

Proto-Quipper with dynamic lifting

PENG FU

University of South Carolina, USA
(email: pfu@cse.sc.edu)

KOHEI KISHIDA

University of Illinois Urbana-Champaign, USA
(email: kkishida@illinois.edu)

NEIL J. ROSS

Dalhousie University, Canada
(email: neil.jr.ross@dal.ca)

PETER SELINGER

Dalhousie University, Canada
(email: selinger@mathstat.dal.ca)

Abstract

Quipper is a functional language for programming quantum circuits. Proto-Quipper is a family of languages aiming to provide a formal foundation for Quipper. In this paper, we extend Proto-Quipper-M with a construct called *dynamic lifting*, which is present in Quipper. By virtue of being a circuit description language, Proto-Quipper has two separate runtimes: circuit generation time and circuit execution time. Values that are known at circuit generation time are called *parameters*, and values that are known at circuit execution time are called *states*. Dynamic lifting is an operation that enables a state, such as the result of a measurement, to be lifted to a parameter, where it can influence the generation of the next portion of the circuit. As a result, dynamic lifting enables Proto-Quipper programs to interleave classical and quantum computation. We describe the syntax of a language we call Proto-Quipper-Dyn. Its type system uses a system of modalities to keep track of the use of dynamic lifting. We then provide an operational semantics, as well as a notion of categorical model for dynamic lifting. We also construct a concrete instance of such a model based on *biset-enrichment*. We prove that both the type system and the operational semantics are sound with respect to our categorical semantics. Finally, we give some examples of Proto-Quipper-Dyn programs that make essential use of dynamic lifting.

1 Introduction

1.1 Quipper and Proto-Quipper

Quipper is a functional programming language for quantum computing [16, 17]. The overall aim of Quipper is to allow quantum algorithms to be specified at a level of abstraction that is similar to how the algorithm might be described in a research paper, and to compile this down to the level of individual quantum gates, producing a logical quantum circuit. Quipper has been used to program a set of nontrivial algorithms from the quantum computing literature, and it has been used to generate quantum circuits consisting of trillions of gates. As a circuit description language, Quipper shares some of the traits of hardware description languages. In particular, it has two notions of runtime: The first of these is *circuit generation time*. This is



© 2026 Copyright held by the owner/author(s).

This work is licensed under a Creative Commons Attribution 4.0 International License.

when a Quipper program is run to generate a quantum circuit. The second is *circuit execution time*. This is when a quantum circuit is executed by a quantum computer or a simulator.

Quipper is a practical language, implemented as an embedded language in Haskell. As such, it lacks formal foundations such as operational and denotational semantics. This motivates the development of Proto-Quipper, a family of experimental languages that aim to provide formal semantics for fragments of Quipper. Proto-Quipper-S features a linear type system with subtyping as well as an operational semantics [38]. Proto-Quipper-M has a linear type system without subtyping, but with a sound categorical semantics in addition to its operation semantics [37]. More recently, Proto-Quipper-D was introduced, which features a type system with linear dependent types as well as a fibrational categorical semantics [12, 15].

1.2 Dynamic lifting and the interaction of the two runtimes

Proto-Quipper, like Quipper, distinguishes two runtimes. Moreover, Proto-Quipper gives a formal account of parameters and states. A *parameter* is a value that is known at circuit generation time, such as a boolean value for an if-then-else expression. A *state* is a value that is only known at circuit execution time, such as the actual state of a qubit or classical bit in a circuit. The type system of Proto-Quipper reflects this distinction. Among the types, there is a subset of *parameter types*, such as **Nat** and **Bool**, whose elements can be duplicated and discarded. There is also a subset of *state types*, such as **Qubit** and **Bit**, which are linear so that their elements cannot in general be duplicated or discarded. One of the fundamental design decisions of Proto-Quipper is that parameter types and state types belong to the same universe of types, so that one can form compound types that are part parameter and part state. An example of this is the type **Bool** $\otimes$ **Qubit**, whose elements are pairs of a boolean (a parameter) and a qubit (a state). Another example is the type of lists of qubits. Here, the length of the list is a parameter (known at circuit generation time), but the actual qubits in the list are states (known at circuit execution time). In this way, Proto-Quipper differs, e.g., from QWire, an embedded quantum circuit description language in which parameters and states belong to separate universes [31].

In Quipper, the two runtimes can interact with each other. A priori, it is clear that states can depend on parameters. For example, we can initialize a qubit based on a boolean parameter, simply by inserting a gate at circuit generation time to initialize the qubit in one state or another. The opposite direction is more complicated. Usually, circuit execution happens *after* circuit generation, and in this case, it is clear that a state cannot be converted to a parameter. However, there are some quantum algorithms that require circuit generation and circuit execution to be interleaved. Here, a state, such as the outcome of a measurement in a circuit, may be used to inform the generation of the next part of the circuit. To enable such interleaving, Quipper provides a construct called *dynamic lifting*, which enables a state to be lifted to a parameter in certain situations. For example, dynamic lifting permits the result of a measurement, which is a state of type **Bit**, to be lifted to a parameter of type **Bool**. It is important to note that dynamic lifting is an *expensive* operation, as it requires control to pass from circuit evaluation time back to circuit generation time. This requires the real-time quantum computer to put all of its active qubits into long-term storage while spending an indeterminate amount of time awaiting further instructions from the classical computer in charge of circuit generation.

Dynamic lifting is important because it can be used to express quantum algorithms that require interleaving circuit execution time and circuit generation time. While there are many quantum algorithms that do not require such interleaving, there are some that do. One example is the so-called *repeat-until-success* procedure [30]. It involves the design of a special circuit such that when we measure some particular subset of the qubits, if the measurement results are all zeros, then the unmeasured qubits are in the desirable state; if some measurements are non-zero, then the unmeasured qubits can be returned to their original state and the circuit is run again. This is repeated until all the measurements are zero, i.e., “until success”. In this case, the quantum computer must maintain the state of the unmeasured qubits between measurements.

Another example is *magic state distillation* [5]. Here, the goal is to prepare a qubit in some target state. We start with a large number of qubits, say n of them, each of which is a rough approximation of the target state. We then apply a probabilistic “distillation” procedure which yields on average, say, $n/4$ qubits that are better approximations of the target state; the remaining qubits are wasted. By repeated distillation steps, we eventually wind up with a small number of qubits that are excellent approximations of the target state. In such a situation, dynamic lifting is essential because after each distillation step, we must throw away the wasted qubits, but we do not know ahead of time which ones (or indeed, how many) there will be. Thus which future gates will be applied depends on the outcomes of previous measurements. With the help of dynamic lifting, these algorithms can be naturally expressed as functions in the programming language.

The concept of dynamic lifting is different from measurement, and the two should not be confused. Measurement is merely a gate in a circuit, turning a quantum bit (a state) into a classical bit (also a state). Dynamic lifting is an operation of the programming language, turning a classical bit (a state) into a boolean (a parameter). The common concept of “measurement that returns a result” is measurement-as-a-gate followed by dynamic lifting. There is no conceptual difference between whether it occurs in the middle of a circuit or at the end, except that when dynamic lifting occurs in the middle of a circuit, the quantum computer must store the remaining quantum state, whereas if it occurs at the end of a circuit, there is no remaining quantum state to store. Since the writing of this paper, IBM has implemented dynamic lifting on some of their newer hardware and Qiskit has been updated to support it under the name of “dynamic circuits” [18].

1.3 A type system for dynamic lifting

Previous versions of Proto-Quipper lacked dynamic lifting. Modeling dynamic lifting is a challenging problem. To better understand the issues involved, it is useful to know that there are two things that can be done with circuits in Proto-Quipper. On the one hand, circuits can be run on a quantum device. On the other hand, circuits can be *boxed*. A boxed circuit is a data structure that contains a circuit that has *already been generated*, i.e., an actual list of gates, rather than merely instructions for how to generate such a list. As a result, a boxed circuit can be used as a building block for all kinds of things. In the simplest case, it can be re-used in the construction of other circuits. But once boxed circuits are available, one can also have other operations on them. Many such operations are implemented in Quipper, although we do not specifically focus on them in this paper. Examples include: applying gate transformations (systematically replacing gates by other gates), adding things like error

correction, or rewriting the circuit to simplify it. Boxed circuits can also be reversed, which is used in many quantum algorithms, for example to uncompute ancillas. The ability to box circuits is crucial to Quipper’s ability to express algorithms at a natural level of abstraction, because algorithms are often described in terms of meta-operations on circuits.

Now it is clear that dynamic lifting only makes sense in the context of a circuit that is actually being executed, rather than one that is merely being boxed. We will keep track of this in the programming language by adding a modality to the type system and a corresponding monad to the semantics. The modality should be thought of as denoting “boxability”. For example, a function of type $\mathbf{Qubit} \multimap_1 \mathbf{Qubit}$ represents a circuit that can be boxed or executed, i.e., that does not use dynamic lifting, whereas a function of type $\mathbf{Qubit} \multimap_0 \mathbf{Qubit}$ represents a quantum operation that can only be executed but not boxed.

Before we can describe the operational or denotational semantics of Proto-Quipper-Dyn, we must be more precise about what we mean by a “circuit”. We must also specify what it means to “execute” a circuit. There are many different notions of circuits, differing, for example, in which collection of gates is provided. Rather than specializing to one of these, we take a more general point of view: a *circuit* is simply a morphism in a small symmetric monoidal category $\mathbf{M}$, which we assume to be given ahead of time, but otherwise arbitrary (subject to some properties). Similarly, for the execution of circuits, we assume that we are given another small symmetric monoidal category $\mathbf{Q}$ of *quantum operations*. Conceptually, the morphisms of $\mathbf{M}$ are *syntactic* entities; thus, $\mathbf{M}$ is typically a category that is freely generated by a collection of wire types such as $\mathbf{Bit}$ and $\mathbf{Qubit}$, and a collection of gates. On the other hand, we think of the morphisms of $\mathbf{Q}$ as *physical* operations, which can be performed on a quantum computer. The categories $\mathbf{M}$ and $\mathbf{Q}$ have the same objects, and there is a symmetric monoidal *interpretation functor* $J : \mathbf{M} \rightarrow \mathbf{Q}$.

Operationally, dynamic lifting is an operation that reads the state of a bit in $\mathbf{Q}$, and returns a boolean value. Since a bit state can be the result of a measurement, the read operation for dynamic lifting is nondeterministic, i.e., it can return different boolean values with probabilities governed by measurements. The nondeterministic nature of the dynamic lifting suggests that it should be modeled as a monadic operation [27].

