typed

From Theorem 74 we have
$\overline{(\!|E_s|\!)} = \overline{(\lambda x, x_1 \ldots x_n.t)(\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)} =$
$\lambda p.\mathsf{release} - = p \mathsf{~in~} \mathsf{bind}\, a = \mathsf{store}() \mathsf{~in~} \mathsf{bind}\, b = e_{t1}\ a \mathsf{~in~} \mathsf{bind}\, c = \mathsf{store!}() \mathsf{~in~} \mathsf{bind}\, d = \mathsf{store}() \mathsf{~in~} b\ (coerce1\ !e_{t2}\ c)\ d$
where
$e_{t1} = \overline{(\lambda x, x_1 \ldots x_n.t)(\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}'|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_{m-1}}|\!)}$
$e_{t2} = \overline{(\!|\mathtt{C}_{\theta_m}|\!)}$ (S-A3)

From (S-A2.1) it is clear that
$\overline{(\!|E_s|\!)}\ () \Downarrow - \Downarrow^j v_1$
And finally from Theorem 99 we have $\forall s.v_1 \overset{s}{\approx}_{aV} v_1$

$|D_s| > |E_s|$ holds directly from the Definition of $|-|$

3. Fix:
Given: $(\mathsf{fix} x.t, \rho, \theta) \to (t, (\mathsf{fix} x.t, \rho).\rho, \theta)$
Let $D'_s = (\mathsf{fix} x.t, \rho, \epsilon)$ and $E'_s = (t, (\mathsf{fix} x.t, \rho).\rho, \epsilon)$

Since we are given that D_s and E_s are well-typed therefore from Lemma 105 we know that D'_s and E'_s are well-typed too.

Also since we know that $e_t\ () \Downarrow v_a \Downarrow^j v_1$ therefore from Lemma 106 we also know that
$\exists j_e.\overline{(\!|D'_s|\!)} \Downarrow - \Downarrow^{j_e} v_e$
From Lemma 109 we know that $\overline{(\!|E'_s|\!)}\ () \Downarrow v'_e \Downarrow^{j_e} v_e$

And then from Lemma 107 we know that $\overline{(\!|E_s|\!)} \Downarrow v_b \Downarrow^j v_2$ s.t $\forall s.v_1 \overset{s}{\approx}_{aV} v_2$

$|D_s| > |E_s|$ holds directly from the Definition of $|-|$

4. Var:
   Given: $D_s = (x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \theta)$ and $E_s = (t_x, \rho_x, \theta)$
   Let $D'_s = (x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \epsilon)$ and $E'_s = (t_x, \rho_x, \epsilon)$

   Since we are given that D_s and E_s are well-typed therefore from Lemma 105 we know that D'_s and E'_s are well-typed too.

   Also since we know that $e_t\ () \Downarrow - \Downarrow^j v_1$ therefore from Lemma 106 we also know that
   $\exists j_e.\overline{\llparenthesis D'_s \rrparenthesis} \Downarrow - \Downarrow^{j_e} v_e$
   From Lemma 111 we know that $\overline{\llparenthesis E'_s \rrparenthesis} \Downarrow - \Downarrow^{j_e - 1} v_e$

   And then from Lemma 110 we know that $\overline{\llparenthesis E_s \rrparenthesis} \Downarrow - \Downarrow^{j-1} v_2$ s.t $\forall s. v_1 \overset{s}{\approx}_{aV} v_2$

   $|E_s| < |D_s| + |e_s|$ holds directly from the Definition of $|-|$ and from Lemma 4.2 in [9]

$\square$

Lemma 105 (ϵ typing)**.** $\forall \Theta, \Delta, I, e, \rho, \theta.$
$\Theta; \Delta \vdash_{-} (e, \rho, \theta) : - \implies \Theta; \Delta \vdash_{-} (e, \rho, \epsilon) : -$

Proof. Main derivation:

$$\dfrac{\dfrac{\overline{\Theta; \Delta \vdash_I (e, \rho, \theta) : \tau}\ \text{Given}}{\Theta; \Delta \vdash_J (e, \rho) : \sigma}\ \text{By inversion} \qquad \overline{\Theta; \Delta \vdash_0 \epsilon : (\sigma, \sigma)}}{\Theta; \Delta \vdash_J (e, \rho, \epsilon) : \sigma}$$

$\square$

Lemma 106 (ϵ reduction)**.** $\forall e, \rho, \theta.$
(e, ρ, θ) *is well typed* $\wedge\ \overline{\llparenthesis (e, \rho, \theta) \rrparenthesis}\ () \Downarrow - \Downarrow^{-} - \implies \overline{\llparenthesis (e, \rho, \epsilon) \rrparenthesis}\ () \Downarrow - \Downarrow^{-} -$

Proof. Since (e, ρ, θ) is well typed therefore from Lemma 105 we also know that
(e, ρ, ϵ) is well typed

From Theorem 94 we know that $\llparenthesis (e, \rho, \epsilon) \rrparenthesis$ is also well typed

From Definition 92 we know that $\llparenthesis (e, \rho, \epsilon) \rrparenthesis = \llparenthesis (e, \rho) \rrparenthesis$

Let $\theta = \mathtt{C}_1 \dots \mathtt{C}_n$
Similarly from Definition 92 we also know that
$\llparenthesis (e, \rho, \theta) \rrparenthesis = \llparenthesis (e, \rho, \mathtt{C}_1 \dots \mathtt{C}_n) \rrparenthesis =$
$\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis, [], \mathtt{C}_2 \dots \mathtt{C}_n) \rrparenthesis =$
$\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \dots \llparenthesis \mathtt{C}_n \rrparenthesis, [], \epsilon) \rrparenthesis =$
$\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \dots \llparenthesis \mathtt{C}_n \rrparenthesis) \rrparenthesis$

From Theorem 74 we know that
$\overline{\llparenthesis (e, \rho, \theta) \rrparenthesis} =$
$\overline{\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \dots \llparenthesis \mathtt{C}_n \rrparenthesis) \rrparenthesis} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d$
where
$e_{t1} = \overline{\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \dots \llparenthesis \mathtt{C}_{n-1} \rrparenthesis) \rrparenthesis}$
$e_{t2} = \llparenthesis \mathtt{C}_n \rrparenthesis \qquad$ (E0)

Since $\overline{\llparenthesis (\llparenthesis e, \rho \rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \dots \llparenthesis \mathtt{C}_n \rrparenthesis) \rrparenthesis} \Downarrow - \Downarrow^{-} -$, therefore we also know that $\overline{\llparenthesis \llparenthesis e, \rho \rrparenthesis \rrparenthesis} \Downarrow - \Downarrow^{-} -$

$\square$

Lemma 107 (Lemma for fix : non-empty stack)**.** $\forall t, \rho, \theta, j, j', j'', v_{\epsilon 1}, v_{\epsilon 2}, v_{\theta 1}.$
$(\mathsf{fix}x.t, \rho, \epsilon)$ *and* $(t, (\mathsf{fix}x.t, \rho).\rho, \epsilon)$ *are well-typed*
$(\mathsf{fix}x.t, \rho, \theta)$ *and* $(t, (\mathsf{fix}x.t, \rho).\rho, \theta)$ *are well-typed*
$\overline{\llparenthesis (\mathsf{fix}x.t, \rho, \epsilon) \rrparenthesis}\ () \Downarrow - \Downarrow^{j} v_{\epsilon 1} \wedge \overline{\llparenthesis (t, (\mathsf{fix}x.t, \rho).\rho, \epsilon) \rrparenthesis}\ () \Downarrow - \Downarrow^{j'} v_{\epsilon 1} \wedge \forall s. v_{\epsilon 1} \overset{s}{\approx}_{aV} v_{\epsilon 2} \wedge$
$\overline{\llparenthesis (\mathsf{fix}x.t, \rho, \theta) \rrparenthesis}\ () \Downarrow - \Downarrow^{j''} v_{\theta 1} \wedge$
$\implies$
$\exists v_{\theta 2}, j'''.\ \overline{\llparenthesis (t, (\mathsf{fix}x.t, \rho).\rho, \theta) \rrparenthesis}\ () \Downarrow - \Downarrow^{j'''} v_{\theta 2} \wedge \forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2} \wedge (j - j') = (j'' - j''')$

Proof. We prove this by induction on θ

1. Case $\theta = \epsilon$:
   Directly from given
2. Case $\theta = \mathsf{C}'.\theta'$:
   Let $\theta' = \mathsf{C}'_1 \dots \mathsf{C}'_n$ and $\theta'' = \mathsf{C}'_1 \dots \mathsf{C}'_{n-1}$

   Given:
   $\underline{(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')}$ and $(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')$ are well-typed $\wedge$
   $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j''}\ v_{\theta 1}$
   We need to prove that
   $\overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j'''}\ v_{\theta 2} \wedge \forall s.v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2} \wedge (j - j') = (j'' - j''')$ (ET-0)

   From IH we know
   $\underline{(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta'')}$ and $(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta'')$ are well-typed,
   $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta'')|\!)}\ () \Downarrow - \Downarrow^{j''_1}\ v_{\theta 11} \implies$
   $\overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta'')|\!)}\ () \Downarrow - \Downarrow^{j'''_1}\ v_{\theta 22} \wedge \forall s.v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1)$ (ET-IH)

   From Definition 91 and Definition 92 we know that
   $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')|\!)} = (\!|\mathsf{fix}x.t, \rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!)$ (ET-1)

   Since $\underline{(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')}$ is well typed therefore we know that
   $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')|\!)} = \overline{(\!|\mathsf{fix}x.t, \rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!)} =$
   $\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{t2}\ c)\ d$
   where
   $e_{t1} = \overline{(\!|\mathsf{fix}x.t, \rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)}$
   $e_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$ (ET-1.1)

   Since we know that $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j''}\ v_{\theta 1}$ therefore we also know that
   $\exists j''_1, v'_1.e_{t1} () \Downarrow - \Downarrow^{j''_1}\ v_{\theta 11}$

   Also since we know that
   $(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')$ and $(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')$ are well-typed
   therefore from Lemma 108 we also know that
   $(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta'')$ and $(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta'')$ are well-typed

   Therefore from (ET-IH) we have
   $\exists v_{\theta 22}, j'''_1.\ \overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta'')|\!)} \Downarrow - \Downarrow^{j'''_1}\ v_{\theta 22} \wedge \forall s.v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1)$ (ET-2)

   From Definition 91 we know that
   $\overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')|\!)} = \overline{((\!|t, (\mathsf{fix}x.t, \rho).\rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))}$

   Since $\underline{(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')}$ is well typed therefore we know that
   $\overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta')|\!)} =$
   $\overline{((\!|t, (\mathsf{fix}x.t, \rho).\rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))} =$
   $\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b' = e'_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b'\ (coerce1\ !e'_{t2}\ c)\ d$
   where
   $e'_{t1} = \overline{((\!|t, (\mathsf{fix}x.t, \rho).\rho|\!)\ (\!|\mathsf{C}'|\!) \dots (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_{n-1}|\!))}$
   $e'_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$

   Since from (ET-2) we know that $\overline{(\!|(t, (\mathsf{fix}x.t, \rho).\rho, \mathsf{C}'.\theta'')|\!)} \Downarrow - \Downarrow^{j'''_1}\ v_{\theta 22}$

   Therefore it suffices to prove that
   $v_{\theta 22}\ (coerce1\ !e'_{t2}\ c)\ d \Downarrow - \Downarrow^{j''' - j'''_1}\ v_{\theta 2}$ and $\forall s.v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$ (ET-p)

   Since we are given that $\overline{(\!|(\mathsf{fix}x.t, \rho, \mathsf{C}'.\theta')|\!)}$ this means from (ET-1.1) we have
   $\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in\ } b\ (coerce1\ !e_{t2}\ c)\ d \Downarrow - \Downarrow^{j''}\ v_{\theta 1}$

   This means
   1) $e_{t1}\ () \Downarrow - \Downarrow^{j''_1}\ v_{\theta 11}$ and
   2) This means $v_{\theta 11}\ (coerce1\ !e_{t2}\ c)\ d \Downarrow - \Downarrow^{y}\ v_{\theta 1}$ for some y s.t $y + j''_1 = j''$

Since from (ET-2) we know that $\forall s.v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22}$ and since $e_{t2} = e'_{t2} = \overline{\llparenthesis \mathtt{C}_n \rrparenthesis}$ therefore from Definition 95 and Lemma 101 we have
$v_{\theta 22}$ $(coerce1\ !e'_{t2}\ c)\ d \Downarrow - \Downarrow^{j''-j''_1} v_{\theta 2}$ and $\forall s.v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$
This means
$j'' - j''_1 = j''' - j'''_1 =$
$j'' - j''' = j''_1 - j'''_1 =$
$j'' - j''' = j - j'$ (From IH)

$\square$

Lemma 108. $\forall \mathtt{C}, \theta.$
$\theta.\mathtt{C}$ is well-typed $\implies$ θ is well-typed

Proof. Proof by induction on θ

1. Base case $\theta = \epsilon$:
   Directly from the typing rule for ϵ
2. Case $\theta = \mathtt{C}'.\theta'$
   This means we have $\mathtt{C}'.\theta'.\mathtt{C}$ is well-typed. This means from the stack typing rule for closure we know that $\theta'.\mathtt{C}$ is well-typed.
   From IH we know that θ' is well-typed.
   Since $\mathtt{C}'$ is well tped and θ' is well-typed therefore $\mathtt{C}'.\theta'$ is well-typed.

$\square$

Lemma 109 (Lemma for fix : empty stack)**.** $\forall t, \rho, \theta.$
$\llparenthesis(\mathsf{fix} x.t, \rho, \epsilon)\rrparenthesis$ *is well-typed* $\wedge$
$\llparenthesis(t, (\mathsf{fix} x.t, \rho).\rho, \epsilon)\rrparenthesis$ *is well-typed* $\wedge$
$\overline{\llparenthesis(\mathsf{fix} x.t, \rho, \epsilon)\rrparenthesis}\ () \Downarrow - \Downarrow^j v_1 \implies$
$\overline{\llparenthesis(t, (\mathsf{fix} x.t, \rho).\rho, \epsilon)\rrparenthesis}\, () \Downarrow - \Downarrow^j v_2 \wedge \forall s.v_1 \overset{s}{\approx}_{aV} v_2$

Proof. Let $\rho = (\mathtt{C}_1, \ldots, \mathtt{C}_n)$
Since we know that $\llparenthesis(\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis$ is well-typed and
$\llparenthesis(\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis = ((\lambda x_1 \ldots x_n.\mathsf{fix} x.t)\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)$
Therefore from Theorem 74 we know that
$\overline{\llparenthesis(\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis} =$
$\overline{((\lambda x_1 \ldots x_n.\mathsf{fix} x.t)\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\ a = \mathsf{store}() \mathsf{\ in\ bind}\ b = e_{t1}\ a \mathsf{\ in\ bind}\ c = \mathsf{store!}() \mathsf{\ in\ bind}\ d = \mathsf{store}() \mathsf{\ in}\ b\ (coerce1\ !e_{t2}\ c)\ d$
where
$e_{t1} = \overline{((\lambda x_1 \ldots x_n.\mathsf{fix} x.t)\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_{n-1} \rrparenthesis)}$
$e_{t2} = \overline{\llparenthesis \mathtt{C}_n \rrparenthesis}$ (F1)

Since we know that
$\overline{((\lambda x_1 \ldots x_n.\mathsf{fix} x.t)\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)}\ () \Downarrow - \Downarrow^j v_1$
Therefore from E-release, E-store, E-bind, E-subExpE and E-app we know that
$\bar{t}[\mathsf{fix} x.\bar{t}[\overline{\llparenthesis \mathtt{C}_1 \rrparenthesis}()/x_1] \ldots [\overline{\llparenthesis \mathtt{C}_n \rrparenthesis}()/x_n]\ ()/x] \Downarrow - \Downarrow^j v_1$ (F2)

Similarly since we know that $\llparenthesis(t, (\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)).(\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis$ is well-typed and
$\llparenthesis(t, (\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)).(\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis = ((\lambda x, x_1 \ldots x_n.t)\ \llparenthesis \mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)\rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)$
Therefore from Theorem 74 we know that
$\overline{\llparenthesis(t, (\mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)).(\mathtt{C}_1, \ldots, \mathtt{C}_n), \epsilon)\rrparenthesis} =$
$\overline{((\lambda x, x_1 \ldots x_n.t)\ \llparenthesis \mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)\rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)} =$
$\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\ a = \mathsf{store}() \mathsf{\ in\ bind}\ b = e'_{t1}\ a \mathsf{\ in\ bind}\ c = \mathsf{store!}() \mathsf{\ in\ bind}\ d = \mathsf{store}() \mathsf{\ in}\ b\ (coerce1\ !e'_{t2}\ c)\ d$
where
$e'_{t1} = \overline{((\lambda x, x_1 \ldots x_n.t)\ \llparenthesis \mathsf{fix} x.t, (\mathtt{C}_1, \ldots, \mathtt{C}_n)\rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_{n-1} \rrparenthesis)}$
$e'_{t2} = \overline{\llparenthesis \mathtt{C}_n \rrparenthesis}$ (F3)

We need to prove that
$\overline{((\lambda x, x_1 \ldots x_n.t)\ \llparenthesis \mathsf{fix} x.t, \rho\rrparenthesis\ \llparenthesis \mathtt{C}_1 \rrparenthesis \ldots\ \llparenthesis \mathtt{C}_n \rrparenthesis)} \Downarrow - \Downarrow^j v_2$

This means it suffices to prove that
$\bar{t}[\mathsf{fix} x.\bar{t}[\overline{\llparenthesis \mathtt{C}_1 \rrparenthesis}()/x_1] \ldots [\overline{\llparenthesis \mathtt{C}_n \rrparenthesis}()/x_n]\ ()/x] \Downarrow - \Downarrow^j v_2$
We get this directly from (F2) and Lemma 100 $\square$

Lemma 110 (Lemma for var : non-empty stack)**.** $\forall t, \rho, \theta, j, j', j'', v_{\epsilon 1}, v_{\epsilon 2}, v_{\theta 1}.$
$(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \epsilon)$ *and* (t_x, ρ_x, ϵ) *are well-typed*
$(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \theta)$ *and* $(t, (\mathsf{fix} x.t, \rho).\rho, \theta)$ *are well-typed*
$\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \epsilon)|\!)}\ () \Downarrow - \Downarrow^{j} v_{\epsilon 1} \wedge \overline{(\!|(t_x, \rho_x, \epsilon)|\!)}\ () \Downarrow - \Downarrow^{j'} v_{\epsilon 1} \wedge$
$\forall s. v_{\epsilon 1} \overset{s}{\approx}_{aV} v_{\epsilon 2} \wedge$
$\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \theta)|\!)}\ () \Downarrow - \Downarrow^{j''} v_{\theta 1} \wedge$
$\Longrightarrow$
$\exists v_{\theta 2}, j'''. \ \overline{(\!|(t_x, \rho_x, \theta)|\!)}\ () \Downarrow - \Downarrow^{j'''} v_{\theta 2} \wedge \forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2} \wedge (j - j') = (j'' - j''')$

Proof. We prove this by induction on θ

1. Case $\theta = \epsilon$:
   Directly from given
2. Case $\theta = \mathsf{C}'.\theta'$:
   Let $\theta' = \mathsf{C}'_1 \dots \mathsf{C}'_n$ and $\theta'' = \mathsf{C}'_1 \dots \mathsf{C}'_{n-1}$

   Given:
   $(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')$ and $(t_x, \rho_x, \mathsf{C}'.\theta')$ are well-typed $\wedge$
   $\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j''} v_{\theta 1}$
   We need to prove that
   $\overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j'''} v_{\theta 2} \wedge \forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2} \wedge (j - j') = (j'' - j''')$ (ET-0)

   From IH we know
   $(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta'')$ and $(t_x, \rho_x, \mathsf{C}'.\theta'')$ are well-typed,
   $\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta'')|\!)}\ () \Downarrow - \Downarrow^{j''_1} v_{\theta 11} \implies$
   $\overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta'')|\!)}\ () \Downarrow - \Downarrow^{j'''_1} v_{\theta 22} \wedge \forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1)$ (ET-IH)

   From Definition 91 and Definition 92 we know that
   $(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')|\!) = (\!|x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n)|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!)$ (ET-1)

   Since $(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')$ is well typed therefore we know that
   $\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')|\!)} = \overline{(\!|x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n)|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!)} =$
   $\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b = e_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in}\ b\ (coerce1\ !e_{t2}\ c)\ d$
   where
   $e_{t1} = \overline{(\!|x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n)|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)}$
   $e_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$ (ET-1.1)

   Since we know that $\overline{(\!|(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')|\!)}\ () \Downarrow - \Downarrow^{j''} v_{\theta 1}$ therefore we also know that
   $\exists j''_1, v'_1. e_{t1}() \Downarrow - \Downarrow^{j''_1} v'_1$

   Also since we know that
   $(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')$ and $(t, (\mathsf{fix} x.t, \rho).\rho, \mathsf{C}'.\theta')$ are well-typed
   therefore from Lemma 108 we also know that
   $(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta'')$ and $(t_x, \rho_x, \mathsf{C}'.\theta'')$ are well-typed

   Therefore from (ET-IH) we have
   $\exists v_{\theta 22}, j'''_1. \ \overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta'')|\!)} \Downarrow - \Downarrow^{j'''_1} v_{\theta 22} \wedge \forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22} \wedge (j - j') = (j''_1 - j'''_1)$ (ET-2)

   From Definition 91 we know that
   $\overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta')|\!)} = \overline{((\!|t_x, \rho_x|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))}$

   Since $(t_x, \rho_x, \mathsf{C}'.\theta')$ is well typed therefore we know that
   $\overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta')|\!)} =$
   $\overline{((\!|t_x, \rho_x|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_n|\!))} =$
   $\lambda p.\mathsf{release} - = p \mathsf{\ in\ bind}\, a = \mathsf{store}() \mathsf{\ in\ bind}\, b' = e'_{t1}\ a \mathsf{\ in\ bind}\, c = \mathsf{store!}() \mathsf{\ in\ bind}\, d = \mathsf{store}() \mathsf{\ in}\ b'\ (coerce1\ !e'_{t2}\ c)\ d$
   where
   $e'_{t1} = \overline{((\!|t_x, \rho_x|\!)\ (\!|\mathsf{C}'|\!) \dots\ (\!|\mathsf{C}_{n-1}|\!)\ (\!|\mathsf{C}_{n-1}|\!))}$
   $e'_{t2} = \overline{(\!|\mathsf{C}_n|\!)}$

   Since from (ET-2) we know that $\overline{(\!|(t_x, \rho_x, \mathsf{C}'.\theta'')|\!)} \Downarrow - \Downarrow^{j'''_1} v_{\theta 22}$

   Therefore it suffices to prove that

$v_{\theta 22}\ (coerce1\ !e'_{t2}\ c)\ d \Downarrow - \Downarrow^{j'''-j_1'''} v_{\theta 2}$ and $\forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$ (ET-p)

Since we are given that $\llparenthesis(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta')\rrparenthesis \Downarrow - \Downarrow^{j''} v_{\theta 1}$ this means from (ET-1.1) we have
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2}\ c)\ d \Downarrow - \Downarrow^{j''} v_{\theta 1}$

