All Hail Kleisli: An Enriched Yoneda Embedding of Indexed Modalities into Indexed Kleisli Categories

Peng Fu
University of South Carolina, USA

Kohei Kishida
University of Illinois at Urbana-Champaign, USA

1 Introduction

When formalizing programs, data, processes, effects, etc., in category theory or type theory, it may turn out crucial to keep track of different modalities of morphisms or terms: for instance, some quantum circuits may be harder to classically simulate than others because they involve certain gates; or, for information security, different data may have different levels of security. Such a family of modalities naturally gives rise to a family of different categories that correspond to the different modalities. This paper provides a theorem stating that, once we have such a family of categories, we can embed them into a single category so that the modalities can be modelled by a family of monads.

After reviewing several examples of families of modalities (Subsection 1.1), we illustrate a construction based on enriched category theory that, given a modality family of this sort, the indexed category arising from this family can be Yoneda-embedded into an indexed Kleisli category (Section 2). We then demonstrate the utility of this construction by defining a generic linear/nonlinear type theory of a modality family and showing how to interpret it in the constructed indexed Kleisli category (Section 3).

1.1 Examples of Modality Families in Application

Let us review some examples of application contexts in which it is crucial to keep track of different modalities of morphisms or terms. We will see that poset-indexed categories underly all these examples.

We take three examples involving quantum circuits.

Example 1.1 (Reversibility and controllability of circuits)**.**
Quantum programming languages often allow the programmer to apply high-level operations on quantum circuits. Examples of such operations include reversing and control. Programming errors can arise, however, from the fact that certain gates, such as state preparations and measurements, can make circuits irreversible or uncontrollable. It is therefore desirable to equip quantum programming languages with a type system that can keep track of the *modalities* of reversibility and controllability and thereby prevent the programmer from trying to reverse the irreversible or to control the uncontrollable.

Since quantum circuits are controllable only if reversible, there are three kinds of circuits, the controllable (and therefore also reversible) ones, the reversible ones, and all the circuits in general. They form three categories, **Ctrl**, **Rev**, and **Gen**, respectively, that are connected by identity-on-objects inclusion functors, giving us the following diagram of categories.

$$\mathbf{Ctrl} \longrightarrow \mathbf{Rev} \longrightarrow \mathbf{Gen}$$

We can also take a poset $P = \{ctrl \leqslant rev \leqslant gen\}$ of the shape of this diagram and regard it as expressing the modalities—being controllable, reversible, and general—and the implication among them. Then the indexed category $\mathbf{C} : P \to \mathbf{Cat}$ that maps P to the diagram above provides a semantics for the modalities.

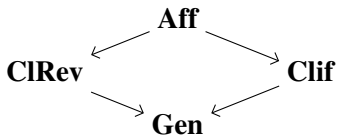
Submitted to:
ACT 2026

Among the languages from the Quipper [7] and Proto-Quipper family, Proto-Quipper-C [6] has a type system that keeps track of this family of modalities P, as well as a categorical semantics based on $\mathbf{C}$ (except that it is not assumed in [6] that the functor $\mathbf{C}_{(ctrl)(rev)} : \mathbf{Ctrl} \to \mathbf{Rev}$ is faithful, as one may add extra pieces of data to morphisms of **Ctrl**). A typing judgment is indexed by a modality $\alpha \in P$ to have the form $\Gamma \vdash_\alpha M : A$. The evaluation of the term M goes by expanding a circuit, and the index α means that only gates of modality α are used in this evaluation. In coordination with this idea, some types are indexed by modalities as well, such as the function types $A \multimap_\alpha B$. Another is the circuit type, $\mathbf{Circ}_\alpha(S,U)$, the type of circuits (with input type S and output type U) of modality α. The typing rules then dictate that reverse can only be applied to circuits of type $\mathbf{Circ}_\alpha(S,U)$ with $\alpha \leqslant rev$.

Example 1.2 (Simulating circuits)**.**

Many quantum programming languages come with a range of classical simulators. For example, Qiskit [10] has simulators based on state-vector simulation and stabilizer simulation. Quipper [7] also has simulators, one dedicated to classical reversible circuits, one to Clifford circuits, and one for general quantum circuits. Since the dedicated simulators can simulate the kinds of circuits they are dedicated to more efficiently than the general simulator can (see e.g. [1]), it would make sense to keep track of the kinds of circuits so that a given circuit can be sent to its most efficient simulator.

This example involves the following four kinds of quantum circuits: the classical reversible ones; the Clifford ones; the affine reversible circuits, i.e., the classical reversible and Clifford ones; and the circuits in general. The first two kinds are generated by certain sets of gates, which means that the four kinds of circuits form categories, so that we have the following diagram of categories, again with identity-on-objects inclusion functors. Note that, since not all classical reversible circuits are Clifford and vice versa, the shape of the diagram is not linear as in Example 1.1.



Example 1.3 (Circuits and physical processes)**.**

In the two previous examples, the involved functors are inclusions, basically (though one functor in [6] is not assumed to be). Here is an example in which the functor is *meant not* to be faithful.

Proto-Quipper-Dyn [5] has a type system for dynamic lifting, by which non-deterministic measurement outcomes obtained by executing a circuit are used as parameters to generate the next circuit. The operational semantics of the language involves probability distributions, and therefore the categorical semantics takes advantage of not only the category **Circ** of circuits, but also the category **Phys** of quantum physical operations and the functor $I : \mathbf{Circ} \to \mathbf{Phys}$ that sends circuits to the operations that underly them. I is not faithful, since the same physical operation can underly several different circuits.

The last example is from classical computing.

Example 1.4 (Language-based information security)**.**

In language-based information security [8, 13], one of the primary concerns is the control of information flow, especially to prevent high-level (e.g. secret) information from flowing to a lower (e.g. public) level. The basic information flow policy is called noninterference, it prohibits all explicit, implicit, and internal timing information flows from secret to public [12].

Security levels are ordered, with $\ell \leqslant \ell'$ meaning that ℓ' is a higher level than ℓ. The set of security levels is generally assumed to form a lattice. Programs and types are then labelled with security levels.

For example, in $\lambda^{\mathrm{REF}}_{\mathrm{SEC}}$ [13], one works with a typing judgment of the form $\Gamma[\ell] \vdash e : s$, in which ℓ is a security level, e a term, and s a type labelled with a security level. The typing judgment indicates that e will not modify a memory location with a security level lower than ℓ. The function types in $\lambda^{\mathrm{REF}}_{\mathrm{SEC}}$ are labeled with security levels (or latent effects in the sense of [11]) ℓ, to have the form $[\ell]s \to s'$. This means that a function $f : [\ell]s \to s'$ may write to a memory location that has security level ℓ. It is therefore not safe to call f at a higher level than ℓ. The type system makes sure that f cannot be called at a higher level, thereby preventing higher level information from getting leaked to lower levels.

Since this example involves modalities applied to types, it may not seem to have the same sort of structure as the other examples, in which we explicitly see an indexed category $\mathbf{C} : P \to \mathbf{Cat}$. Our result, however, suggests that the current example can be treated in the same fashion. See Remark 3.2 (3).

1.2 How Our Result Helps

In all the examples above, we see poset-indexed categories $\mathbf{C} : P \to \mathbf{Cat}$. The principal result of this paper is to show that, when P is a join-semilattice and all $\mathbf{C}_{ij}$ are identity on objects, we can use (generalized) Yoneda embeddings and embed $\mathbf{C}$ into an "indexed Kleisli category", i.e., a P-shaped family of Kleisli categories $Kl(\mathbf{T}_i)$ of monads $\mathbf{T}_i$ in a category $\mathbf{A}$. A particular case of this construction was given in the authors' previous work [6] for the P and $\mathbf{C}$ of Example 1.1, and the result of this paper can be seen as providing a theorem generalizing it.

Let us lay out two virtues of our result by focusing on two aspects of the construction: (a) that it embeds $\mathbf{C}_i$'s by Yoneda embeddings, and (b) that it embeds $\mathbf{C}_i$'s into the Kleisli categories of monads in a single category $\mathbf{A}$. We will actually demonstrate these virtues in Section 3.

(a) The first virtue concerns contexts in which $\mathbf{C}_i$'s are not (cartesian or monoidal) closed. For instance, if we generate the symmetric monoidal categories of quantum circuits syntactically by sets of gates, they are not monoidal closed. Our generalized Yoneda embeddings then embed $\mathbf{C}_i$'s into closed categories, thereby making a lambda calculus available. The target categories can be made monoidal closed as well, so that the embeddings are strong monoidal, serving the quantum cases of application.