We therefore conceptualize the types of Proto-Quipper-Dyn as the objects of a single category $\mathbf{A}$, with a monad $T : \mathbf{A} \rightarrow \mathbf{A}$, called the *dynamic lifting monad*. This will be done in such a way that $\mathbf{M}$ is fully embedded in $\mathbf{A}$, and $\mathbf{Q}$ is fully embedded in the Kleisli category $Kl_T(\mathbf{A})$, in a way that makes the following diagram commute.

$$\begin{array}{ccc} \mathbf{M} & \xhookrightarrow{\psi} & \mathbf{A} \\ \downarrow J & & \downarrow E \\ \mathbf{Q} & \xhookrightarrow{\phi} & Kl_T(\mathbf{A}) \end{array}$$

Here, J is the given interpretation functor, and E is the canonical functor from $\mathbf{A}$ to $Kl_T(\mathbf{A})$. We then model dynamic lifting as a map $\text{dynlift} : \mathbf{Bit} \rightarrow T \mathbf{Bool} \in Kl_T(\mathbf{A})$, which lifts a state ($\mathbf{Bit}$) to a parameter ($\mathbf{Bool}$), such that the following diagram commutes.

$$\begin{array}{ccc} & & \mathbf{Bit} \\ & \nearrow \text{init} & \downarrow \text{dynlift} \\ \mathbf{Bool} & \xrightarrow{\eta} & T \mathbf{Bool} \end{array}$$

Note that dynamic lifting is a morphism of the Kleisli category; this makes sense because it is essentially a side-effecting read operation. More generally, any computation that potentially uses dynamic lifting will have type $A \rightarrow TB$.

As mentioned above, our type system must also distinguish quantum circuits that are being executed from quantum circuits that are being boxed. Naturally, since the latter may not use dynamic lifting, they are maps in the category $\mathbf{A}$ while the former are maps in the Kleisli category $Kl_T(\mathbf{A})$. As a practical matter for programmer convenience, it would be awkward to have T as an explicit type constructor that must be mentioned everywhere in the program. Instead, we use a system of modalities to keep track of the dynamic lifting monad T . More specifically, we annotate a typing judgment with a modality, i.e., $\Gamma \vdash_\alpha M : A$, where $\alpha \in \{0, 1\}$. When $\alpha = 0$, it means that the term M represents a morphism $[\![\Gamma]\!] \rightarrow T[\![A]\!]$ in the Kleisli category $Kl_T(\mathbf{A})$. When $\alpha = 1$, it means that the term M represents a morphism $[\![\Gamma]\!] \rightarrow [\![A]\!]$ in $\mathbf{A}$. An example of the typing rule for dynamic lifting is the following (where $\text{Meas} : \mathbf{Qubit} \rightarrow \mathbf{Bit}$ represents the measurement gate).

$$\frac{\ell : \mathbf{Qubit} \vdash_1 \text{Meas}(\ell) : \mathbf{Bit}}{\ell : \mathbf{Qubit} \vdash_0 \text{dynlift}(\text{Meas}(\ell)) : \mathbf{Bool}}$$

If we have a quantum circuit $\mathbf{Qubit} \rightarrow \mathbf{Bit}$, it can be run by a quantum computer and the measurement result of type $\mathbf{Bit}$ will be lifted to a parameter of type $\mathbf{Bool}$. Note that the `dynlift` operation sets the modality of the typing judgment to 0, and as a result, we have a map $\mathbf{Qubit} \rightarrow T\mathbf{Bool}$ in the Kleisli category. The use of modalities in our type system ensures that the term $\text{Meas}(\ell)$ can be turned into a boxed circuit, whereas it will be a compile time typing error to try to box the term $\text{dynlift}(\text{Meas}(\ell))$.

1.4 Operational semantics

Next, let us take a look at the operational semantics of Proto-Quipper-Dyn. In previous versions of Proto-Quipper, the operational semantics used configurations of the form (C, M) , where C is the circuit being currently constructed, and M is a term. On the other hand, in the quantum lambda calculus [40], which is not a circuit construction language but intended to run directly on a quantum computer, the operational semantics used configurations of the form (Q, M) , where Q is the current quantum state and M is a term.

In a sense, Proto-Quipper-Dyn is a combination of these prior languages: it is a language for circuit construction (via the *boxing* operation), but it is also a language for running quantum operations (as otherwise dynamic lifting would not be possible). Consequently, our operational semantics uses *both* kinds of configurations: those of the form (Q, M) are only used for top-level computations that actually run on a quantum device, and those of the form (C, M) are used during boxing. These two kinds of configurations correspond closely to the two runtimes, since configuration of the form (C, M) are used for circuit construction and those of the form (Q, M) are used for circuit execution. They also correspond to the two categories $\mathbf{M}$ and $\mathbf{Q}$.

Consequently, the evaluation rules take two different forms. Evaluation at circuit generation time takes the form $(C, M) \Downarrow (C', V)$, where C is a circuit, M is a term, and V is a value. The type system ensures that such an evaluation does not involve dynamic lifting, so it can be done entirely with a classical computer and the evaluation is deterministic. On the other hand, evaluation at circuit execution time takes the form $(Q, M) \Downarrow \sum_i p_i(Q_i, V_i)$, where Q represents a quantum state. Since M can use dynamic lifting, the result of such an evaluation

rule is probabilistic, with outcome (Q_i, V_i) happening with probability p_i . Note that the probabilities here arise from measurements.

1.5 Related work

The feature of dynamic lifting first appeared in the Quipper programming language [16]. The various versions of Proto-Quipper [15, 37] each formalize some aspects of Quipper, but the present Proto-Quipper-D is the first of these to include dynamic lifting in the formalization.

A common misunderstanding is that dynamic lifting means performing measurements during the execution of a quantum program. That would not be a new feature; indeed, the ability to perform on-the-fly measurements was already present in the earliest quantum programming languages, such as Ömer’s QCL [1998], Selinger’s quantum flow charts [2004] and Selinger and Valiron’s quantum lambda calculus [2009]. Rather, dynamic lifting is an operation that only makes sense in the context of a *circuit description language*, where circuits are not executed during the circuit generation phase. Dynamic lifting is the transfer of information from the circuit execution environment back to the circuit generation environment. Therefore, in the following discussion of related work, we do not include comparisons with most papers on languages that include measurement but do not have separate circuit generation and circuit execution times.

One of the features of the present work, and of Proto-Quipper in general, is that it works with the standard notion of quantum circuits [28], which are basically lists of gates, or more precisely, gates that have been composed using the laws of symmetric monoidal categories. By contrast, some of the other notions of dynamic lifting that appear in the literature (and which we discuss in more detail below) not only add features to the programming language, but also to the generated circuits themselves. Relatedly, one of the features that makes boxed circuits useful in Proto-Quipper is that they are actual data structures. Here, by a “data structure”, we mean data that can be queried, for example via a case distinction or pattern matching. This is different from a “thunk”, such as a lambda abstraction, which represents a suspended computation. Some of the alternative notions of dynamic lifting that appear in the literature make dynamic lifting part of the circuit language, allowing circuits containing dynamic lifting to be boxed. This turns circuits into thunks.

The original related work is of course Quipper [16], where the concept of dynamic lifting was first introduced. One of the main shortcomings of Quipper is that, as an embedded language, it is not strongly typed; in Quipper, it is a runtime error when one attempts to box a circuit that contains dynamic lifting. By contrast, in Proto-Quipper-D, it is a typing error.

In recent work, Lee et al. [23] extended Proto-Quipper with a version of dynamic lifting. They work with a single runtime modeled by a category of *quantum channels*, which are generalizations of quantum circuits with a notion of branching for measurement results. A quantum channel is a list of gates like a quantum circuit, with the important exception that if the current gate is a measurement, the list has two tails, one for each possible measurement outcome. Consequently, the channels of Lee et al. must either be implemented as thunks, or as data structures that are exponentially large. The main difference with our work is that in our setting, the type system ensures that circuits containing dynamic lifting cannot be boxed, and therefore boxed circuits are data structures that contain only one branch.

Another version of Proto-Quipper incorporating a form of dynamic lifting was proposed by Colledan and Lago [6]. Their language uses a very general version of dynamic lifting,

which is even more general than the one present in the Quipper language, and allows for the output *type* of their circuit to depend on the outcomes of the measurements specified in the computation. They also work with a single runtime where dynamic lifting is part of their generalized notion of quantum circuits. While this alternative notion of dynamic lifting is interesting in its own right, their language does not come equipped with a denotational semantics.

QWire [31, 33, 34] is a quantum programming language that also supports dynamic lifting. QWire has a host language and a circuit language. The host language describes the computation of the classical computer, while the circuit language describes the computation of the quantum computer. QWire has a denotational semantics for the circuit language, but not for the host language. Dynamic lifting is part of the syntax of the circuit language, and therefore, QWire’s notion of quantum circuits differs from Proto-Quipper’s notion. Besides dynamic lifting, QWire also has a notion of *static lifting* in the form of a “run” function. This allows measuring all the qubits in a circuit, returning boolean values to the host language, without leaving any unmeasured quantum state. By contrast, Proto-Quipper-Dyn does not require a run function, since it does not have separate host and circuit languages. All circuits that are not constructed inside a box are automatically executed, and dynamic lifting can be used to bring measurement results into the control flow of the language.

We use enriched category theory to construct a categorical model for dynamic lifting. There are some existing works that also use enriched categories in the context of quantum programming languages. For example, Lindenhovius et al. [24] use CPO-enrichment to model a version of Proto-Quipper-M with recursion. The main difference between our model and that of Lindenhovius et al. is that our model accounts for dynamic lifting while their model accounts for recursion. Rennela and Staton [35] give a categorical model for a QWire-like language that also uses enriched categories. Their language allows boxing a circuit that uses dynamic lifting, which is quite different from how boxing works in Proto-Quipper. As a result, circuits in their setting are thunks and not data structures. Also, in Rennela and Staton’s EWire language, the host language does not include wire types such as a type of qubits, whereas Proto-Quipper does not have separate host and circuit languages, and includes all types in a single language. Consequently, Proto-Quipper has a linear type system, whereas the EWire host language does not. This difference is also reflected in the model: in Rennela and Staton’s semantics, programs are interpreted in a cartesian-closed category, whereas in our model, they are interpreted in a monoidal category.

The fact that Proto-Quipper has two distinct runtimes (circuit generation time and circuit execution time) suggests a possible connection to another computational paradigm that also has multiple runtimes, namely *multi-staged computation* [26, 41], and more recently, *two-level type theory* [1, 21]. However, there are some important differences. One of them is that multi-staged computation, such as in MetaML [41], deals with potentially many levels, but all of the levels share the same operations and the same hardware; the primary purpose of staging is to precisely orchestrate the order in which operations are evaluated. On the other hand, the main purpose of dynamic lifting in Proto-Quipper is to interleave computations from two different hardware models. The “meta-language” of Proto-Quipper terms has almost nothing in common with the “object language” of circuits. Each of the two stages has its own distinct operations: classical expressions and control flow for the meta-language, and gates

and measurements for the object language. In particular, quantum circuits are not just code for expressions of the meta-language.