This means
1) $\overline{\llparenthesis(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \mathsf{C}'.\theta'')\rrparenthesis}\ () \Downarrow - \Downarrow^{j_1''} v_{\theta 11}$ and
2) This means $v_{\theta 11}\ (coerce1\ !e_{t2}\ c)\ d \Downarrow - \Downarrow^{y} v_{\theta 1}$ for some y s.t $y + j_1'' = j''$

Since from (ET-2) we have $\forall s. v_{\theta 11} \overset{s}{\approx}_{aV} v_{\theta 22} \wedge$ and since $e_{t2} = e'_{t2} = \overline{\llparenthesis \mathsf{C}_n \rrparenthesis}$ therefore from Definition 95 and Lemma 101 we have
$v_{\theta 22}\ (coerce1\ !e'_{t2}\ c)\ d \Downarrow - \Downarrow^{j''-j_1''} v_{\theta 2}$ and $\forall s. v_{\theta 1} \overset{s}{\approx}_{aV} v_{\theta 2}$
This means
$j'' - j_1'' = j''' - j_1''' =$
$j'' - j''' = j_1'' - j_1''' =$
$j'' - j''' = j - j'$ (From IH)

□

Lemma 111 (Lemma for var : empty stack)**.** $\forall t, \rho, \theta.$
$\Theta; \Delta; . \vdash_{-} \llparenthesis(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \epsilon)\rrparenthesis : - \wedge$
$\Theta; \Delta; . \vdash_{-} \llparenthesis(t_x, \rho_x, \epsilon)\rrparenthesis : - \wedge$
$\overline{\llparenthesis(x, (t_0, \rho_0) \dots (t_x, \rho_x) \dots (t_n, \rho_n), \epsilon)\rrparenthesis}\ () \Downarrow - \Downarrow^{j} v \implies$
$\overline{\llparenthesis(t_x, \rho_x, \epsilon)\rrparenthesis}\ () \Downarrow - \Downarrow^{j-1} v$

Proof. From Definition 92 we also have
$\llparenthesis(x, (t_0, \rho_0), \dots (t_x, \rho_x), \dots (t_n, \rho_n), \epsilon)\rrparenthesis$
$= \llparenthesis x, (t_0, \rho_0), \dots (t_x, \rho_x), \dots (t_n, \rho_n)\rrparenthesis$
$= (\lambda x_1 \dots x \dots x_n.x)\ \llparenthesis(t_0, \rho_0)\rrparenthesis \dots\ \llparenthesis(t_n, \rho_n)\rrparenthesis$

Similarly from Definition 92 we also have
$\llparenthesis(t_x, \rho_x, \epsilon)\rrparenthesis = \llparenthesis(t_x, \rho_x)\rrparenthesis$ (S-V1)

Therefore from Theorem 74 we know that
$\overline{\llparenthesis(x, ((t_1, \rho_1), \dots (t_x, \rho_x) \dots, (t_n, \rho_n)), \epsilon)\rrparenthesis} =$
$\overline{((\lambda x_1 \dots x \dots x_n.x)\ \llparenthesis(t_1, \rho_1)\rrparenthesis \dots \llparenthesis(t_x, \rho_x)\rrparenthesis \dots\ \llparenthesis(t_n, \rho_n)\rrparenthesis)} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1,n}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2,n}\ c)\ d$
where
$e_{t1,n} = \overline{((\lambda x_1 \dots x \dots x_n.x)\ \llparenthesis(t_1, \rho_1)\rrparenthesis \dots \llparenthesis(t_x, \rho_x)\rrparenthesis \dots\ \llparenthesis(t_{n-1}, \rho_{n-1})\rrparenthesis)}$
$e_{t2,n} = \overline{\llparenthesis(t_n, \rho_n)\rrparenthesis}$ (V4)

Similalrly
$e_{t1,n} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1,n-1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2,n-1}\ c)\ d$
where
$e_{t1,n-1} = \overline{((\lambda x_1 \dots x \dots x_n.x)\ \llparenthesis(t_1, \rho_1)\rrparenthesis \dots \llparenthesis(t_x, \rho_x)\rrparenthesis \dots\ \llparenthesis(t_{n-2}, \rho_{n-2})\rrparenthesis)}$
$e_{t2,n-1} = \overline{\llparenthesis(t_{n-1}, \rho_{n-1})\rrparenthesis}$

In the same way we have
$e_{t1,1} =$
$\lambda p.\mathsf{release} - = p \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } \mathsf{bind}\, b = e_{t1,1}\ a \text{ in } \mathsf{bind}\, c = \mathsf{store!}() \text{ in } \mathsf{bind}\, d = \mathsf{store}() \text{ in } b\ (coerce1\ !e_{t2,1}\ c)\ d$
where
$e_{t1,1} = \overline{(\lambda x_1 \dots x \dots x_n.x)}$
$e_{t2,1} = \overline{\llparenthesis(t_1, \rho_1)\rrparenthesis}$

Similalry we also get
$e_{t1,1} =$
$\lambda p_1.\ \mathsf{ret}\ \lambda y.\lambda p_2.\ \mathsf{let}\,!\, x = y \text{ in } \mathsf{release} - = p_1 \text{ in } \mathsf{release} - = p_2 \text{ in } \mathsf{bind}\, a = \mathsf{store}() \text{ in } e_{l,1}\ a$
where
$e_{l,1} = \overline{((\lambda x_2 \dots x \dots x_n.x))}$

and
$e_{l,n} =$
$\lambda p_1.\,\mathsf{ret}\,\lambda y.\lambda p_2.\,\mathsf{let}\,!\,x = y\ \mathsf{in}\ \mathsf{release} - = p_1\ \mathsf{in}\ \mathsf{release} - = p_2\ \mathsf{in}\ \mathsf{bind}\,a = \mathsf{store}()\ \mathsf{in}\ e_{l,n}\ a$
where
$e_{l,n} = \overline{x} = \lambda p.\mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind} - = \uparrow^1\ \mathsf{in}\ x$

Since we know that
$\overline{((\lambda x_1 \ldots x \ldots x_n.x)\ (\!|(t_0,\rho_0)|\!) \ldots\ (\!|(t_n,\rho_n)|\!))}\ ()\Downarrow - \Downarrow^j v$
this means from E-release, E-bind, E-store, E-app that
$(\mathsf{bind} - = \uparrow^1\ \mathsf{in}\ \overline{(\!|(t_x,\rho_x)|\!)}\ ())\Downarrow - \Downarrow^j v$
Therefore from E-bind, E-step and E-app we know that $\overline{(\!|(t_x,\rho_x)|\!)}\ ()\Downarrow - \Downarrow^{j-1} v$ □

Theorem 112 (Rederiving dlPCF's soundness)**.** $\forall t, I, \tau, \rho.$
$\vdash_I (t,\epsilon,\epsilon) : \tau \wedge (t,\epsilon,\epsilon) \xrightarrow{n} (v,\rho,\epsilon) \implies n \leq |t| * (I+1)$

Proof. Let us rename t to t_1 and v to t_{n+1} then we know that
$(t_1,\epsilon,\epsilon) \rightarrow (t_2,\rho_2,\theta_2) \ldots (t_n,\rho_n,\theta_n) \rightarrow (t_{n+1},\rho,\epsilon)$

Since we are given that (t,ϵ,ϵ) is well-typed therefore from dlPCF's subject reduction we know that (t_2,ρ_2,θ_2) to (t_n,ρ_n,θ_n) and (t_{n+1},ρ,ϵ) are all well-typed.

From Theorem 115 we know that $\forall 1 \leq i \leq n.(\!|(t_i,\rho_i,\theta_i)|\!) \xrightarrow{*} -$
Also from Theorem 94 we know that $\forall 1 \leq i \leq n.(\!|(t_i,\rho_i,\theta_i)|\!)$ is well typed

So now we can apply Theorem 88 and from Definition 86 to get
$\forall 1 \leq i \leq n+1.\exists j_i.\overline{(\!|(t_i,\rho_i,\theta_i)|\!)}\ ()\Downarrow - \Downarrow^{j_i} -$

Next we apply Theorem 104 for every step of the reduction starting from (t_1,ϵ,ϵ) and we know that either the cost reduces by 1 and the size increases by $|t|$ or cost remains the same and the size reduces.

Thus we know that size can vary from t to 1 and cost can vary from j_1 to 0. Therefore, the number of reduction steps are bounded by $|t| * (j_1+1)$
From Theorem 72 we know that $j_1 < I$ therefore we have $n \leq |t| * (I+1)$ □

B.5.4 Cross-language model: Krivine to dlPCF

Definition 113 (Cross language logical realtion: Krivine to dlPCF)**.**

$$(v_k,\rho,\epsilon) \sim_v v_d \triangleq v_d = v_k\ \rho$$

$$(e_k,\rho,\theta) \sim_e e_d \triangleq \forall v_k,\rho'.(e_k,\rho,\theta) \xrightarrow{*} (v_k,\rho',\epsilon) \implies \exists v_d.e_d \xrightarrow{*} v_d \wedge (v_k,\rho',\epsilon) \sim_v v_d$$

Lemma 114. $\forall e_k, \rho, \theta, e'_k, \rho', \theta'.$
$(e_k,\rho,\theta) \xrightarrow{*} (e'_k,\rho',\theta') \implies \exists e'_d.(\!|(e_k,\rho,\theta)|\!) \xrightarrow{*} e'_d \wedge e'_d = (\!|(e'_k,\rho',\theta')|\!)$

Proof. Given: $(e_k,\rho,\theta) \xrightarrow{*} (e'_k,\rho',\theta')$
To prove: $\exists e'_d.(\!|(e_k,\rho,\theta)|\!) \xrightarrow{*} e'_d \wedge e'_d = (\!|(e'_k,\rho',\theta')|\!)$

Lets assume it takes n steps for $(e_k,\rho,\theta) \xrightarrow{n} (e'_k,\rho',\theta')$

We induct on n
Base case $(n = 1)$

1. App1:
   In this case we are given $(t\ u,\rho,\theta) \rightarrow (t,\rho,(u,\rho).\theta)$
   Let $\rho = \mathtt{C}_{\rho_1} \ldots \mathtt{C}_{\rho_n}$ and $\theta = \mathtt{C}_{\theta_1} \ldots \mathtt{C}_{\theta_m}$

   From Definition 92 we know that
   $(\!|(e_k,\rho,\theta)|\!) =$
   $(\lambda x_1 \ldots x_n.t\ u)(\!|\mathtt{C}_{\rho_1}|\!) \ldots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)$

   From dlPCF's app rule we know that
   $(\lambda x_1 \ldots x_n.t\ u)\mathtt{C}_{\rho_1} \ldots \mathtt{C}_{\rho_n}\ \mathtt{C}_{\theta_1} \ldots \mathtt{C}_{\theta_m} \xrightarrow{*}$
   $t[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \ldots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n]\ u[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \ldots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n]\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)$

   We choose e'_d as $t[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \ldots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n]\ u[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \ldots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n]\ (\!|\mathtt{C}_{\theta_1}|\!) \ldots (\!|\mathtt{C}_{\theta_m}|\!)$ and we get thedesired from Definition 92

2. App2:
   In this case we are given $(\lambda x.t, \rho, \mathtt{C}.\theta) \to (t, \mathtt{C}.\rho, \theta)$
   Let $\rho = \mathtt{C}_{\rho_1} \dots \mathtt{C}_{\rho_n}$ and $\theta = \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   From Definition 92 we know that
   $(\!|(\lambda x.t, \rho, \mathtt{C}.\theta)|\!) =$
   $(\lambda x_1 \dots x_n.\lambda x.t)(\!|\mathtt{C}_{\rho_1}|\!) \dots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!)$

   From dlPCF's app rule we know that
   $(\lambda x_1 \dots x_n.\lambda x.t)(\!|\mathtt{C}_{\rho_1}|\!) \dots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!) \xrightarrow{*}$
   $t[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n][(\!|\mathtt{C}|\!)/x]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   We choose e'_d as $t[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n][(\!|\mathtt{C}|\!)/x]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$ and we get the desired from Definition 92
3. Var:
   In this case we are given $(x, (t_0, \rho_0) \dots (t_n, \rho_n), \theta) \to (t_x, \rho_x, \theta)$
   Let $\theta = \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   From Definition 92 we know that
   $(\!|(x, (t_0, \rho_0) \dots (t_n, \rho_n), \theta)|\!) =$
   $(\lambda x_1 \dots x_n.\lambda x.t)(\!|\mathtt{C}_{\rho_1}|\!) \dots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!)$

   From dlPCF's app rule we know that
   $(\lambda x_1 \dots x_n.x)(\!|(t_0, \rho_0)|\!) \dots (\!|(t_n, \rho_n)|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!) \xrightarrow{*}$
   $(\!|(t_x, \rho_x)|\!)\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   Let $\rho_x = \mathtt{C}_{x_1} \dots \mathtt{C}_{x_k}$ therefore from Definition 92 we know that
   $(\!|(t_x, \rho_x)|\!)\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m} =$
   $\lambda x_{x_1} \dots x_{x_k}.t_x\ (\!|\mathtt{C}_{x_1}|\!) \dots (\!|\mathtt{C}_{x_k}|\!)\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   Therefore from dlPCF's app rule we know that
   $(\!|(t_x, \rho_x)|\!)\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m} \xrightarrow{*} t_x[(\!|\mathtt{C}_{x_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{x_k}|\!)/x_k]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   We choose e'_d as $t_x[(\!|\mathtt{C}_{x_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{x_k}|\!)/x_k]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$ and we get the desired from Definition 92
4. Fix:
   In this case we are given $(\mathsf{fix}x.t, \rho, \theta) \to (t, (\mathsf{fix}x.t, \rho).\rho, \theta)$
   Let $\rho = \mathtt{C}_{\rho_1} \dots \mathtt{C}_{\rho_n}$ and $\theta = \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   From Definition 92 we know that
   $(\!|(\mathsf{fix}x.t, \rho, \theta)|\!) =$
   $(\lambda x_1 \dots x_n.\mathsf{fix}x.t)(\!|\mathtt{C}_{\rho_1}|\!) \dots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|(\mathsf{fix}x.t, \rho)|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!)$

   From dlPCF's app and fix rule we know that
   $(\lambda x_1 \dots x_n.\mathsf{fix}x.t)(\!|\mathtt{C}_{\rho_1}|\!) \dots (\!|\mathtt{C}_{\rho_n}|\!)\ (\!|\mathtt{C}|\!)\ (\!|\mathtt{C}_{\theta_1}|\!) \dots (\!|\mathtt{C}_{\theta_m}|\!) \xrightarrow{*}$
   $\mathsf{fix}x.t[(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n][(\!|(\mathsf{fix}x.t, \rho)|\!)/x]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m} \to$
   $t\ [(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n][(\!|(\mathsf{fix}x.t, \rho)|\!)/x]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$

   We choose e'_d as $t\ [(\!|\mathtt{C}_{\rho_1}|\!)/x_1] \dots [(\!|\mathtt{C}_{\rho_n}|\!)/x_n][(\!|(\mathsf{fix}x.t, \rho)|\!)/x]\ \mathtt{C}_{\theta_1} \dots \mathtt{C}_{\theta_m}$ and we get the desired from Definition 92

   Inductive case
   We get this directly from IH and the base case

□

Theorem 115 (Fundamental theorem)**.** $\forall e_k, \rho, \theta.\ (e_k, \rho, \theta) \sim_e (\!|(e_k, \rho, \theta)|\!)$

Proof. From Definition 113 it suffices to prove that
$\forall v_k, \rho'.(e_k, \rho, \theta) \xrightarrow{*} (v_k, \rho', \epsilon) \implies \exists v_d.e_d \xrightarrow{*} v_d \land (v_k, \rho', \epsilon) \sim_v v_d$

This means athat given some v_k, ρ' s.t $(e_k, \rho, \theta) \xrightarrow{*} (v_k, \rho', \epsilon)$ it suffices to prove that
$\exists v_d.e_d \xrightarrow{*} v_d \land (v_k, \rho', \epsilon) \sim_v v_d$

From Lemma 114 we know that
$\exists e'_d.(\!|(e_k, \rho, \theta)|\!) \xrightarrow{*} e'_d \land e'_d = (\!|(v_k, \rho', \epsilon)|\!)$

Let $\rho' = \mathtt{C}_1 \dots \mathtt{C}_n$ therefore from Definition 92 we know that
$(\!|(v_k, \rho', \epsilon)|\!) = (\lambda x_1 \dots x_n v_k)\ (\!|\mathtt{C}_1|\!) \dots (\!|\mathtt{C}_n|\!)$

Therefore from dlPCF's app rule we know that
$(\!|(v_k, \rho', \epsilon)|\!) \xrightarrow{*} v_k[(\!|\mathtt{C}_1|\!)/x_1] \dots [(\!|\mathtt{C}_n|\!)/x_n]$

We choose v_d as $v_k[(\!|\mathtt{C}_1|\!)/x_1] \dots [(\!|\mathtt{C}_n|\!)/x_n]$ and we get the desired from Definition 113

□

C Examples

C.1 Church numerals

$\mathsf{Nat} = \lambda_t n.\forall \alpha : \mathbb{N} \to \mathit{Type}.\forall C : \mathbb{N} \to \mathbb{N}.$
$!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap \mathbb{M}\,0\,((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n-1) + n)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ n))$

$e_1 \uparrow^1 e_2 \triangleq \mathsf{bind} - = \uparrow^1 \mathsf{in}\ e_1\ e_2$

$$\frac{\dfrac{}{\Psi; \Theta; \Delta; \Omega_1; \Gamma_1 \vdash e_1 : \tau_1 \multimap \mathbb{M}(n)\,\tau_2} \qquad \dfrac{}{\Psi; \Theta; \Delta; \Omega_2; \Gamma_2 \vdash e_2 : \tau_1}}{\Psi; \Theta; \Delta; \Omega_1 \oplus \Omega_2; \Gamma_1 \oplus \Gamma_2 \vdash e_1 \uparrow^1 e_2 : \mathbb{M}(n+1)\,\tau_2}$$

<u>Type derivation for $\overline{0}$</u>

$\overline{0} = \Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : \mathsf{Nat}\ 0$

$T_0 =$
$\forall \alpha.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap \mathbb{M}\,0\,((\alpha\ 0 \otimes [0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 0))$
$T_{0.1} = \forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap \mathbb{M}\,0\,((\alpha\ 0 \otimes [0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 0))$
$T_{0.2} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap \mathbb{M}\,0\,((\alpha\ 0 \otimes [0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 0))$
$T_{0.3} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1))))$
$T_1 = \mathbb{M}\,0\,((\alpha\ 0 \otimes [0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 0))$
$T_{1.1} = ((\alpha\ 0 \otimes [0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 0))$
$T_2 = (\alpha\ 0 \otimes [0]\,\mathbf{1})$
$T_{2.1} = \alpha\ 0$
$T_{2.2} = [0]\,\mathbf{1}$
$T_3 = \mathbb{M}\,0\,(\alpha\ 0)$
$TI = \alpha : \mathbb{N} \to \mathit{Type}; C : \mathbb{N} \to \mathit{Sort}$
D1:

$$\frac{}{TI; .; .; f : T_{0.3}, y_1 : T_{2.1}, y_2 : T_{2.2} \vdash \mathsf{ret}\,y_1 : \mathbb{M}\,0\,T_{2.1}}$$

D0:

$$\frac{}{TI; .; .; f : T_{0.3}, x : T_2 \vdash x : T_2}$$

Main derivation:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{D0 \qquad D1}{TI; .; .; f : T_{0.3}, x : T_2 \vdash \mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : T_3}}{TI; .; .; f : T_{0.3} \vdash \lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : T_{1.1}}}{TI; .; .; f : T_{0.3} \vdash \mathsf{ret}\,\lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : T_1}}{TI; .; .; . \vdash \lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : T_{0.2}}}{\cfrac{\alpha : \mathbb{N} \to \mathit{Type}; .; .; ; .; . \vdash \Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1 : T_{0.1}}{.; .; .; .; . \vdash \Lambda.\Lambda.(\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{ret}\,y_1) : T_0}}$$

<u>Type derivation for $\overline{1}$</u>

$\overline{1} = \Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : \mathsf{Nat}\ 1$
where
$E_1 = \mathsf{bind}\,a = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\!\langle y_1, a\rangle\!\rangle$

$T_0 = \forall \alpha : \mathbb{N} \to \mathit{Type}.\forall C : \mathbb{N} \to \mathit{Sort}.$
$!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1))$
$T_{0.1} = \forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1))$
$T_{0.2} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1))$
$T_{0.3} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1))))$
$T_{0.4} = (\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1))))$
$T_{0.5} = (\alpha\ 0 \otimes [C\ 0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (0+1))$
$T_1 = \mathbb{M}\,0\,((\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1))$
$T_{1.1} = ((\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1))$
$T_2 = (\alpha\ 0 \otimes [C\ 0+1]\,\mathbf{1})$

$T_{2.1} = \alpha\ 0$
$T_{2.2} = [C\ 0+1]\,\mathbf{1}$
$T_3 = \mathbb{M}\,0\,(\alpha\ 1)$
$TI = \alpha : \mathbb{N} \to Type; C : \mathbb{N} \to Sort$
D7:

$$\frac{}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, a : [C\ 0]\,\mathbf{1} \vdash \langle\langle y_1, a\rangle\rangle : (T_{2.1} \otimes [C\ 0]\,\mathbf{1})}$$

D6:

$$\frac{}{TI; .; f_u : T_{0.4}; . \vdash f_u\ []\ : T_{0.5}}$$

D5:

$$\frac{D6 \qquad D7}{TI; .; f_u : T_{0.4}; y_1 : T_{2.2}, a : [C\ 0]\,\mathbf{1} \vdash f_u\ []\ \uparrow^1 \langle\langle y_1, a\rangle\rangle : \mathbb{M}\,1\,\alpha\ 1}$$

D4:

$$\frac{\dfrac{D4}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, y_2 : T_{2.2} \vdash \mathsf{store}() : \mathbb{M}(C\ 0)\,[C\ 0]\,\mathbf{1}} \qquad D5}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, y_2 : T_{2.2} \vdash \mathsf{bind}\ a = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\langle y_1, a\rangle\rangle : \mathbb{M}(C\ 0+1)\,\alpha\ 1}$$

D3:

$$\frac{D4}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1} \vdash E_1 : \mathbb{M}(C\ 0+1)\,\alpha\ 1}$$

D2:

$$\frac{\dfrac{}{TI; .; f_u : T_{0.4}; y_2 : T_{2.2} \vdash y_2 : T_{2.2}} \qquad D3}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, y_2 : T_{2.2} \vdash \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_3}$$

D1:

$$\frac{\dfrac{}{TI; .; f_u : T_{0.4}; x : T_2 \vdash x : T_2} \qquad D2}{TI; .; f_u : T_{0.4}; x : T_2 \vdash \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_3}$$

D0:

$$\frac{}{TI; .; .; f : T_{0.3} \vdash f : T_{0.3}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D0 \qquad D1}{TI; .; .; f : T_{0.3}, x : T_2 \vdash \mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_3}}{TI; .; .; f : T_{0.3} \vdash \lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_{1.1}}}{TI; .; .; f : T_{0.3} \vdash \mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_1}}{TI; .; .; . \vdash \lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_{0.2}}}{.; \alpha : \mathbb{N} \to Type; .; . \vdash \Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_{0.1}}}{.; .; .; . \vdash \Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_1 : T_0}$$