(b) The second virtue is about the use of a single category $\mathbf{A}$ and a P-indexed monad $\mathbf{T}$, as opposed to a family of categories. For the sake of argument, let us assume $\mathbf{C}_i$'s are closed. Even then, a family of categories $\mathbf{C}_i$ is often inconvenient for modelling a type system. One may try to interpret a typing judgment $\Gamma \vdash_i M : A$ as a morphism in $\mathbf{C}_i$. Nonetheless, the type system may have the following typing rule for lambda-abstraction, for the bottom element 0 of P and $i \neq 0$.

$$\frac{\Gamma, x : A \vdash_i M : B}{\Gamma \vdash_0 \lambda x.M : A \multimap_i B}$$

Indeed, Proto-Quipper-C does have this rule with $0 = ctrl$ in Example 1.1. This is because the evaluation of $\lambda x.M$ does not use any gates that break controllability or reversibility: $\lambda x.M$ is a value and its trivial evaluation does not use any gates at all. Now we would interpret the top judgment, of modality i, by $[\![M]\!] : [\![\Gamma]\!] \otimes [\![A]\!] \to [\![B]\!]$ that is a morphism of $\mathbf{C}_i$. But how to interpret the bottom judgment of modality 0? The assumed monoidal closure of $\mathbf{C}_i$ would only give us a morphism $\widetilde{[\![M]\!]} : [\![\Gamma]\!] \to [\![A]\!] \multimap [\![B]\!]$ of $\mathbf{C}_i$, but there is no obvious functor from $\mathbf{C}_i$ to $\mathbf{C}_0$ that would enable us to interpret $\lambda x.M$ in $\mathbf{C}_0$. This difficulty is resolved by interpreting judgments of all modalities in the single category $\mathbf{A}$.

Before carrying on to stating and proving our principal theorem, let us justify its assumptions. The first assumption is that all the functors $\mathbf{C}_{ij}$ are identity on objects. This holds in all the four examples above, and it should be justified as long as we are concerned with modalities of morphisms. The second

assumption is that P is a join-semilattice. Again, this holds in all the examples, but a stronger justification comes from the fact that joins naturally arise from various typing rules, such as

$$\frac{\Phi,\Gamma_1 \vdash_i M : A \multimap_k B \quad \Phi,\Gamma_2 \vdash_j N : A}{\Phi,\Gamma_1,\Gamma_2 \vdash_{i\vee j\vee k} MN : B}$$

This rule, in terms of Example 1.1, states that there are three ways in which the evaluation of MN involves gates that break controllability: the evaluation of M does; the evaluation of N does; or M is such that the evaluation of its application involves gates that break controllability. This means that the modality of MN is the lowest upperbound of that of the judgment on M, that of the judgment on N, and the latent effect k in $M : A \multimap_k B$. In the same vein, the generic type theory for modality families that we define in Subsection 3.1 has several typing rules that presuppose joins of modalities. In fact, the concept of join-semilattice-indexed monad appears in other forms of applications than the examples above (see, e.g., Subsection 2.5 of [9]); our result can be seen as providing such applications with a method of canonically constructing a join-semilattice-indexed monad.

2 Embedding into an Indexed Kleisli Category

Here is a statement of the principal theorem of this paper.

Definition 2.1. Let P be any poset. A *P-indexed category* is a P-shaped diagram $\mathbf{C} : P \to \mathbf{Cat}$ of categories. Given an (ordinary) category $\mathbf{D}$, a *P-indexed monad* in $\mathbf{D}$ is a diagram $\mathbf{T} : P \to \mathbf{Mnd}(\mathbf{D})$ in the category $\mathbf{Mnd}(\mathbf{D})$ of monads in $\mathbf{D}$ and their transformations, and the *Kleisli category* of $\mathbf{T}$ is a P-indexed category $\mathbf{E} : P \to \mathbf{Cat}$ such that

1. $\mathbf{E}_i = Kl(\mathbf{T}_i)$ for each $i \in P$,
2. $\mathbf{E}_{ij} : \mathbf{E}_i \to \mathbf{E}_j$ for each $i \leqslant j$ is the identity on objects that has

$$\mathbf{E}_{ij}(f : X \to \mathbf{T}_i Y) : X \xrightarrow{\quad f \quad} \mathbf{T}_i Y \xrightarrow{\quad \mathbf{T}_{ij,Y} \quad} \mathbf{T}_j Y$$

Theorem 2.2. *Let L be any join-semilattice (including a bottom 0), and $\mathbf{C} : L \to \mathbf{Cat}$ be an L-indexed category such that all the $\mathbf{C}_i$'s ($i \in L$) share the same objects and $\mathbf{C}_{ij} : \mathbf{C}_i \to \mathbf{C}_j$ for each $i \leqslant j$ is an identity on objects. Then $\mathbf{C}$ fully embeds into the Kleisli category $\mathbf{E} : L \to \mathbf{Cat}$ of an L-indexed monad $\mathbf{T}$ in some category $\mathbf{A}$ in the following manner.*

1. *(a)* $\mathbf{E}_0 = \mathbf{A}$ *with* $\mathbf{T}_0 = \mathrm{id}_{\mathbf{A}}$.
   (b) $\mathbf{T}_i \circ \mathbf{T}_j = \mathbf{T}_j \circ \mathbf{T}_i = \mathbf{T}_j$ *whenever* $i \leqslant j$.
   (c) $\mathbf{T}_{ij,X}$ *for each X equals the following for the unit $\eta^{\mathbf{T}_j}$ of $\mathbf{T}_j$.*

$$\eta^{\mathbf{T}_j}_{\mathbf{T}_i X} : \mathbf{T}_i X \longrightarrow \mathbf{T}_j \mathbf{T}_i X = \mathbf{T}_j X$$

2. *There is a natural transformation $\mathbf{H} : \mathbf{C} \to \mathbf{E}$ such that each $\mathbf{H}_i : \mathbf{C}_i \to \mathbf{E}_i$ is full and faithful and indeed is a (generalized) Yoneda embedding.*

Throughout this section, we fix L and $\mathbf{C}$ as in the theorem. Let us write $\mathrm{Obj}_{\mathbf{C}}$ for the set of objects of the $\mathbf{C}_i$'s.

2.1 Bases of Enrichment

We construct the structure **A**, **T**, **E** in Theorem 2.2 by taking advantage of enriched categories. Let us first introduce the bases of enrichment that we will employ.

For each $i \in L$, let us write $L_i = L(i,-) = \{\, j \in L \mid i \leqslant j \,\}$. Clearly, $L_0 = L$. When $i \leqslant j$, it induces an adjoint pair of monotone maps,

$$L_i \underset{\text{inc}_{ij}}{\overset{\text{join}_{ij}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} L_j$$

where inc_{ij} is the inclusion while $\text{join}_{ij} = - \vee j : L_i \to L_j :: k \mapsto k \vee j$. For each $i \in L$, we have the presheaf topos $\mathbf{Sets}^{L_i}$ of L_i-indexed sets. For each $i \leqslant j$, the adjoint pair $\text{join}_{ij} \dashv \text{inc}_{ij}$ induces a geometric morphism $U_{ij} \dashv \Delta_{ij}$ from $\mathbf{Sets}^{L_j}$ to $\mathbf{Sets}^{L_i}$

$$\mathbf{Sets}^{L_j} \underset{\Delta_{ij}}{\overset{U_{ij}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{Sets}^{L_i}$$

by precomposition, i.e., $U_{ij} = - \circ \text{inc}_{ij}$ and $\Delta_{ij} = - \circ \text{join}_{ij}$. (This implies that U_{ij} has a left adjoint and Δ_{ij} has a right adjoint.) More explicitly, given $X : L_i \to \mathbf{Sets}$, we have $U_{ij}X = X \circ \text{inc}_{ij} : L_j \to \mathbf{Sets}$, so that $(U_{ij}X)_k = X_k$, and given $Y : L_j \to \mathbf{Sets}$, we have $\Delta_{ij}Y = Y \circ \text{join}_{ij} : L_i \to \mathbf{Sets}$, so that $(\Delta_{ij}Y)_k = Y_{k \vee j}$. In particular, if $j \leqslant k$ then $(\Delta_{ij}Y)_k = Y_k$. This implies $U_{ij} \circ \Delta_{ij} = \text{id}_{\mathbf{Sets}^{L_j}}$. Note also that U and Δ are functorial, i.e., $U_{jk} \circ U_{ij} = U_{ik}$ and $\Delta_{ij} \circ \Delta_{jk} = \Delta_{ik}$ whenever $i \leqslant j \leqslant k$.

We write $T_{ij} = \Delta_{ij} \circ U_{ij}$ for the monad of $U_{ij} \dashv \Delta_{ij}$. It follows from $U_{ij} \circ \Delta_{ij} = \text{id}_{\mathbf{Sets}^{L_j}}$ that T_{ij} is idempotent. More generally, when $i \leqslant j \leqslant k$, we have $T_{ij} \circ T_{ik} = T_{ik} \circ T_{ij} = T_{ik}$ (by $U_{ij} \circ \Delta_{ij} = \text{id}_{\mathbf{Sets}^{L_j}}$ as well as the functoriality of U and Δ). One consequence of T_{ij} being idempotent is that its multiplication $\mu^{T_{ij}} : {T_{ij}}^2 \to T_{ij}$ is the identity. The unit $\eta^{T_{ij}} : 1_{\mathbf{Sets}^{L_i}} \to T_{ij}$ is such that, for each L_i-indexed set $X : L_i \to \mathbf{Sets}$, the X-component $\eta^{T_{ij}}_X : X \to T_{ij}X$ is a natural transformation whose k-component is

$$\eta^{T_{ij}}_{X,k} = X_{k(k \vee j)} : X_k \to X_{k \vee j} = (T_{ij}X)_k.$$

Another consequence of T_{ij} being an idempotent monad is that $\Delta_{ij} : \mathbf{Sets}^{L_j} \to \mathbf{Sets}^{L_i}$ is a reflective subcategory that is both the Eilenberg-Moore category and the Kleisli category of T_{ij}.