Finally, we will mention the quantum programming language Silq [3]. Like Proto-Quipper-Dyn, Silq also uses modalities to keep track of the use of certain operations. For example, the modality “mfree” in Silq is used to indicate whether a computation uses measurement. The difference is that Silq is not a circuit description language, so it does not have a notion of boxed circuits or dynamic lifting.

1.6 Contributions

In this paper, we describe the syntax and type system of an extension of Proto-Quipper with dynamic lifting, called Proto-Quipper-Dyn. The type system uses a system of modalities to keep track of the use of dynamic lifting. We also provide an operational semantics, using two different kinds of configurations to model circuit generation time and circuit execution time. We further provide an abstract categorical semantics for this language, in which dynamic lifting is modeled by a map $\mathbf{Bit} \rightarrow T \mathbf{Bool}$, where T is a monad encapsulating circuit execution. By an “abstract” categorical semantics, we mean that we only state the properties that a categorical model must satisfy to give a sound interpretation of the language. We also give a construction of a concrete categorical model. Lastly, we have implemented a prototype for Proto-Quipper-Dyn, which is available at [11].

This paper is an extended version of two conference papers [13, 14], with more details and proofs. Compared to this earlier work, we have further simplified and improved the categorical semantics. Unlike in Fu et al. [14], we no longer require enrichment in the abstract categorical semantics, which allows us to simplify the axioms. We still use enrichment to construct the concrete categorical model.

The rest of the paper is organized as follows: In Section 2, we give some examples of dynamic lifting in Proto-Quipper-Dyn. In Section 3, we give an axiomatization of a categorical semantics for dynamic lifting. In Section 4, we define a type system for dynamic lifting that uses a system of modalities. We then show how a typing judgment with modalities is interpreted as a morphism in our categorical model. In Section 5, we define a call-by-value big-step operational semantics for our language. We show that the operational semantics satisfies type preservation and that the type system guarantees error freeness. We also show that the operational semantics is sound with respect to the categorical semantics. In Section 6, we construct a concrete categorical model based on biset-enrichment. We finish the paper with some concluding remarks in Section 7.

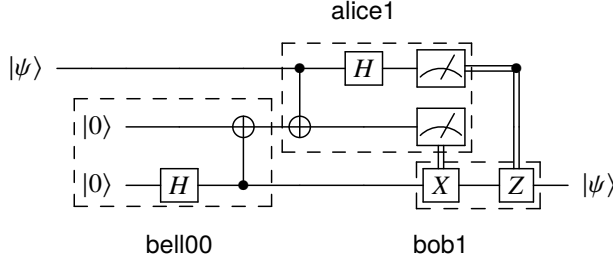
2 Dynamic lifting in Proto-Quipper

We now discuss several Proto-Quipper-Dyn programs that make use of dynamic lifting. A prototype implementation of Proto-Quipper-Dyn is available at [11], and the programs in this section have been tested with that implementation. In Section 2.1, we give an example of quantum teleportation. In Section 2.2, we give an example of magic state distillation, and in Section 2.3, we give an example of repeat-until-success circuits.

Note that several of the following example programs make use of recursion. While we do not formally treat recursion in this paper, it is included in the prototype implementation. See Section 7 for more discussion.

2.1 Quantum teleportation

The following circuit implements a one-qubit quantum teleportation protocol.



This circuit is generated by the following Proto-Quipper-Dyn functions.

```
alice1 : !(Qubit -> Qubit -> Bit * Bit)
```

```
alice1 a q =
```

```
  let (a, q) = CNotGate a q
```

```
    q = HGate q
```

```
  in (Meas a, Meas q)
```

```
bob1 : !(Qubit -> Bit -> Bit -> Qubit)
```

```
bob1 q x y =
```

```
  let (q, x) = C_XGate q x
```

```
    (q, y) = C_ZGate q y
```

```
    _ = Discard x
```

```
    _ = Discard y
```

```
  in q
```

```
bell00 : !(Unit -> Qubit * Qubit)
```

```
bell00 u =
```

```
  let a = Init0 ()
```

```
    b = Init0 ()
```

```
  in CNotGate b (HGate a)
```

```
tele1 : !(Qubit -> Qubit)
```

```
tele1 q =
```

```
  let (b, a) = bell00 ()
```

```
    (x, y) = alice1 a q
```

```
    z = bob1 b x y
```

```
  in z
```

```
boxTele : Circ(Qubit, Qubit)
```

```
boxTele = box Qubit tele1
```

In the above functions, the `!` indicates a type is reusable. `Discard` is a gate of type `!(Bit  $\rightarrow$  Unit)`. As its name indicates, it discards a bit. The modality information is not visible to the programmer. Because the programs do not use dynamic lifting, the modalities in the fully annotated types are all 1. For example, the fully annotated type of `tele1` is

$!_1(\mathbf{Qubit} \multimap_1 \mathbf{Qubit})$. We can therefore box `tele1` into a quantum circuit. The evaluation of `boxTele` occurs on a classical computer and generates the circuit diagram above.

For comparison, let us consider the following Proto-Quipper-Dyn programs that implement quantum teleportation using dynamic lifting.

```
alice2 : !(Qubit -> Qubit -> Bool * Bool)
alice2 a q =
  let (a, q) = CNotGate a q
      q = HGate q
  in (dynlift (Meas a), dynlift (Meas q))

bob2 : !(Qubit -> Bool -> Bool -> Qubit)
bob2 q x y =
  let q = if x then XGate q else q
      q = if y then ZGate q else q
  in q

tele2 : !(Qubit -> Qubit)
tele2 q =
  let (b, a) = bell00 ()
      (x, y) = alice2 a q
      z = bob2 b x y
  in z

test : Bool
test = dynlift (Meas (tele2 (Init0 ())))

-- The following causes a typing error
boxAttempt : Circ(Qubit, Qubit)
boxAttempt = box Qubit tele2
```

As before, the code contains no modality annotations. In the `alice2` function, dynamic lifting is used right after the measurement gate `Meas : Qubit  $\rightarrow$  Bit`. Accordingly, the fully annotated type of `alice2` is $!_1(\mathbf{Qubit} \multimap_1 \mathbf{Qubit} \multimap_0 \mathbf{Bool} * \mathbf{Bool})$. The `bob2` function then uses if-then-else expressions to decide whether to apply the gates `QNot` and `ZGate`, rather than applying the bit-controlled gates `C.XGate` and `C.ZGate`, as in the `bob1` function.

The `tele2` function calls the `bob2` function with the booleans provided by the `alice2` function. Hence, the `tele2` function implicitly uses dynamic lifting. Its fully annotated type is $!_1(\mathbf{Qubit} \multimap_0 \mathbf{Qubit})$. Because of the modality inference, the type checker will issue a typing error for the `boxAttempt` function. According to the typing rule for `box`, the `box Qubit` function requires an argument of type $!_1(\mathbf{Qubit} \multimap_1 \mathbf{Qubit})$, which is distinct from the type of `tele2`. This error is sensible because the `tele2` function does not correspond to a circuit.

The `test` function applies `tele2` to an input qubit in the $|0\rangle$ state. The output value of `test` should then be `False` with probability 1. Note that the evaluation of `test` requires access to a quantum computer or a simulator.

2.2 Magic state distillation

Magic states are quantum states that can be used, in conjunction with Clifford gates, to perform universal quantum computing fault tolerantly [5]. For example, there is a standard method to implement a T gate using the magic state $(|0\rangle + e^{\frac{\pi i}{4}}|1\rangle)/\sqrt{2}$, along with Clifford gates and measurements. This enables the application of any operation from the Clifford+ T gate set, a well-known universal set of quantum gates [28].

The process of producing a magic state such as $(|0\rangle + e^{\frac{\pi i}{4}}|1\rangle)/\sqrt{2}$ from several imperfect states is called *magic state distillation* [5]. In order to distill a magic state $|M\rangle$, one first prepares several qubits in a state that approximates $|M\rangle$ up to an error rate ϵ . A carefully designed quantum circuit is then applied to these qubits and some of them are measured. If all of the measurement results are 0, then the remaining qubits are guaranteed to be in a state that approximates $|M\rangle$ up to an improved error rate $\epsilon' < \epsilon$. If any one of the measurement results is 1, then all of the qubits are discarded and the entire process is restarted. In practice, several rounds of distillation are required to obtain a state that approximates $|M\rangle$ up to an acceptable error rate.

A Proto-Quipper-Dyn implementation of Bravyi and Kitaev's distillation algorithm is given in the following.

```
-- A five qubit error correcting code.
fiveQubits : ! (Qubit * Qubit * Qubit * Qubit * Qubit
               -> Qubit * Qubit * Qubit * Qubit * Qubit)

fiveQubits input =
  let (a1, a2, a3, a4, a5) = input
    (a1, a5) = CZ a1 a5
    (a2, a5) = CZ a2 a5
    (a3, a5) = CZ a3 a5
    (a2, a4) = CZ a2 a4
    (a3, a4) = CZ a3 a4
    (a5, a4) = CY a5 a4
    (a1, a3) = CZ a1 a3
    (a4, a3) = CZ a4 a3
    (a5, a3) = CY a5 a3
    (a1, a2) = CZ a1 a2
    (a5, a2) = CY a5 a2
    (a3, a1) = CZ a3 a1
    (a5, a1) = CNotGate a5 a1
  in (HGate a1, HGate a2, HGate a3, HGate a4, ZGate a5)

prepMixedState : ! (Unit -> Qubit)
prepMixedState u = TGate (HGate (Init0 ()))

distill : !(Qubit * Qubit * Qubit * Qubit * Qubit -> Maybe Qubit)
distill input =
  let (a1, a2, a3, a4, a5) = fiveQubits input
    a1' = dynlift (Meas a1)
```

```

    a2' = dynlift (Meas a2)
    a3' = dynlift (Meas a3)
    a4' = dynlift (Meas a4)
  in if a1' || a2' || a3' || a4'
    then let a = dynlift (Meas a5) in Nothing
    else Just a5

distillation : !(Nat -> Qubit)
distillation n =
  case n of
    Z -> prepMixedState ()
    S n' ->
      let q1 = distillation n'
          q2 = distillation n'
          q3 = distillation n'
          q4 = distillation n'
          q5 = distillation n'
      in
        case distill (q1, q2, q3, q4, q5) of
          Nothing -> distillation n
          Just q -> q

```

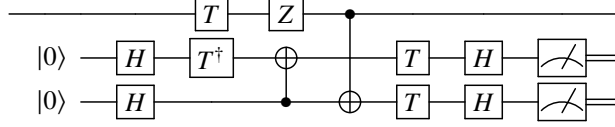
In the `distill` function, we first apply a five-qubit error correction circuit `fiveQubits` to the inputs, then measure the qubits and, through dynamic lifting, promote the resulting bits to booleans. If all of the booleans are `False`, the distillation was successful and we return the remaining qubit. Otherwise the distillation failed, so we discard the unmeasured qubit and return nothing. Dynamic lifting is essential for defining the `distill` function because in this case the if-then-else expression cannot be implemented as a circuit.