Type derivation for $\overline{2}$

$\overline{2} = \Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ \mathsf{bind}\ b = E_1\ \mathsf{in}\ E_2 : \mathsf{Nat}\ 2$
where
$E_1 = \mathsf{bind}\ a = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\langle y_1, a\rangle\rangle$
$E_2 = \mathsf{bind}\ c = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\langle b, c\rangle\rangle$

$T_0 =$
$\forall\alpha : \mathbb{N} \to Type.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2))$
$T_{0.1} = \forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2))$
$T_{0.2} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap ((\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2))$
$T_{0.3} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1))))$
$T_{0.4} = (\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1))))$
$T_{0.5} = (\alpha\ 0 \otimes [C\ 0]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 1)$
$T_{0.6} = (\alpha\ 1 \otimes [C\ 1]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2)$
$T_1 = \mathbb{M}\,0\,((\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2))$
$T_{1.1} = ((\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ 2))$

$T_2 = (\alpha\ 0 \otimes [C\ 0 + C\ 1 + 2]\ \mathbf{1})$
$T_{2.1} = \alpha\ 0$
$T_{2.2} = [C\ 0 + C\ 1 + 2]\ \mathbf{1}$
$T_3 = \mathbb{M}\, 1\, (\alpha\ 2)$
$T_{3.1} = \mathbb{M}(C\ 0 + C\ 1 + 2)\, (\alpha\ 2)$
$TI = \alpha : \mathbb{N} \to Type; C : \mathbb{N} \to Sort$
D5.22

$$\frac{}{TI; .; f_u : T_{0.4}; b : \alpha\ 1, c : [(C\ 1)]\ \mathbf{1} \vdash \langle\langle b, c\rangle\rangle : (\alpha\ 1 \otimes [(C\ 1)]\ \mathbf{1})}$$

D5.21

$$\frac{}{TI; .; f_u : T_{0.4}; . \vdash f_u\ [] : T_{0.6}}$$

D5.2

$$\frac{D5.21 \qquad D5.22}{TI; .; f_u : T_{0.4}; b : \alpha\ 1, c : [(C\ 1)]\ \mathbf{1} \vdash f_u\ []\ \uparrow^1\ \langle\langle b, c\rangle\rangle : T_3}$$

D5.1

$$\frac{}{TI; .; f_u : T_{0.4}; . \vdash \mathsf{store}() : \mathbb{M}(C\ 1)\ [(C\ 1)]\ \mathbf{1}}$$

D5:

$$\frac{\dfrac{D5.1 \qquad D5.2}{TI; .; f_u : T_{0.4}; b : \alpha\ 1 \vdash \mathsf{bind}\, c = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \langle\langle b, c\rangle\rangle : \mathbb{M}(C\ 1 + 1)\, (\alpha\ 2)}}{TI; .; f_u : T_{0.4}; b : \alpha\ 1 \vdash E_2 : \mathbb{M}(C\ 1 + 1)\, (\alpha\ 2)}$$

D4.12:

$$\frac{}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, a : [(C\ 0)]\ \mathbf{1} \vdash \langle\langle y_1, a\rangle\rangle : (T_{2.1} \otimes [(C\ 0)]\ \mathbf{1})}$$

D4.11:

$$\frac{}{TI; .; f_u : T_{0.4}; . \vdash f_u\ [] : T_{0.5}}$$

D4.1:

$$\frac{D4.11 \qquad D4.12}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, a : [(C\ 0)]\ \mathbf{1} \vdash f_u\ []\ \uparrow^1\ \langle\langle y_1, a\rangle\rangle : \mathbb{M}\, 1\, (\alpha\ 1)}$$

D4:

$$\frac{\dfrac{\dfrac{}{TI; .; f_u : T_{0.4}; . \vdash \mathsf{store}() : \mathbb{M}(C\ 0)\ [(C\ 0)]\ \mathbf{1}} \qquad D4.1}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1} \vdash \mathsf{bind}\, a = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\langle y_1, a\rangle\rangle : \mathbb{M}(C\ 0 + 1)\, (\alpha\ 1)}}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1} \vdash E_1 : \mathbb{M}(C\ 0 + 1)\, (\alpha\ 1)}$$

D3.2:

$$\frac{D4 \qquad D5}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1} \vdash \mathsf{bind}\, b = E_1\ \mathsf{in}\ E_2 : T_{3.1}}$$

D3.1:

$$\frac{}{TI; .; f_u : T_{0.4}; y_2 : T_{2.2} \vdash y_2 : T_{2.2}}$$

D3:

$$\frac{D3.1 \qquad D3.2}{TI; .; f_u : T_{0.4}; y_1 : T_{2.1}, y_2 : T_{2.2} \vdash \mathsf{release} - = y_2\ \mathsf{in}\ \ \mathsf{bind}\, b = E_1\ \mathsf{in}\ E_2 : T_3}$$

D2:

$$\frac{}{TI; .; f_u : T_{0.4}; x : T_2 \vdash x : T_2}$$

D1:

$$\frac{D2 \qquad D3}{TI; .; f_u : T_{0.4}; x : T_2 \vdash \mathsf{let} \langle\langle y_1, y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ \ \mathsf{bind}\, b = E_1\ \mathsf{in}\ E_2 : T_3}$$

D0:

$$\frac{}{TI; .; .; f : T_{0.3} \vdash f : T_{0.3}}$$

D0.0:

$$\dfrac{\dfrac{\dfrac{\dfrac{D0 \qquad D1}{TI; .; .; f : T_{0.3}, x : T_2 \vdash \quad \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_3}}{TI; .; .; f : T_{0.3} \vdash \quad \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_{1.1}}}{TI; .; .; f : T_{0.3} \vdash \quad \mathsf{ret}\, \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_1}}{TI; .; .; . \vdash \quad \lambda f.\, \mathsf{ret}\, \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_{0.2}}$$

Main derivation:

$$\dfrac{\dfrac{D0.0}{\begin{array}{c} .; \alpha : \mathbb{N} \to Type; .; . \vdash \\ \Lambda C.\lambda f.\, \mathsf{ret}\, \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_{0.1}\end{array}}}{.; .; .; . \vdash \Lambda.\Lambda.\lambda f.\, \mathsf{ret}\, \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b = E_1 \;\mathsf{in}\; E_2 : T_0}$$

Type derivation for $succ : \forall n.\, [2]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n+1)))$

$succ = \Lambda.\lambda p.\, \mathsf{ret}\, \lambda \overline{N}.\, \mathsf{ret}\, \Lambda.\Lambda.\lambda f.\, \mathsf{ret}\, \lambda x.\, \mathsf{let}\,!\,f_u = f \;\mathsf{in}\; \mathsf{let}\langle\langle y_1, y_2\rangle\rangle = x \;\mathsf{in}\; \mathsf{release} - = y_2 \;\mathsf{in}\; E_0$
where
$E_0 = \mathsf{release} - = p \;\mathsf{in}\; \mathsf{bind}\, a = E_1 \;\mathsf{in}\; E_2$
$E_1 = \mathsf{bind}\, b = \mathsf{store}() \;\mathsf{in}\; \mathsf{bind}\, b_1 = (\overline{N}\ []\ []\ \uparrow^1 !f_u) \;\mathsf{in}\; b_1\ \uparrow^1 \langle\langle y_1, b\rangle\rangle$
$E_2 = \mathsf{bind}\, c = \mathsf{store}() \;\mathsf{in}\; \mathsf{ret}\, f_u\ []\ \uparrow^1 \langle\langle a, c\rangle\rangle$

$T_p = [2]\, \mathbf{1}$
$T_0 = \forall n.T_p \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n] \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n+1]))$
$T_{0.0} = T_p \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n] \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n+1]))$
$T_{0.01} = \mathbb{M}\, 0\, (\mathsf{Nat}[n] \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n+1]))$
$T_{0.1} = \mathsf{Nat}[n] \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n+1])$
$T_{0.2} = \mathbb{M}\, 0\, (\mathsf{Nat}[n+1])$
$T_{0.11} = \mathsf{Nat}[n]$
$T_{0.12} =$
$\forall \alpha : \mathbb{N} \to Type.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap$
$\mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap \mathbb{M}\, 0\, ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.13} = \forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.14} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.15} = \mathbb{M}\, 0\, ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.151} = \mathbb{M}\, 1\, ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.16} = ((\alpha\ 0 \otimes [C\ 0 + \ldots + C\ (n-1) + n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n))$
$T_{0.2} = \mathsf{Nat}[n+1]$
$T_1 =$
$\forall \alpha : \mathbb{N} \to Type.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (n+1)))$
$T_{1.1} = \forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (n+1)))$
$T_{1.2} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (n+1)))$
$T_{1.3} = !(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1))))$
$T_{1.31} = (\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n+1))))$
$T_{1.40} = \mathbb{M}\, 0\, ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (n+1)))$
$T_{1.4} = ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (n+1)))$
$T_{1.41} = (\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1})$
$T_{1.411} = \alpha\ 0$
$T_{1.412} = [(C\ 0 + \ldots + C\ (n) + (n+1))]\, \mathbf{1}$
$T_{1.42} = \mathbb{M}\, 0\, (\alpha\ (n+1))$
$T_{1.43} = \mathbb{M}(C\ 0 + \ldots + C\ (n) + (n+1))\, (\alpha\ (n+1))$
$T_{1.431} = \mathbb{M}(C\ 0 + \ldots + C\ (n) + (n+1) + 2)\, (\alpha\ (n+1))$
$T_{1.44} = \mathbb{M}(C\ 0 + \ldots + C\ (n-1) + n + 2)\, (\alpha\ n)$
$T_{1.45} = \mathbb{M}(C\ n + 1)\, (\alpha\ (n+1))$
$TI = \alpha; n, C$

D3.1:

$$\frac{}{TI; .; f_u : T_{1.31}; a : \alpha\ n, c : [(C\ n)]\,\mathbf{1} \vdash f_u\ []\ \uparrow^1 \langle\!\langle a, c\rangle\!\rangle : \mathbb{M}\,1\,\alpha\ (n+1)}$$

D3:

$$\frac{\dfrac{}{TI; .; f_u : T_{1.31}; . \vdash \mathsf{store}() : \mathbb{M}(C\ n)\,[(C\ n)]\,\mathbf{1}} \qquad D3.1}{TI; .; f_u : T_{1.31}; a : \alpha\ n \vdash \mathsf{bind}\, c = \mathsf{store}()\ \mathsf{in}\ f_u\ []\ \uparrow^1 \langle\!\langle a, c\rangle\!\rangle : T_{1.45}}$$

D2.3:

$$\frac{\dfrac{\dfrac{}{TI; .; f_u : T_{1.31}; y_1 : T_{1.411}, b : [n * C]\,\mathbf{1}, b_1 : T_{0.16} \vdash b_1 : T_{0.16}}}{\begin{array}{c} TI; .; f_u : T_{1.31}; y_1 : T_{1.411}, b : [(C\ 0 + \ldots + C\ (n-1) + (n))]\,\mathbf{1}, b_1 : T_{0.16} \vdash \\ \langle\!\langle y_1, b\rangle\!\rangle : (T_{1.411} \otimes [(C\ 0 + \ldots + C\ (n-1) + (n))]\,\mathbf{1}) \end{array}}}{TI; .; f_u : T_{1.31}; y_1 : T_{1.411}, b : [(C\ 0 + \ldots + C\ (n-1) + (n))]\,\mathbf{1}, b_1 : T_{0.16} \vdash \quad b_1\ \uparrow^1 \langle\!\langle y_1, b\rangle\!\rangle : \mathbb{M}\,1\,\alpha\ n}$$

D2.2

$$\frac{}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11} \vdash \overline{N}\ []\ []\ \uparrow^1 ! f_u : T_{0.151}}$$

D2.1:

$$\frac{D2.2 \qquad D2.3}{\begin{array}{c} TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, y_1 : T_{1.411}, b : [(C\ 0 + \ldots + C\ (n-1) + (n))]\,\mathbf{1} \vdash \\ \mathsf{bind}\, b_1 = (\overline{N}\ []\ []\ \uparrow^1 ! f_u)\ \mathsf{in}\ b_1\ \uparrow^1 \langle\!\langle y_1, b\rangle\!\rangle : \mathbb{M}\,2\,\alpha\ n \end{array}}$$

D2:

$$\frac{\dfrac{}{TI; .; f_u : T_{1.31}; . \vdash \mathsf{store}() : \mathbb{M}(C\ 0 + \ldots + C\ (n-1) + (n))\,[(C\ 0 + \ldots + C\ (n-1) + (n))]\,\mathbf{1}} \qquad D2.1}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, y_1 : T_{1.411} \vdash \mathsf{bind}\, b = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\, b_1 = (\overline{N}\ []\ []\ \uparrow^1 ! f_u)\ \mathsf{in}\ b_1\ \uparrow^1 \langle\!\langle y_1, b\rangle\!\rangle : T_{1.44}}$$

D1.5:

$$\frac{\dfrac{D2}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, y_1 : T_{1.411} \vdash E_1 : T_{1.44}} \qquad \dfrac{D3}{TI; .; f_u : T_{1.31}; a : \alpha\ n \vdash E_2 : T_{1.45}}}{TI; .; f_u : T_{1.31}; y_1 : T_{1.411} \vdash \mathsf{bind}\, a = E_1\ \mathsf{in}\ E_2 : T_{1.431}}$$

D1.4:

$$\frac{}{TI; .; f_u : T_{1.31}; p : T_p \vdash p : T_p}$$

D1.3

$$\frac{\dfrac{D1.4 \qquad D1.5}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, p : T_p, y_1 : T_{1.411} \vdash \mathsf{release} - = p\ \mathsf{in}\ \mathsf{bind}\, a = E_1\ \mathsf{in}\ E_2 : T_{1.43}}}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, p : T_p, y_1 : T_{1.411} \vdash E_0 : T_{1.43}}$$

D1.2

$$\frac{\dfrac{}{TI; .; f_u : T_{1.31}; y_2 : T_{1.412} \vdash y_2 : T_{1.412}} \qquad D1.3}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, p : T_p, y_1 : T_{1.411}, y_2 : T_{1.412} \vdash \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.42}}$$

D1.1

$$\frac{}{TI; .; f_u : T_{1.31}; x : T_{1.41} \vdash x : T_{1.41}}$$

D1:

$$\frac{D1.1 \qquad D1.2}{TI; .; f_u : T_{1.31}; \overline{N} : T_{0.11}, p : T_p, x : T_{1.41} \vdash \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.42}}$$

D0:

$$\frac{}{TI; .; .; f : T_{1.3} \vdash f : T_{1.3}}$$

D0.0:

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D0 \qquad D1}{TI; .; .; \overline{N} : T_{0.11}, p : T_p, f : T_{1.31}, x : T_{1.41} \vdash \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.42}}}{TI; .; .; \overline{N} : T_{0.11}, p : T_p, f : T_{1.31} \vdash \lambda x.\, \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.4}}}{TI; .; .; \overline{N} : T_{0.11}, p : T_p, f : T_{1.31} \vdash \mathsf{ret}\ \lambda x.\, \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.40}}}{TI; .; .; \overline{N} : T_{0.11}, p : T_p \vdash \lambda f.\, \mathsf{ret}\ \lambda x.\, \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{1.2}}}{.; n; .; .; \overline{N} : T_{0.11}, p : T_p \vdash \Lambda.\Lambda.\lambda f.\, \mathsf{ret}\ \lambda x.\, \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_1}}{.; n; .; .; \overline{N} : T_{0.11}, p : T_p \vdash \mathsf{ret}\ \Lambda.\Lambda.\lambda f.\, \mathsf{ret}\ \lambda x.\, \mathsf{let}\, ! f_u = f\ \mathsf{in}\ \mathsf{let} \langle\!\langle y_1, y_2\rangle\!\rangle = x\ \mathsf{in}\ \mathsf{release} - = y_2\ \mathsf{in}\ E_0 : T_{0.2}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{D0.0}{.;n;.;.;p:T_p \vdash \lambda\overline{N}.\,\mathsf{ret}\,\Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1,y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release}-=y_2\ \mathsf{in}\ E_0 : T_{0.1}}}{.;n;.;.;p:T_p \vdash \mathsf{ret}\,\lambda\overline{N}.\,\mathsf{ret}\,\Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1,y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release}-=y_2\ \mathsf{in}\ E_0 : T_{0.01}}}{.;n;.;.;. \vdash \lambda p.\,\mathsf{ret}\,\lambda\overline{N}.\,\mathsf{ret}\,\Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1,y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release}-=y_2\ \mathsf{in}\ E_0 : T_{0.0}}}{.;.;.;.;. \vdash \Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N}.\,\mathsf{ret}\,\Lambda.\Lambda.\lambda f.\,\mathsf{ret}\,\lambda x.\,\mathsf{let}\,!\,f_u = f\ \mathsf{in}\ \mathsf{let}\langle\langle y_1,y_2\rangle\rangle = x\ \mathsf{in}\ \mathsf{release}-=y_2\ \mathsf{in}\ E_0 : T_0}$$

Type derivation for $add : \forall n_1, n_2.\,[(n_1 * 3 + n_1 + 2)]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))))$

$add = \Lambda.\Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0$
where
$E_0 = \mathsf{release}-=p\ \mathsf{in}\ \mathsf{bind}\,a = E_1\ \mathsf{in}\ E_2$
$E0.1 = \mathsf{release}-=y_2\ \mathsf{in}\ \mathsf{bind}\,b_1 = (\mathsf{bind}\,b_2 = \mathsf{store}\,()\ \mathsf{in}\ succ\,[]\,b_2)\ \mathsf{in}\ b_1 \uparrow^1 y_1$
$E_1 = \overline{N_1}\ []\ []\ \uparrow^1 !(\Lambda.\lambda t.\,\mathsf{let}\langle\langle y_1, y_2\rangle\rangle = t\ \mathsf{in}\ E0.1)$
$E_2 = \mathsf{bind}\,b = \mathsf{store}()\ \mathsf{in}\ a \uparrow^1 \langle\langle \overline{N_2}, b\rangle\rangle$

$T_p = [(n_1 * 3 + n_1 + 2)]\,\mathbf{1}$
$T_0 = \forall n_1, n_2.T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))))$
$T_{0.1} = \forall n_2.T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))))$
$T_{0.2} = T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))))$
$T_{0.20} = \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_1+n_2])))$
$T_{0.21} = (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_1+n_2])))$
$T_{0.3} = \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2)))$
$T_{0.31} = \mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))$
$T_{0.4} = \mathbb{M}\,1\,(\mathsf{Nat}\ (n_1+n_2))$
$T_{0.40} = \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1+n_2))$
$T_{0.5} = \mathbb{M}(n_1 * 3 + n_1 + 1)\,(\mathsf{Nat}\ (n_1+n_2))$
$T_{0.6} = \mathbb{M}(n_1 * 3 + n_1 + 2)\,(\mathsf{Nat}\ (n_1+n_2))$
$T_1 =$
$\forall\alpha : \mathbb{N} \to Type.\forall C.!(\forall k.((\alpha\ k \otimes [C\ k]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (k+1)))) \multimap$
$\mathbb{M}\,0\,((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n_1-1) + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (n_1)))$
$a_f = \lambda k.\mathsf{Nat}\ (n_2+k)$
$T_{1.1} = \forall C.!(\forall k.((a_f\ k \otimes [C\ k]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(a_f\ (k+1)))) \multimap$
$\mathbb{M}\,0\,((a_f\ 0 \otimes [(C\ 0 + \ldots + C\ (n_1-1) + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(a_f\ n_1))$
$T_{1.2} = \forall C.!(\forall k.((\mathsf{Nat}\ (n_2+k)\ \otimes [C\ k]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+(k+1))))) \multimap$
$\mathbb{M}\,0\,((\mathsf{Nat}\ (n_2+0) \otimes [(C\ 0 + \ldots + C\ (n_1-1) + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+n_1)))$
$T_{1.21} = !(\forall k.((\mathsf{Nat}\ (n_2+k)\ \otimes [C\ k]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+(k+1))))) \multimap$
$\mathbb{M}\,0\,((\mathsf{Nat}\ (n_2+0) \otimes [(C\ 0 + \ldots + C\ (n_1-1) + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+n_1)))[(\lambda_s - .3)/C]$
$T_{1.22} = !(\forall k.((\mathsf{Nat}\ (n_2+k)\ \otimes [3]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2+(k+1)])))$
$T_{1.23} = (\forall k.((\mathsf{Nat}\ (n_2+k)\ \otimes [3]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2+(k+1)])))$
$T_{1.24} = ((\mathsf{Nat}\ (n_2+k)\ \otimes [3]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+(k+1))))$
$T_{1.241} = (\mathsf{Nat}\ (n_2+k)\ \otimes [3]\,\mathbf{1})$
$T_{1.2411} = (\mathsf{Nat}\ (n_2+k)$
$T_{1.2412} = [3]\,\mathbf{1}$
$T_{1.242} = \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+(k+1)))$
$T_{1.3} = \mathbb{M}\,0\,((\mathsf{Nat}\ (n_2+0) \otimes [(n_1 * 3 + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+n_1)))$
$T_{1.30} = \mathbb{M}\,1\,((\mathsf{Nat}\ (n_2+0) \otimes [(n_1 * 3 + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+n_1)))$
$T_{1.31} = ((\mathsf{Nat}\ (n_2+0) \otimes [(n_1 * 3 + n_1)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+n_1)))$
$T_2 = \mathsf{Nat}\ n_2$
$T_3 = (\mathsf{Nat}\ (n_2+k) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_2+k+1)))$

D3:

$$\dfrac{}{.;n_1,n_2;.;.;\overline{N_1} : T_1 \vdash \overline{N_1} : T_1}$$

D2.10:

$$\dfrac{\dfrac{D3 \qquad \dfrac{}{.;n_1,n_2;.;.;. \vdash (\lambda_t k.\mathsf{Nat}[n_2+k]) : \mathbb{N} \to Type}}{.;n_1,n_2;.;.;\overline{N_1} : T_1 \vdash \overline{N_1}\ [] : T_{1.1}}}{.;n_1,n_2;.;.;\overline{N_1} : T_1 \vdash \overline{N_1}\ [] : T_{1.2}}$$

D2:

$$\frac{D2.10 \qquad \frac{}{.; n_1, n_2; .; .; . \vdash (\lambda_s - .3) : \mathbb{N} \to \mathbb{N}}}{.; n_1, n_2; .; .; \overline{N_1} : T_1 \vdash \overline{N_1}\ []\ [] : T_{1.21}}$$

D1.32:

$$\frac{}{.; n_1, n_2, k; .; .; b_2 : [2]\,\mathbf{1} \vdash succ\ []\ b_2 : \mathbb{M}\,0\,T_3}$$

D1.31:

$$\frac{\frac{}{.; n_1, n_2, k; .; .; . \vdash \mathsf{store}() : \mathbb{M}\,2\,[2]\,\mathbf{1}} \qquad D1.32}{.; n_1, n_2, k; .; .; . \vdash (\mathsf{bind}\, b_2 = \mathsf{store}()\ \mathsf{in}\ succ\ []\ b_2\ ) : \mathbb{M}\,2\,T_3}$$