For each $i \in L$, since $\mathbf{Sets}^{L_i}$ is cartesian closed, we consider enrichment of categories in $\mathbf{Sets}^{L_i}$. In particular, we take $\mathbf{Sets}^{L_i}$ as enriched in itself. Let us write $\mathscr{V}_i$ for $\mathbf{Sets}^{L_i}$ regarded as a category enriched in itself. Its hom-object $\mathscr{V}_i(X,Y)$ is the exponential $X \Rightarrow_i Y$ in $\mathbf{Sets}^{L_i}$; or, described differently, it is the L_i-indexed set $\mathscr{V}_i(X,Y) : L_i \to \mathbf{Sets}$ generated from the hom-set $\mathbf{Sets}^{L_i}(X,Y)$ by the functors $U_{jk} : \mathbf{Sets}^{L_j} \to \mathbf{Sets}^{L_k}$, meaning that

$$\begin{aligned} \mathscr{V}_i(X,Y)_j &= \mathbf{Sets}^{L_j}(U_{ij}X, U_{ij}Y), \\ \mathscr{V}_i(X,Y)_{jk} &= U_{jk,(U_{ij}X)(U_{ij}Y)} : \mathbf{Sets}^{L_j}(U_{ij}X, U_{ij}Y) \to \mathbf{Sets}^{L_k}(U_{ik}X, U_{ik}Y). \end{aligned}$$

Note that $U_{ij}(\mathscr{V}_i(X,Y)) = \mathscr{V}_j(U_{ij}X, U_{ij}Y)$ when $i \leqslant j$.

Indeed, when $i \leqslant j$, any $\mathscr{V}_j$-category can be regarded as enriched in $\mathscr{V}_i$, by sending hom-objects in $\mathscr{V}_j$ to $\mathscr{V}_i$ by Δ_{ij}. In particular, we can regard $\mathscr{V}_j$ as enriched in $\mathscr{V}_i$. Then the ordinary functor $U_{ij} : \mathbf{Sets}^{L_i} \to \mathbf{Sets}^{L_j}$ can be regarded as a $\mathscr{V}_i$-functor $\mathscr{V}_{ij} : \mathscr{V}_i \to \mathscr{V}_j$; it sends X to $U_{ij}X$, and its (X,Y)-component is

$$\eta^{T_{ij}}_{\mathscr{V}_i(X,Y)} : \mathscr{V}_i(X,Y) \to T_{ij}(\mathscr{V}_i(X,Y)) = \Delta_{ij}(\mathscr{V}_j(U_{ij}X, U_{ij}Y)),$$

so that the underlying ordinary functor of $\mathscr{V}_{ij}$ is U_{ij}. The right adjoint $\Delta_{ij} : \mathbf{Sets}^{L_j} \to \mathbf{Sets}^{L_i}$ is also enriched in $\mathscr{V}_i$;[1] we use the same notation for the $\mathscr{V}_i$-enriched $\Delta_{ij} : \mathscr{V}_j \to \mathscr{V}_i$. It sends X to $\Delta_{ij}X$, and its (X,Y)-component $\Delta_{ij,XY} : \Delta_{ij}(\mathscr{V}_j(X,Y)) \to \mathscr{V}_i(\Delta_{ij}X, \Delta_{ij}Y)$ is given by k-components

$$\Delta_{k(k\vee j),(U_{j(k\vee j)}X)(U_{j(k\vee j)}Y)} : \mathbf{Sets}^{L_{k\vee j}}(U_{j(k\vee j)}X, U_{j(k\vee j)}Y) \to \mathbf{Sets}^{L_k}(\Delta_{k(k\vee j)}U_{j(k\vee j)}X, \Delta_{k(k\vee j)}U_{j(k\vee j)}Y).$$

Again, the underlying ordinary functors of the $\mathscr{V}_i$-functor Δ_{ij} is the ordinary Δ_{ij}.

2.2 Encapsulating the Modality Family

We then introduce a family of $\mathscr{V}_i$-categories ($i \in L$), $\mathbf{D}_i$, each of which has $\mathbf{C}_i$ as its underlying ordinary category but takes advantage of enrichment in $\mathscr{V}_i$ to encapsulate not only $\mathbf{C}_i$ but all $\mathbf{C}_j$'s ($j \in L_i$).

We define $\mathbf{D}_i$ with $\mathrm{Obj}_{\mathbf{C}}$ and hom-objects $\mathbf{D}_i(A,B)$ that are generated in a manner similar to $\mathscr{V}_i(X,Y)$, from the hom-set $\mathbf{C}_i(A,B)$ by the functors $\mathbf{C}_{jk} : \mathbf{C}_j \to \mathbf{C}_k$.

Definition 2.3. For each $i \in L$, we define $\mathbf{D}_i$ to be the $\mathscr{V}_i$-category whose set of objects is $\mathrm{Obj}_{\mathbf{C}}$ and whose hom-object $\mathbf{D}_i(A,B)$ for $A,B \in \mathrm{Obj}_{\mathbf{C}}$ is the L_i-indexed set $\mathbf{D}_i(A,B) : L_i \to \mathbf{Sets}$ that has

$$\begin{aligned}\mathbf{D}_i(A,B)_j &= \mathbf{C}_j(A,B)\\ \mathbf{D}_i(A,B)_{jk} &= \mathbf{C}_{jk,AB} : \mathbf{C}_j(A,B) \to \mathbf{C}_k(A,B).\end{aligned}$$

Observe that $U_{ij}(\mathbf{D}_i(A,B)) = \mathbf{D}_j(A,B)$. Moreover,

Definition 2.4. When $i \leqslant j$, we can define a $\mathscr{V}_i$-functor $\mathbf{D}_{ij} : \mathbf{D}_i \to \mathbf{D}_j$ (where we regard $\mathbf{D}_j$ as enriched in $\mathscr{V}_i$ via Δ_{ij}) as an identity on objects whose (A,B)-component is a natural transformation

$$\mathbf{D}_{ij,AB} = \eta^{T_{ij}}_{\mathbf{D}_i(A,B)} : \mathbf{D}_i(A,B) \to T_{ij}(\mathbf{D}_i(A,B)) = \Delta_{ij}(\mathbf{D}_j(A,B)).$$

Note that the underlying ordinary functor of $\mathbf{D}_{ij} : \mathbf{D}_i \to \mathbf{D}_j$ is $\mathbf{C}_{ij} : \mathbf{C}_i \to \mathbf{C}_j$.

2.3 Enriched Yoneda Embeddings

For each $i \in L$, we will consider $\mathscr{V}_i$-valued and $\mathscr{V}_i$-enriched presheaves over $\mathbf{D}_i$, so that we can Yoneda-embed $\mathbf{D}_i$ into the category $\widehat{\mathbf{D}}_i$ of those presheaves.

Indeed, the operation $\widehat{\mathbf{D}}$ is functorial, not only in the ordinary way but also in the enriched way.

Lemma 2.5. *When $i \leqslant j$, there is an ordinary functor $\widehat{\mathbf{D}}_{ij} : \widehat{\mathbf{D}}_i \to \widehat{\mathbf{D}}_j$ that sends $F : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$ to the unique $\widehat{\mathbf{D}}_{ij}F$ making the following commute as a diagram of $\mathscr{V}_i$-functors.*

$$\begin{array}{ccc} \mathbf{D}_i{}^{\mathrm{op}} & \xrightarrow{\quad F \quad} & \mathscr{V}_i \\ {\scriptstyle \mathbf{D}_{ij}}\downarrow & \swarrow\!\!\swarrow & \downarrow{\scriptstyle \mathscr{V}_{ij}} \\ \mathbf{D}_j{}^{\mathrm{op}} & \xrightarrow[\widehat{\mathbf{D}}_{ij}F]{} & \mathscr{V}_j \end{array} \tag{1}$$

Here we only provide a proof sketch. See Appendix A for a fuller proof.

[1] This requires that L have finite joins. Induced by precomposition, $U_{ij} = - \circ \mathrm{inc}_{ij}$ has a right adjoint Δ_{ij} regardless of joins, defined by $(\Delta_{ij}Y)_k = \varprojlim_{j,k\leqslant\ell} Y_\ell$. But Δ_{ij} thus defined would not be $\mathscr{V}_i$-enriched if not for joins $j \vee k$.