The `distillation` function performs n rounds of magic state distillation. The function `prepMixedState` prepares an initial imperfect state. The `distillation` function is a recursive function that assumes five successful distillations from the previous round and then applies the `distill` function to the resulting qubits. If that function returns a qubit, the n -th round of distillation was successful, otherwise it will restart the whole process.

2.3 Repeat-until-success

The *repeat-until-success* paradigm provides a technique to apply a unitary that cannot be implemented exactly, at the cost of potentially running the same circuit multiple times. In order to apply a non-Clifford+T gate N to a target qubit $|\phi\rangle$, one first initializes several ancillary qubits before applying a well-chosen Clifford+T circuit C to the target and the ancillas and measuring the ancillas. If all of the measurement results are 0, the target qubit is guaranteed to be in the state $N|\phi\rangle$. Otherwise, a correction is applied to the target to return it to its initial state and the process is repeated.

Consider the following circuit used by Paetznick and Svore [30] to illustrate the implementation of the gate $V_3 = \frac{I+2iZ}{\sqrt{5}}$ using the repeat-until-success method.



The top wire is the target qubit, while the wires below it are the ancillas. We apply a sequence of gates (H , H , T^* , $CNOT$, T , and H) to the ancillas before measuring the first ancilla. If, as we assume here, the measurement result is 0, then we apply a sequence of gates (T , Z , $CNOT$, T , and H) to the target qubit and the second ancilla before measuring the second ancilla. Assuming, again, that the measurement result is 0, we then know that the target qubit is in the desired state. Note that the circuit above is not a representation of the entire repeat-until-success protocol. Instead, it is the circuit constructed in the event that both measurement results are 0 (which can be shown to occur with probability $5/8$). If the measurements yield different results, the circuit constructed by the repeat-until-success protocol is different. For example, if the result of the second measurement is 1, a Z gate must be applied to the target qubit to return it to its initial state.

The following function gives a precise description of the implementation of V_3 in Proto-Quipper-Dyn.

```
v3 : !(Qubit -> Qubit)
v3 q =
  let a1 = TGate_inv (HGate (Init0 ()))
      a2 = HGate (Init0 ())
      (a1, a2) = CNotGate a1 a2
      a1 = HGate (TGate a1)
  in if dynlift (Meas a1)
    then
      let _ = Discard (Meas a2)
      in v3 q
    else let q = ZGate (TGate q)
          (a2, q) = CNotGate a2 q
          a2 = H (TGate a2)
          in if dynlift (Meas a2)
            then v3 (ZGate q)
            else q
```

Once again, dynamic lifting plays an essential role here. Note that the $v3$ function has type $!_1(\mathbf{Qubit} \multimap_0 \mathbf{Qubit})$; it is a quantum computation, rather than a quantum circuit.

3 A categorical semantics for dynamic lifting

In this section we will give a general categorical semantics for dynamic lifting. Our definition takes the form of a sequence of properties (a)-(f), which will be dispersed throughout this section, culminating in Definition 3.5 of a model for Proto-Quipper with dynamic lifting.

Definition 3.1. A category $\mathbf{A}$ is a *linear-non-linear programming language model* if

- (a) $\mathbf{A}$ has coproducts and is symmetric monoidal closed, i.e., it is symmetric monoidal and there is an adjunction $- \otimes A \dashv A \multimap -$ for each $A \in \mathbf{A}$.
- (b) $\mathbf{A}$ is equipped with an adjunction

$$p : \mathbf{Set} \rightarrow \mathbf{A} \dashv b : \mathbf{A} \rightarrow \mathbf{Set}$$

such that p is a strong monoidal functor.

- Remarks.**
- The requirement that $\mathbf{A}$ has coproducts and is symmetric monoidal closed implies that it can model function types and sum types in a functional programming language. Moreover, since $- \otimes A$ is a left adjoint functor for any $A \in \mathbf{A}$, it preserves the coproducts, so the tensor products distribute over coproducts in $\mathbf{A}$.
 - The adjunction in (b) is often called a *linear-non-linear adjunction* [2]. Here, the assumption that p is a strong monoidal functor means that there exist isomorphisms $e : I \rightarrow p1$ and $m : pX \otimes pY \rightarrow p(X \times Y)$ making some diagrams commute (see Appendix A).
 - Since p is strong monoidal and $\mathbf{Set}$ is cartesian, for any $X \in \mathbf{Set}$, there are maps $\text{discard}_X : pX \rightarrow I$ and $\text{dup}_X : pX \rightarrow pX \otimes pX$ in $\mathbf{A}$. Moreover, for any map $f : X \rightarrow Y$ in $\mathbf{Set}$, we have the following in $\mathbf{A}$.

$$\text{dup}_Y \circ pf = (pf \otimes pf) \circ \text{dup}_X$$

We call objects of the form $pX \in \mathbf{A}$ *parameter objects*, since they can be duplicated and discarded. For example, $\mathbf{Bool} := I + I \cong p1 + p1 \cong p(1 + 1)$ is a parameter object.

- For any $X \in \mathbf{Set}$, $B \in \mathbf{A}$, we write δ for the isomorphism $\delta : \mathbf{A}(pX, B) \cong \mathbf{Set}(X, bB)$, and force_B for the counit $\text{force}_B : pbB \rightarrow B$.
- Because $p : \mathbf{Set} \rightarrow \mathbf{A}$ is the left adjoint and it is strong monoidal, we can deduce that

$$p(X) \cong p\left(\sum_{x \in X} 1\right) \cong \sum_{x \in X} p(1) \cong \sum_{x \in X} I.$$

Due to the adjunction $p \dashv b$, we also have $bB \cong \mathbf{Set}(1, bB) \cong \mathbf{A}(I, B)$.

Definition 3.2. A *convex space* is a set X equipped with a *convex sum* operation, which assigns to any $x, y \in X$ and $p, q \in [0, 1]$ such that $p + q = 1$ an element $px + qy \in X$, subject to certain standard conditions, which are detailed in Appendix B. A category is *enriched in convex spaces* if each hom-set is equipped with the structure of a convex space, and moreover, composition is *bilinear* with respect to convex sum, i.e., $(pf + qg) \circ h = p(f \circ h) + q(g \circ h)$ and $h \circ (pf + qg) = p(h \circ f) + q(h \circ g)$.

As mentioned in the introduction, Proto-Quipper-Dyn is parameterized by two small categories $\mathbf{M}$ and $\mathbf{Q}$ of *circuits* and *quantum operations*. Since general quantum operations, such as measurement, are probabilistic, we assume that each set of morphisms in $\mathbf{Q}$ is *convex*. On the other hand, convexity is not assumed for the category $\mathbf{M}$. We now specify the properties that these categories must satisfy.

Assumption 3.3. We assume that we are given two small symmetric monoidal categories $\mathbf{M}$ and $\mathbf{Q}$, satisfying the following properties:

- (1) $\mathbf{M}$ and $\mathbf{Q}$ have the same objects, including a distinguished object called **Bit**. The category $\mathbf{M}$ has distinguished morphisms zero, one : $I \rightarrow \mathbf{Bit}$.

- (2) $\mathbf{Q}$ has a coproduct $\mathbf{Bit} = I + I$, and the tensor product in $\mathbf{Q}$ distributes over this coproduct.
- (3) There is a given strict symmetric monoidal functor $J : \mathbf{M} \rightarrow \mathbf{Q}$ that is the identity on objects and $J(\text{zero}) = \text{inj}_1 : I \rightarrow I + I$, $J(\text{one}) = \text{inj}_2 : I \rightarrow I + I$. We call J the *interpretation functor*.
- (4) The category $\mathbf{Q}$ is enriched in convex spaces.
- (5) For any $A \in \mathbf{Q}$, and $f : I \rightarrow \mathbf{Bit} \otimes A \in \mathbf{Q}$, we have $f = p_1(\text{inj}_1 \otimes f_1) + p_2(\text{inj}_2 \otimes f_2)$, where $\text{inj}_1, \text{inj}_2 : I \rightarrow I + I$ and $p_1, p_2 \in [0, 1]$ are uniquely determined real numbers such that $p_1 + p_2 = 1$. When $p_i \neq 0$, the map $f_i : I \rightarrow A$ is also unique.

The categories $\mathbf{M}$ and $\mathbf{Q}$ are not only used in the categorical semantics, but also in the operational semantics of Proto-Quipper-Dyn (i.e., to run the program, we must know what a circuit is and what a quantum operation is). Therefore, these categories should be regarded as given as part of the language specification, rather than as a degree of freedom in the semantics. On the other hand, nothing in the operational or denotational semantics depends on particular properties of $\mathbf{M}$ and $\mathbf{Q}$ other than properties (1)–(5) above. Therefore, Proto-Quipper-Dyn can handle a wide variety of possible circuit models and physical execution models.

In practice, the category $\mathbf{M}$ will be a category of quantum circuits and the category $\mathbf{Q}$ will be a category of quantum operations. These categories will typically have additional objects, such as **Qubit** and perhaps **Qutrit**, and additional morphisms, such as $H : \mathbf{Qubit} \rightarrow \mathbf{Qubit}$ and $\text{Meas} : \mathbf{Qubit} \rightarrow \mathbf{Bit}$. Assumption (5) means that any morphism with domain I and a bit state in its codomain is a convex sum of two morphisms. This property is used in the rule for dynamic lifting in the operational semantics.

To support dynamic lifting, we require the following monad, which allows us to combine the behaviors of $\mathbf{M}$ and $\mathbf{Q}$ into a single category.

Definition 3.4. A category $\mathbf{A}$ satisfying (a) and (b) has a *dynamic lifting monad* T if the following holds.

- (c) T is a commutative strong monad on $\mathbf{A}$. For any $A, B \in \mathbf{A}$, we write $t_{A,B} : A \otimes TB \rightarrow T(A \otimes B)$ for the strength and $s_{A,B} : TA \otimes B \rightarrow T(A \otimes B)$ for the costrength. As usual, we write $\eta : A \rightarrow TA$ and $\mu : T^2A \rightarrow TA$ for the structural maps of the monad.
- (d) The Kleisli category $Kl_T(\mathbf{A})$ of the monad T is enriched in convex spaces. In other words, for any $A, B \in \mathbf{A}$, if $f, g : A \rightarrow TB$ and $p, q \in [0, 1]$, $p + q = 1$, then there exists a convex sum $pf + qg : A \rightarrow TB$. Moreover, for any $h : C \rightarrow TA$, $e : B \rightarrow TD$, we have the following:

$$\begin{aligned} \mu \circ T(pf + qg) \circ h &= p(\mu \circ Tf \circ h) + q(\mu \circ Tg \circ h) \\ \mu \circ Te \circ (pf + qg) &= p(\mu \circ Te \circ f) + q(\mu \circ Te \circ g). \end{aligned}$$

- (e) There are fully faithful embeddings $\mathbf{M} \xrightarrow{\psi} \mathbf{A}$ and $\mathbf{Q} \xrightarrow{\phi} Kl_T(\mathbf{A})$. These embedding functors are strong monoidal, and ϕ preserves the convex sum. Moreover, the following diagram commutes for any $S, U \in \mathbf{M}$.

$$\begin{array}{ccc} \mathbf{M}(S, U) & \xrightarrow{\psi_{S,U}} & \mathbf{A}(S, U) \\ \downarrow J_{S,U} & & \downarrow E_{S,U} \\ \mathbf{Q}(S, U) & \xrightarrow{\phi_{S,U}} & Kl_T(\mathbf{A})(S, U) \end{array}$$

Here, $E : \mathbf{A} \rightarrow Kl_T(\mathbf{A})$ is the canonical embedding of $\mathbf{A}$ into the Kleisli category, which is the identity on objects.