D1.3:

$$\frac{D1.31 \qquad \frac{}{.; n_1, n_2, k; .; .; y_1 : T_{1.2411}, b_1 : T_3 \vdash b_1 \uparrow^1 y_1 : \mathbb{M}\,1\,\mathsf{Nat}[n_2 + k + 1]}}{.; n_1, n_2, k; .; .; y_1 : T_{1.2411} \vdash \mathsf{bind}\, b_1 = (\mathsf{bind}\, b_2 = \mathsf{store}\,()\ \mathsf{in}\ succ\ []\ b_2\ )\ \mathsf{in}\ b_1 \uparrow^1 y_1 : \mathbb{M}(3)\,\mathsf{Nat}[n_2 + k + 1]}$$

D1.2:

$$\frac{\frac{}{.; n_1, n_2, k; .; .; y_2 : T_{1.2412} \vdash y_2 : T_{1.2412}} \qquad D1.3}{\begin{array}{c} .; n_1, n_2, k; .; .; y_1 : T_{1.2411}, y_2 : T_{1.2412} \vdash \\ \mathsf{release} - = y_2\ \mathsf{in}\ \ \mathsf{bind}\, b_1 = (\mathsf{bind}\, b_2 = \mathsf{store}\,()\ \mathsf{in}\ succ\ []\ b_2)\ \mathsf{in}\ b_1 \uparrow^1 y_1 : \mathbb{M}\,0\,\mathsf{Nat}[n_2 + k + 1] \end{array}}$$

D1.1:

$$\frac{\frac{\frac{}{.; n_1, n_2, k; .; .; t : T_{1.241} \vdash t : T_{1.241}} \qquad \frac{D1.2}{.; n_1, n_2, k; .; .; y_1 : T_{1.2411}, y_2 : T_{1.2412} \vdash E0.1 : T_{1.242}}}{.; n_1, n_2, k; .; .; t : T_{1.241} \vdash \mathsf{let}\langle\!\langle y_1, y_2 \rangle\!\rangle = t\ \mathsf{in}\ E0.1 : T_{1.242}}}{.; n_1, n_2, k; .; .; . \vdash \lambda t.\, \mathsf{let}\langle\!\langle y_1, y_2 \rangle\!\rangle = t\ \mathsf{in}\ E0.1) : T_{1.24}}$$

D1:

$$\frac{D2 \qquad \frac{\frac{D1.1}{.; n_1, n_2; .; .; . \vdash (\Lambda.\lambda t.\, \mathsf{let}\langle\!\langle y_1, y_2 \rangle\!\rangle = t\ \mathsf{in}\ E0.1) : T_{1.23}}}{.; n_1, n_2; .; .; . \vdash !(\Lambda.\lambda t.\, \mathsf{let}\langle\!\langle y_1, y_2 \rangle\!\rangle = t\ \mathsf{in}\ E0.1) : T_{1.22}}}{.; n_1, n_2; .; .; \overline{N_1} : T_1 \vdash \overline{N_1}\ []\ [] \uparrow^1 !(\Lambda.\lambda t.\, \mathsf{let}\langle\!\langle y_1, y_2 \rangle\!\rangle = t\ \mathsf{in}\ E0.1) : T_{1.30}}$$

D0.1

$$\frac{D1}{.; n_1, n_2; .; .; \overline{N_1} : T_1, \overline{N_2} : T_2 \vdash E_1 : T_{1.30}}$$

D2.1:

$$\frac{}{.; n_1, n_2; .; .; \overline{N_2} : T_2, a : T_{1.31}, b : [(n_1 * 3 + n_1)]\,\mathbf{1} \vdash a \uparrow^1 \langle\!\langle \overline{N_2}, b \rangle\!\rangle : T_{0.4}}$$

D2.0:

$$\frac{\frac{}{.; n_1, n_2; .; .; . \vdash \mathsf{store}() : \mathbb{M}(n_1 * 3 + n_1)\,[(n_1 * 3 + n_1)]\,\mathbf{1}} \qquad D2.1}{.; n_1, n_2; .; .; \overline{N_1} : T_1, \overline{N_2} : T_2, a : T_{1.31} \vdash \mathsf{bind}\, b = \mathsf{store}()\ \mathsf{in}\ a \uparrow^1 \langle\!\langle \overline{N_2}, b \rangle\!\rangle : T_{0.5}}$$

D0.2:

$$\frac{D2.0}{.; n_1, n_2; .; .; \overline{N_1} : T_1, \overline{N_2} : T_2, a : T_{1.31} \vdash E_2 : T_{0.5}}$$

D0:

$$\frac{D0.1 \qquad D0.2}{.; n_1, n_2; .; .; \overline{N_1} : T_1, \overline{N_2} : T_2 \vdash \mathsf{bind}\, a = E_1\ \mathsf{in}\ \ E_2 : T_{0.6}}$$

D0.0

$$\frac{\frac{}{.; n_1, n_2; .; .; p : T_p \vdash p : T_p} \qquad D0}{.; n_1, n_2; .; .; p : T_p, \overline{N_1} : T_1, \overline{N_2} : T_2 \vdash \mathsf{release} - = p\ \mathsf{in}\ \ \mathsf{bind}\, a = E_1\ \mathsf{in}\ \ E_2 : T_{0.40}}$$

Main derivation:

$$\begin{array}{c} D0.0 \\ \hline .; n_1, n_2; .; .; p : T_p, \overline{N_1} : T_1 \vdash \lambda \overline{N_2}.E_0 : T_{0.31} \\ \hline .; n_1, n_2; .; .; p : T_p, \overline{N_1} : T_1 \vdash \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_{0.3} \\ \hline .; n_1, n_2; .; .; p : T_p \vdash \lambda \overline{N_1}.\, \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_{0.21} \\ \hline .; n_1, n_2; .; .; p : T_p \vdash \mathsf{ret}\, \lambda \overline{N_1}.\, \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_{0.20} \\ \hline .; n_1, n_2; .; .; . \vdash \lambda p.\, \mathsf{ret}\, \lambda \overline{N_1}.\, \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_{0.2} \\ \hline .; n_1; .; .; . \vdash \Lambda.\lambda p.\, \mathsf{ret}\, \lambda \overline{N_1}.\, \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_{0.1} \\ \hline .; .; .; .; . \vdash \Lambda.\Lambda.\lambda p.\, \mathsf{ret}\, \lambda \overline{N_1}.\, \mathsf{ret}\, \lambda \overline{N_2}.E_0 : T_0 \end{array}$$

Type derivation for *mult*

$mult : \forall n_1, n_2.\, [(n_1 * (n_2 * 3 + n_2 + 4) + n_1 + 2)]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2))))$

$mult = \Lambda.\Lambda.\lambda p.\, \mathsf{ret}\, \lambda \overline{N_1}.\, \mathsf{ret}(\lambda \overline{N_2}.E_0)$

where

$E_0 = \mathsf{release} - = p \text{ in } \mathsf{bind}\, a = E_1 \text{ in } E_2$

$E0.1 = \mathsf{release} - = y_2 \text{ in } \mathsf{bind}\, b_1 = (\mathsf{bind}\, b_2 = \mathsf{store}\,() \text{ in } add\ []\ []\ b_2 \uparrow^1 \overline{N_2}) \text{ in } b_1 \uparrow^1 y_1$

$E_1 = \overline{N_1}\ []\ []\ \uparrow^1\ !(\Lambda.\lambda t.\, \mathsf{let} \langle\!\langle y_1, y_2 \rangle\!\rangle = t \text{ in } E0.1)$

$E_2 = \mathsf{bind}\, b = \mathsf{store}() \text{ in } a \uparrow^1 \langle\!\langle \overline{0}, b \rangle\!\rangle$

$T_p = [(n_1 * (n_2 * 3 + n_2 + 4) + n_1 + 2)]\, \mathbf{1}$

$T_0 = \forall n_1, n_2.T_p \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2))))$

$T_{0.1} = \forall n_2.T_p \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2))))$

$T_{0.2} = T_p \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2))))$

$T_{0.21} = \mathbb{M}\, 0\, (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_1 * n_2])))$

$T_{0.22} = (\mathsf{Nat}\ n_1 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2))))$

$T_{0.3} = \mathbb{M}\, 0\, (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2)))$

$T_{0.31} = (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_1 * n_2)))$

$T_{0.4} = \mathbb{M}\, 1\, (\mathsf{Nat}\ (n_1 * n_2))$

$T_{0.5} = \mathbb{M}(n_1 * (n_2 * 3 + n_2 + 4) + n_1 + 1)\, (\mathsf{Nat}\ (n_1 * n_2))$

$T_{0.6} = \mathbb{M}(n_1 * (n_2 * 3 + n_2 + 4) + n_1 + 2)\, (\mathsf{Nat}\ (n_1 * n_2))$

$T_1 =$
$\forall \alpha : \mathbb{N} \to Type.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ (j_n + 1)))) \multimap$
$\mathbb{M}\, 0\, ((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n_1 - 1) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\alpha\ n_1))$

$a_f = \lambda k.\mathsf{Nat}[n_2 * k]$

$T_{1.1} = \forall C.!(\forall j_n.((a_f\ j_n \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (a_f\ (j_n + 1)))) \multimap$
$\mathbb{M}\, 0\, ((a_f\ 0 \otimes [(C\ 0 + \ldots + C\ (n_1 - 1) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (a_f\ n_1))$

$T_{1.2} = \forall C.!(\forall j_n.((\mathsf{Nat}[n_2 * j_n] \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (j_n + 1)]))) \multimap$
$\mathbb{M}\, 0\, ((\mathsf{Nat}[n_2 * 0] \otimes [(C\ 0 + \ldots + C\ (n_1 - 1) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_2 * n_1)))$

$T_{1.21} = !(\forall j_n.((\mathsf{Nat}[n_2 * j_n] \otimes [C\ j_n]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (j_n + 1)]))) \multimap$
$\mathbb{M}\, 0\, ((\mathsf{Nat}[n_2 * 0] \otimes [(C\ 0 + \ldots + C\ (n_1 - 1) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_2 * n_1)))[C/(\lambda_.(n_2 * 3 + n_2 + 4))]$

$T_{1.22} = !(\forall j_n.((\mathsf{Nat}[n_2 * j_n] \otimes [(n_2 * 3 + n_2 + 4)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (j_n + 1)])))$

$T_{1.23} = (\forall j_n.((\mathsf{Nat}[n_2 * j_n] \otimes [(n_2 * 3 + n_2 + 4)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (j_n + 1)])))$

$T_{1.24} = ((\mathsf{Nat}[n_2 * k] \otimes [(n_2 * 3 + n_2 + 4)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (k + 1)]))$

$T_{1.241} = (\mathsf{Nat}[n_2 * k] \otimes [(n_2 * 3 + n_2 + 4)]\, \mathbf{1})$

$T_{1.2411} = (\mathsf{Nat}[n_2 * k]$

$T_{1.2412} = [(n_2 * 3 + n_2 + 4)]\, \mathbf{1}$

$T_{1.242} = \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (k + 1)])$

$T_{1.3} = \mathbb{M}\, 0\, ((\mathsf{Nat}[n_2 * 0] \otimes [(n_1 * (n_2 * 3 + n_2 + 4) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_2 * n_1)))$

$T_{1.30} = \mathbb{M}\, 1\, ((\mathsf{Nat}[n_2 * 0] \otimes [(n_1 * (n_2 * 3 + n_2 + 4) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_2 * n_1)))$

$T_{1.31} = ((\mathsf{Nat}[n_2 * 0] \otimes [(n_1 * (n_2 * 3 + n_2 + 4) + n_1)]\, \mathbf{1}) \multimap \mathbb{M}\, 0\, (\mathsf{Nat}\ (n_2 * n_1)))$

$T_2 = \mathsf{Nat}\ n_2$

$T_3 = (\mathsf{Nat}[n_2 * k] \multimap \mathbb{M}\, 0\, (\mathsf{Nat}[n_2 * (k + 1)]))$

D3:

$$\frac{}{.; n_1, n_2; .; .; \overline{N_1} : T_1 \vdash \overline{N_1} : T_1}$$

D2.10:

$$\dfrac{\dfrac{D3 \qquad \dfrac{}{.;n_1,n_2;.;.;.\vdash (\lambda_t k.\mathsf{Nat}[n_2*k]):\mathbb{N}\to Type}}{.;n_1,n_2;.;.;\overline{N_1}:T_1\vdash \overline{N_1}\ []:T_{1.1}}}{.;n_1,n_2;.;.;\overline{N_1}:T_1\vdash \overline{N_1}\ []:T_{1.2}}$$

D2:

$$\dfrac{D2.10 \qquad \dfrac{}{.;n_1,n_2;.;.;.\vdash(\lambda_s-.(n_2*3+n_2+4)):\mathsf{S}\to\mathsf{S}}}{.;n_1,n_2;.;.;\overline{N_1}:T_1\vdash\overline{N_1}\ []\ []:T_{1.21}}\ \text{T-iapp}$$

D1.32

$$\dfrac{}{.;n_1,n_2,k;.;.;.\vdash add\ []\ []\ b_2\uparrow^1\overline{N_2}:\mathbb{M}\,1\,T_3}$$

D1.31

$$\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;.\vdash \mathsf{store}\,():\mathbb{M}(n_2*3+n_2+2)\,[(n_2*3+n_2+2)]\,\mathbf{1}}\qquad D1.32}{.;n_1,n_2,k;.;.;y_1:T_{1.241},b_1:T_3\vdash(\mathsf{bind}\,b_2=\mathsf{store}()\ \mathsf{in}\ add\ []\ []\ b_2\uparrow^1\overline{N_2}):\mathbb{M}(n_2*3+n_2+3)\,T_3}$$

D1.3

$$\dfrac{D1.31\qquad \dfrac{}{.;n_1,n_2,k;.;.;y_1:T_{1.241},b_1:T_3\vdash b_1\uparrow^1 y_1:\mathbb{M}\,1\,\mathsf{Nat}[n_2*(k+1)]}}{\begin{array}{c}.;n_1,n_2,k;.;.;y_1:\vdash \mathsf{bind}\,b_1=(\mathsf{bind}\,b_2=\mathsf{store}\,()\ \mathsf{in}\ add\ []\ []\ b_2\uparrow^1\overline{N_2})\ \mathsf{in}\ b_1\uparrow^1 y_1:\\ \mathbb{M}(n_2*3+n_2+4)\,\mathsf{Nat}[n_2*(k+1)]\end{array}}$$

D1.2:

$$\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;y_2:T_{1.2412}\vdash y_2:T_{1.2412}}\qquad D1.3}{\begin{array}{c}.;n_1,n_2,k;.;.;y_1:T_{1.2411},y_2:T_{1.2412}\vdash\\ \mathsf{release}-=y_2\ \mathsf{in}\ \ \mathsf{bind}\,b_1=(\mathsf{bind}\,b_2=\mathsf{store}\,()\ \mathsf{in}\ add\ []\ []\ b_2\uparrow^1\overline{N_2})\ \mathsf{in}\ b_1\uparrow^1 y_1:\mathbb{M}\,0\,\mathsf{Nat}[n_2*(k+1)]\end{array}}$$

D1.1

$$\dfrac{\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;t:T_{1.241}\vdash t:T_{1.241}}\qquad \dfrac{D1.2}{.;n_1,n_2,k;.;.;y_1:T_{1.2411},y_2:T_{1.2412}\vdash E0.1:T_{1.242}}}{.;n_1,n_2,k;.;.;t:T_{1.241}\vdash \mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.242}}}{.;n_1,n_2,k;.;.;.\vdash\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.24}}$$

D1:

$$\dfrac{D2\qquad\dfrac{\dfrac{D1.1}{.;n_1,n_2;.;.;.\vdash(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.23}}}{.;n_1,n_2;.;.;.\vdash!(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.22}}}{.;n_1,n_2;.;.;\overline{N_1}:T_1\vdash\overline{N_1}\ []\ []\uparrow^1!(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.30}}$$

D0.1:

$$\dfrac{D1}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2\vdash E_1:T_{1.30}}$$

D2.1:

$$\dfrac{}{.;n_1,n_2;.;.;\overline{N_2}:T_2,a:T_{1.31},b:[(n_1*(n_2*3+n_2+4)+n_1)]\,\mathbf{1}\vdash a\uparrow^1\langle\!\langle\overline{0},b\rangle\!\rangle:T_{0.4}}$$

D2.0:

$$\dfrac{\dfrac{}{.;n_1,n_2;.;.;.\vdash \mathsf{store}():\mathbb{M}(n_1*(n_2*3+n_2+4)+n_1)\,[(n_1*(n_2*3+n_2+4)+n_1)]\,\mathbf{1}}\qquad D2.1}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2,a:T_{1.31}\vdash \mathsf{bind}\,b=\mathsf{store}()\ \mathsf{in}\ a\uparrow^1\langle\!\langle\overline{0},b\rangle\!\rangle:T_{0.5}}$$

D0.2:

$$\dfrac{D2.0}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2,a:T_{1.31}\vdash E_2:T_{0.5}}$$

D0:

$$\dfrac{D0.1\qquad D0.2}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2\vdash \mathsf{bind}\,a=E_1\ \mathsf{in}\ \ E_2:T_{0.6}}$$

D0.0

$$\dfrac{\dfrac{}{.;n_1,n_2;.;.;p:T_p \vdash p:T_p} \qquad D0}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1,\overline{N_2}:T_2 \vdash \mathsf{release} - = p \;\mathsf{in}\; \mathsf{bind}\, a = E_1 \;\mathsf{in}\;\; E_2 : T_{0.4}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D0.0}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1 \vdash \lambda\overline{N_2}.E_0 : T_{0.31}}}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1 \vdash \mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.3}}}{.;n_1,n_2;.;.;p:T_p \vdash \lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.22}}}{.;n_1,n_2;.;.;p:T_p \vdash \mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.21}}}{.;n_1,n_2;.;.;. \vdash \lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.2}}}{.;n_1;.;.;. \vdash \Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.1}}}{.;.;.;.;. \vdash \Lambda.\Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_0}$$

Type derivation for *exp*

$exp : \forall n_1, n_2.\, [\sum_{i \in \{0,n_2-1\}} (\lambda k.(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)\ (i)) + n_2 + 2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2}))))$

$exp = \Lambda.\Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0$

where

$E_0 = \mathsf{release} - = p \;\mathsf{in}\; \mathsf{bind}\, a = E_1 \;\mathsf{in}\;\; E_2$

$E_{0.1} = \mathsf{release} - = y_2 \;\mathsf{in}\; \mathsf{bind}\, b_1 = (\mathsf{bind}\, b_2 = \mathsf{store}\,()\;\mathsf{in}\; mult\,[]\,[]\,b_2 \uparrow^1 \overline{N_1}) \;\mathsf{in}\; b_1 \uparrow^1 y_1$

$E_1 = \overline{N_2}\ []\ []\ \uparrow^1\ !(\Lambda.\lambda t.\,\mathsf{let}\langle\langle y_1, y_2\rangle\rangle = t \;\mathsf{in}\; E0.1)$

$E_2 = \mathsf{bind}\, b = \mathsf{store}\,\mathbf{1} \;\mathsf{in}\; a \uparrow^1 \langle\langle \overline{1}, b\rangle\rangle$

$P = \sum_{i \in \{0,n_2-1\}} (\lambda k.(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)\ i) + n_2 + 2$

$T_p = [P]\,\mathbf{1}$

$T_b = [P-1]\,\mathbf{1}$

$T_0 = \forall n_1, n_2.T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2}))))$

$T_{0.1} = \forall n_2.T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2}))))$

$T_{0.2} = T_p \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2}))))$

$T_{0.20} = \mathbb{M}\,0\,(\mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_1^{n_2}])))$

$T_{0.21} = \mathsf{Nat}\ n_1 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$

$T_{0.3} = \mathbb{M}\,0\,(\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$

$T_{0.31} = (\mathsf{Nat}\ n_2 \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$

$T_{0.4} = \mathbb{M}\,1\,(\mathsf{Nat}\ (n_1^{n_2}))$

$T_{0.5} = \mathbb{M}(P-1)\,(\mathsf{Nat}\ (n_1^{n_2}))$

$T_{0.6} = \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2}))$

$T_1 =$
$\forall \alpha : \mathbb{N} \to Type.\forall C.!(\forall j_n.((\alpha\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ (j_n+1)))) \multimap$
$\mathbb{M}\,0\,((\alpha\ 0 \otimes [(C\ 0 + \ldots + C\ (n_2-1) + n_2)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\alpha\ n_2))$

$a_f = \lambda k.\mathsf{Nat}[n_2^k]$

$T_{1.1} =$
$\forall C.!(\forall j_n.((a_f\ j_n \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(a_f\ (j_n+1)))) \multimap$
$\mathbb{M}\,0\,((a_f\ 0 \otimes [(C\ 0 + \ldots + C\ (n_2-1) + n_2)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(a_f\ n_2))$

$T_{1.2} =$
$\forall C.!(\forall j_n.((\mathsf{Nat}[n_2^{j_n}]\ \otimes [C\ j_n]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(j_n+1)}]))) \multimap$
$\mathbb{M}\,0\,((\mathsf{Nat}[n_2^0] \otimes [(C\ 0 + \ldots + C\ (n_2-1) + n_2)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$

$T_{1.21} =$
$!(\forall j_n.((\mathsf{Nat}[n_2^{j_n}]\ \otimes [(n_1 * (n_1^{j_n} * 3 + n_1^{j_n} + 4) + n_1 + 4)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(j_n+1)}]))) \multimap$
$\mathbb{M}\,0\,((\mathsf{Nat}[n_2^0] \otimes [P]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$

$P =$
$(\lambda k.(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4))\ 0 + \ldots + (\lambda k.(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4))\ (n_2 - 1) + n_2$

$T_{1.22} =!(\forall k.((\mathsf{Nat}[n_2^k]\ \otimes [(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(k+1)}])))$

$T_{1.23} = (\forall k.((\mathsf{Nat}[n_2^k]\ \otimes [(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(k+1)}])))$

$T_{1.24} = ((\mathsf{Nat}[n_2^k]\ \otimes [(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(k+1)}]))$

$T_{1.241} = (\mathsf{Nat}[n_2^k]\ \otimes [(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)]\,\mathbf{1})$

$T_{1.2411} = \mathsf{Nat}[n_2^k]$

$T_{1.2412} = ([(n_1 * (n_1^k * 3 + n_1^k + 4) + n_1 + 4)]\,\mathbf{1})$

$T_{1.242} = \mathbb{M}\,0\,(\mathsf{Nat}[n_2^{(k+1)}])$
$T_{1.3} = \mathbb{M}\,0\,((\mathsf{Nat}[n_2^0] \otimes [P]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$
$T_{1.30} = \mathbb{M}\,1\,((\mathsf{Nat}[n_2^0] \otimes [P]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$
$T_{1.31} = ((\mathsf{Nat}[n_2^0] \otimes [P]\,\mathbf{1}) \multimap \mathbb{M}\,0\,(\mathsf{Nat}\ (n_1^{n_2})))$
$T_2 = \mathsf{Nat}\ n_1$
$T_3 = (\mathsf{Nat}[n_1^k] \multimap \mathbb{M}\,0\,(\mathsf{Nat}[n_1^{(k+1)}]))$

D3:

$$\frac{}{.;n_1,n_2;.;.;\overline{N_2}:T_1 \vdash \overline{N_2}:T_1}$$

D2.1:

$$\dfrac{\dfrac{D3 \qquad \dfrac{}{.;n_1,n_2;.;.;.\vdash (\lambda_t k.\mathsf{Nat}[n_2^k]):\mathbb{N}\to Type}}{.;n_1,n_2;.;.;\overline{N_2}:T_1\vdash \overline{N_2}\ []:T_{1.1}}}{.;n_1,n_2;.;.;\overline{N_2}:T_1\vdash \overline{N_2}\ []:T_{1.2}}$$