Proof sketch. Given $F : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$, let us define $\widehat{\mathbf{D}}_{ij}F : \mathbf{D}_j{}^{\mathrm{op}} \to \mathscr{V}_j$ by $(\widehat{\mathbf{D}}_{ij}F)(A) = U_{ij}FA$ and $(\widehat{\mathbf{D}}_{ij}F)_{AB} = U_{ij}F_{AB} : \mathbf{D}_j(B,A) \to \mathscr{V}_j(U_{ij}FA, U_{ij}FB)$. Fixing any $G : \mathbf{D}_j{}^{\mathrm{op}} \to \mathscr{V}_j$, it makes (1) commute in place of $\widehat{\mathbf{D}}_{ij}F$, i.e. $G \circ \mathbf{D}_{ij} = U_{ij} \circ F$, iff $G = \widehat{\mathbf{D}}_{ij}F$. Moreover, given $\alpha : F \to G$ be a natural transformation of $\mathscr{V}_i$-presheaves $F, G : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$, define $\widehat{\mathbf{D}}_{ij}\alpha : \widehat{\mathbf{D}}_{ij}F \to \widehat{\mathbf{D}}_{ij}G$ with A-components

$$(\widehat{\mathbf{D}}_{ij}\alpha)_A = U_{ij}\alpha_A : U_{ij}FA \to U_{ij}GA. \qquad \square$$

Lemma 2.5 helps us to describe $\widehat{\mathbf{D}}_i$ as a $\mathscr{V}_i$-category, by generating the hom-object $\widehat{\mathbf{D}}_i(F,G) : L_i \to \mathbf{Sets}$ from the hom-set $\mathscr{V}_i^{\mathbf{D}_i{}^{\mathrm{op}}}(F,G)$ by the functors $\widehat{\mathbf{D}}_{jk} : \widehat{\mathbf{D}}_j \to \widehat{\mathbf{D}}_k$, i.e.,

$$\begin{aligned}
\widehat{\mathbf{D}}_i(F,G)_j &= \mathscr{V}_j^{\mathbf{D}_j{}^{\mathrm{op}}}(\widehat{\mathbf{D}}_{ij}F, \widehat{\mathbf{D}}_{ij}G), \\
\widehat{\mathbf{D}}_i(F,G)_{jk} &= \widehat{\mathbf{D}}_{jk,(\widehat{\mathbf{D}}_{ij}F)(\widehat{\mathbf{D}}_{ij}G)} : \mathscr{V}_j^{\mathbf{D}_j{}^{\mathrm{op}}}(\widehat{\mathbf{D}}_{ij}F, \widehat{\mathbf{D}}_{ij}G) \to \mathscr{V}_k^{\mathbf{D}_k{}^{\mathrm{op}}}(\widehat{\mathbf{D}}_{ik}F, \widehat{\mathbf{D}}_{ik}G).
\end{aligned}$$

It is then immediate that

Lemma 2.6. *The functor $\widehat{\mathbf{D}}_{ij} : \widehat{\mathbf{D}}_i \to \widehat{\mathbf{D}}_j$ of Lemma 2.5 is enriched in $\mathscr{V}_i$.*

The uniqueness part of Lemmas 2.5–2.6 entails

Lemma 2.7. $\widehat{\mathbf{D}}_{jk} \circ \widehat{\mathbf{D}}_{ij} = \widehat{\mathbf{D}}_{ik}$.

Here is how the generalized Yoneda embedding $y_i : \mathbf{D}_i \to \widehat{\mathbf{D}}_i$, as a $\mathscr{V}_i$-functor, works on $A \in \mathrm{Obj}_{\mathbf{C}}$: it gives $y_iA : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$ such that

- $(y_iA)(B) = \mathbf{D}_i(B,A)$.
- $(y_iA)_{BC} : \mathbf{D}_i(C,B) \to \mathscr{V}_i(\mathbf{D}_i(B,A), \mathbf{D}_i(C,A))$ is the map of L_i-indexed sets whose j-component is the function $(y_iA)_{BC,j} : \mathbf{C}_j(C,B) \to \mathbf{Sets}^{L_j}(\mathbf{D}_j(B,A), \mathbf{D}_j(C,A))$ as follows: for each $f : \mathbf{C}_j(C,B)$, note that $\mathbf{C}_{jk}(f) : \mathbf{C}_k(C,B)$, and $(y_iA)_{BC,j}(f) : \mathbf{D}_j(B,A) \to \mathbf{D}_j(C,A)$ is the natural transformation whose k-component is the precomposition function

$$((y_iA)_{BC,j}(f))_k : \mathbf{C}_k(B,A) \to \mathbf{C}_k(C,A) :: g \mapsto g \circ \mathbf{C}_{jk}(f).$$

Observe that, when $i, j \leqslant k$, we have $(y_iA)_{BC,k} = (y_jA)_{BC,k}$ since $((y_iA)_{BC,k}(f))_\ell = ((y_jA)_{BC,k}(f))_\ell$ for all $k \leqslant \ell$, and therefore that $U_{ij}((y_iA)_{BC}) = (y_jA)_{BC}$ when $i \leqslant j$. Based on this observation, here is a crucial fact connecting $\widehat{\mathbf{D}}$ and the Yoneda embeddings.

Lemma 2.8. *Whenever $i \leqslant j$, we have $\widehat{\mathbf{D}}_{ij}(y_iA) = y_jA$ for every $A \in \mathrm{Obj}_{\mathbf{C}}$.*

See Appendix A for a proof.

Looking at the diagram (1), one may wonder if, when $i \leqslant j$, we need to more generally consider $\mathscr{V}_j$-valued and $\mathscr{V}_i$-enriched presheaves over $\mathbf{D}_i$, where we regard $\mathscr{V}_j$ as enriched in $\mathscr{V}_i$ via Δ_{ij}. For the $\mathscr{V}_i$-category $\mathscr{V}_j^{\mathbf{D}_i{}^{\mathrm{op}}}$ of those presheaves, let us write $\widehat{\mathbf{D}}_j{}^i$; so, in particular, $\widehat{\mathbf{D}}_j = \widehat{\mathbf{D}}_j{}^j$. As we are about to show (Lemma 2.11), however, the superscripts make no essential difference, as $\widehat{\mathbf{D}}_j{}^i \cong \widehat{\mathbf{D}}_j$.

Let us first introduce

Definition 2.9. For each $i \leqslant j$, precomposition with the $\mathscr{V}_i$-functor $\mathbf{D}_{ij} : \mathbf{D}_i \to \mathbf{D}_j$ and postcomposition with the $\mathscr{V}_i$-functors

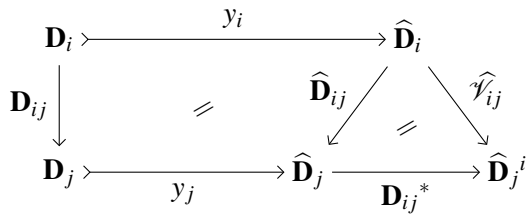
$$\mathscr{V}_j \underset{\Delta_{ij}}{\overset{\mathscr{V}_{ij}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathscr{V}_i$$

give us the following $\mathscr{V}_i$-functors:

$$\widehat{\mathbf{D}}_j = \mathscr{V}_j^{\mathbf{D}_j{}^{\mathrm{op}}} \xrightarrow{\mathbf{D}_{ij}{}^* = - \circ \mathbf{D}_{ij}} \widehat{\mathbf{D}}_j{}^i = \mathscr{V}_j^{\mathbf{D}_i{}^{\mathrm{op}}} \underset{\widehat{\Delta}_{ij} = \Delta_{ij} \circ -}{\overset{\widehat{\mathscr{V}}_{ij} = \mathscr{V}_{ij} \circ -}{\xleftarrow[\qquad\bot\qquad]{}}} \widehat{\mathbf{D}}_i = \mathscr{V}_i^{\mathbf{D}_i{}^{\mathrm{op}}}$$

It is crucial to observe that the property $\mathscr{V}_{ij} \circ \Delta_{ij} = \mathrm{id}_{\mathscr{V}_j}$ lifts to $\widehat{\mathscr{V}}_{ij} \circ \widehat{\Delta}_{ij} = \mathrm{id}_{\widehat{\mathbf{D}}_j{}^i}$ whenever $i \leqslant j$. Definition 2.9 helps us to rewrite Lemma 2.8 and the commutativity part of Lemmas 2.5–2.6 into

Lemma 2.10. *The following diagram of $\mathscr{V}_i$-functors commutes.*



(2)

Proof. The commutativity part of Lemmas 2.5–2.6 means that the triangle commutes. Lemma 2.8 means that the trapezoid on the left side commutes. □

Moreover, the bases of the triangle in (2) is an isomorphism.

Lemma 2.11. *The following opposite pair of $\mathscr{V}_i$-functors is mutually inverse, making $\widehat{\mathbf{D}}_j{}^i \cong \widehat{\mathbf{D}}_j$.*

$$\widehat{\mathbf{D}}_j \underset{\mathbf{D}_{ij}{}^*}{\overset{\widehat{\mathbf{D}}_{ij} \circ \widehat{\Delta}_{ij}}{\curvearrowleft}} \widehat{\mathbf{D}}_j{}^i$$

Proof. On the one hand, the commutativity part of Lemmas 2.5–2.6—i.e. the triangle in (2)—implies that $\mathbf{D}_{ij}{}^* \circ \widehat{\mathbf{D}}_{ij} \circ \widehat{\Delta}_{ij} = \widehat{\mathscr{V}}_{ij} \circ \widehat{\Delta}_{ij} = \mathrm{id}_{\widehat{\mathbf{D}}_j{}^i}$. On the other, we have $\widehat{\mathbf{D}}_{ij} \circ \widehat{\Delta}_{ij} \circ \mathbf{D}_{ij}{}^* = \mathrm{id}_{\widehat{\mathbf{D}}_j}$ because, for any $G : \mathbf{D}_j{}^{\mathrm{op}} \to \mathscr{V}_j$, we have $\mathscr{V}_{ij} \circ \Delta_{ij} \circ G \circ \mathbf{D}_{ij} = G \circ \mathbf{D}_{ij}$, and this implies $\widehat{\mathbf{D}}_{ij} \circ \widehat{\Delta}_{ij} \circ \mathbf{D}_{ij}{}^*(G) = \widehat{\mathbf{D}}_{ij}(\Delta_{ij} \circ G \circ \mathbf{D}_{ij}) = G$ by the uniqueness part of Lemmas 2.5–2.6. □

2.4 An Indexed Kleisli Category

The underlying ordinary versions of $\widehat{\mathbf{D}}_i$ and y_i's serve as the $\mathbf{E}$ and $\mathbf{H}$ of Theorem 2.2, essentially. We henceforth take the underlying categories and functors of the ones we discussed so far, recycling the same notation for the underlying versions rather than introducing new notation.