- (f) There are maps $\text{dynlift} : \mathbf{Bit} \rightarrow T\mathbf{Bool}$ and $\text{init} : \mathbf{Bool} \rightarrow \mathbf{Bit}$ in $\mathbf{A}$ such that the following diagram commutes.

$$\begin{array}{ccc} & \mathbf{Bit} & \\ \text{init} \nearrow & & \downarrow \text{dynlift} \\ \mathbf{Bool} & \xrightarrow{\eta} & T\mathbf{Bool} \end{array}$$

Here, $\mathbf{Bool} = pI + pI$, and $\mathbf{Bit}$ is the object of $\mathbf{M}$ and $\mathbf{Q}$ described in Assumption 3.3.

Remarks

- For any $S, U \in \text{obj}(\mathbf{M})$, the map $\text{unbox} : \mathbf{M}(S, U) \rightarrow \mathbf{b}(S \multimap U)$ is defined by

$$\mathbf{M}(S, U) \xrightarrow{\cong} \mathbf{A}(S, U) \xrightarrow{\text{curry}} \mathbf{A}(I, S \multimap U) \xrightarrow{\cong} \mathbf{b}(S \multimap U).$$

Similarly, the map $\text{box} : \mathbf{b}(S \multimap U) \rightarrow \mathbf{M}(S, U)$ is defined by

$$\mathbf{b}(S \multimap U) \xrightarrow{\cong} \mathbf{A}(I, S \multimap U) \xrightarrow{\text{uncurry}} \mathbf{A}(S, U) \xrightarrow{\cong} \mathbf{M}(S, U).$$

So box and unbox form an isomorphism between $\mathbf{M}(S, U)$ and $\mathbf{b}(S \multimap U)$.

- If a category $\mathbf{A}$ satisfies (a), (b), and has a fully faithful embedding $\psi : \mathbf{M} \rightarrow \mathbf{A}$ (the first part of (e)), then it is a model for Proto-Quipper *without* dynamic lifting. For example, the category $\overline{\mathbf{M}}$ of Rios and Selinger [37] is such a model. The rest of the structure, concerning $\mathbf{Q}$ and the monad T , is what is needed to support dynamic lifting.

- The objects of the Kleisli category $Kl_T(\mathbf{A})$ are the same as the objects of $\mathbf{A}$, and the hom-set is given by $Kl_T(\mathbf{A})(A, B) := \mathbf{A}(A, TB)$ for any $A, B \in \mathbf{A}$.
- The Kleisli category $Kl_T(\mathbf{A})$ is monoidal because T is a commutative strong monad and $\mathbf{A}$ is monoidal. For any $f : A_1 \rightarrow TB_1$ and $g : A_2 \rightarrow TB_2$ in $Kl_T(\mathbf{A})$, we define $f \otimes g \in Kl_T(\mathbf{A})(A_1 \otimes A_2, B_1 \otimes B_2)$ by

$$A_1 \otimes A_2 \xrightarrow{f \otimes g} TB_1 \otimes TB_2 \xrightarrow{s} T(B_1 \otimes TB_2) \xrightarrow{Tt} TT(B_1 \otimes B_2) \xrightarrow{\mu} T(B_1 \otimes B_2).$$

- Since $\psi(S) = \phi(S)$ for any $S \in \mathbf{M}, \mathbf{Q}$, we define $\mathbf{Bit} = \psi(\mathbf{Bit}) = \phi(\mathbf{Bit}) \in \mathbf{A}$.
- Condition (e) expresses the requirement that the category $\mathbf{A}$ must combine both categories $\mathbf{M}$ and $\mathbf{Q}$, i.e., they are subcategories of $\mathbf{A}$ and its Kleisli category, respectively. Thus $\mathbf{A}$ has both quantum circuits and quantum operations. The commutative diagram implies that a circuit in $\mathbf{A}$ can be used as a quantum operation.
- Condition (f) gives a categorical characterization of dynamic lifting. The map dynlift is not in the image of ϕ or ψ , and therefore it is neither a quantum circuit nor a quantum operation.

Definition 3.5. We say a category $\mathbf{A}$ is a *model for Proto-Quipper with dynamic lifting* if it satisfies (a)–(f).

We have now axiomatized a general categorical model for Proto-Quipper with dynamic lifting. In Section 6, we will give a construction of a concrete model based on *biset-enrichment*.

4 A type system for dynamic lifting

In this section, we present the syntax of Proto-Quipper-Dyn and a type system for dynamic lifting. Our typing judgments have the form $\Gamma \vdash_\alpha M : A$, where $\alpha ::= 0 \mid 1$ is a modality used to keep track of dynamic lifting. When $\alpha = 1$, the term M is guaranteed not to perform any dynamic lifting operations while it is being reduced to a value. Such computations can

<i>Modality</i>	α, β	$::=$	$0 \mid 1$
<i>Types</i>	A, B	$::=$	Unit $\mid$ Qubit $\mid$ Bit $\mid$ Bool $\mid$ $!_{\alpha}A \mid A \multimap_{\alpha} B \mid \mathbf{Circ}(S, U) \mid A \otimes B$
<i>Parameter Types</i>	P, R	$::=$	Unit $\mid$ Bool $\mid$ $!_{\alpha}A \mid \mathbf{Circ}(S, U) \mid P \otimes R$
<i>Simple Types</i>	S, U	$::=$	Unit $\mid$ Qubit $\mid$ Bit $\mid S \otimes U$
<i>Terms</i>	M, N	$::=$	$c \mid x \mid \lambda x. M \mid M N \mid \mathbf{Unit} \mid (a, C, b) \mid \mathbf{apply}(M, N) \mid \mathbf{force} M$ $\mid \mathbf{lift} M \mid \mathbf{box} U M \mid (M, N) \mid \mathbf{let} (x, y) = N \text{ in } M \mid \mathbf{dynlift} M$
<i>Simple Terms</i>	a, b	$::=$	$\ell \mid \mathbf{Unit} \mid (a, b)$
<i>Contexts</i>	Γ	$::=$	$\cdot \mid x : A, \Gamma \mid \ell : \mathbf{Qubit}, \Gamma \mid \ell : \mathbf{Bit}, \Gamma$
<i>Parameter contexts</i>	Φ	$::=$	$\cdot \mid x : P, \Phi.$
<i>Label Contexts</i>	Σ	$::=$	$\cdot \mid \ell : \mathbf{Qubit}, \Sigma \mid \ell : \mathbf{Bit}, \Sigma$
<i>Values</i>	V	$::=$	$x \mid \ell \mid \lambda x. M \mid \mathbf{lift} M \mid (a, C, b) \mid (V, V') \mid \mathbf{Unit}$
<i>Circuits</i>	$C, \mathcal{D} : \Sigma \rightarrow \Sigma'$		

Fig. 1. The syntax of Proto-Quipper-Dyn

therefore be carried out at circuit generation time. When $\alpha = 0$, M may invoke dynamic lifting so the evaluation of M needs to be performed at circuit execution time.

4.1 Syntax

Definition 4.1 (Syntax). The syntax of Proto-Quipper-Dyn is in Figure 1.

The modality α appears in the linear function type $A \multimap_{\alpha} B$ and the linear exponential type $!_{\alpha}A$. This is because the values of $A \multimap_{\alpha} B$ and $!_{\alpha}A$ are *thunks* and we use the modality α in the types to keep track of the dynamic lifting within the thunks. $\mathbf{Circ}(S, U)$ denotes a type of circuits with input S and output U . The values of this type are boxed quantum circuits that can be constructed by the $\mathbf{box}$ term. They can be further manipulated by meta-operations such as circuit reversal, circuit iteration, or printing; these operations are treated as constants in the language, i.e., we do not fix a particular set of such operations, but assume that they would be defined in a standard library that comes with any particular instance of Proto-Quipper-Dyn.

The terms of our language are similar to the ones from Rios and Selinger [37], with the addition of a term construct for dynamic lifting $\mathbf{dynlift} M$, which will be evaluated to a boolean value. Lift and force are from linear lambda calculus, and are used to construct and deconstruct a term of type $!_{\alpha}A$. Values of type $\mathbf{Circ}(S, U)$ are represented as (a, C, b) , where a, b are simple terms and C is a morphism in $\mathbf{M}$. The term $\mathbf{apply}(M, N)$ attaches a boxed circuit M of type $\mathbf{Circ}(S, U)$ to the wires described by N . The term c ranges over constants such as booleans, natural numbers, and built-in functions. A subset of the types are designated as *parameter types*; these are types whose values are known at circuit generation time. Unlike general values, values of parameter type can be duplicated and discarded. A boxed circuit is considered to be a parameter and can be used multiple times. Parameter types are closed under tensor product.

A value of simple type corresponds to a state. Our language and semantics can accommodate coproducts (sum types), but we elide the treatment here for the sake of simplicity.