D2:

$$\dfrac{D2.1 \qquad \dfrac{}{.;n_1,n_2;.;.;.\vdash(\lambda_s k.(n_1*(n_1^k*3+n_1^k+4)+n_1+2)):\mathbb{N}\to\mathbb{N}}}{.;n_1,n_2;.;.;\overline{N_2}:T_1\vdash\overline{N_2}\ []\ []:T_{1.21}}$$

D1.32

$$\frac{}{.;n_1,n_2,k;.;.;b_2:[((n_1*(n_1^k*3+n_1^k+4)+n_1+2))]\,\mathbf{1}\vdash mult\ []\ []\ b_2\ \uparrow^1\overline{N_1}:\mathbb{M}\,1\,T_3}$$

D1.31

$$\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;.\vdash \mathsf{store}\,():\mathbb{M}((n_1*(n_1^k*3+n_1^k+4)+n_1+2))\,[((n_1*(n_1^k*3+n_1^k+4)+n_1+2))]\,\mathbf{1}} \qquad D1.32}{\begin{array}{c}.;n_1,n_2,k;.;.;y_1:T_{1.241},b_1:T_3\vdash(\mathsf{bind}\,b_2=\mathsf{store}\,()\ \mathsf{in}\ mult\ []\ []\ b_2\ \uparrow^1\overline{N_1}):\\ \mathbb{M}(n_1*(n_1^k*3+n_1^k+4)+n_1+3)\,T_3\end{array}}$$

D1.3

$$\dfrac{D1.31 \qquad \dfrac{}{.;n_1,n_2,k;.;.;y_1:T_{1.241},b_1:T_3\vdash b_1\ \uparrow^1 y_1:\mathbb{M}\,1\,\mathsf{Nat}[n_2^{(k+1)}]}}{\begin{array}{c}.;n_1,n_2,k;.;.;y_1:\vdash \mathsf{bind}\,b_1=(\mathsf{bind}\,b_2=\mathsf{store}\,()\ \mathsf{in}\ mult\ []\ []\ b_2\ \uparrow^1\overline{N_1})\ \mathsf{in}\ b_1\ \uparrow^1 y_1:\\ \mathbb{M}(n_1*(n_1^k*3+n_1^k+4)+n_1+4)\,\mathsf{Nat}[n_2^{(k+1)}]\end{array}}$$

D1.2:

$$\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;y_2:T_{1.2412}\vdash y_2:T_{1.2412}} \qquad D1.3}{\begin{array}{c}.;n_1,n_2,k;.;.;y_1:T_{1.2411},y_2:T_{1.2412}\vdash\\ \mathsf{release}-=y_2\ \mathsf{in}\ \ \mathsf{bind}\,b_1=(\mathsf{bind}\,b_2=\mathsf{store}\,()\ \mathsf{in}\ mult\ b_2\ n_1\ (n_1^k)\ \uparrow^1\overline{N_1})\ \mathsf{in}\ b_1\ \uparrow^1 y_1:\mathbb{M}\,0\,\mathsf{Nat}[n_2^{(k+1)}]\end{array}}$$

D1.1

$$\dfrac{\dfrac{\dfrac{}{.;n_1,n_2,k;.;.;t:T_{1.241}\vdash t:T_{1.241}} \qquad \dfrac{D1.2}{.;n_1,n_2,k;.;.;y_1:T_{1.2411},y_2:T_{1.2412}\vdash E0.1:T_{1.242}}}{.;n_1,n_2,k;.;.;t:T_{1.241}\vdash \mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.242}}}{.;n_1,n_2,k;.;.;.\vdash\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.24}}$$

D1:

$$\dfrac{D2 \qquad \dfrac{\dfrac{D1.1}{.;n_1,n_2;.;.;.\vdash(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.23}}}{.;n_1,n_2;.;.;.\vdash!(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.22}}}{.;n_1,n_2;.;.;\overline{N_1}:T_1\vdash\overline{N_1}\ []\ []\ \uparrow^1!(\Lambda.\lambda t.\,\mathsf{let}\langle\!\langle y_1,y_2\rangle\!\rangle=t\ \mathsf{in}\ E0.1):T_{1.30}}$$

D0.1:

$$\dfrac{D1}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2\vdash E_1:T_{1.30}}$$

D2.1:

$$\frac{}{.;n_1,n_2;.;.;\overline{N_2}:T_2,a:T_{1.31},b:T_b\vdash a\ \uparrow^1\langle\!\langle\overline{1},b\rangle\!\rangle:T_{0.4}}$$

D2.0:

$$\dfrac{\dfrac{}{.;n_1,n_2;.;.;.\vdash \mathsf{store}() : \mathbb{M}(P-2)\,T_b} \qquad D2.1}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2,a:T_{1.31}\vdash \mathsf{bind}\,b=\mathsf{store}()\ \mathsf{in}\ a\ \uparrow^1\langle\langle\overline{1},b\rangle\rangle : T_{0.5}}$$

D0.2:

$$\dfrac{D2.0}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2,a:T_{1.31}\vdash E_2 : T_{0.5}}$$

D0:

$$\dfrac{D0.1 \qquad D0.2}{.;n_1,n_2;.;.;\overline{N_1}:T_1,\overline{N_2}:T_2\vdash \mathsf{bind}\,a=E_1\ \mathsf{in}\ \ E_2 : T_{0.5}}$$

D0.0

$$\dfrac{\dfrac{}{.;n_1,n_2;.;.;p:T_p\vdash p:T_p} \qquad D0}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1,\overline{N_2}:T_2\vdash \mathsf{release}-=p\ \mathsf{in}\ \ \mathsf{bind}\,a=E_1\ \mathsf{in}\ \ E_2 : T_{0.6}}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D0.0}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1\vdash \lambda\overline{N_2}.E_0 : T_{0.31}}}{.;n_1,n_2;.;.;p:T_p,\overline{N_1}:T_1\vdash \mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.3}}}{.;n_1,n_2;.;.;p:T_p\vdash \lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.21}}}{.;n_1,n_2;.;.;p:T_p\vdash \mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.20}}}{.;n_1,n_2;.;.;.\vdash \lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.2}}}{.;n_1;.;.;.\vdash \Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_{0.1}}}{.;.;.;.;.\vdash \Lambda.\Lambda.\lambda p.\,\mathsf{ret}\,\lambda\overline{N_1}.\,\mathsf{ret}\,\lambda\overline{N_2}.E_0 : T_0}$$

C.2 Map

$map : \forall n,c.!(\tau_1 \multimap \mathbb{M}\,c\,\tau_2) \multimap L^n([c]\,\tau_1) \multimap \mathbb{M}\,0\,(L^n\tau_2)$
$map \triangleq$
$\mathsf{fix}f.\Lambda.\Lambda.\lambda gl.\mathsf{let}\,!\,g_u = g\ \mathsf{in}\ E_0$
$E_0 = \mathsf{match}\ l\ \mathsf{with}\ |nil \mapsto E_{0.1}\ |h :: t \mapsto E_{0.2}$
$E_{0.1} = \mathsf{ret}\,nil$
$E_{0.2} = \mathsf{release}\,h_e = h\ \mathsf{in}\ \ E_{0.3}$
$E_{0.3} = \mathsf{bind}\,h_n = g_u\ h_e\ \mathsf{in}\ E_{0.4}$
$E_{0.4} = \mathsf{bind}\,t_n = f[][]\ !g_u\ t\ \mathsf{in}\ \mathsf{ret}\,h_n :: t_n$

Typing derivation

$E = \mathsf{fix}f.\Lambda.\Lambda.\lambda gl.\mathsf{let}\,!\,g_u = g\ \mathsf{in}\ E_0$
$E_0 = \mathsf{match}\ l\ \mathsf{with}\ |nil \mapsto E_{0.1}\ |h :: t \mapsto E_{0.2}$
$E_{0.1} = \mathsf{ret}\,nil$
$E_{0.2} = \mathsf{release}\,h_e = h\ \mathsf{in}\ \ E_{0.3}$
$E_{0.3} = \mathsf{bind}\,h_n = g_u\ h_e\ \mathsf{in}\ E_{0.4}$
$E_{0.4} = \mathsf{bind}\,t_n = f[][]\ !g_u\ t\ \mathsf{in}\ \mathsf{ret}\,h_n :: t_n$
$E_1 = \Lambda.\Lambda.\lambda gl.\mathsf{let}\,!\,g_u = g\ \mathsf{in}\ E_0$
$E_2 = \lambda gl.\mathsf{let}\,!\,g_u = g\ \mathsf{in}\ E_0$
$E_3 = \mathsf{let}\,!\,g_u = g\ \mathsf{in}\ E_0$

$T_0 = \forall n,c.!(\tau_1 \multimap \mathbb{M}\,c\,\tau_2) \multimap L^n([c]\,\tau_1) \multimap \mathbb{M}\,0\,(L^n\tau_2)$
$T_1 =!(\tau_1 \multimap \mathbb{M}\,c\,\tau_2) \multimap L^n([c]\,\tau_1) \multimap \mathbb{M}\,0\,(L^n\tau_2)$
$T_{1.1} = (\tau_1 \multimap \mathbb{M}\,c\,\tau_2)$
$T_{1.2} = L^n([c]\,\tau_1)$
$T_{1.3} = \mathbb{M}\,0\,(L^n\tau_2)$

D1.2:

$$\dfrac{}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h_n:\tau_2,t_n:L^i\tau_2\vdash \mathsf{ret}\,h_n :: t_n : \mathbb{M}\,0\,L^n\tau_2}$$

D1.1:

$$\dfrac{\dfrac{}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h_n:\tau_2\vdash f[][]\ !g_u\ t : \mathbb{M}\,0\,L^i\tau_2} \qquad D1.2}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h_n:\tau_2,t:L^i([c]\,\tau_1)\vdash E_{0.4} : \mathbb{M}\,0\,L^n\tau_2}$$

D1.0:

$$\dfrac{\dfrac{}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h_e:\tau_1 \vdash (g_u\ h_e):\mathbb{M}\,c\,\tau_2} \qquad D1.1}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h_e:\tau_1,t:L^i([c]\,\tau_1)\vdash E_{0.3}:\mathbb{M}\,c\,L^n\tau_2}$$

D1:

$$\dfrac{\dfrac{}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h:[c]\,\tau_1 \vdash h:[c]\,\tau_1} \qquad D1.0}{.;n,c,i;n=i+1;f:T_0,g_u:T_{1.1};h:[c]\,\tau_1,t:L^i([c]\,\tau_1)\vdash E_{0.2}:\mathbb{M}\,0\,L^n\tau_2}$$

D0:

$$\dfrac{\dfrac{\dfrac{}{.;n,c;n=0;f:T_0,g_u:T_{1.1};.\vdash nil:L^0\tau_2}}{.;n,c;n=0;f:T_0,g_u:T_{1.1};.\vdash \mathsf{ret}\,nil:\mathbb{M}\,0\,L^n\tau_2}}{.;n,c;n=0;f:T_0,g_u:T_{1.1};.\vdash E_{0.1}:\mathbb{M}\,0\,L^n\tau_2}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{.;n,c;f:T_0;g:!T_{1.1}\vdash g:!T_{1.1}} \qquad \dfrac{\dfrac{}{.;n,c;.;f:T_0,g_u:T_{1.1};l:T_{1.2}\vdash l:T_{1.2}} \quad D0 \quad D1}{.;n,c;.;f:T_0,g_u:T_{1.1};l:T_{1.2}\vdash E_0:\mathbb{M}\,0\,L^n\tau_2}}{.;n,c;.;f:T_0;g:!T_{1.1},l:T_{1.2}\vdash E_3:\mathbb{M}\,0\,L^n\tau_2}}{.;n,c;.;f:T_0;.\vdash E_2:T_1}}{.;.;.;f:T_0;.\vdash E_1:T_0}}{.;.;.;.;.\vdash E:T_0}$$

C.3 Append

$append : \forall s_1,s_2.L^{s_1}[1]\,\tau \multimap L^{s_2}\tau \multimap \mathbb{M}\,0\,(L^{s_1+s_2}\tau)$
$append \triangleq \mathsf{fix} f.\Lambda.\Lambda.\lambda l_1 l_2.E_0$
$E_0 = \mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_{0.1}\ |h :: t \mapsto E_{0.2}$
$E_{0.1} = \mathsf{ret}\,nil :: l_2$
$E_{0.2} = \mathsf{release}\,h_e = h\ \mathsf{in}\ \mathsf{bind}\,t_e = f[\,][\,]\ t\ l_2\ \mathsf{in}\ E_{0.3}$
$E_{0.3} = \mathsf{bind} - = \uparrow^1\ \mathsf{in}\ \mathsf{ret}\,h_e :: t_e$

Typing derivation

$E_0 = \mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_{0.1}\ |h :: t \mapsto E_{0.2}$
$E_{0.1} = \mathsf{ret}\,nil :: l_2$
$E_{0.2} = \mathsf{release}\,h_e = h\ \mathsf{in}\ \mathsf{bind}\,t_e = f[\,][\,]\ t\ l_2\ \mathsf{in}\ E_{0.3}$
$E_{0.3} = \mathsf{bind} - = \uparrow^1\ \mathsf{in}\ \mathsf{ret}\,h_e :: t_e$

$T_0 = \forall s_1,s_2.L^{s_1}[1]\,\tau \multimap L^{s_2}\tau \multimap \mathbb{M}\,0\,(L^{s_1+s_2}\tau)$
$T_1 = L^{s_1}[1]\,\tau \multimap L^{s_2}\tau \multimap \mathbb{M}\,0\,(L^{s_1+s_2}\tau)$
$T_{1.1} = L^{s_1}[1]\,\tau$
$T_{1.2} = L^{s_2}\tau$
$T_{1.3} = \mathbb{M}\,0\,(L^{s_1+s_2}\tau)$
$T_2 = L^{s_2}\tau \multimap \mathbb{M}\,s_1\,(L^{s_1+s_2}\tau)$

D1.2:

$$\dfrac{\dfrac{}{.;s_1,s_2,i;s_1=i+1;f:T_0;h_e:\tau,t_e:L^{i+s_2}\tau\vdash (h_e :: t_e):L^{i+1+s_2}\tau}}{.;s_1,s_2,i;s_1=i+1;f:T_0;h_e:\tau,t_e:L^{i+s_2}\tau\vdash \mathsf{ret}(h_e :: t_e):\mathbb{M}\,0\,(L^{s_1+s_2}\tau)}$$

D1.1:

$$\dfrac{\dfrac{}{.;s_1,s_2,i;s_1=i+1;.\vdash \uparrow^1:\mathbb{M}\,1\,\mathbf{1}} \qquad D1.2}{.;s_1,s_2,i;s_1=i+1;f:T_0;h_e:\tau,t_e:L^{i+s_2}\tau\vdash \mathsf{bind} - = \uparrow^1\ \mathsf{in}\ \mathsf{ret}\,h_e :: t_e:\mathbb{M}\,1\,(L^{s_1+s_2}\tau)}$$

D1.0:

$$\dfrac{\dfrac{}{.;s_1,s_2,i;s_1=i+1;f:T_0;t:L^i\tau,l_2:L^{s_2}\tau\vdash f[\,][\,]\ t\ l_2:\mathbb{M}(0)\,(L^{i+s_2}\tau)} \qquad D1.1}{.;s_1,s_2,i;s_1=i+1;f:T_0;h_e:\tau,t:L^i\tau,l_2:L^{s_2}\tau\vdash \mathsf{bind}\,t_e = f[\,][\,]\ t\ l_2\ \mathsf{in}\ \mathsf{ret}(h_e :: t_e):\mathbb{M}\,1\,(L^{s_1+s_2}\tau)}$$

D1:

$$\dfrac{\dfrac{}{.;s_1,s_2,i;s_1=i+1;f:T_0;h:[1]\,\tau\vdash h:[1]\,\tau} \qquad D1.0}{.;s_1,s_2,i;s_1=i+1;f:T_0;h:[1]\,\tau,t:L^i\tau,l_2:T_{1.2}\vdash E_{0.2}:\mathbb{M}\,0\,(L^{(s_1+s_2)}\tau)}$$

D0:

$$\dfrac{\dfrac{}{.;s_1,s_2;s_1=0;f:T_0;l_2:T_{1.2}\vdash l_2:L^{s_2}\tau}}{.;s_1,s_2;s_1=0;f:T_0;l_2:T_{1.2}\vdash \mathsf{ret}\,l_2:\mathbb{M}\,0\,(L^{s_1+s_2})\tau}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{.;s_1,s_2;.;f:T_0;l_1:T_{1.1}\vdash l_1:T_{1.1}} \quad D0 \quad D1}{.;s_1,s_2;.;f:T_0;l_1:T_{1.1},l_2:T_{1.2}\vdash E_0:\mathbb{M}\,0\,(L^{s_1+s_2}\tau)}}{.;s_1,s_2;.;f:T_0;.\vdash \lambda l_1 l_2.E_0:T_1}}{.;.;.;f:T_0;.\vdash \Lambda.\Lambda.\lambda l_1 l_2.E_0:T_0}}{.;.;.;.;.\vdash \mathsf{fix}f.\Lambda.\Lambda.\lambda l_1 l_2.E_0:T_0}$$

C.4 Eager functional queue

$enqueue : \forall m,n.\,[3]\,\mathbf{1} \multimap \tau \multimap L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$enqueue \triangleq \Lambda.\lambda\ p\ a\ l_1\ l_2.\mathsf{release} - = p \text{ in } \mathsf{bind}\, x = \mathsf{store}\, a \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$

Typing derivation for enqueue *enqueue*

$T_0 = \forall m,n.\,[3]\,\mathbf{1} \multimap \tau \multimap L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_1 = [3]\,\mathbf{1} \multimap \tau \multimap L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_{1.0} = [3]\,\mathbf{1}$
$T_2 = \tau \multimap L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_3 = L^n([2]\,\tau) \multimap L^m\tau \multimap \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_{3.1} = L^n([2]\,\tau)$
$T_{3.2} = L^m\tau$
$T_4 = \mathbb{M}\,0\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_5 = \mathbb{M}\,1\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$
$T_6 = \mathbb{M}\,3\,(L^{n+1}([2]\,\tau)\otimes L^m\tau)$

$enqueue = \Lambda.\lambda\ p\ a\ l_1\ l_2.\mathsf{release} - = p \text{ in } \mathsf{bind}\, x = \mathsf{store}\, a \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$
$E_1 = \lambda\ p\ a\ l_1\ l_2.\mathsf{release} - = p \text{ in } \mathsf{bind}\, x = \mathsf{store}\, a \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$
$E_2 = \mathsf{release} - = p \text{ in } \mathsf{bind}\, x = \mathsf{store}\, a \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$
$E_3 = \mathsf{bind}\, x = \mathsf{store}\, a \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$
$E_4 = \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$
$E_5 = \mathsf{ret}\langle\langle(x :: l_1), l_2\rangle\rangle$

D2:

$$\dfrac{}{.;m,n;.;.;x:[2]\,\tau,l_1:L^n([2]\,\tau),l_2:L^m\tau\vdash E_5:T_4}$$

D1:

$$\dfrac{\dfrac{}{.;m,n;.;.;.\vdash \uparrow^1:\mathbb{M}\,1\,\mathbf{1}} \quad D2}{.;m,n;.;.;x:[2]\,\tau,l_1:L^n([2]\,\tau),l_2:L^m\tau\vdash E_4:T_5}$$

D0:

$$\dfrac{\dfrac{}{.;m,n;.;.;a:\tau\vdash \mathsf{store}\,a:\mathbb{M}\,2\,([2]\,\tau)} \quad D1}{.;m,n;.;.;a:\tau,l_1:L^n([2]\,\tau),l_2:L^m\tau\vdash E_3:T_6}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{.;m,n;.;.;p:T_{1.0}\vdash p:T_{1.0}} \quad D0}{.;m,n;.;.;p:T_{1.0},a:\tau,l_1:L^n([2]\,\tau),l_2:L^m\tau\vdash E_2:T_4}}{.;m,n;.;.;.\vdash E_1:T_1}}{.;.;.;.;.\vdash enqueue:T_0}$$

$Dq : \forall m,n.(m+n>0) \Rightarrow [1]\,\mathbf{1} \multimap L^m([2]\,\tau) \multimap L^n\tau \multimap$
$\mathbb{M}\,0\,(\exists m',n'.((m'+n'+1)=(m+n))\&(L^{m'}[2]\,\tau\otimes L^{n'}\tau))$

$Dq \triangleq \Lambda.\Lambda.\Lambda.\lambda\ p\ l_1\ l_2.\mathsf{match}\ l_2\ \mathsf{with}\ |nil \mapsto E_1\ |h_2 :: l_2' \mapsto E_2$
$E_1 = \mathsf{bind}\, l_r = M\ [][]\ l_1\ nil \text{ in } \mathsf{match}\ l_r\ \mathsf{with}\ |nil \mapsto -\ |h_r :: l_r' \mapsto E_{1.1}$
$E_{1.1} = \mathsf{release} - = p \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\,\Lambda.\langle\langle nil, l_r'\rangle\rangle$
$E_2 = \mathsf{release} - = p \text{ in } \mathsf{bind} - = \uparrow^1 \text{ in } \mathsf{ret}\,\Lambda.\langle\langle l_1, l_2'\rangle\rangle$

Typing derivation for dequeue Dq

$T_0 = \forall m, n.(m+n>0) \Rightarrow [1]\,\mathbf{1} \multimap L^m([2]\,\tau) \multimap L^n\tau \multimap$
$\mathbb{M}\,0\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_1 = (m+n>0) \Rightarrow [1]\,\mathbf{1} \multimap L^m([2]\,\tau) \multimap L^n\tau \multimap$
$\mathbb{M}\,0\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_2 = [1]\,\mathbf{1} \multimap L^m([2]\,\tau) \multimap L^n\tau \multimap \mathbb{M}\,0\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_{2.1} = L^m([2]\,\tau)$

$T_3 = L^n\tau \multimap \mathbb{M}\,0\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_{3.1} = L^n\tau$

$T_4 = \mathbb{M}\,0\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_{4.1} = \mathbb{M}\,1\,(\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_5 = (\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))$

$T_{5.1} = (\exists m', n'.(m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau))[m/m'][i/n']$

$T_{5.2} = (L^m[2]\,\tau \otimes L^n\tau)$

$T_6 = (m'+n'+1) = (m+n)\&(L^{m'}[2]\,\tau \otimes L^{n'}\tau)[0/m'][i/n']$

$T_7 = (L^0[2]\,\tau \otimes L^i\tau)$

$E_0 = \Lambda.\Lambda.\Lambda.\lambda\; l_1\; l_2.\mathsf{match}\; l_2\; \mathsf{with}\; |nil \mapsto E_1\; |h_2 :: l_2' \mapsto E_2$

$E_{0.1} = \lambda\; p\; l_1\; l_2.\mathsf{match}\; l_2\; \mathsf{with}\; |nil \mapsto E_1\; |h_2 :: l_2' \mapsto E_2$

$E_{0.2} = \mathsf{match}\; l_2\; \mathsf{with}\; |nil \mapsto E_1\; |h_2 :: l_2' \mapsto E_2$