Let us write $\widehat{T}_{ij}$ for the monad $T_{ij} \circ -$ of $\widehat{\mathscr{V}}_{ij} \dashv \widehat{\Delta}_{ij}$. It is also the monad of $\widehat{\mathbf{D}}_{ij} \dashv R_{ij} = \widehat{\Delta}_{ij} \circ (\mathbf{D}_{ij}{}^*)^{-1}$ by Lemmas 2.10 and 2.11. The properties of idempotence and $\widehat{T}_{ij} \circ \widehat{T}_{ik} = \widehat{T}_{ik} \circ \widehat{T}_{ij} = \widehat{T}_{ik}$ lift from T_{ij}. The idempotence means that $R_{ij} : \widehat{\mathbf{D}}_j \to \widehat{\mathbf{D}}_i$ as well as $\widehat{\Delta}_{ij} : \widehat{\mathbf{D}}_j{}^i \to \widehat{\mathbf{D}}_i$ is a reflective subcategory that is both the Eilenberg-Moore category and the Kleisli category of $\widehat{T}_{ij}$. Thus, $\widehat{\mathbf{D}}$ is *essentially* the Kleisli category of the L-indexed monad $\widehat{T}_{0-}$ in $\widehat{\mathbf{D}}_0$, so that the natural transformation y_- Yoneda-embeds the L-indexed category $\mathbf{C}$ into $\widehat{\mathbf{D}}$.

One (evil) problem is that $\widehat{\mathbf{D}}_i$ as the Kleisli category of $\widehat{T}_{0i}$ is the category of free algebras, rather than the category of Kleisli morphisms between the same objects as $\widehat{\mathbf{D}}_0$, and hence that $\widehat{\mathbf{D}}$ fails to satisfy Definition 2.1. Nevertheless, we have

Fact 2.12. For $i \in L$, write $\mathbf{T}_i$ for $\widehat{T}_{0i}$, and $\mathbf{E}_i$ for the category $Kl(\mathbf{T}_i)$ of Kleisli morphisms of $\mathbf{T}_i$. Then $\mathbf{T}_0 = \mathrm{id}_{\widehat{\mathbf{D}}_0}$ and $\mathbf{E}_0 = \widehat{\mathbf{D}}_0$. When $i \leqslant j$, define $\mathbf{T}_{ij} : \mathbf{T}_i \to \mathbf{T}_j$ by $\mathbf{T}_{ij,X} = \eta^{\mathbf{T}_j}_{\mathbf{T}_i X}$ as in Theorem 2.2 (1c), and let $\mathbf{E}_{ij} : \mathbf{E}_i \to \mathbf{E}_j$ be the identity on objects with $\mathbf{E}_{ij}(f : X \to \mathbf{T}_i Y) = \mathbf{T}_{ij,Y} \circ f$ as in Definition 2.1 (2), so that $\mathbf{T}$ is an L-indexed monad in $\widehat{\mathbf{D}}_0$ and $\mathbf{E}$ its Kleisli category. Then $e_i : \mathbf{E}_i \to \widehat{\mathbf{D}}_i$ such that $e_i X = \widehat{\mathbf{D}}_{0i} X$ and

$$e_{i,XY} : \mathbf{E}_i(X,Y) = \widehat{\mathbf{D}}_0(X, \widehat{T}_{0i} Y) \to \widehat{\mathbf{D}}_i(\widehat{\mathbf{D}}_{0i} X, \widehat{\mathbf{D}}_{0i} Y)$$

is the isomorphism of $\widehat{\mathbf{D}}_{0i} \dashv R_{0i}$ is (part of) an equivalence of categories, and e_i and e_j make the right square below commute whenever $i \leqslant j$.

$$\begin{array}{ccccc}
\mathbf{C}_i & \xrightarrow{y_i} & \widehat{\mathbf{D}}_i & \xleftarrow{e_i} & \mathbf{E}_i \\
{\scriptstyle \mathbf{C}_{ij}}\downarrow & \swarrow & {\scriptstyle \widehat{\mathbf{D}}_{ij}}\downarrow & \searrow & \downarrow{\scriptstyle \mathbf{E}_{ij}} \\
\mathbf{C}_j & \xrightarrow[y_j]{} & \widehat{\mathbf{D}}_j & \xleftarrow[e_j]{} & \mathbf{E}_j
\end{array}$$

This enables us to "divert" y_i to $\mathbf{E}_i$ by pulling back $y_i A$ along $e_{i,XY}$; i.e., we can define $\mathbf{H}_i : \mathbf{C}_i \to \mathbf{E}_i$ by $y_i A = y_0 A$ and

$$\mathbf{H}_{i,AB} : \mathbf{C}_i(A,B) \xrightarrow{y_{i,AB}} \widehat{\mathbf{D}}_i(y_i A, y_i B) \xrightarrow{e_{i,(y_0 A)(y_0 B)}{}^{-1}} \mathbf{E}_i(y_0 A, y_0 B)$$

and this makes $\mathbf{H}_i$ essentially Yoneda as $y_i = e_i \circ \mathbf{H}_i$. This establishes Theorem 2.2.

2.5 Extra Structures

It should be emphasized for the sake of application that, being essentially Yoneda, the embedding $\mathbf{H} : \mathbf{C} \to \mathbf{E}$ in our construction accommodates several extra structures.

We will give a generic linear/non-linear type theory of modality families in Subsection 3.1, and provide it with a categorical semantics in Subsection 3.2. This semantics will take advantage of coproducts in $\mathbf{E}_0$—which are guaranteed to exist, since the presheaf categories $\widehat{\mathbf{D}}_i$'s, and $\mathbf{E}_i$'s by Fact 2.12, come with all limits and colimits as well as exponentials.

For other structures that will figure significantly in Subsection 3.2, we have

Lemma 2.13. *Suppose a category $\mathbf{C} : L \to \mathbf{Cat}$ indexed by a join-semilattice L is such that each $\mathbf{C}_i$ is symmetric monoidal and each $\mathbf{C}_{ij}$ is monoidal. Then the L-indexed monad $\mathbf{T}$ and its Kleisli category $\mathbf{E}$ given by Theorem 2.2 satisfy the following properties: For each i, we can define a monoidal product $\otimes_i$ and unit I_i in $\mathbf{E}_i$ so that*

1. *Each $\mathbf{E}_{ij} : (\mathbf{E}_i, \otimes_i, I_i) \to (\mathbf{E}_j, \otimes_j, I_j)$ is strong monoidal.*
2. *Each $\mathbf{H}_i : \mathbf{C}_i \to (\mathbf{E}_i, \otimes_i, I_i)$ is strong monoidal.*
3. *Each $(\mathbf{E}_i, \otimes_i, I_i)$ is monoidal closed.*
4. *Each $\mathbf{T}_i$ is commutative strong regarding $\otimes_i$ and I_i.*
5. *For each i, there is a strong monoidal functor p_i and a right adjoint $\flat_i$ that constitute a linear/nonlinear adjunction (à la Benton [2]):*

$$\mathbf{Sets} \underset{\flat_i}{\overset{p_i}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} (\mathbf{E}_i, \otimes_i, I_i)$$

Proof. We prove the equivalent cases for $\widehat{\mathbf{D}}$ instead of $\mathbf{E}$. Day's convolution [3] gives $\otimes_i$ and I_i satisfying (2) and (3), and it is then straightforward to check (1). The proof of (4), commutative strength, is similar to the one in [4]. For (5) $p_i \dashv \flat$, define

$$p_i : \mathbf{Sets} \to \widehat{\mathbf{D}}_i :: S \mapsto \sum_{x \in S} I_i, \qquad \flat_i : \widehat{\mathbf{D}}_i \to \mathbf{Sets} :: X \mapsto \widehat{\mathbf{D}}_i(I_i, X).$$

(We write $\widehat{\mathbf{D}}_i(I_i,X)$, etc., for hom-sets.) This makes p_i strongly monoidal by the definition of $\otimes_i$ and I_i. We moreover have $p_i \dashv \flat_i$ because

$$\begin{aligned}\widehat{\mathbf{D}}_i(p_i S, X) &\cong \mathscr{V}_i(1, \widehat{\mathbf{D}}_i(p_i S, X)) = \mathscr{V}_i(1, \widehat{\mathbf{D}}_i(\sum_{x \in S} I_i, X)) \cong \mathscr{V}_i(1, \prod_{x \in S} \widehat{\mathbf{D}}_i(I_i, X)) \cong \prod_{x \in S} \mathscr{V}_i(1, \widehat{\mathbf{D}}_i(I_i, X)) \\ &\cong \mathbf{Sets}(S, \widehat{\mathbf{D}}_i(I_i, X)) = \mathbf{Sets}(S, \flat_i X).\end{aligned}$$

□

3 A Generic Type Theory for Modality Families

In this section, we first give a generic linear/non-linear type system of a modality family. We then show how typing judgments of modality i can be interpreted by Kleisli morphisms of the monad $\mathbf{T}_i$.