We make a distinction between variables and labels. A label ℓ corresponds to a wire in a circuit (when generating circuits), or to an address of a bit or qubit state (when running

$$\begin{array}{c}
\frac{}{\Phi, x : A \vdash_1 x : A} \textit{var} \qquad \frac{\Gamma \vdash_\alpha M : !_1(S \multimap_1 U)}{\Gamma \vdash_\alpha \textit{box } S M : \mathbf{Circ}(S, U)} \textit{box} \\
\\
\frac{\Gamma_1 \vdash_{\alpha_1} M : A \multimap_\beta B \quad \Gamma_2 \vdash_{\alpha_2} N : A}{\Gamma_1 + \Gamma_2 \vdash_{\alpha_1 \& \alpha_2 \& \beta} MN : B} \textit{app} \qquad \frac{\Gamma, x : A \vdash_\alpha M : B}{\Gamma \vdash_1 \lambda x. M : A \multimap_\alpha B} \textit{lambda} \\
\\
\frac{\Gamma_1 \vdash_\alpha M : \mathbf{Circ}(S, U) \quad \Gamma_2 \vdash_\beta N : S}{\Gamma_1 + \Gamma_2 \vdash_{\alpha \& \beta} \textit{apply}(M, N) : U} \textit{apply} \qquad \frac{\Gamma \vdash_\beta M : !_\alpha A}{\Gamma \vdash_{\alpha \& \beta} \textit{force } M : A} \textit{force} \\
\\
\frac{\Sigma_1 \vdash_1 a : S \quad \Sigma_2 \vdash_1 b : U \quad C : \Sigma_1 \rightarrow \Sigma_2}{\Phi \vdash_1 (a, C, b) : \mathbf{Circ}(S, U)} \textit{circ} \qquad \frac{\Phi \vdash_\alpha M : A}{\Phi \vdash_1 \textit{lift } M : !_\alpha A} \textit{lift} \\
\\
\frac{}{\Phi, \ell : \mathbf{Qubit}|\mathbf{Bit} \vdash_1 \ell : \mathbf{Qubit}|\mathbf{Bit}} \textit{label} \qquad \frac{}{\Phi \vdash c : A_c} \textit{const} \\
\\
\frac{\Gamma_1 \vdash_{\alpha_1} M : A \quad \Gamma_2 \vdash_{\alpha_2} N : B}{\Gamma_1 + \Gamma_2 \vdash_{\alpha_1 \& \alpha_2} (M, N) : A \otimes B} \textit{pair} \qquad \frac{\Gamma \vdash_\alpha M : \mathbf{Bit}}{\Gamma \vdash_0 \textit{dynlift } M : \mathbf{Bool}} \textit{dynlift} \\
\\
\frac{\Gamma_1, x : A, y : B \vdash_{\alpha_1} M : C \quad \Gamma_2 \vdash_{\alpha_2} N : A \otimes B}{\Gamma_1 + \Gamma_2 \vdash_{\alpha_1 \& \alpha_2} \textit{let } (x, y) = N \textit{ in } M : C} \textit{let}
\end{array}$$

Fig. 2. The typing rules for Proto-Quipper-Dyn

circuits). Consequently, a label can only have type **Bit** or **Qubit**. Labels can only be renamed, not substituted. Every label context Σ has an obvious interpretation $\llbracket \Sigma \rrbracket$ in the category $\mathbf{M}$ as a tensor of the appropriate sequence of the objects **Qubit** and **Bit**. We write $\mathcal{D} : \Sigma \rightarrow \Sigma'$ to denote a quantum circuit, i.e., a morphism $\mathcal{D} : \llbracket \Sigma \rrbracket \rightarrow \llbracket \Sigma' \rrbracket$.

4.2 The typing rules

Definition 4.2 (Typing). The typing rules are in Figure 2.

We write $\alpha \& \beta$ for the boolean conjunction of α and β so that, e.g., $0 \& 1 = 0$. If $\Gamma_1 = \Phi, \Gamma'_1$ and $\Gamma_2 = \Phi, \Gamma'_2$ and Γ'_1, Γ'_2 are disjoint, we write $\Gamma_1 + \Gamma_2$ for $\Phi, \Gamma'_1, \Gamma'_2$.

The typing judgments are of the form $\Gamma \vdash_\alpha M : A$, where α is the modality of M . When $\alpha = 0$, this means that dynamic lifting may occur during the evaluation of M to a value V . This is distinct from any modality that may occur in the type A , which refers to dynamic lifting that occurs when the value V is used. In the *var* rule, we require a parameter context Φ . In the *const* rule, A_c is the type of the constant symbol c . In the *lift* and *lambda* rules, the modality α is moved to the type and the current modality (i.e., modality in the conclusion) is set to 1. This is because the lift and lambda terms are values, and values do not perform dynamic lifting. In fact, all values have modality 1. The modalities in types, e.g., $A \multimap_\alpha B$ in the *lambda* rule, are for recording the modality of the term M under the lambda abstraction.

In elimination rules such as *app* and *force*, the modality in the type affects the current modality of the typing judgment through boolean conjunction. This is related to how the

evaluations are performed for these terms. For example, when evaluating the term MN , we will first evaluate M , then evaluate N and finally perform a beta-reduction. Thus, the evaluation of MN could perform dynamic lifting of $\alpha_1 = 0$, $\alpha_2 = 0$, or $\beta = 0$. Consequently, the modality for the typing judgment of MN is the boolean conjunction of all these related modalities.

By the *dynlift* rule, an application of dynamic lifting sets the current modality to 0, signifying that a dynamic lifting is performed. In the *box* rule, a term M can only be boxed into a circuit if it has type $!_1(S \multimap_1 U)$. This ensures that the value of M (denoted by V) does not use dynamic lifting. Thus, when evaluating the term $(\text{box } S \ V)$, a dynamic lifting cannot occur. This prevents a class of runtime errors in Quipper that are caused by boxing functions that use dynamic lifting.

In the *apply* rule, depending on the modality $\alpha_1 \ \& \ \alpha_2$, the term $\text{apply}(M, N)$ either appends the quantum circuit M to N , which is done at circuit generation time, or applies the quantum operation M to N , which is done at circuit execution time. The *circ* rule defines a well-typed quantum circuit. In practice, we often assume that a set of well-typed quantum gates is provided as pre-defined constants of the language, for example as part of a standard library. For this reason, programmers do not usually need to write *circ* terms directly; such terms would only be written by the implementors of the standard library.

The following lemma shows that a value can only have modality 1 and that the free variables of a parameter must come from a parameter context.

Lemma 4.3. *If $\Gamma \vdash_\alpha V : B$, then $\alpha = 1$. Moreover, if $B = P$ is a parameter type, then $\Gamma = \Phi$ is a parameter context.*

PROOF. By induction on the derivation of $\Gamma \vdash_\alpha V : B$. □

The following lemma shows that the type system has the usual substitution property.

Lemma 4.4 (Substitution). *If $\Gamma_1, x : A, \Gamma'_1 \vdash_\alpha M : B$ and $\Gamma_2 \vdash_1 V : A$, then $\Gamma_1, \Gamma_2, \Gamma'_1 \vdash_\alpha [V/x]M : B$.*

PROOF. By induction on the typing derivation of M . □

4.3 Interpretation of the typing rules

The modality is a syntactic device to track the dynamic lifting monad T . We will interpret $\Gamma \vdash_1 M : A$ as a map $[\Gamma] \rightarrow [A]$ in $\mathbf{A}$, and $\Gamma \vdash_0 M : A$ as a map $[\Gamma] \rightarrow T[A]$. To make the notation more uniform, we write $\alpha[A]$ to mean $T[A]$ if $\alpha = 0$, and just $[A]$ if $\alpha = 1$. With this notation, the interpretation of $\Gamma \vdash_\alpha M : A$ is a map $[\Gamma] \rightarrow \alpha[A]$. Similarly, the modalities in types such as $A \multimap_\alpha B$ and $!_\alpha A$ also indicate optional occurrences of the dynamic lifting monad T . Recall from Definition 3.1 that $\mathbf{A}$ is equipped with functors $p : \mathbf{Set} \rightarrow \mathbf{A}$ and $\flat : \mathbf{A} \rightarrow \mathbf{Set}$.

Definition 4.5. We interpret types as objects in $\mathbf{A}$.

$$\begin{aligned}
 [A \multimap_\alpha B] &= [A] \multimap \alpha[B] \\
 [A \otimes B] &= [A] \otimes [B] \\
 [!_\alpha A] &= p^{\flat \alpha} [A] \\
 [\text{Circ}(S, U)] &= p\mathbf{A}([S], [U]) \\
 [\text{Bool}] &= p(1 + 1) \\
 [\text{Bit}] &= \mathbf{Bit} \\
 [\text{Qubit}] &= \mathbf{Qubit}
 \end{aligned}$$

It is easy to check that for a parameter type P , there exists $X \in \mathbf{Set}$ such that $\llbracket P \rrbracket = pX$; in other words, $\llbracket P \rrbracket$ is a parameter object. Similarly, for a simple type S , there exists $Y \in \mathbf{M}$ such that $\llbracket S \rrbracket = \psi Y$, where $\psi : \mathbf{M} \rightarrow \mathbf{A}$ is the fully faithful embedding of Definition 3.4. We call objects of the form ψY *simple objects*. We interpret a context Γ as a tensor product of all objects in Γ (denoted by $\llbracket \Gamma \rrbracket$). The interpretation of parameter context $\llbracket \Phi \rrbracket$ is a parameter object and the interpretation of a label context $\llbracket \Sigma \rrbracket$ is a simple object. Without loss of generality, we assume that if $\llbracket \Sigma \rrbracket = \llbracket \Sigma' \rrbracket$, then $\Sigma = \Sigma'$ (this condition can always be ensured by making additional isomorphic copies of objects, if necessary).

The interpretation of typing judgements is defined as follows. Here, for convenience, we write $\eta_\alpha : \alpha \llbracket A \rrbracket \rightarrow T \llbracket A \rrbracket$, which is the unit of the monad when $\alpha = 1$ and the identity when $\alpha = 0$.

Definition 4.6 (Interpretation). To each valid typing judgement $\Gamma \vdash_\alpha M : A$, we associate a map $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \alpha \llbracket A \rrbracket$ in $\mathbf{A}$, called its *interpretation*. Note that $\llbracket M \rrbracket$ here is an abbreviation for $\llbracket \Gamma \vdash_\alpha M : A \rrbracket$.

The interpretation is defined by induction on the derivation of $\Gamma \vdash_\alpha M : A$.

- Case

$$\overline{\Phi, x : A \vdash_1 x : A.}$$

We define $\llbracket x \rrbracket$ as $\llbracket \Phi \rrbracket \otimes \llbracket A \rrbracket \xrightarrow{\text{discard} \otimes \llbracket A \rrbracket} \llbracket A \rrbracket$.

- Case

$$\overline{\Phi, \ell : \mathbf{Qubit} \vdash_1 \ell : \mathbf{Qubit}.}$$

We define $\llbracket \ell \rrbracket$ as $\llbracket \Phi \rrbracket \otimes \llbracket \mathbf{Qubit} \rrbracket \xrightarrow{\text{discard} \otimes \llbracket \mathbf{Qubit} \rrbracket} \mathbf{Qubit} \xrightarrow{\text{id}} \mathbf{Qubit}$.