$E_1 = \mathsf{bind}\, l_r = M\;[][]\; l_1\; nil\; \mathsf{in\; match}\; l_r\; \mathsf{with}\; |nil \mapsto -\; |h_r :: l_r' \mapsto E_{1.1}$

$E_{1.1} = \mathsf{release} - = p\; \mathsf{in}\; \mathsf{bind}\, x = \uparrow^1 \mathsf{in}\; \Lambda.\,\mathsf{ret}\langle\langle nil, l_r'\rangle\rangle$

$E_2 = \mathsf{release} - = p\; \mathsf{in}\; \mathsf{bind}\, x = \uparrow^1 \mathsf{in}\; \Lambda.\,\mathsf{ret}\langle\langle l_1, l_2'\rangle\rangle$

D1.3:

$$\frac{}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^i\tau,l_1:T_{2.1} \vdash \langle\langle l_1,l_2'\rangle\rangle : T_{5.2}}$$

D1.2:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{.;m,n,i;(j+1=n),(m+n)>0 \models (m+i+1)=(m+n)} \quad D1.3}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^i\tau,l_1:T_{2.1} \vdash \Lambda.\langle\langle l_1,l_2'\rangle\rangle : T_{5.1}}}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^i\tau,l_1:T_{2.1} \vdash \Lambda.\langle\langle l_1,l_2'\rangle\rangle : T_5}}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^i\tau,l_1:T_{2.1} \vdash \mathsf{ret}\,\Lambda.\langle\langle l_1,l_2'\rangle\rangle : T_4}$$

D1.1:

$$\cfrac{\cfrac{}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^j\tau \vdash \uparrow^1 : \mathbb{M}\,1\,\mathbf{1}} \quad D1.2}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^j\tau,l_1:T_{2.1} \vdash \mathsf{bind} - = \uparrow^1 \mathsf{in\; ret}\,\Lambda.\langle\langle l_1,l_2'\rangle\rangle : T_{4.1}}$$

D1:

$$\cfrac{\cfrac{\cfrac{}{.;m,n,i;(j+1=n),(m+n)>0;.;p:[1]\,\mathbf{1} \vdash p:[1]\,\mathbf{1}} \quad D1.1}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^j\tau,l_1:T_{2.1} \vdash \mathsf{release} - = p\; \mathsf{in}\; \mathsf{bind} - = \uparrow^1 \mathsf{in\; ret}\,\Lambda.\langle\langle l_1,l_2'\rangle\rangle : T_4}}{.;m,n,i;(j+1=n),(m+n)>0;.;h_2:\tau,l_2':L^j\tau,l_1:T_{2.1} \vdash E_2 : T_4}$$

D0.05:

$$\frac{}{.;m,n,i;(n=0),(i+1=m),(m+n)>0,(0+u+1)=(m+n);.;h_r:\tau,l_r':L^i\tau \vdash \langle\langle nil,l_r'\rangle\rangle : T_7}$$

D0.04:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{.;m,n,i;(n=0),(i+1=m),(m+n)>0 \models (0+i+1)=(m+n)} \quad D0.05}{.;m,n,i;(n=0),(i+1=m),(m+n)>0;.;h_r:\tau,l_r':L^i\tau \vdash \Lambda.\langle\langle nil,l_r'\rangle\rangle : T_6}}{.;m,n,i;(n=0),(i+1=m),(m+n)>0;.;h_r:\tau,l_r':L^i\tau \vdash \Lambda.\langle\langle nil,l_r'\rangle\rangle : T_5}}{.;m,n,i;(n=0),(i+1=m),(m+n)>0;.;h_r:\tau,l_r':L^i\tau \vdash \mathsf{ret}\,\Lambda.\langle\langle nil,l_r'\rangle\rangle : T_4}$$

D0.03:

$$\cfrac{\cfrac{}{.;m,n,i;(n=0),(i+1=m),(m+n)>0;.;h_r:\tau,l_r':L^i\tau \vdash \uparrow^1 : \mathbb{M}\,1\,\mathbf{1}} \quad D0.04}{.;m,n,i;(n=0),(i+1=m),(m+n)>0;.;h_r:\tau,l_r':L^i\tau \vdash \mathsf{bind} - = \uparrow^1 \mathsf{in\; ret}\,\Lambda.\langle\langle nil,l_r'\rangle\rangle : T_{4.1}}$$

D0.02:

$$\frac{\dfrac{}{.; m, n; (n = 0), (i + 1 = m), (m + n) > 0; .; p : [1]\,\mathbf{1} \vdash p : [1]\,\mathbf{1}} \quad D0.03}{.; m, n; (n = 0), (i + 1 = m), (m + n) > 0; .; h_r : \tau, l'_r : L^i\tau, p : [1]\,\mathbf{1} \vdash E_{1.1} : T_4}$$

D0.01:

$$\frac{}{.; m, n; (n = 0), (m + n) > 0; .; . \vdash \mathsf{fix}\ x.x : T_4}$$

D0.0:

$$\frac{\dfrac{}{.; m, n; (n = 0), (m + n) > 0; .; l_r : L^m\tau \vdash l_r : L^m\tau} \quad D0.01 \quad D0.02}{.; m, n; (n = 0), (m + n) > 0; .; l_r : L^m\tau, p : [1]\,\mathbf{1} \vdash \mathsf{match}\ l_r\ \mathsf{with}\ |nil \mapsto -\ |h_r :: l'_r \mapsto E_{1.1} : T_4}$$

D0:

$$\frac{\dfrac{}{.; m, n; (n = 0), (m + n) > 0; .; l_1 : T_{2.1} \vdash M\ [][]\ l_1\ nil : \mathbb{M}\,0\,(L^m\tau)} \quad D0.0}{.; m, n; (n = 0), (m + n) > 0; .; l_1 : T_{2.1}, p : [1]\,\mathbf{1} \vdash E_1 : T_4}$$

Main derivation:

$$\cfrac{\cfrac{\cfrac{\dfrac{}{.; m, n; (m + n) > 0; .; l_2 : T_{3.1} \vdash l_2 : T_{3.1}} \quad D0 \quad D1}{.; m, n; (m + n) > 0; .; l_1 : T_{2.1}, l_2 : T_{3.1}, p : [1]\,\mathbf{1} \vdash E_{0.2} : T_0}}{.; m, n; (m + n) > 0; .; . \vdash E_{0.1} : T_0}}{.; .; .; .; . \vdash E_0 : T_0}$$

$Move : \forall m, n.L^m([2]\,\tau) \multimap L^n\tau \multimap \mathbb{M}\,0\,(L^{m+n}\tau)$
$Move \triangleq \mathsf{fix} M\Lambda.\Lambda.\lambda\ l_1\ l_2.\mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_1\ |h_1 :: l'_1 \mapsto E_2$
$E_1 = \mathsf{ret}(l_2)$
$E_2 = \mathsf{release} - = h\ \mathsf{in}\ \mathsf{bind} - = \uparrow^2\ \mathsf{in}\ M\ [][]\ l'_1\ (h_1 :: l_2)$

Typing derivation for $Move$

$T_0 = \forall m, n.L^m([2]\,\tau) \multimap L^n\tau \multimap \mathbb{M}\,0\,(L^{m+n}\tau)$
$T_1 = L^m([2]\,\tau) \multimap L^n\tau \multimap \mathbb{M}\,0\,(L^{m+n}\tau)$
$T_{1.1} = L^m([2]\,\tau)$
$T_2 = L^n\tau \multimap \mathbb{M}\,0\,(L^{m+n}\tau)$
$T_{2.1} = L^n\tau$
$T_3 = \mathbb{M}\,0\,(L^{m+n}\tau)$
$T_4 = \mathbb{M}\,0\,(L^{i+n+1}\tau)$
$T_5 = \mathbb{M}\,2\,(L^{m+n}\tau)$

$E_0 = \mathsf{fix}\ M.\Lambda.\Lambda.\lambda\ l_1\ l_2.\mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_1\ |h_1 :: l'_1 \mapsto E_2$
$E_{0.0} = \Lambda.\Lambda.\lambda\ l_1\ l_2.\mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_1\ |h_1 :: l'_1 \mapsto E_2$
$E_{0.1} = \lambda\ l_1\ l_2.\mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_1\ |h_1 :: l'_1 \mapsto E_2$
$E_{0.2} = \mathsf{match}\ l_1\ \mathsf{with}\ |nil \mapsto E_1\ |h_1 :: l'_1 \mapsto E_2$
$E_1 = \mathsf{ret}(l_2)$
$E_2 = \mathsf{release} - = h\ \mathsf{in}\ \mathsf{bind} - = \uparrow^2\ \mathsf{in}\ M\ [][]\ l'_1\ (h_1 :: l_2)$
$E_{2.1} = \mathsf{bind} - = \uparrow^2\ \mathsf{in}\ M\ [][]\ l'_1\ (h_1 :: l_2)$
$E_{2.2} = M\ [][]\ l'_1\ (h_1 :: l_2)$

D3:

$$\cfrac{\dfrac{}{.; m, n, i; i + 1 = m; M : T_0; l'_1 : L^i[2]\,\tau, l_2 : T_{2.1} \vdash M\ [][]\ l'_1\ (h_1 :: l_2) : T_4}}{.; m, n, i; i + 1 = m; M : T_0; l'_1 : L^i[2]\,\tau, l_2 : T_{2.1} \vdash M\ [][]\ l'_1\ (h_1 :: l_2) : T_3}$$

D2:

$$\frac{\dfrac{}{.; m, n, i; i + 1 = m; M : T_0; . \vdash \uparrow^2 : \mathbb{M}\,2\,\mathbf{1}} \quad D3}{.; m, n, i; i + 1 = m; M : T_0; l'_1 : L^i[2]\,\tau, l_2 : T_{2.1} \vdash E_{2.1} : T_5}$$

D1:

$$\frac{\dfrac{}{.; m, n, i; i + 1 = m; M : T_0; h_1 : [2]\,\tau \vdash h_1 : [2]\,\tau} \quad D2}{.; m, n, i; i + 1 = m; M : T_0; h_1 : [2]\,\tau, l'_1 : L^i[2]\,\tau, l_2 : T_{2.1} \vdash E_2 : T_3}$$

D0:

$$\frac{}{.;m,n;.;M:T_0;l_2:T_{2.1}\vdash E_1:T_3}$$

Main derivation:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{.;m,n;.;M:T_0;l_1:T_{1.1}\vdash l_1:T_{1.1}}\quad D0\quad D1}{.;m,n;.;M:T_0;l_1:T_{1.1},l_2:T_{2.1}\vdash E_{0.2}:T_1}}{.;m,n;.;M:T_0;.\vdash E_{0.2}:T_1}}{.;.;.;M:T_0;.\vdash E_{0.1}:T_{0.0}}}{.;.;.;.;.\vdash E_0:T_0}}{.;.;.;.;.\vdash Move:T_0}$$

C.5 Okasaki's implicit queue

Typing rules for value constructors and case analysis

$$\frac{}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C0}:\mathit{Queue}\ \tau}\ \text{T-C0}\qquad \frac{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash e:\tau}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C1}\ e:\mathit{Queue}\ \tau}\ \text{T-C1}$$

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash e:[1]\,\mathbf{1}\multimap\mathbb{M}\,0\,(\tau\otimes \mathit{Queue}\ (\tau\otimes\tau))}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C2}\ e:\mathit{Queue}\ \tau}\ \text{T-C2}$$

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash e:[0]\,\mathbf{1}\multimap\mathbb{M}\,0\,((\tau\otimes \mathit{Queue}\ (\tau\otimes\tau))\otimes\tau)}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C3}\ e:\mathit{Queue}\ \tau}\ \text{T-C3}$$

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash e:[2]\,\mathbf{1}\multimap\mathbb{M}\,0\,((\tau\otimes\tau)\otimes \mathit{Queue}\ (\tau\otimes\tau))}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C4}\ e:\mathit{Queue}\ \tau}\ \text{T-C4}$$

$$\frac{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash e:[1]\,\mathbf{1}\multimap\mathbb{M}\,0\,(((\tau\otimes\tau)\otimes \mathit{Queue}\ (\tau\otimes\tau))\otimes\tau)}{\Psi;\Theta;\Delta;\Omega;\Gamma\vdash \mathit{C5}\ e:\mathit{Queue}\ \tau}\ \text{T-C5}$$

$$\frac{\begin{array}{c}\Psi;\Theta;\Delta;\Omega;\Gamma_1\vdash e:(\mathit{Queue}\ \tau)\qquad \Psi;\Theta;\Delta;\Omega;\Gamma_2\vdash e_0:\tau'\\ \Psi;\Theta;\Delta;\Omega;\Gamma_2,x:\tau\vdash e_1:\tau'\qquad \Psi;\Theta;\Delta;\Omega;\Gamma_2,x:[1]\,\mathbf{1}\multimap\mathbb{M}\,0\,(\tau\otimes \mathit{Queue}\ (\tau\otimes\tau))\vdash e_2:\tau'\\ \Psi;\Theta;\Delta;\Omega;\Gamma_2,x:[0]\,\mathbf{1}\multimap\mathbb{M}\,0\,((\tau\otimes \mathit{Queue}\ (\tau\otimes\tau))\otimes\tau)\vdash e_3:\tau'\\ \Psi;\Theta;\Delta;\Omega;\Gamma_2,x:[2]\,\mathbf{1}\multimap\mathbb{M}\,0\,((\tau\otimes\tau)\otimes \mathit{Queue}\ (\tau\otimes\tau))\vdash e_4:\tau'\\ \Psi;\Theta;\Delta;\Omega;\Gamma_2,x:[1]\,\mathbf{1}\multimap\mathbb{M}\,0\,(((\tau\otimes\tau)\otimes \mathit{Queue}\ (\tau\otimes\tau))\otimes\tau)\vdash e_5:\tau'\end{array}}{\Psi;\Theta;\Delta;\Omega;\Gamma_1\oplus\Gamma_2\vdash\quad \mathsf{case}\ e\ \mathsf{of}\ |\mathit{C0}\mapsto e_0\ |\mathit{C1}\ x\mapsto e_1\ |\mathit{C2}\ x\mapsto e_2\ |\mathit{C3}x\mapsto e_3\ |\mathit{C4}x\mapsto e_4\ |\mathit{C5}x\mapsto e_5:\tau'}\ \text{T-caseIQ}$$

$\mathit{snoc}:[2]\,\mathbf{1}\multimap\forall\alpha.\mathit{Queue}\ \alpha\multimap\alpha\multimap\mathbb{M}\,0\,\mathit{Queue}\ \alpha$
$\mathsf{fix}\ \mathit{snoc}.\lambda p.\Lambda.\lambda\ q\ a.$
$-=\mathsf{release}\,p\ \mathsf{in}\ -=\uparrow^1;\ \mathsf{ret}$
$\quad\mathsf{case}\ q\ \mathsf{of}$
$\quad\quad|\mathit{C0}\mapsto\mathsf{ret}\ \mathit{C1}\ \ a$

$\quad\quad|\mathit{C1}\ \ x\mapsto\mathsf{ret}\ \mathit{C4}\ \ (\lambda p''.\,\mathsf{ret}\langle\!\langle\langle\!\langle x,a\rangle\!\rangle,\mathit{C0}\rangle\!\rangle)$

$\quad\quad|\mathit{C2}\ \ x\mapsto$
$\quad\quad\quad\mathsf{bind}\,p'=\mathsf{store}()\ \mathsf{in}$
$\quad\quad\quad\quad\mathsf{bind}\,x'=x\ \ p'\ \mathsf{in}$
$\quad\quad\quad\quad\quad\mathsf{let}\langle\!\langle f,m\rangle\!\rangle=x'\ \mathsf{in}$
$\quad\quad\quad\quad\quad\quad\mathsf{ret}(\mathit{C3}\ \ (\lambda p''.\langle\!\langle\langle\!\langle f,m\rangle\!\rangle,a\rangle\!\rangle)$

$\quad\quad|\mathit{C3}\ \ x\mapsto$
$\quad\quad\quad\mathsf{bind}\,p'=\mathsf{store}()\ \mathsf{in}$
$\quad\quad\quad\quad\mathsf{bind}\,x'=x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\!\langle fm,r\rangle\!\rangle=x'\ \mathsf{in}$
$\quad\quad\quad\quad\quad\mathsf{let}\langle\!\langle f,m\rangle\!\rangle=fm\ \mathsf{in}\ \ \mathsf{bind}\,p_o=\mathsf{store}()\ \mathsf{in}$
$\quad\quad\quad\quad\quad\quad\mathsf{ret}\ \mathit{C2}\ \ (\lambda p''.$
$\quad\quad\quad\quad\quad\quad\quad\quad -=\mathsf{release}\,p_o\ \mathsf{in}\ -=\mathsf{release}\,p''\ \mathsf{in}\ \mathsf{bind}\,p'''=\mathsf{store}()\ \mathsf{in}$
$\quad\quad\quad\quad\quad\quad\quad\quad \mathsf{bind}\,m'=\mathit{snoc}\ \ p'''\ \ m\ \ (r,a))\ \mathsf{in}\ \mathsf{ret}\langle\!\langle f,m'\rangle\!\rangle$

```
|C4  x ↦
  bind p′ = store() in
    ret C5  (λp″.
                    − = release p′ in − = release p″ in
                    bind p‴ = store() in let⟪f, m⟫ = x  p‴ in
                    ret⟪⟪f, m⟫, a⟫

|C5  x ↦
  bind p′ = store() in
    bind x′ = x  p′ in
      let⟪fm, r⟫ = x′ in let⟪f, m⟫ = fm in
        ret(C4  (λp″.
                    bind m′ = snoc  p″  m in ret⟪f, m′⟫)
```

Listing 4: snoc function

$E_{0.0} = - = \mathsf{release}\, p\ \mathsf{in} E_{0.1}$
$E_{0.1} = - = \uparrow^1; E_{0.2}$
$E_{0.2} = \mathsf{case}\ q\ \mathsf{of} | \mathit{C0} \mapsto E_0 | \mathit{C1}\ x \mapsto E_1 | \mathit{C2}\ x \mapsto E_2 | \mathit{C3}\ x \mapsto E_3 | \mathit{C4}\ x \mapsto E_4 | \mathit{C5}\ x \mapsto E_5$
$E_0 = \mathsf{ret}(\mathit{C1}\ a)$
$E_1 = \mathsf{ret}\ \mathit{C4}\ (\lambda p''. \mathsf{ret}\langle\langle\langle\langle x, a\rangle\rangle, \mathit{C0}\rangle\rangle)$
$E_2 = \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ E_{2.1}$
$E_{2.1} = \mathsf{bind}\, x' = x\ p'\ \mathsf{in}\ E_{2.2}$
$E_{2.2} = \mathsf{let}\langle\langle f, m\rangle\rangle = x'\ \mathsf{in}\ E_{2.3}$
$E_{2.3} = \mathsf{ret}(\mathit{C3}\ (\lambda p''.\langle\langle\langle\langle f, m\rangle\rangle, a\rangle\rangle))$
$E_3 = \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ E_{3.1}$
$E_{3.1} = \mathsf{bind}\, x' = x\ p'\ \mathsf{in}\ E_{3.2}$
$E_{3.2} = \mathsf{let}\langle\langle fm, r\rangle\rangle = x'\ \mathsf{in}\ E_{3.3}$
$E_{3.3} = \mathsf{let}\langle\langle f, m\rangle\rangle = fm\ \mathsf{in}\ E_{3.31}$
$E_{3.31} = \mathsf{bind}\, p_o = \mathsf{store}()\ \mathsf{in}\ E_{3.4}$
$E_{3.4} = \mathsf{ret}\ \mathit{C2}\ (\lambda p''.E_{3.41})$
$E_{3.41} = - = \mathsf{release}\, p_o\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in}\ \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.42}$
$E_{3.42} = \mathsf{bind}\, m' = \mathit{snoc}\ p'''\ m\ (r, a)\ \mathsf{in}\ \mathsf{ret}\langle\langle f, m'\rangle\rangle$
$E_4 = \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ E_{4.1}$
$E_{4.1} = \mathsf{ret}\ \mathit{C5}\ (\lambda p''.E_{4.11})$
$E_{4.11} = - = \mathsf{release}\, p'\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in}\ E_{4.12}$
$E_{4.12} = \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ \mathsf{let}\langle\langle f, m\rangle\rangle = x\ p'''\ \mathsf{in}\ E_{4.13}$
$E_{4.13} = \mathsf{ret}\langle\langle\langle\langle f, m\rangle\rangle, a\rangle\rangle$
$E_5 = \mathsf{bind}\, p' = \mathsf{store}()\ \mathsf{in}\ E_{5.1}$
$E_{5.1} = \mathsf{bind}\, x' = x\ p'\ \mathsf{in}\ E_{5.2}$
$E_{5.2} = \mathsf{let}\langle\langle fm, r\rangle\rangle = x'\ \mathsf{in}\ E_{5.3}$
$E_{5.3} = \mathsf{let}\langle\langle f, m\rangle\rangle = fm\ \mathsf{in}\ E_{5.4}$
$E_{5.4} = \mathsf{ret}(\mathit{C4}\ (\lambda p''.\ \mathsf{bind}\, m' = \mathit{snoc}\ p''\ m\ \mathsf{in}\ \mathsf{ret}\langle\langle f, m'\rangle\rangle))$

$T_{0.0} = [2]\, \mathbf{1} \multimap \forall\alpha. \mathit{Queue}\ \alpha \multimap \alpha \multimap \mathbb{M}\, 0\ \mathit{Queue}\ \alpha$
$T_0 = \mathbb{M}\, 0\ \mathit{Queue}\ \alpha$
$T_1 = \mathbb{M}\, 1\ \mathit{Queue}\ \alpha$
$T_2 = \mathbb{M}\, 2\ \mathit{Queue}\ \alpha$
$T_3 = \mathbb{M}\, 0\, (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{3.1} = (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{3.2} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_4 = \mathbb{M}\, 0\, (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha)$
$T_{4.1} = (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha)$
$T_{4.2} = \alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{4.3} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_5 = [2]\, \mathbf{1} \multimap \mathbb{M}\, 0\, (\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{5.1} = \mathbb{M}\, 0\, (\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{5.2} = (\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{5.3} = (\alpha \otimes \alpha)$
$T_{5.4} = \mathit{Queue}\ (\alpha \otimes \alpha)$

$T_6 = [1]\,\mathbf{1} \multimap \mathbb{M}\,0\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha)$
$T_{6.1} = \mathbb{M}\,0\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha)$
$T_{6.2} = ((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha)$
$T_{6.3} = (\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{6.4} = (\alpha \otimes \alpha)$
$T_{6.5} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_7 = \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{7.1} = \mathbb{M}\,1\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{7.2} = \mathbb{M}\,2\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_8 = \mathbb{M}\,0\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{8.1} = ((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha$
$T_9 = \mathbb{M}\,0\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{9.1} = ((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$

D5.5:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : T_{6.4}, m : T_{6.5}, p'' : [2]\,\mathbf{1}, m' : \mathit{Queue}\ (\alpha \otimes \alpha) \vdash \langle\!\langle f, m' \rangle\!\rangle : T_{9.1}}}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : T_{6.4}, m : T_{6.5}, p'' : [2]\,\mathbf{1}, m' : \mathit{Queue}\ (\alpha \otimes \alpha) \vdash \mathsf{ret}\langle\!\langle f, m' \rangle\!\rangle : T_9}$$