3.1 Type System

Syntax Fixing a join-semilattice L, we use i, j, k to denote elements in L. We write 0 for the bottom and $i \vee j$ for joins. We work with type expressions A generated from some base types by monoidal product types $A \otimes B$, linear function types $A \multimap_i B$, and linear exponential types $!_i A$. We write Γ for a typing environment of the form $x_1 : A_1, \ldots, x_n : A_n$. We assume there are some base types that are considered non-resource (or parameter). We write P for a non-resource type generated from some base non-resource type by $\otimes$ and $!_i$. We write Φ for a parameter context of the form $x_1 : P_1, \ldots, x_n : P_n$.

The typing judgment $\Gamma \vdash_i M : A$ is labelled by $i \in L$, meaning that the term M has modality i. The following typing rules define a linear/nonlinear type system for tracking modalities.

Definition 3.1 (Typing rules)**.**

$$\frac{}{\Phi, x : A \vdash_0 x : A}\ \mathit{var}$$

$$\frac{\Gamma, x : A \vdash_i M : B}{\Gamma \vdash_0 \lambda x.M : A \multimap_i B}\ \mathit{lambda} \qquad \frac{\Phi, \Gamma_1 \vdash_i M : A \multimap_j B \quad \Phi, \Gamma_2 \vdash_k N : A}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j \vee k} MN : B}\ \mathit{app}$$

$$\frac{\Phi \vdash_i M : A}{\Phi \vdash_0 \mathsf{lift} M : !_i A}\ \mathit{lift} \qquad \frac{\Gamma \vdash_i M : !_j A}{\Gamma \vdash_{i \vee j} \mathsf{force} M : A}\ \mathit{force}$$

$$\frac{\Phi, \Gamma_1 \vdash_i M : A \quad \Phi, \Gamma_2 \vdash_j N : B}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j} (M, N) : A \otimes B}\ \mathit{pair} \qquad \frac{\Phi, \Gamma_1 \vdash_i N : A \otimes B \quad \Phi, \Gamma_2, x : A, y : B \vdash_j M : C}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j} \mathbf{let}\,(x, y) = N\ \mathbf{in}\ M : C}\ \mathit{let}$$

Remark 3.2. Here are some remarks. (3)–(5) each invite further considerations, but we do not delve into them in this paper.

1. All the values have modality 0, the bottom, as dictated by the rules *var*, *lambda*, and *lift*. We justified this stipulation for quantum circuit scenarios in Subsection 1.2. For information security scenarios, values would have the lowest level, and therefore it is safe to use them in any higher security environment.

2. The types $A \multimap_i B$ and $!_i A$ have modality i as latent effect. Their elimination rules, *app* and *force*, show how the latent effect—the modality of the function type or the linear exponential type—gets incorporated into the modality of the resulting term.
3. In language-based security (Example 1.4), one further annotates each type with a modality, so a type generally has the form A_i. This however poses no problem for our semantics, as we can interpret A_i by $\mathbf{T}_i[\![A]\!]$.
4. The order on L induces a subtyping relation, which gives rise to the following typing rule.

$$\frac{\Gamma \vdash_i M : A \quad A \leq B}{\Gamma \vdash_i M : B}$$

   The subtyping relation $A \leq B$ in the semantics can be modeled by a morphism $[\![A]\!] \to [\![B]\!]$.
5. It is also possible to have a typing judgment for sub-modality, i.e.,

$$\frac{\Gamma \vdash_i M : A \quad i \leqslant j}{\Gamma \vdash_j M : B}$$

   This can be modeled by the monad transformation $\mathbf{T}_{ij} : \mathbf{T}_i \to \mathbf{T}_j$ (see Theorem 2.2 (1b)).

3.2 Interpretation

We are to provide a categorical semantics for the linear/nonlinear type system of a given L-indexed *symmetric monoidal* category $\mathbf{C} : L \to \mathbf{SMC}$. Let $\mathbf{A}$, $\mathbf{T} : L \to \mathbf{Mnd}(\mathbf{A})$, and $\mathbf{E} : L \to \mathbf{SMC}$ be the *symmetric monoidal closed* category, the L-indexed monad in $\mathbf{A}$, and the Kleisli category of $\mathbf{T}$, respectively, that are given by Theorem 2.2 and Lemma 2.13. We also have a linear/nonlinear adjunction $p = p_0 \dashv \flat = \flat_0$ between **Sets** and $\mathbf{A}$ by Lemma 2.13. We write ! for the comonad $p \circ \flat$.

We first define the following interpretation of types.

Definition 3.3. We define $[\![A]\!]$ as an object of $\mathbf{A}$ by induction on the structure of A. We assume there are some distinguished objects in $\mathbf{A}$ that serve as interpretation for the base types. Then set

$$\begin{array}{lcl} [\![A \otimes B]\!] & = & [\![A]\!] \otimes [\![B]\!] \\ [\![!_i A]\!] & = & !\,\mathbf{T}_i[\![A]\!] \\ [\![A \multimap_i B]\!] & = & [\![A]\!] \multimap \mathbf{T}_i[\![B]\!] \end{array}$$

For a parameter type P, its interpretation $[\![P]\!]$ is equipped with a comonoid structure, $\Delta : [\![P]\!] \to [\![P]\!] \otimes [\![P]\!]$ and discard : $[\![P]\!] \to I$. When $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, we write $[\![\Gamma]\!]$ for the tensor product $[\![A_1]\!] \otimes \cdots \otimes [\![A_n]\!]$.

The following theorem gives an interpretation of the typing judgment $\Gamma \vdash_i M : A$ as a Kleisli morphism of $\mathbf{T}_i$ (including $\mathbf{T}_0 = \mathrm{id}_{\mathbf{A}}$) in $\mathbf{A}$. See Appendix B for more details.

Theorem 3.4. *If $\Gamma \vdash_i M : A$, then there exists a morphism $[\![\Gamma \vdash_i M : A]\!] : [\![\Gamma]\!] \to \mathbf{T}_i[\![A]\!]$. We often write $[\![M]\!]$ for $[\![\Gamma \vdash_i M : A]\!]$.*

Proof. We proceed by induction on the derivation of $\Gamma \vdash_i M : A$.

- Case.

$$\frac{}{\Phi, x : A \vdash_0 x : A}\ \mathit{var}$$

  Since $\mathbf{T}_0 = \mathrm{id}_{\mathbf{A}}$, we define $[\![\Phi, x : A \vdash_0 x : A]\!]$ by

$$[\![\Phi]\!] \otimes [\![A]\!] \xrightarrow{\mathrm{discard} \otimes [\![A]\!]} I \otimes [\![A]\!] \xrightarrow{\cong} [\![A]\!].$$

- Case.
$$\frac{\Phi,\Gamma_1 \vdash_i M : A \multimap_j B \quad \Phi,\Gamma_2 \vdash_k N : A}{\Phi,\Gamma_1,\Gamma_2 \vdash_{i\vee j\vee k} MN : B}\ app$$
By the induction hypotheses, we have $[\![M]\!] : [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \to \mathbf{T}_i([\![A]\!] \multimap \mathbf{T}_j[\![B]\!])$ and $[\![N]\!] : [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \to \mathbf{T}_k[\![A]\!]$. We therefore define $[\![MN]\!]$ by
$$\begin{array}{lll} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] & \xrightarrow{\Delta} & [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \\ & \xrightarrow{\cong} & [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \\ & \xrightarrow{[\![M]\!]\otimes[\![N]\!]} & \mathbf{T}_i([\![A]\!] \multimap \mathbf{T}_j[\![B]\!]) \otimes \mathbf{T}_k[\![A]\!] \\ & \xrightarrow{s} & \mathbf{T}_i\mathbf{T}_k(([\![A]\!] \multimap \mathbf{T}_j[\![B]\!]) \otimes [\![A]\!]) \\ & \xrightarrow{\mathbf{T}_i\mathbf{T}_k\varepsilon} & \mathbf{T}_i\mathbf{T}_k\mathbf{T}_j[\![B]\!] \\ & \xrightarrow{i} & \mathbf{T}_{i\vee j\vee k}[\![B]\!]. \end{array}$$
Here s is the composition of two strength maps, and i is defined by
$$\mathbf{T}_i\mathbf{T}_k\mathbf{T}_j[\![B]\!] \to \mathbf{T}_{i\vee j\vee k}\mathbf{T}_{i\vee j\vee k}\mathbf{T}_{i\vee j\vee k}[\![B]\!] = \mathbf{T}_{i\vee j\vee k}[\![B]\!],$$
where the arrow is obtained by the monad transformations $\mathbf{T}_i \to \mathbf{T}_{i\vee j\vee k}, \mathbf{T}_k \to \mathbf{T}_{i\vee j\vee k}$ and $\mathbf{T}_j \to \mathbf{T}_{i\vee j\vee k}$, and the equation by idempotence.
- Case.
$$\frac{\Gamma, x : A \vdash_i M : B}{\Gamma \vdash_0 \lambda x.M : A \multimap_i B}\ lambda$$
By the induction hypothesis, we have $[\![M]\!] : [\![\Gamma]\!] \otimes [\![A]\!] \to \mathbf{T}_i[\![B]\!]$. By currying we have $[\![\lambda x.M]\!] = \widetilde{[\![M]\!]} : [\![\Gamma]\!] \to [\![A]\!] \multimap \mathbf{T}_i[\![B]\!]$, a morphism in $\mathbf{A}$. □

4 Conclusion and Future Work

In this paper we provided a construction based on enriched categories that, given a join-semilattice-indexed category $\mathbf{C} : L \to \mathbf{Cat}$ (with all $\mathbf{C}_{ij}$ identities on objects), gives an L-indexed monad $\mathbf{T}$ in a single category so that $\mathbf{C}$ can be Yoneda-embedded into its Kleisli category. This theorem can be applied to provide a categorical semantics for type theories of a family of modalities of morphisms, including, but not limited to, modalities of quantum circuits. We demonstrated the potential of such application by defining a generic linear/nonlinear type theory of modality families and applying the theorem to provide a categorical semantics.