- Case

$$\frac{\Phi, \Gamma_1 \vdash_{\alpha_1} M : A \quad \Phi, \Gamma_2 \vdash_{\alpha_2} N : B}{\Phi, \Gamma_1, \Gamma_2 \vdash_{\alpha_1 \& \alpha_2} (M, N) : A \otimes B.}$$

Suppose $\alpha_1 = \alpha_2 = 0$. The other cases are similar. By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \rightarrow T \llbracket A \rrbracket$ and $\llbracket N \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \rightarrow T \llbracket B \rrbracket$. So we define $\llbracket (M, N) \rrbracket$ as the following composition:

$$\begin{aligned} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket &\xrightarrow{\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\ &\xrightarrow{\cong} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \xrightarrow{\llbracket M \rrbracket \otimes \llbracket N \rrbracket} T \llbracket A \rrbracket \otimes T \llbracket B \rrbracket \\ &\xrightarrow{t} T(T \llbracket A \rrbracket \otimes T \llbracket B \rrbracket) \xrightarrow{Ts} TT(\llbracket A \rrbracket \otimes \llbracket B \rrbracket) \xrightarrow{\mu} T(\llbracket A \rrbracket \otimes \llbracket B \rrbracket). \end{aligned}$$

- Case

$$\frac{\Gamma \vdash_\alpha M : \mathbf{Bit}}{\Gamma \vdash_0 \text{dynlift } M : \mathbf{Bool}.}$$

By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \alpha \llbracket \mathbf{Bit} \rrbracket$. We define $\llbracket \text{dynlift } M \rrbracket$ by

$$\llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \alpha \llbracket \mathbf{Bit} \rrbracket \xrightarrow{\eta_\alpha} T \llbracket \mathbf{Bit} \rrbracket \xrightarrow{T \text{ dynlift}} TT \llbracket \mathbf{Bool} \rrbracket \xrightarrow{\mu} T \llbracket \mathbf{Bool} \rrbracket.$$

Note that $\text{dynlift} : \mathbf{Bit} \rightarrow T \mathbf{Bool}$ is the dynamic lifting morphism defined in Definition 3.4.

- Case

$$\frac{\Gamma, x : A \vdash_\alpha M : B}{\Gamma \vdash_1 \lambda x. M : A \multimap_\alpha B.}$$

By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \otimes \llbracket A \rrbracket \rightarrow \alpha \llbracket B \rrbracket$. Using monoidal closedness, we define $\llbracket \lambda x. M \rrbracket := \text{curry}(\llbracket M \rrbracket) : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket \multimap \alpha \llbracket B \rrbracket$.

- Case

$$\frac{\Phi, \Gamma_1 \vdash_{\alpha_1} M : A \multimap_{\beta} B \quad \Phi, \Gamma_2 \vdash_{\alpha_2} N : A}{\Phi, \Gamma_1, \Gamma_2 \vdash_{\alpha_1 \& \alpha_2 \& \beta} MN : B.} \text{ app}$$

Here we only consider the case where $\alpha_1 = \alpha_2 = \beta = 0$. The other cases are similar. By the induction hypothesis, we have morphisms $\llbracket M \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \rightarrow T(\llbracket A \rrbracket \multimap T\llbracket B \rrbracket)$ and $\llbracket N \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \rightarrow T\llbracket A \rrbracket$. Thus we define $\llbracket MN \rrbracket$ to be the following.

$$\begin{aligned} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket &\xrightarrow{\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\ &\xrightarrow{\cong} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\ &\xrightarrow{\llbracket M \rrbracket \otimes \llbracket N \rrbracket} T(\llbracket A \rrbracket \multimap T\llbracket B \rrbracket) \otimes T\llbracket A \rrbracket \\ &\xrightarrow{t} T(T(\llbracket A \rrbracket \multimap T\llbracket B \rrbracket) \otimes \llbracket A \rrbracket) \xrightarrow{Ts} TT((\llbracket A \rrbracket \multimap T\llbracket B \rrbracket) \otimes \llbracket A \rrbracket) \\ &\xrightarrow{\mu} T((\llbracket A \rrbracket \multimap T\llbracket B \rrbracket) \otimes \llbracket A \rrbracket) \xrightarrow{T\epsilon} TT\llbracket B \rrbracket \xrightarrow{\mu} T\llbracket B \rrbracket. \end{aligned}$$

- Case

$$\frac{\Phi \vdash_{\alpha} M : A}{\Phi \vdash_1 \text{lift } M : !_\alpha A.}$$

By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Phi \rrbracket \rightarrow \alpha \llbracket A \rrbracket$, where $\llbracket \Phi \rrbracket = pX$ for some $X \in \mathbf{Set}$. By the adjunction $p \vdash b$, we have $\delta \llbracket M \rrbracket : X \rightarrow b\alpha \llbracket A \rrbracket$. So we define $\llbracket \text{lift } M \rrbracket := p\delta \llbracket M \rrbracket : pX \rightarrow pb\alpha \llbracket A \rrbracket$.

- Case

$$\frac{\Gamma \vdash_{\beta} M : !_\alpha A}{\Gamma \vdash_{\alpha \& \beta} \text{force } M : A.}$$

We only consider the case where $\alpha = \beta = 0$; the other cases are similar. By the induction hypothesis, we have a map $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow TpbT\llbracket A \rrbracket$. Since there is a natural transformation $\text{force} : TpbT\llbracket A \rrbracket \rightarrow T\llbracket A \rrbracket$, we define $\llbracket \text{force } M \rrbracket$ by

$$\llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} TpbT\llbracket A \rrbracket \xrightarrow{T\text{force}} TT\llbracket A \rrbracket \xrightarrow{\mu} T\llbracket A \rrbracket.$$

- Case

$$\frac{\Gamma \vdash_{\alpha} M : !_1(S \multimap_1 U)}{\Gamma \vdash_{\alpha} \text{box } S M : \mathbf{Circ}(S, U).}$$

Here we only consider the case $\alpha = 0$; the other case is similar. By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow Tpb(\llbracket S \rrbracket \multimap \llbracket U \rrbracket)$. We define $\llbracket \text{box } SM \rrbracket$ by

$$\llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} Tpb(\llbracket S \rrbracket \multimap \llbracket U \rrbracket) \xrightarrow{Tpb\text{box}} Tp\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket).$$

- Case

$$\frac{\Phi, \Gamma_1 \vdash_{\alpha} M : \mathbf{Circ}(S, U) \quad \Phi, \Gamma_2 \vdash_{\beta} N : S}{\Phi, \Gamma_1, \Gamma_2 \vdash_{\alpha \& \beta} \text{apply}(M, N) : U} \text{ apply}$$

Here we only consider the case $\alpha = \beta = 0$. By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Gamma_1 \rrbracket \rightarrow Tp\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket)$ and $\llbracket N \rrbracket : \llbracket \Gamma_2 \rrbracket \rightarrow T\llbracket S \rrbracket$. Thus we define

$\llbracket \text{apply}(M, N) \rrbracket$ by

$$\begin{aligned}
& \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \xrightarrow{\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
& \xrightarrow{\cong} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
& \xrightarrow{\llbracket M \rrbracket \otimes \llbracket N \rrbracket} Tp\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket) \otimes T\llbracket S \rrbracket \\
& \xrightarrow{t} T(Tp\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket) \otimes \llbracket S \rrbracket) \\
& \xrightarrow{Ts} TT(p\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket) \otimes \llbracket S \rrbracket) \\
& \xrightarrow{\mu} T(p\mathbf{A}(\llbracket S \rrbracket, \llbracket U \rrbracket) \otimes \llbracket S \rrbracket) \\
& \xrightarrow{T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket)} T((\llbracket S \rrbracket \multimap \llbracket U \rrbracket) \otimes \llbracket S \rrbracket) \\
& \xrightarrow{T\epsilon} T\llbracket U \rrbracket.
\end{aligned}$$

- Case

$$\frac{\begin{array}{c} \Sigma_1 \vdash_1 a : S \quad \Sigma_2 \vdash_1 b : U \\ C : \Sigma_1 \rightarrow \Sigma_2 \end{array}}{\Phi \vdash_1 (a, C, b) : \mathbf{Circ}(S, U)}$$

Note that $\llbracket a \rrbracket : \llbracket \Sigma_1 \rrbracket \rightarrow \llbracket S \rrbracket$ and $\llbracket b \rrbracket : \llbracket \Sigma_2 \rrbracket \rightarrow \llbracket U \rrbracket$ are isomorphisms. Moreover, $C : \llbracket \Sigma_1 \rrbracket \rightarrow \llbracket \Sigma_2 \rrbracket$ is a morphism in $\mathbf{A}$. We write $\theta_C = \llbracket b \rrbracket \circ C \circ \llbracket a \rrbracket^{-1} : \llbracket S \rrbracket \rightarrow \llbracket U \rrbracket$ and $1_{\theta_C} : 1 \rightarrow \mathbf{M}(\llbracket S \rrbracket, \llbracket U \rrbracket)$ such that $1_{\theta_C}(\ast) = \theta_C$. Thus we define $\llbracket (a, C, b) \rrbracket$ as $p1_{\theta_C} : I \rightarrow p\mathbf{M}(\llbracket S \rrbracket, \llbracket U \rrbracket)$.

- Case

$$\frac{\Phi, \Gamma_1, x : A, y : B \vdash_{\alpha_1} M : C \quad \Phi, \Gamma_2 \vdash_{\alpha_2} N : A \otimes B}{\Phi, \Gamma_1, \Gamma_2 \vdash_{\alpha_1 \& \alpha_2} \text{let } (x, y) = N \text{ in } M : C} \text{let}$$

We only consider the case where $\alpha_1 = \alpha_2 = 0$; the other cases are similar. By the induction hypothesis, we have $\llbracket M \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket A \rrbracket \otimes \llbracket B \rrbracket \rightarrow T\llbracket C \rrbracket$ and $\llbracket N \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \rightarrow T(\llbracket A \rrbracket \otimes \llbracket B \rrbracket)$. Thus we define $\llbracket \text{let } (x, y) = N \text{ in } M \rrbracket$ as the following composition.

$$\begin{aligned}
& \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \xrightarrow{\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
& \xrightarrow{\cong} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
& \xrightarrow{\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket N \rrbracket} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes T(\llbracket A \rrbracket \otimes \llbracket B \rrbracket) \\
& \xrightarrow{t} T(\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket A \rrbracket \otimes \llbracket B \rrbracket) \xrightarrow{T\llbracket M \rrbracket} TT\llbracket C \rrbracket \xrightarrow{\mu} T\llbracket C \rrbracket.
\end{aligned}$$

Our interpretation of the typing rules satisfies the usual semantic substitution theorem.

Theorem 4.7 (Semantic substitution). *Suppose $\Phi, \Gamma_1, x : A, \Gamma_2 \vdash_\alpha M : B$ and $\Phi, \Gamma_3 \vdash_1 V : A$. We have the following equality.*

$$\begin{aligned}
 \llbracket [V/x]M \rrbracket &= \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\text{dup} \otimes \text{id}} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\cong} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\text{id} \otimes \llbracket V \rrbracket \otimes \text{id}} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket A \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\llbracket M \rrbracket} \alpha \llbracket B \rrbracket.
 \end{aligned}$$

PROOF. The proof is by induction on the typing derivation of M . See Appendix C for details. $\square$

The next two lemmas show that values of parameter type are in the image of the functor p , and that values of simple types are isomorphisms.