D5.4:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, m : T_{6.5}, p'' : [2]\,\mathbf{1} \vdash S\ p''\ []\ m\ \langle\!\langle r, a \rangle\!\rangle : \mathbb{M}\,0\,(\mathit{Queue}\ (\alpha \otimes \alpha))} \quad D5.5}{\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, f : T_{6.4}, m : T_{6.5}, p'' : [2]\,\mathbf{1} \vdash \mathsf{bind}\, m' = S\ p''\ []\ m\ \langle\!\langle r, a \rangle\!\rangle \text{ in } \mathsf{ret}\langle\!\langle f, m' \rangle\!\rangle : T_9}}{\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, f : T_{6.4}, m : T_{6.5} \vdash (\lambda p''.\, \mathsf{bind}\, m' = S\ p''\ []\ m\ \langle\!\langle r, a \rangle\!\rangle \text{ in } \mathsf{ret}\langle\!\langle f, m' \rangle\!\rangle)) : [2]\,\mathbf{1} \multimap T_9}}{\begin{array}{c}\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, f : T_{6.4}, m : T_{6.5} \vdash \\ (C4\ (\lambda p''.\, \mathsf{bind}\, m' = S\ p''\ []\ m\ \langle\!\langle r, a \rangle\!\rangle \text{ in } \mathsf{ret}\langle\!\langle f, m' \rangle\!\rangle)) : \mathit{Queue}\ \alpha\end{array}}}{\dfrac{\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, f : T_{6.4}, m : T_{6.5} \vdash \mathsf{ret}(C4\ (\lambda p''.\, \mathsf{bind}\, m' = S\ p''\ []\ m\ \langle\!\langle r, a \rangle\!\rangle \text{ in } \mathsf{ret}\langle\!\langle f, m' \rangle\!\rangle)) : T_0}{\alpha; .; .; S : T_{0.0}; r : \alpha, a : \alpha, f : T_{6.4}, m : T_{6.5} \vdash E_{5.4} : T_0}}$$

D5.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; fm : T_{6.3} \vdash fm : T_{6.3}} \quad D5.4}{\alpha; .; .; S : T_{0.0}; a : \alpha, fm : T_{6.3}, r : \alpha \vdash \mathsf{let}\langle\!\langle f, m \rangle\!\rangle = fm \text{ in } E_{5.4} : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, fm : T_{6.3}, r : \alpha \vdash E_{5.3} : T_0}$$

D5.2:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x' : T_{6.2} \vdash x' : T_{6.2}} \quad D5.3}{\alpha; .; .; S : T_{0.0}; a : \alpha, x' : T_{6.2} \vdash \mathsf{let}\langle\!\langle fm, r \rangle\!\rangle = x' \text{ in } E_{5.3} : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x' : T_{6.2} \vdash E_{5.2} : T_0}$$

D5.1:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x : T_6, p' : [1]\,\mathbf{1} \vdash x\ p' : T_{6.1}} \quad D5.2}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_6, p' : [1]\,\mathbf{1} \vdash \mathsf{bind}\, x' = x\ p' \text{ in } E_{5.2} : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_6, p' : [1]\,\mathbf{1} \vdash E_{5.1} : T_0}$$

D5:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,1\,([1]\,\mathbf{1})} \quad D5.1}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_6 \vdash E_5 : T_1}$$

D4.5:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, f : T_{5.3}, m : T_{5.4} \vdash \langle\!\langle \langle\!\langle f, m \rangle\!\rangle, a \rangle\!\rangle : T_{8.1}}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, f : T_{5.3}, m : T_{5.4} \vdash \mathsf{ret}\langle\!\langle \langle\!\langle f, m \rangle\!\rangle, a \rangle\!\rangle : T_8}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, f : T_{5.3}, m : T_{5.4} \vdash E_{4.13} : T_8}$$

D4.4:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x : T_5, p''' : [2]\,\mathbf{1} \vdash x\ p''' : T_{5.1}} \quad D4.5}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, p''' : [2]\,\mathbf{1} \vdash \mathsf{let}\langle\!\langle f, m \rangle\!\rangle = x\ p''' \text{ in } E_{4.13} : T_8}$$

D4.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5 \vdash \mathsf{store}() : \mathbb{M}\,2\,([2]\,\mathbf{1})} \qquad \mathit{D4.4}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5 \vdash \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = x\ p'''\ \mathsf{in}\ E_{4.13} : T_{8.2}}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5 \vdash E_{4.12} : T_{8.2}}$$

D4.2:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; p'' : [1]\,\mathbf{1} \vdash p'' : [1]\,\mathbf{1}} \qquad \mathit{D4.3}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, p'' :\vdash - = \mathsf{release}\, p''\ \mathsf{in}\ E_{4.12} : T_{8.1}}$$

D4.11:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; p' : [1]\,\mathbf{1} \vdash p' : [1]\,\mathbf{1}} \qquad \mathit{D4.2}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : T_5, p' : [1]\,\mathbf{1}, p'' : [1]\,\mathbf{1} \vdash - = \mathsf{release}\, p'\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in}\ E_{4.12} : T_8}$$

D4.1:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathit{D4.11}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha), p' : [1]\,\mathbf{1}, p'' : [1]\,\mathbf{1} \vdash \quad E_{4.11} : T_8}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha), p' : [1]\,\mathbf{1} \vdash \quad (\lambda p''.E_{4.11}) : [1]\,\mathbf{1} \multimap T_8}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha), p' : [1]\,\mathbf{1} \vdash \mathit{C5}\ (\lambda p''.E_{4.11}) : \mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha), p' : [1]\,\mathbf{1} \vdash \mathsf{ret}\ \mathit{C5}\ (\lambda p''.E_{4.11}) : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha), p' : [1]\,\mathbf{1} \vdash E_{4.1} : T_0}$$

D4:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,1\,([1]\,\mathbf{1})} \qquad \mathit{D4.1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \vdash E_4 : T_1}$$

D3.43:

$$\dfrac{}{\alpha; .; .; S : T_{0.0}; f : \alpha, m' : \mathit{Queue}\ (\alpha \otimes \alpha) \vdash \mathsf{ret}\langle\!\langle f, m'\rangle\!\rangle : T_7}$$

D3.42:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; m : T_{4.3}, r : \alpha, a : \alpha, p''' : [2]\,\mathbf{1} \vdash S\ p'''\ []\ m\ (r, a) : \mathbb{M}\,0\,(\mathit{Queue}\ (\alpha \otimes \alpha))} \qquad \mathit{D3.43}}{\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p''' : [2]\,\mathbf{1} \vdash \quad \mathsf{bind}\, m' = S\ p'''\ []\ m\ (r, a)\ \mathsf{in}\ \mathsf{ret}\langle\!\langle f, m'\rangle\!\rangle : T_7}}{\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p''' : [2]\,\mathbf{1} \vdash E_{3.42} : T_7}$$

D3.41:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,2\,([2]\,\mathbf{1})} \qquad \mathit{D3.42}}{\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha \vdash \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.42} : T_{7.2}}$$

D3.401:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; p'' : [1]\,\mathbf{1} \vdash p'' : [1]\,\mathbf{1}} \qquad \mathit{D3.41}}{\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p'' : [1]\,\mathbf{1} \vdash - = \mathsf{release}\, p''\ \mathsf{in}\ \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.42} : T_{7.1}}$$

D3.40:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; p_o : [1]\,\mathbf{1} \vdash p_o : [1]\,\mathbf{1}} \qquad \mathit{D3.401}}{\begin{array}{c}\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1}, p'' : [1]\,\mathbf{1} \vdash \\ - = \mathsf{release}\, p_o\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in}\ \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.42} : T_7\end{array}}}{\begin{array}{c}\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1} \vdash \\ \lambda p''.- = \mathsf{release}\, p_o\ \mathsf{in}\ - = \mathsf{release}\, p''\ \mathsf{in}\ \mathsf{bind}\, p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.42} : [1]\,\mathbf{1} \multimap T_7\end{array}}}{\alpha; .; .; S : T_{0.0}; f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1} \vdash (\lambda p''.E_{3.41}) : [1]\,\mathbf{1} \multimap T_7}$$

D3.4:

$$\dfrac{\dfrac{\dfrac{\mathit{D3.40}}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1} \vdash \mathit{C2}\ (\lambda p''.E_{3.41}) : \mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1} \vdash \mathsf{ret}\ \mathit{C2}\ (\lambda p''.E_{3.41}) : T_1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha, a : \alpha, p_o : [1]\,\mathbf{1} \vdash E_{3.4} : T_1}$$

D3.31:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha \vdash \mathsf{store}() : \mathbb{M}\,1\,[1]\,\mathbf{1}} \qquad D3.4}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha \vdash \mathsf{bind}\, p_o = \mathsf{store}()\ \mathsf{in}\ E_{3.4} : T_1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, f : \alpha, m : T_{4.3}, r : \alpha \vdash E_{3.31} : T_1}$$

D3.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; fm : T_{4.2} \vdash fm : T_{4.2}} \qquad D3.4}{\alpha; .; .; S : T_{0.0}; a : \alpha, fm : T_{4.2}, r : \alpha \vdash \mathsf{let}\langle\langle f, m\rangle\rangle = fm\ \mathsf{in}\ E_{3.31} : T_1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, fm : T_{4.2}, r : \alpha \vdash E_{3.3} : T_1}$$

D3.2:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x' : T_{4.1} \vdash x' : T_{4.1}} \qquad D3.3}{\alpha; .; .; S : T_{0.0}; . \vdash \mathsf{let}\langle\langle fm, r\rangle\rangle = x'\ \mathsf{in}\ E_{3.3} : T_1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x' : T_{4.1} \vdash E_{3.2} : T_1}$$

D3.1:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x : [0]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha), p' : [0]\,\mathbf{1} \vdash x\ p' : T_4} \qquad D3.2}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [0]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha) \vdash E_{3.1} : T_1}$$

D3:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; \vdash \mathsf{store}() : \mathbb{M}\,0\,([0]\,\mathbf{1})} \qquad D3.1}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : [0]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha) \otimes \alpha) \vdash E_3 : T_1}$$

D2.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; q : \mathit{Queue}\ \alpha, a : \alpha, f : \alpha, m : T_{3.2} \vdash (\mathit{C3}\ (\lambda p''.\langle\langle\langle\langle f, m\rangle\rangle, a\rangle\rangle)) : \mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; q : \mathit{Queue}\ \alpha, a : \alpha, f : \alpha, m : T_{3.2} \vdash \mathsf{ret}(\mathit{C3}\ (\lambda p''.\langle\langle\langle\langle f, m\rangle\rangle, a\rangle\rangle)) : T_0}}{\alpha; .; .; S : T_{0.0}; q : \mathit{Queue}\ \alpha, a : \alpha, f : \alpha, m : T_{3.2} \vdash E_{2.3} : T_0}$$

D2.2:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x' : T_{3.1} \vdash x' : T_{3.1}} \qquad D2.3}{\alpha; .; .; S : T_{0.0}; a : \alpha, x' : T_{3.1} \vdash \mathsf{let}\langle\langle f, m\rangle\rangle = x'\ \mathsf{in}\ E_{2.3} : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x' : T_{3.1} \vdash E_{2.2} : T_0}$$

D2.1:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; x : ([1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))), p' : [1]\,\mathbf{1} \vdash x\ p' : T_3} \qquad D2.2}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : ([1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))), p' : [1]\,\mathbf{1} \vdash E_{2.1} : T_0}$$

D2:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,1\,([1]\,\mathbf{1})} \qquad D2.1}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : ([1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))) \vdash E_2 : T_1}$$

D1:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : \alpha \vdash \mathit{C4}\ (\lambda p''.\,\mathsf{ret}\langle\langle\langle\langle x, a\rangle\rangle, \mathit{C0}\rangle\rangle) : \mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : \alpha \vdash \mathsf{ret}\ \mathit{C4}\ (\lambda p''.\,\mathsf{ret}\langle\langle\langle\langle x, a\rangle\rangle, \mathit{C0}\rangle\rangle) : T_0}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : \alpha \vdash \mathsf{ret}\ \mathit{C4}\ (\lambda p''.\,\mathsf{ret}\langle\langle\langle\langle x, a\rangle\rangle, \mathit{C0}\rangle\rangle) : T_1}}{\alpha; .; .; S : T_{0.0}; a : \alpha, x : \alpha \vdash E_1 : T_1}$$

D0:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; a : \alpha \vdash \mathit{C1}\ a : \mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; a : \alpha \vdash \mathsf{ret}(\mathit{C1}\ a) : \mathbb{M}\,1\,\mathit{Queue}\ \alpha}}{\alpha; .; .; S : T_{0.0}; a : \alpha \vdash E_0 : T_1}$$

D0.2:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; q : \mathit{Queue}\ \alpha \vdash q : \mathit{Queue}\ \alpha} \qquad D0 \qquad D1 \qquad D2 \qquad D3 \qquad D4 \qquad D5}{\alpha; .; .; S : T_{0.0}; q : \mathit{Queue}\ \alpha, a : \alpha \vdash E_{0.2} : T_1}$$

D0.1:

$$\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; . \vdash \uparrow^1 : \mathbb{M}\,1\,\mathbf{1}} \qquad D0.2}{\alpha; .; .; S : T_{0.0}; q : Queue\ \alpha, a : \alpha \vdash E_{0.1} : T_2}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; S : T_{0.0}; p : [2]\,\mathbf{1} \vdash p : [2]\,\mathbf{1}} \qquad D0.1}{\alpha; .; .; S : T_{0.0}; p : [2]\,\mathbf{1}, q : Queue\ \alpha, a : \alpha \vdash E_{0.0} : T_0}}{.; .; .; .; . \vdash \mathsf{fix} f.\lambda p.\Lambda.\lambda q.\lambda a.E_{0.0} : T_{0.0}}$$

$head : [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,\alpha$
$head \triangleq \lambda p.\Lambda.\lambda\ \ q.$
$\quad \mathsf{bind}\, ht = headTail\ \ p\ \ []\ \ q\ \mathsf{in}\ \mathsf{ret}\, \mathsf{fst}(ht)$

Listing 5: head function

$E_0 = \mathsf{bind}\, ht = headTail\ p\ []\ q\ \mathsf{in}\ E_1$
$E_1 = \mathsf{ret}(\mathsf{fst}(ht))$

$T_0 = [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,\alpha$

D0:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash \mathsf{fst}(ht) : \alpha}}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash \mathsf{ret}(\mathsf{fst}(ht)) : \mathbb{M}\,0\,\alpha}}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash E_1 : \mathbb{M}\,0\,\alpha}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; .; q : Queue\ \alpha \vdash headTail\ p\ []\ q : \mathbb{M}\,0\,(\alpha \otimes Queue\ \alpha)} \qquad D0}{\alpha; .; .; .; q : Queue\ \alpha \vdash \mathsf{bind}\, ht = headTail\ p\ []\ q\ \mathsf{in}\ E_1 : \mathbb{M}\,0\,\alpha}}{\alpha; .; .; .; p : [3]\,\mathbf{1}, q : Queue\ \alpha \vdash E_0 : \mathbb{M}\,0\,\alpha}}{.; .; .; .; . \vdash \lambda p.\Lambda.\lambda q.E_0 : T_0}$$

$tail : [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,(Queue\ \ \alpha)$
$tail \triangleq \lambda p.\Lambda.\lambda\ \ q.$
$\quad \mathsf{bind}\, ht = headTail\ \ p\ \ []\ \ q\ \mathsf{in}\ \mathsf{ret}\, \mathsf{snd}(ht)$

Listing 6: tail function

$E_0 = \mathsf{bind}\, ht = headTail\ p\ []\ q\ \mathsf{in}\ E_1$
$E_1 = \mathsf{ret}(\mathsf{snd}(ht))$

$T_0 = [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,(Queue\ \alpha)$

D0:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash \mathsf{snd}(ht) : Queue\ \alpha}}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash \mathsf{ret}(\mathsf{snd}(ht)) : \mathbb{M}\,0\,(Queue\ \alpha)}}{\alpha; .; .; .; q : Queue\ \alpha, ht : (\alpha \otimes Queue\ \alpha) \vdash E_1 : \mathbb{M}\,0\,(Queue\ \alpha)}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; .; q : Queue\ \alpha \vdash headTail\ p\ []\ q : \mathbb{M}\,0\,(\alpha \otimes Queue\ \alpha)} \qquad D0}{\alpha; .; .; .; q : Queue\ \alpha \vdash \mathsf{bind}\, ht = headTail\ p\ []\ q\ \mathsf{in}\ E_1 : \mathbb{M}\,0\,(Queue\ \alpha)}}{\alpha; .; .; .; p : [3]\,\mathbf{1}, q : Queue\ \alpha \vdash E_0 : \mathbb{M}\,0\,(Queue\ \alpha)}}{.; .; .; .; . \vdash \lambda p.\Lambda.\lambda q.E_0 : T_0}$$

$$
\begin{array}{l}
headTail : [3]\,\mathbf{1} \multimap \forall\alpha.Queue\ \ \alpha \multimap \mathbb{M}\,0\,(\alpha \otimes Queue\ \ \alpha)\\
headTail \triangleq \mathsf{fix}\ \ HT.\lambda p.\Lambda.\lambda\ \ q.\\
- = \mathsf{release}\,p\ \mathsf{in}\ \ - = \uparrow^1;\ \mathsf{ret}\\
\quad \mathsf{case}\ \ q\ \ \mathsf{of}\\
\qquad |\,C0 \mapsto \mathsf{fix}x.x\\
\\
\qquad |\,C1\ \ x \mapsto \mathsf{ret}\langle\!\langle x, C0\rangle\!\rangle\\
\\
\qquad |\,C2\ \ x \mapsto\\
\qquad\quad \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\,p_o = \mathsf{store}()\ \mathsf{in}\\
\qquad\qquad \mathsf{bind}\,x' = x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = x'\ \mathsf{in}\\
\qquad\qquad\quad \mathsf{ret}\langle\!\langle f, (C4\ \ (\lambda p''.- = \mathsf{release}\,p_o\ \mathsf{in}\ - = \mathsf{release}\,p''\ \mathsf{in}\ \mathsf{bind}\,p_r = \mathsf{store}()\ \mathsf{in}\ HT\ \ p_r\ \ []\ \ m))\rangle\!\rangle\\
\\
\qquad |\,C3\ \ x \mapsto\\
\qquad\quad \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\,p_o = \mathsf{store}()\ \mathsf{in}\\
\qquad\qquad \mathsf{bind}\,x' = x\ \ p'\ \mathsf{in}\ \ \mathsf{let}\langle\!\langle fm, r\rangle\!\rangle = x'\ \mathsf{in}\ \ \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = fm\ \mathsf{in}\\
\qquad\qquad \mathsf{ret}\langle\!\langle f, (C5\ \ (\lambda p''.- = \mathsf{release}\,p_o\ \mathsf{in}\ - = \mathsf{release}\,p''\ \mathsf{in}\\
\qquad\qquad\qquad\qquad\qquad\qquad \mathsf{bind}\,p''' = \mathsf{store}()\ \mathsf{in}\ \ \mathsf{bind}\,ht = HT\ \ p'''\ \ []\ \ m\ \mathsf{in}\ \mathsf{ret}\langle\!\langle ht, r\rangle\!\rangle))\rangle\!\rangle\\
\\
\qquad |\,C4\ \ x \mapsto\\
\qquad\quad \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\,x' = x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = x'\ \mathsf{in}\ \mathsf{let}\langle\!\langle f_1, f_2\rangle\!\rangle = f\ \mathsf{in}\\
\qquad\quad \mathsf{ret}\langle\!\langle f_1, C2\ \ (\lambda p''.\,\mathsf{ret}\langle\!\langle f_2, m\rangle\!\rangle)\rangle\!\rangle\\
\\
\qquad |\,C5\ \ x \mapsto\\
\qquad\quad \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ \mathsf{bind}\,x' = x\ \ p'\ \mathsf{in}\ \mathsf{let}\langle\!\langle fm, r\rangle\!\rangle = x'\ \mathsf{in}\ \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = fm\ \mathsf{in}\ \mathsf{let}\langle\!\langle f_1, f_2\rangle\!\rangle = f\ \mathsf{in}\\
\qquad\qquad\quad \mathsf{ret}\langle\!\langle f_1, (C3\ \ (\lambda p''.\,\mathsf{ret}\langle\!\langle\langle\!\langle f_2, m\rangle\!\rangle, r\rangle\!\rangle))\rangle\!\rangle
\end{array}
$$

Listing 7: head and tail function

$$
\begin{array}{l}
E_{0.0} = \mathsf{fix}HT.\lambda p.\Lambda.\lambda\ q.E_{0.1}\\
E_{0.1} = - = \mathsf{release}\,p\ \mathsf{in} - = \uparrow^1; E_{0.2}\\
E_{0.2} = \mathsf{case}\ q\ \mathsf{of}|C0 \mapsto E_0|C1\ x \mapsto E_1|C2\ x \mapsto E_2|C3\ x \mapsto E_3|C4\ x \mapsto E_4|C5\ x \mapsto E_5\\
E_0 = \mathsf{fix}x.x\\
E_1 = \mathsf{ret}\langle\!\langle x, C0\rangle\!\rangle\\
E_2 = \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ E_{2.0}\\
E_{2.0} = \mathsf{bind}\,p_o = \mathsf{store}()\ \mathsf{in}\ E_{2.1}\\
E_{2.1} = \mathsf{bind}\,x' = x\ p'\ \mathsf{in}\ E_{2.11}\\
E_{2.11} = \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = x'\ \mathsf{in}\ E_{2.2}\\
E_{2.2} = \mathsf{ret}\langle\!\langle f, (C4\ \ (\lambda p''.E_{2.3}))\rangle\!\rangle\\
E_{2.3} = - = \mathsf{release}\,p_o\ \mathsf{in}\ E_{2.4}\\
E_{2.4} = - = \mathsf{release}\,p''\ \mathsf{in}\ E_{2.5}\\
E_{2.5} = \mathsf{bind}\,p_r = \mathsf{store}()\ \mathsf{in}\ HT\ p_r\ []\ m\\
E_3 = \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ E_{3.0}\\
E_{3.0} = \mathsf{bind}\,p_o = \mathsf{store}()\ \mathsf{in}\ E_{3.1}\\
E_{3.1} = \mathsf{bind}\,x' = x\ p'\ \mathsf{in}\ E_{3.11}\\
E_{3.11} = \mathsf{let}\langle\!\langle fm, r\rangle\!\rangle = x'\ \mathsf{in}\ E_{3.12}\\
E_{3.12} = \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = fm\ \mathsf{in}\ E_{3.2}\\
E_{3.2} = \mathsf{ret}\langle\!\langle f, E_{3.3}\rangle\!\rangle\\
E_{3.3} = C5\ \ (\lambda p''.E_{3.31})\\
E_{3.4} = - = \mathsf{release}\,p_o\ \mathsf{in}\ E_{3.41}\\
E_{3.41} = \mathsf{release}\,p''\ \mathsf{in}\ E_{3.5}\\
E_{3.5} = \mathsf{bind}\,p''' = \mathsf{store}()\ \mathsf{in}\ E_{3.6}\\
E_{3.6} = \mathsf{bind}\,ht = HT\ p'''\ []\ m\ \mathsf{in}\ \mathsf{ret}\langle\!\langle ht, r\rangle\!\rangle\\
E_4 = \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ E_{4.1}\\
E_{4.1} = \mathsf{bind}\,x' = x\ p'\ \mathsf{in}\ E_{4.2}\\
E_{4.2} = \mathsf{let}\langle\!\langle f, m\rangle\!\rangle = x'\ \mathsf{in}\ E_{4.3}\\
E_{4.3} = \mathsf{let}\langle\!\langle f_1, f_2\rangle\!\rangle = f\ \mathsf{in}\ E_{4.4}\\
E_{4.4} = \mathsf{ret}\langle\!\langle f_1, C2\ (\lambda p''.\,\mathsf{ret}\langle\!\langle f_2, m\rangle\!\rangle)\rangle\!\rangle\\
E_5 = \mathsf{bind}\,p' = \mathsf{store}()\ \mathsf{in}\ E_{5.1}\\
E_{5.1} = \mathsf{bind}\,x' = x\ p'\ \mathsf{in}\ E_{5.2}\\
E_{5.2} = \mathsf{let}\langle\!\langle fm, r\rangle\!\rangle = x'\ \mathsf{in}\ E_{5.3}
\end{array}
$$