There are various directions in which to extend the result of this paper. Remark 3.2 contained three such directions as (3)–(5). Another direction concerns extra structures, or how to extend the results in Lemma 2.13. For instance, Yoneda embeddings do not preserve coproducts, whereas the semantics of Proto-Quipper-Dyn (Example 1.3) needed them to be preserved. This is why the construction in [4] takes advantage of the Lambek embeddings rather than the Yoneda. Extending our general construction recipe to cover more tailor-made embeddings and thereby to accommodate more extra structures can be a significant project for the sake of application.

References

[1] Scott Aaronson & Daniel Gottesman (2004): *Improved simulation of stabilizer circuits*. *Physical Review A—Atomic, Molecular, and Optical Physics* 70(5), p. 052328.

[2] P Nick Benton (1994): *A mixed linear and non-linear logic: Proofs, terms and models*. In: *International Workshop on Computer Science Logic*, Springer, pp. 121–135.

[3] Brian Day (2006): *On closed categories of functors*. In: *Reports of the Midwest Category Seminar IV*, Springer, pp. 1–38.

[4] Peng Fu, Kohei Kishida, Neil J. Ross & Peter Selinger (2023): *A biset-enriched categorical model for Proto-Quipper with dynamic lifting*. In: *Proceedings of the 19th International Conference on Quantum Physics and Logic, QPL 2022, Oxford, Electronic Proceedings in Theoretical Computer Science* 394, pp. 302–342, doi:https://doi.org/10.4204/EPTCS.394.16.

[5] Peng Fu, Kohei Kishida, Neil J. Ross & Peter Selinger (2023): *Proto-Quipper with dynamic lifting*. In: *Proceedings of the 50th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2023, Boston, Proceedings of the ACM on Programming Languages* 7 (POPL), pp. 309–334, doi:https://doi.org/10.1145/3571204. Also available from arXiv:2204.13041.

[6] Peng Fu, Kohei Kishida, Neil J. Ross & Peter Selinger (2025): *Proto-Quipper with reversing and control*. Proceedings of the 22nd International Conference on Quantum Physics and Logic (QPL 2025), Varna, Bulgaria.

[7] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger & Benoît Valiron (2013): *Quipper: a scalable quantum programming language*. In: *Proceedings of the 34th Annual ACM SIGPLAN Conference on Programming Language Design and Implementation, ACM SIGPLAN Notices* 48, ACM, pp. 333–342.

[8] Nevin Heintze & Jon G Riecke (1998): *The SLam calculus: programming with secrecy and integrity*. In: *Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp. 365–377.

[9] Shin-ya Katsumata (2014): *Parametric effect monads and semantics of effect systems*. *SIGPLAN Not.* 49(1), p. 633–645, doi:https://doi.org/10.1145/2578855.2535846. Available at https://doi.org/10.1145/2578855.2535846.

[10] *Qiskit*. https://www.ibm.com/quantum/qiskit. Accessed: 2026-03-22.

[11] Jean-Pierre Talpin & Pierre Jouvelot (1992): *Polymorphic type, region and effect inference*. *Journal of functional programming* 2(3), pp. 245–271.

[12] Stephan Arthur Zdancewic (2002): *Programming languages for information security*. Cornell University.

[13] Steve Zdancewic (2003): *A type system for robust declassification*. *Electronic Notes in Theoretical Computer Science* 83, pp. 263–277.

A Proofs for Lemmas 2.5 and 2.8

Here is a proof for Lemma 2.5.

Proof. Given $F : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$, let us define $\widehat{\mathbf{D}}_{ij}F : \mathbf{D}_j{}^{\mathrm{op}} \to \mathscr{V}_j$ by $(\widehat{\mathbf{D}}_{ij}F)(A) = U_{ij}FA$ and $(\widehat{\mathbf{D}}_{ij}F)_{AB} = U_{ij}F_{AB} : \mathbf{D}_j(B,A) \to \mathscr{V}_j(U_{ij}FA, U_{ij}FB)$. Fixing any $G : \mathbf{D}_j{}^{\mathrm{op}} \to \mathscr{V}_j$, we show that it makes (1) commute in place of $\widehat{\mathbf{D}}_{ij}F$, i.e. $G \circ \mathbf{D}_{ij} = U_{ij} \circ F$, iff $G = \widehat{\mathbf{D}}_{ij}F$. On $A \in \mathrm{Obj}_{\mathbf{C}}$, since $\mathbf{D}_{ij}$ is an identity on objects we have $G \circ \mathbf{D}_{ij}(A) = U_{ij}FA$ iff $GA = U_{ij}FA = (\widehat{\mathbf{D}}_{ij}F)(A)$. On hom-objects, on the one hand, the (A,B)-

component of (1) is the following, which commutes by the naturality of $\eta^{T_{ij}} : \mathrm{id}_{\mathbf{Sets}^{L_i}} \to T_{ij}$.

$$\begin{array}{ccc} \mathbf{D}_i(B,A) & \xrightarrow{\quad F_{AB} \quad} & \mathscr{V}_i(FA,FB) \\ {\scriptstyle \eta^{T_{ij}}_{\mathbf{D}_i(B,A)}}\downarrow & \eqslantless & \downarrow{\scriptstyle \eta^{T_{ij}}_{\mathscr{V}_i(FA,FB)}} \\ T_{ij}(\mathbf{D}_i(B,A)) & \xrightarrow[\Delta_{ij}((\widehat{\mathbf{D}}_{ij}F)_{AB}) = T_{ij}(F_{AB})]{} & T_{ij}(\mathscr{V}_i(FA,FB)) \end{array}$$

On the other hand, if G makes this square commute in place of $\widehat{\mathbf{D}}_{ij}F$, then, for each $j \leqslant k$, the vertical arrows of the k-component of the square are identities and therefore imply that the horizontal arrows, $F_{AB,k} = (\widehat{\mathbf{D}}_{ij}F)_{AB,k}$ above and $G_{AB,k}$ below, are identical.

We now show that $\widehat{\mathbf{D}}_{ij}$ extends to an ordinary functor. Let $\alpha : F \to G$ be a natural transformation of $\mathscr{V}_i$-presheaves $F, G : \mathbf{D}_i{}^{\mathrm{op}} \to \mathscr{V}_i$. Then define $\widehat{\mathbf{D}}_{ij}\alpha : \widehat{\mathbf{D}}_{ij}F \to \widehat{\mathbf{D}}_{ij}G$ with A-components

$$(\widehat{\mathbf{D}}_{ij}\alpha)_A = U_{ij}\alpha_A : U_{ij}FA \to U_{ij}GA.$$

This is natural since applying U_{ij} to the following hexagon, which commutes by the naturality of α, gives the commuting hexagon that expresses the naturality of $\widehat{\mathbf{D}}_{ij}\alpha$.

$$\begin{array}{ccccc} & & \mathbf{D}_i(B,A) \times 1 & \xrightarrow{F_{AB} \times \alpha_A} & \mathscr{V}_i(FA,FB) \times \mathscr{V}_i(FB,GB) \\ & \langle \mathrm{id}, ! \rangle \nearrow & & & \searrow \circ \\ \mathbf{D}_i(B,A) & & \eqslantless & & \mathscr{V}_i(FA,GB) \\ & \langle !, \mathrm{id} \rangle \searrow & & & \nearrow \circ \\ & & 1 \times \mathbf{D}_i(B,A) & \xrightarrow[\alpha_B \times G_{AB}]{} & \mathscr{V}_i(FA,GA) \times \mathscr{V}_i(GA,GB) \end{array} \qquad \square$$

Here is a proof for Lemma 2.8.