$E_{5.3} = \mathsf{let}\langle\langle f, m\rangle\rangle = fm \;\mathsf{in}\; E_{5.4}$
$E_{5.4} = \mathsf{let}\langle\langle f_1, f_2\rangle\rangle = f \;\mathsf{in}\; E_{5.5}$
$E_{5.5} = \mathsf{ret}\langle\langle f_1, (\mathit{C3}\ (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle))\rangle\rangle$

$T_{0.0} = [3]\,\mathbf{1} \multimap \forall\alpha.\mathit{Queue}\ \alpha \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ \alpha)$
$T_{0.2} = [1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.21} = \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.22} = (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.23} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{0.3} = [0]\,\mathbf{1} \multimap \mathbb{M}\,0\,((\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.31} = \mathbb{M}\,0\,((\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.32} = ((\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.33} = (\alpha \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.34} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{0.4} = [2]\,\mathbf{1} \multimap \mathbb{M}\,0\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.41} = \mathbb{M}\,0\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.411} = \mathbb{M}\,1\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.413} = \mathbb{M}\,3\,((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.42} = ((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.43} = (\alpha \otimes \alpha)$
$T_{0.44} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_{0.5} = [1]\,\mathbf{1} \multimap \mathbb{M}\,0\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.51} = \mathbb{M}\,0\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.511} = \mathbb{M}\,1\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.512} = \mathbb{M}\,2\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.513} = \mathbb{M}\,3\,(((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.52} = (((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha)) \otimes \alpha)$
$T_{0.53} = ((\alpha \otimes \alpha) \otimes \mathit{Queue}\ (\alpha \otimes \alpha))$
$T_{0.54} = (\alpha \otimes \alpha)$
$T_{0.55} = \mathit{Queue}\ (\alpha \otimes \alpha)$
$T_0 = \mathbb{M}\,0\,(\alpha \otimes \mathit{Queue}\ \alpha)$
$T_1 = \mathbb{M}\,1\,(\alpha \otimes \mathit{Queue}\ \alpha)$
$T_2 = \mathbb{M}\,2\,(\alpha \otimes \mathit{Queue}\ \alpha)$

D5.51:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.55}, r : \alpha, p'' : [0]\,\mathbf{1} \vdash \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle T_{0.31}}}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.55}, r : \alpha \vdash (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle) : T_{0.3}}}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.55}, r : \alpha \vdash (\mathit{C3}\ (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle)) : \mathit{Queue}\ \alpha}$$

D5.5:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha \vdash f_1 : \alpha} \qquad D5.51}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.55}, r : \alpha \vdash \quad \langle\langle f_1, (\mathit{C3}\ (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle))\rangle\rangle : \alpha \otimes \mathit{Queue}\ \alpha}}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.55}, r : \alpha \vdash \mathsf{ret}\langle\langle f_1, (\mathit{C3}\ (\lambda p''.\, \mathsf{ret}\langle\langle\langle\langle f_2, m\rangle\rangle, r\rangle\rangle))\rangle\rangle : T_1}}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.55}, r : \alpha \vdash E_{5.5} : T_1}$$

D5.4:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f : T_{0.54} \vdash f : T_{0.54}} \qquad D5.5}{\alpha; .; .; HT : T_{0.0}; f : T_{0.54}, m : T_{0.55}, r : \alpha \vdash \mathsf{let}\langle\langle f_1, f_2\rangle\rangle = f \;\mathsf{in}\; E_{5.5} : T_1}}{\alpha; .; .; HT : T_{0.0}; f : T_{0.54}, m : T_{0.55}, r : \alpha \vdash E_{5.4} : T_1}$$

D5.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.53} \vdash fm : T_{0.53}} \qquad D5.4}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.53}, r : \alpha \vdash \mathsf{let}\langle\langle f, m\rangle\rangle = fm \;\mathsf{in}\; E_{5.4} : T_1}}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.53}, r : \alpha \vdash E_{5.3} : T_1}$$

D5.2:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.52} \vdash x' : T_{0.52}} \quad D5.3}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.52} \vdash \mathsf{let}\langle\langle fm, r\rangle\rangle = x' \ \mathsf{in}\ E_{5.3} : T_1}}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.52} \vdash E_{5.2} : T_1}$$

D5.1:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : T_{0.5}, p' : [1]\, \mathbf{1} \vdash x\ p' : T_{0.51}} \quad D5.2}{\alpha; .; .; HT : T_{0.0}; x : T_{0.5}, p' : [1]\, \mathbf{1} \vdash \mathsf{bind}\, x' = x\ p' \ \mathsf{in}\ E_{5.2} : T_1}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.5}, p' : [1]\, \mathbf{1} \vdash E_{5.1} : T_1}$$

D5:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\, 1\, ([1]\, \mathbf{1})} \quad D5.1}{\alpha; .; .; HT : T_{0.0}; x : T_{0.5} \vdash E_5 : T_2}$$

D4.41:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.44}, p'' : [1]\, \mathbf{1} \vdash \mathsf{ret}\langle\langle f_2, m\rangle\rangle : T_{0.21}}}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.44} \vdash (\lambda p''. \mathsf{ret}\langle\langle f_2, m\rangle\rangle) : T_{0.2}}}{\alpha; .; .; HT : T_{0.0}; f_2 : \alpha, m : T_{0.44} \vdash \mathit{C2}\ (\lambda p''. \mathsf{ret}\langle\langle f_2, m\rangle\rangle) : \mathit{Queue}\ \alpha}$$

D4.4:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha \vdash f_1 : \alpha} \quad D4.41}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.44} \vdash \langle\langle f_1, \mathit{C2}\ (\lambda p''. \mathsf{ret}\langle\langle f_2, m\rangle\rangle)\rangle\rangle : \alpha \otimes \mathit{Queue}\ \alpha}}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.44} \vdash \mathsf{ret}\langle\langle f_1, \mathit{C2}\ (\lambda p''. \mathsf{ret}\langle\langle f_2, m\rangle\rangle)\rangle\rangle : T_0}}{\alpha; .; .; HT : T_{0.0}; f_1 : \alpha, f_2 : \alpha, m : T_{0.44} \vdash E_{4.4} : T_0}$$

D4.3:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f : T_{0.43} \vdash f : T_{0.43}} \quad D4.4}{\alpha; .; .; HT : T_{0.0}; f : T_{0.43}, m : T_{0.44} \vdash \mathsf{let}\langle\langle f_1, f_2\rangle\rangle = f \ \mathsf{in}\ E_{4.4} : T_0}}{\alpha; .; .; HT : T_{0.0}; f : T_{0.43}, m : T_{0.44} \vdash E_{4.3} : T_0}$$

D4.2:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.42} \vdash x' : T_{0.42}} \quad D4.3}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.42} \vdash \mathsf{let}\langle\langle f, m\rangle\rangle = x' \ \mathsf{in}\ E_{4.3} : T_0}}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.42} \vdash E_{4.2} : T_0}$$

D4.1:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : T_{0.4}, p' : [2]\, \mathbf{1} \vdash x\ p' : T_{0.41}} \quad D4.2}{\alpha; .; .; HT : T_{0.0}; x : T_{0.4}, p' : [2]\, \mathbf{1} \vdash \mathsf{bind}\, x' = x\ p' \ \mathsf{in}\ E_{4.2} : T_0}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.4}, p' : [2]\, \mathbf{1} \vdash E_{4.1} : T_0}$$

D4:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\, 2\, [2]\, \mathbf{1}} \quad D4.1}{\alpha; .; .; HT : T_{0.0}; x : T_{0.4} \vdash E_4 : T_2}$$

D3.61:

$$\dfrac{}{\alpha; .; .; HT : T_{0.0}; r : \alpha, ht : T_{0.53} \vdash \mathsf{ret}\langle\langle ht, r\rangle\rangle : T_{0.51}}$$

D3.6:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p''' : [3]\, \mathbf{1} \vdash HT\ p'''\ []\ m : \mathbb{M}\, 0\, T_{0.53}} \quad D3.61}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p''' : [3]\, \mathbf{1} \vdash \mathsf{bind}\, ht = HT\ p'''\ []\ m \ \mathsf{in}\ \mathsf{ret}\langle\langle ht, r\rangle\rangle : T_{0.51}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p''' : [3]\, \mathbf{1} \vdash E_{3.6} : T_{0.51}}$$

D3.5:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : [3]\, [3]\, \mathbf{1}} \quad D3.6}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha \vdash \mathsf{bind}\, p''' = \mathsf{store}() \ \mathsf{in}\ E_{3.6} : T_{0.511}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha \vdash E_{3.5} : T_{0.513}}$$

D3.41:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; p'' : [1]\,\mathbf{1} \vdash p'' : [1]\,\mathbf{1}}\quad D3.5}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p'' : [1]\,\mathbf{1} \vdash - = \mathsf{release}\, p'' \;\mathsf{in}\; E_{3.5} : T_{0.512}}$$

D3.4:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; p_o : [2]\,\mathbf{1} \vdash p_o : [2]\,\mathbf{1}}\quad D3.41}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p_o : [2]\,\mathbf{1}, p'' : [1]\,\mathbf{1} \vdash - = \mathsf{release}\, p_o \;\mathsf{in}\; E_{3.41} : T_{0.51}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha, p'' : [1]\,\mathbf{1} \vdash E_{3.4} : T_{0.51}}$$

D3.3:

$$\dfrac{\dfrac{\dfrac{D3.4}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha \vdash (\lambda p''.E_{3.4}) : T_{0.5}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha \vdash \mathit{C5}\ (\lambda p''.E_{3.4}) : \mathit{Queue}\ \alpha}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.34}, r : \alpha \vdash E_{3.3} : \mathit{Queue}\ \alpha}$$

D3.2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f : \alpha \vdash f : \alpha}\quad D3.3}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.34}, r : \alpha \vdash \langle\!\langle f, E_{3.3} \rangle\!\rangle : (\alpha \otimes \mathit{Queue}\ \alpha)}}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.34}, r : \alpha \vdash \mathsf{ret} \langle\!\langle f, E_{3.3} \rangle\!\rangle : T_2}}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.34}, r : \alpha \vdash E_{3.2} : T_2}$$

D3.12:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.33} \vdash fm : T_{0.33}}\quad D3.2}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.33}, r : \alpha \vdash \mathsf{let} \langle\!\langle f, m \rangle\!\rangle = fm \;\mathsf{in}\; E_{3.2} : T_2}}{\alpha; .; .; HT : T_{0.0}; fm : T_{0.33}, r : \alpha \vdash E_{3.12} : T_2}$$

D3.11:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.32} \vdash x' : T_{0.32}}\quad D3.12}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.32} \vdash \mathsf{let} \langle\!\langle fm, r \rangle\!\rangle = x' \;\mathsf{in}\; E_{3.12} : T_2}}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.32} \vdash E_{3.11} : T_2}$$

D3.1:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3}, p' : [0]\,\mathbf{1} \vdash x\ p' : T_{0.31}}\quad D3.11}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3}, p' : [0]\,\mathbf{1} \vdash \mathsf{bind}\, x' = x\ p' \;\mathsf{in}\; E_{3.11} : T_2}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3}, p' : [0]\,\mathbf{1}, p_o : [2]\,\mathbf{1} \vdash E_{3.1} : T_2}$$

D3.0:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3} \vdash \mathsf{store}() : \mathbb{M}\,2\,[2]\,\mathbf{1}}\quad D3.1}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3}, p' : [0]\,\mathbf{1} \vdash \mathsf{bind}\, p_o = \mathsf{store}() \;\mathsf{in}\; E_{3.1} : T_2}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3}, p' : [0]\,\mathbf{1} \vdash E_{3.0} : T_2}$$

D3:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,0\,\mathbf{1}}\quad D3.0}{\alpha; .; .; HT : T_{0.0}; x : T_{0.3} \vdash E_3 : T_2}$$

D2.51:

$$\dfrac{}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p_r : [3]\,\mathbf{1} \vdash HT\ p_r\ []\ m : T_{0.41}}$$

D2.5:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23} \vdash \mathsf{store}() : \mathbb{M}\,3\,[3]\,\mathbf{1}}\quad D2.51}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23} \vdash \mathsf{bind}\, p_r = \mathsf{store}() \;\mathsf{in}\; HT\ p_r\ []\ m : T_{0.413}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23} \vdash E_{2.5} : T_{0.413}}$$

D2.4:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; p'' : [2]\,\mathbf{1} \vdash p'' : [2]\,\mathbf{1}}\quad D2.5}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p'' : [2]\,\mathbf{1} \vdash \mathsf{release}\, p'' \;\mathsf{in}\; E_{2.5} : T_{0.411}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p'' : [2]\,\mathbf{1} \vdash E_{2.4} : T_{0.411}}$$

D2.3:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; p_o : [1]\,\mathbf{1} \vdash p_o : [1]\,\mathbf{1}} \qquad D2.4}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p_o : [1]\,\mathbf{1}, p'' : [2]\,\mathbf{1} \vdash - = \mathsf{release}\, p_o \;\mathsf{in}\; E_{2.4} : T_{0.41}}$$

D2.21:

$$\dfrac{\dfrac{\dfrac{D2.3}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p_o : [1]\,\mathbf{1}, p'' : [2]\,\mathbf{1} \vdash E_{2.3} : T_{0.41}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p_o : [1]\,\mathbf{1} \vdash \lambda p''.E_{2.3} : T_{0.4}}}{\alpha; .; .; HT : T_{0.0}; m : T_{0.23}, p_o : [1]\,\mathbf{1} \vdash \mathit{C4}\ (\lambda p''.E_{2.3}) : \mathit{Queue}\ \alpha}$$

D2.2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; f : \alpha \vdash f : \alpha} \qquad D2.21}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.23}, p_o : [1]\,\mathbf{1} \vdash \langle\langle f, (\mathit{C4}\ (\lambda p''.E_{2.3}))\rangle\rangle : (\alpha \otimes \mathit{Queue}\ \alpha)}}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.23}, p_o : [1]\,\mathbf{1} \vdash \mathsf{ret}\langle\langle f, (\mathit{C4}\ (\lambda p''.E_{2.3}))\rangle\rangle : T_0}}{\alpha; .; .; HT : T_{0.0}; f : \alpha, m : T_{0.23}, p_o : [1]\,\mathbf{1} \vdash E_{2.2} : T_0}$$

D2.11:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.22} \vdash x' : T_{0.22}} \qquad D2.2}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.22}, p_o : [1]\,\mathbf{1} \vdash \mathsf{let}\langle\langle f, m\rangle\rangle = x' \;\mathsf{in}\; E_{2.2} : T_0}}{\alpha; .; .; HT : T_{0.0}; x' : T_{0.22}, p_o : [1]\,\mathbf{1} \vdash E_{2.11} : T_0}$$

D2.1:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2}, p' : [1]\,\mathbf{1} \vdash x\ p' : T_{0.21}} \qquad D2.11}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2}, p_o : [1]\,\mathbf{1}, p' : [1]\,\mathbf{1} \vdash \mathsf{bind}\, x' = x\ p' \;\mathsf{in}\; E_{2.11} : T_0}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2}, p_o : [1]\,\mathbf{1}, p' : [1]\,\mathbf{1} \vdash E_{2.1} : T_0}$$

D2.0:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,1\,[1]\,\mathbf{1}} \qquad D2.1}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2}, p' : [1]\,\mathbf{1} \vdash \mathsf{bind}\, p_o = \mathsf{store}() \;\mathsf{in}\; E_{2.1} : T_1}}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2}, p' : [1]\,\mathbf{1} \vdash E_{2.0} : T_1}$$

D2:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{store}() : \mathbb{M}\,1\,[1]\,\mathbf{1}} \qquad D2.0}{\alpha; .; .; HT : T_{0.0}; x : T_{0.2} \vdash E_2 : T_2}$$

D1:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; x : \alpha \vdash \mathsf{ret}\ \langle\langle x, \mathit{C0}\rangle\rangle : T_2}}{\alpha; .; .; HT : T_{0.0}; x : \alpha \vdash E_1 : T_2}$$

D0:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \mathsf{fix}x.x : T_2}}{\alpha; .; .; HT : T_{0.0}; . \vdash E_0 : T_2}$$

D0.2:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; q : \mathit{Queue}\ \alpha \vdash q : \mathit{Queue}\ \alpha} \qquad D0 \qquad D1 \qquad D2 \qquad D3 \qquad D4 \qquad D5}{\alpha; .; .; HT : T_{0.0}; q : \mathit{Queue}\ \alpha \vdash E_{0.2} : T_2}$$

D0.1:

$$\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; . \vdash \uparrow^1 : \mathbb{M}\,1\,\mathbf{1}} \qquad D0.2}{\alpha; .; .; HT : T_{0.0}; q : \mathit{Queue}\ \alpha \vdash - = \uparrow^1; E_{0.2} : T_3}$$

Main derivation:

$$\dfrac{\dfrac{\dfrac{}{\alpha; .; .; HT : T_{0.0}; p : [3]\,\mathbf{1}, q : \mathit{Queue}\ \alpha \vdash p : [3]\,\mathbf{1}} \qquad D0.1}{\alpha; .; .; HT : T_{0.0}; p : [3]\,\mathbf{1}, q : \mathit{Queue}\ \alpha \vdash E_{0.1} : T_0}}{.; .; .; .; . \vdash E_{0.0} : T_{0.0}}$$

References

[1] Ahmed, A. J. *Semantics of types for mutable state.* PhD thesis, Princeton university, 2004.

[2] Atkey, R. Syntax and semantics of quantitative type theory. In *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (2018).

[3] Avanzini, M., and Dal Lago, U. Automating sized-type inference for complexity analysis. *Proc. ACM Program. Lang. 1*, ICFP (2017).

[4] Carbonneaux, Q., Hoffmann, J., and Shao, Z. Compositional certified resource bounds. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)* (2015).

[5] Charguéraud, A., and Pottier, F. Verifying the correctness and amortized complexity of a union-find implementation in separation logic with time credits. *J. Autom. Reasoning 62*, 3 (2019).

[6] Çiçek, E., Barthe, G., Gaboardi, M., Garg, D., and Hoffmann, J. Relational cost analysis. In *Proceedings of the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (2017).

[7] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009.

[8] Crary, K., and Weirich, S. Resource bound certification. In *POPL 2000, Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Boston, Massachusetts, USA, January 19-21, 2000* (2000).

[9] Dal Lago, U., and Gaboardi, M. Linear dependent types and relative completeness. *Logical Methods in Computer Science 8*, 4 (2011).

[10] Dal Lago, U., and Petit, B. Linear dependent types in a call-by-value scenario. *Science of Computer Programming 84* (2012).

[11] Danielsson, N. A. Lightweight semiformal time complexity analysis for purely functional data structures. In *Proceedings of the 35th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (2008).

[12] Danner, N., Licata, D. R., and Ramyaa. Denotational cost semantics for functional languages with inductive types. In *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming (ICFP)* (2015), pp. 140–151.

[13] Felleisen, M., and Friedman, D. P. Control operators, the secd-machine, and the λ-calculus. In *Proceedings of the IFIP Working Conference on Formal Description of Programming Concepts* (1987).

[14] Gaboardi, M., Katsumata, S.-y., Orchard, D., Breuvart, F., and Uustalu, T. Combining effects and coeffects via grading. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP)* (2016).

[15] Girard, J.-Y., Scedrov, A., and Scott, P. J. Bounded linear logic: a modular approach to polynomial-time computability. *Theoretical Computer Science 97*, 1 (1992).

[16] Hoffman, J. *Types with Potential: Polynomial Resource Bounds via Automatic Amortized Analysis.* PhD thesis, Ludwig-Maximilians-Universität München, 2011.

[17] Hoffmann, J., Aehlig, K., and Hofmann, M. Multivariate amortized resource analysis. In *Proceedings of the Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (2011).

[18] Hoffmann, J., Das, A., and Weng, S.-C. Towards automatic resource bound analysis for ocaml. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)* (2017).

[19] Hoffmann, J., and Hofmann, M. Amortized resource analysis with polynomial potential: A static inference of polynomial bounds for functional programs. In *Proceedings of the 19th European Conference on Programming Languages and Systems (ESOP)* (2010).

[20] Hofmann, M., and Jost, S. Static prediction of heap space usage for first-order functional programs. In *Proceedings of the 30th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (2003).

[21] Jost, S., Hammond, K., Loidl, H.-W., and Hofmann, M. Static determination of quantitative resource usage for higher-order programs. In *Proceedings of the Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (2010).

[22] Jost, S., Loidl, H., Hammond, K., Scaife, N., and Hofmann, M. "carbon credits" for resource-bounded computations using amortised analysis. In *Proceedings of Formal Methods (FM)* (2009).

[23] Jost, S., Vasconcelos, P., Florido, M., and Hammond, K. Type-based cost analysis for lazy functional languages. *J. Autom. Reason. 59*, 1 (2017).

[24] Kavvos, G. A., Morehouse, E., Licata, D. R., and Danner, N. Recurrence extraction for functional programs through call-by-push-value. *PACMPL 4*, POPL (2020).

[25] Krivine, J.-L. A call-by-name lambda-calculus machine. *Higher Order Symbolic Computation 20*, 3 (2007).

[26] Madhavan, R., Kulal, S., and Kuncak, V. Contract-based resource verification for higher-order functions with memoization. In *Proceedings of the ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)* (2017).

[27] Mével, G., Jourdan, J.-H., and Pottier, F. Time credits and time receipts in Iris. In *European Symposium on Programming (ESOP)* (2019).

[28] Moggi, E. Notions of computation and monads. *Information and Computation 93*, 1 (1991).

[29] Neis, G., Dreyer, D., and Rossberg, A. Non-parametric parametricity. *J. Funct. Program. 21*, 4-5 (2011).

[30] Okasaki, C. *Purely Functional Data Structures.* PhD thesis, Carnegie Mellon University, 1996.

[31] Pym, D. J., O'Hearn, P. W., and Yang, H. Possible worlds and resources: the semantics of bi. *Theoretical Computer Science 315*, 1 (2004).

[32] Tarjan, R. E. Amortized computational complexity. *SIAM Journal on Algebraic and Discrete Methods 6*, 2 (1985).

[33] Xi, H. Dependent ML an approach to practical programming with dependent types. *J. Funct. Program. 17*, 2 (2007).