Proof. By the uniqueness part of Lemma 2.5, it is enough to show that y_iA and y_jA make (1) commute in place of F and $\widehat{\mathbf{D}}_{ij}F$, i.e., that $(y_jA) \circ \mathbf{D}_{ij} = U_{ij} \circ (y_iA)$. For every $B \in \mathrm{Obj}_{\mathbf{C}}$, we have $(U_{ij} \circ (y_iA))(B) = U_{ij}(\mathbf{D}_i(B,A)) = \mathbf{D}_j(B,A) = (y_jA)(B)$ since $\mathbf{D}_{ij}$ is an identity on objects. For any $B, C \in \mathrm{Obj}_{\mathbf{C}}$, the (B,C)-component of (1) (with y_iA, y_jA in place of F, $\widehat{\mathbf{D}}_{ij}F$) is the following because $U_{ij}((y_iA)_{BC}) = (y_jA)_{BC}$, and it commutes by the naturality of $\eta^{T_{ij}} : \mathrm{id}_{\mathbf{Sets}^{L_i}} \to T_{ij}$.

$$\begin{array}{ccc} \mathbf{D}_i(C,B) & \xrightarrow{\quad (y_iA)_{BC} \quad} & \mathscr{V}_i(\mathbf{D}_i(B,A), \mathbf{D}_i(C,A)) \\ {\scriptstyle \eta^{T_{ij}}_{\mathbf{D}_i(C,B)}}\downarrow & \eqslantless & \downarrow{\scriptstyle \eta^{T_{ij}}_{\mathscr{V}_i(\mathbf{D}_i(B,A),\mathbf{D}_i(C,A))}} \\ T_{ij}(\mathbf{D}_i(C,B)) & \xrightarrow[\Delta_{ij}((y_jA)_{BC}) = T_{ij}((y_iA)_{BC})]{} & T_{ij}(\mathscr{V}_i(\mathbf{D}_i(B,A), \mathbf{D}_i(C,A))) \end{array} \qquad \square$$

B Proof for Theorem 3.4

Theorem B.1. *If $\Gamma \vdash_i M : A$, then there exists a morphism $[\![\Gamma \vdash_i M : A]\!] : [\![\Gamma]\!] \to \mathbf{T}_i[\![A]\!]$. We often write $[\![M]\!]$ for $[\![\Gamma \vdash_i M : A]\!]$.*

Proof. We proceed by induction on the derivation of $\Gamma \vdash_i M : A$.

- Case.
$$\frac{}{\Phi, x : A \vdash_0 x : A}\ var$$
Since $\mathbf{T}_0 = \mathrm{id}_{\mathbf{A}}$, we define $[\![\Phi, x : A \vdash_0 x : A]\!]$ by
$$[\![\Phi]\!] \otimes [\![A]\!] \xrightarrow{\mathrm{discard} \otimes [\![A]\!]} I \otimes [\![A]\!] \xrightarrow{\cong} [\![A]\!].$$

- Case
$$\frac{\Phi, \Gamma_1 \vdash_i M : A \multimap_j B \quad \Phi, \Gamma_2 \vdash_k N : A}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j \vee k} MN : B}\ app$$
By the induction hypotheses, we have $[\![M]\!] : [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \to \mathbf{T}_i([\![A]\!] \multimap \mathbf{T}_j[\![B]\!])$ and $[\![N]\!] : [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \to \mathbf{T}_k[\![A]\!]$. We therefore define $[\![MN]\!]$ by
$$\begin{array}{lll} [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] & \xrightarrow{\Delta} & [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \\ & \xrightarrow{\cong} & [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \\ & \xrightarrow{[\![M]\!] \otimes [\![N]\!]} & \mathbf{T}_i([\![A]\!] \multimap \mathbf{T}_j[\![B]\!]) \otimes \mathbf{T}_k[\![A]\!] \\ & \xrightarrow{s} & \mathbf{T}_i\mathbf{T}_k(([\![A]\!] \multimap \mathbf{T}_j[\![B]\!]) \otimes [\![A]\!]) \\ & \xrightarrow{\mathbf{T}_i\mathbf{T}_k\varepsilon} & \mathbf{T}_i\mathbf{T}_k\mathbf{T}_j[\![B]\!] \\ & \xrightarrow{i} & \mathbf{T}_{i \vee j \vee k}[\![B]\!]. \end{array}$$
Here s is the composition of two strength maps, and i is defined by
$$\mathbf{T}_i\mathbf{T}_k\mathbf{T}_j[\![B]\!] \to \mathbf{T}_{i \vee j \vee k}\mathbf{T}_{i \vee j \vee k}\mathbf{T}_{i \vee j \vee k}[\![B]\!] = \mathbf{T}_{i \vee j \vee k}[\![B]\!],$$
where the arrow is obtained by the monad transformations $\mathbf{T}_i \to \mathbf{T}_{i \vee j \vee k}$, $\mathbf{T}_k \to \mathbf{T}_{i \vee j \vee k}$, and $\mathbf{T}_j \to \mathbf{T}_{i \vee j \vee k}$.

- Case.
$$\frac{\Gamma, x : A \vdash_i M : B}{\Gamma \vdash_0 \lambda x.M : A \multimap_i B}\ lambda$$
By the induction hypothesis, we have $[\![M]\!] : [\![\Gamma]\!] \otimes [\![A]\!] \to \mathbf{T}_i[\![B]\!]$. By currying we have $[\![\lambda x.M]\!] = \widetilde{[\![M]\!]} : [\![\Gamma]\!] \to [\![A]\!] \multimap \mathbf{T}_i[\![B]\!]$, a morphism in $\mathbf{A}$.

- Case.
$$\frac{\Gamma \vdash_i M : !_j A}{\Gamma \vdash_{i \vee j} \mathsf{force} M : A}\ force$$
By the induction hypothesis, we have $[\![M]\!] : [\![\Gamma]\!] \to \mathbf{T}_i\, !\, \mathbf{T}_j[\![A]\!]$. We therefore define $[\![\mathsf{force} M]\!]$ by the following.
$$[\![\Gamma]\!] \xrightarrow{[\![M]\!]} \mathbf{T}_i\, !\, \mathbf{T}_j[\![A]\!] \xrightarrow{T_i \mathsf{force}} \mathbf{T}_i\mathbf{T}_j[\![A]\!] \xrightarrow{i} \mathbf{T}_{i \vee j}[\![A]\!].$$
We write $\mathsf{force} : !X \to X$ for the counit of the comonad $!$.

- Case.
$$\frac{\Phi \vdash_i M : A}{\Phi \vdash_0 \mathsf{lift}\, M : !_i A}\ \mathit{lift}$$
By the induction hypothesis, we have $[\![M]\!] : [\![\Phi]\!] \to \mathbf{T}_i[\![A]\!]$. Since $[\![\Phi]\!] = p(X)$ for some $X \in \mathbf{Sets}$, by the adjunction $p \dashv \flat$ we have $\overline{[\![M]\!]} : X \to \flat \mathbf{T}_i[\![A]\!]$. We therefore define $[\![\mathsf{lift}\, M]\!] = p\overline{[\![M]\!]} : [\![\Phi]\!] \to !\,\mathbf{T}_i[\![A]\!]$.

- Case.
$$\frac{\Phi, \Gamma_1 \vdash_i M : A \quad \Phi, \Gamma_2 \vdash_j N : B}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j} (M, N) : A \otimes B}\ \mathit{pair}$$
By the induction hypotheses, we have $[\![M]\!] : [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \to \mathbf{T}_i[\![A]\!]$ and $[\![N]\!] : [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \to \mathbf{T}_j[\![B]\!]$. We define $[\![(M,N)]\!]$ by
$$\begin{array}{lll}
[\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] & \xrightarrow{\Delta} & [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \\
& \xrightarrow{\cong} & [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \\
& \xrightarrow{[\![M]\!] \otimes [\![N]\!]} & \mathbf{T}_i[\![A]\!] \otimes \mathbf{T}_j[\![B]\!] \\
& \xrightarrow{s} & \mathbf{T}_i \mathbf{T}_j([\![A]\!] \otimes [\![B]\!]) \\
& \xrightarrow{i} & \mathbf{T}_{i \vee j}([\![A]\!] \otimes [\![B]\!])
\end{array}$$

- Case.
$$\frac{\Phi, \Gamma_1 \vdash_i N : A \otimes B \quad \Phi, \Gamma_2, x : A, y : B \vdash_j M : C}{\Phi, \Gamma_1, \Gamma_2 \vdash_{i \vee j} \mathbf{let}\,(x,y) = N\ \mathbf{in}\ M : C}\ \mathit{let}$$
By the induction hypotheses, we have $[\![N]\!] : [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \to \mathbf{T}_i([\![A]\!] \otimes [\![B]\!])$ and $[\![M]\!] : [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A]\!] \otimes [\![B]\!] \to \mathbf{T}_j[\![C]\!]$. We define $[\![\mathbf{let}\,(x,y) = N\ \mathbf{in}\ M]\!]$ by
$$\begin{array}{lll}
[\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] & \xrightarrow{\Delta} & [\![\Phi]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \\
& \xrightarrow{\cong} & [\![\Phi]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \\
& \xrightarrow{[\![N]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!]} & \mathbf{T}_i([\![A]\!] \otimes [\![B]\!]) \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!] \\
& \xrightarrow{s} & \mathbf{T}_i([\![A]\!] \otimes [\![B]\!] \otimes [\![\Phi]\!] \otimes [\![\Gamma_2]\!]) \\
& \xrightarrow{\cong} & \mathbf{T}_i([\![\Phi]\!] \otimes [\![\Gamma_2]\!] \otimes [\![A]\!] \otimes [\![B]\!]) \\
& \xrightarrow{\mathbf{T}_i[\![M]\!]} & \mathbf{T}_i \mathbf{T}_j[\![C]\!] \\
& \xrightarrow{i} & \mathbf{T}_{i \vee j}[\![C]\!].
\end{array}$$

□