Lemma 4.8. *If $\Phi \vdash_1 V : P$ for some parameter context Φ and parameter type P , then $\llbracket V \rrbracket = pf : pX \rightarrow pY$ for some $f : X \rightarrow Y \in \mathbf{Set}$ such that $\llbracket \Phi \rrbracket = pX$, $\llbracket P \rrbracket = pY$.*

PROOF. By induction on $\Phi \vdash_1 V : P$. $\square$

Lemma 4.9. *Suppose $\Sigma \vdash_1 a : S$ for some label context Σ and simple type S , then $\llbracket a \rrbracket : \llbracket \Sigma \rrbracket \rightarrow \llbracket S \rrbracket$ is an isomorphism in $\mathbf{A}$.*

PROOF. By induction on $\Sigma \vdash_1 a : S$. $\square$

Since the embedding $\psi : \mathbf{M} \hookrightarrow \mathbf{A}$ is full and faithful and $\llbracket \Sigma \rrbracket$ and $\llbracket S \rrbracket$ are in the image of ψ , there is a corresponding isomorphism in $\mathbf{M}$ which we also denote as $\llbracket a \rrbracket : \llbracket \Sigma \rrbracket \rightarrow \llbracket S \rrbracket$.

4.4 Discussion: implementing the type checker

While typing judgments and certain types ($A \multimap_\alpha B$ and $!_\alpha A$) are annotated with a modality, information about this modality is meant to be hidden from the programmer unless an error occurs. For example, if one attempt to box a function which uses dynamic lifting, the type checker will raise a modality error. As a result, such programming errors are caught at compile time in Proto-Quipper-Dyn, whereas they are only caught at runtime in Quipper.

The modality inference can readily be integrated into bi-directional type checking, which uses a pair of recursively defined functions for type checking and type inference [10, 32]. To work with modalities, the type checking function not only takes a term and a type as inputs, but also the current modality of the typing judgment. For example, when checking a term $\lambda x.M$ against a type $A \multimap_\alpha B$ with current modality β , the type checking function first ensures that the current modality β is 1, then extends the current typing environment with $x : A$ and recursively checks the term M against the type B , with the modality α . The type inference function takes a term as input and outputs the inferred type as well as the inferred modality. For example, when inferring the type for a term MN , the type inference function first infers a type $A \multimap_\alpha B$ and a modality β for M , and then it infers a type A and a modality γ for the term N , so the type inference function will return the type B and the inferred modality $\alpha \& \beta \& \gamma$.

$$\begin{array}{c}
\frac{(C_1, M) \Downarrow (C_2, \lambda x.M') \quad (C_2, N) \Downarrow (C_3, V) \quad (C_3, [V/x]M') \Downarrow (C_4, V')}{(C_1, MN) \Downarrow (C_4, V')} \text{ app} \\
\\
\frac{(C, M) \Downarrow (C', \text{lift } M') \quad (C', M') \Downarrow (C'', V)}{(C, \text{force } M) \Downarrow (C'', V)} \text{ lift-force} \\
\\
\frac{(C, N) \Downarrow (C', (V_1, V_2)) \quad (C', [V_1/x, V_2/y]M) \Downarrow (C'', V)}{(C, \text{let } (x, y) = N \text{ in } M) \Downarrow (C'', V)} \text{ let}
\end{array}
\qquad
\begin{array}{c}
\frac{(C_1, M) \Downarrow (C_2, (a, \mathcal{D}, b)) \quad (C_2, N) \Downarrow (C_3, c) \quad \text{append}(\mathcal{D}, C_3, a, c) = C'}{(C_1, \text{apply}(M, N)) \Downarrow (C', b)} \text{ apply} \\
\\
\frac{(C, M) \Downarrow (C', \text{lift } M') \quad \text{gen}(S) = a \quad (\text{id}_S, M' a) \Downarrow (\mathcal{D}, b)}{(C, \text{box } S M) \Downarrow (C', (a, \mathcal{D}, b))} \text{ box} \\
\\
\frac{(C, M) \Downarrow (C', V_1) \quad (C', N) \Downarrow (C'', V_2)}{(C, (M, N)) \Downarrow (C'', (V_1, V_2))} \text{ pair}
\end{array}$$

Fig. 3. Operational semantics for circuit generation time

5 Operational semantics and soundness

In this section, we will specify an operational semantics for Proto-Quipper-Dyn and show that it is sound with respect to the categorical model $\mathbf{A}$ for dynamic lifting of Definition 3.5.

We distinguish two kinds of evaluation in order to model Quipper’s two runtimes. The evaluation rules for circuit generation time will work with morphisms in $\mathbf{M}$, i.e., quantum circuits. On the other hand, the evaluation rules for circuit execution time will work with morphisms in $\mathbf{Q}$, i.e., quantum operations. Because of the embeddings $\psi : \mathbf{M} \hookrightarrow \mathbf{A}$ and $\phi : \mathbf{Q} \hookrightarrow \text{Kl}_T(\mathbf{A})$, we are able to interpret the configurations for these two runtimes as maps in the category $\mathbf{A}$.

5.1 Operational semantics for circuit generation time

First of all, we specify the meaning of appending circuits in the category $\mathbf{M}$.

Definition 5.1 (Circuit append). Suppose $C : \Sigma \rightarrow \Sigma_1, \Sigma_2$ and $\mathcal{D} : \Sigma'_1 \rightarrow \Sigma_3$ are morphisms in $\mathbf{M}$ and there are typing judgments $\Sigma_1 \vdash_1 a : S$ and $\Sigma'_1 \vdash_1 a' : S$. We define $\text{append}(\mathcal{D}, C, a', a)$ to be the following morphism in $\mathbf{M}$. Note that $\llbracket a' \rrbracket$ is invertible by Lemma 4.9.

$$((\mathcal{D} \circ \llbracket a' \rrbracket^{-1} \circ \llbracket a \rrbracket) \otimes \llbracket \Sigma_2 \rrbracket) \circ C : \Sigma \rightarrow \Sigma_3, \Sigma_2$$

Thus $\text{append}(\mathcal{D}, C, a', a)$ is the result of appending the circuit $\mathcal{D}$ to C by connecting the interfaces a' and a .

The following are the evaluation rules for circuit generation time, where the underlying states are given by morphisms in $\mathbf{M}$. The evaluation is defined over *configurations* of the form (C, M) , where C is a morphism in $\mathbf{M}$ and M is a term.

Definition 5.2 (Circuit generation time evaluation). The operational semantics for circuit generation time is defined in Figure 3.

In the rule *box*, we write $\text{gen}(S) = a$ to mean that the a is a globally fresh simple term of type S . Recall from Figure 1 that a simple term is essentially a tuple of labels, so $\text{gen}(S)$ is a side-effecting function that generates an appropriate number of previously unused labels.

Note that the evaluation of $(C, M) \Downarrow (C', V)$ does not account for dynamic lifting, and the underlying states are circuits. So it is the same set of evaluation rules as in Rios and Selinger [37]. The evaluation comes with the following notion of circuit configuration.

Definition 5.3 (Well-typed circuit configuration). We write $\Sigma \vdash_1 (C, M) : A; \Sigma'$ to mean there exists Σ'' such that $C : \Sigma \rightarrow \Sigma', \Sigma''$ and $\Sigma'' \vdash_1 M : A$.

A well-typed circuit configuration requires a typed term with modality 1, i.e., $\vdash_1 M : A$. It is a runtime error if a term with dynamic lifting is encountered when using the evaluation rules in Definition 5.2. Our type system and the following type preservation theorem ensures that this cannot happen.

Theorem 5.4. *If $\Sigma \vdash_1 (C, M) : A; \Sigma'$ and $(C, M) \Downarrow (C', V)$, then $\Sigma \vdash_1 (C', V) : A; \Sigma'$.*

PROOF. By induction on $(C, M) \Downarrow (C', V)$. □

In the following we define the interpretation $\llbracket C, M \rrbracket$ as a morphism in the category $\mathbf{A}$.

Definition 5.5. Suppose $\Sigma \vdash_1 (C, M) : A; \Sigma'$. We have maps $\psi C : \llbracket \Sigma \rrbracket \rightarrow \llbracket \Sigma' \rrbracket \otimes \llbracket \Sigma'' \rrbracket$ and $\llbracket M \rrbracket : \llbracket \Sigma'' \rrbracket \rightarrow \llbracket A \rrbracket$ in $\mathbf{A}$. We define $\llbracket C, M \rrbracket$ as follows:

$$\llbracket \Sigma \rrbracket \xrightarrow{\psi C} \llbracket \Sigma' \rrbracket \otimes \llbracket \Sigma'' \rrbracket \xrightarrow{\llbracket \Sigma' \rrbracket \otimes \llbracket M \rrbracket} \llbracket \Sigma' \rrbracket \otimes \llbracket A \rrbracket.$$

The following theorem shows that the evaluation rules for circuit generation time are sound with respect to the categorical model $\mathbf{A}$.

Theorem 5.6. *If $\Sigma \vdash_1 (C, M) : A; \Sigma'$ and $(C, M) \Downarrow (C', V)$, then $\llbracket C, M \rrbracket = \llbracket C', V \rrbracket$.*

PROOF. By induction on $(C, M) \Downarrow (C', V)$. Since the reduction rules in Definition 5.2 do not include a rule for dynamic lifting, the proof is similar to the one in Rios and Selinger [37] and Rios [36]. □

5.2 Operational semantics for circuit execution time

Since dynamic lifting requires the ability to access the states in $\mathbf{Q}$, we first define the concepts of *state* and *addresses*.

Definition 5.7 (State and addresses). For any object $S \in \mathbf{Q}$, a *state* is a morphism $Q : I \rightarrow S \in \mathbf{Q}$. We write $\text{addr}(Q) = \Sigma$ if $\phi(S) = \llbracket \Sigma \rrbracket$, we call Σ the *addresses* of Q . (Recall that we have, for convenience and without loss of generality, assumed that the interpretation function $\llbracket - \rrbracket$ is one-to-one on label contexts).

We often write $Q : I \rightarrow \Sigma \in \mathbf{Q}$ for $Q : I \rightarrow S$, where $\phi(S) = \llbracket \Sigma \rrbracket$. The following *read* operation will be used to define the operational semantics for dynamic lifting.

Definition 5.8 (Read operation). Suppose $\text{addr}(Q) = \Sigma, \ell : \mathbf{Bit}$ and $Q = p_1(\text{inj}_1 \otimes Q_1) + p_2(\text{inj}_2 \otimes Q_2)$, where $\text{addr}(Q_1) = \text{addr}(Q_2) = \Sigma$ and $p_1, p_2 \in [0, 1]$ and $p_1 + p_2 = 1$. We define a formal sum $\text{read}(Q, \ell) = p_1(Q_1, \text{False}) + p_2(Q_2, \text{True})$, where $\text{False}, \text{True} : \mathbf{Bool}$.

Note that by the last condition in Assumption 3.3, we know that $Q = p_1(\text{inj}_1 \otimes Q_1) + p_2(\text{inj}_2 \otimes Q_2) : I \rightarrow \Sigma \otimes \mathbf{Bit}$ for some essentially uniquely determined $Q_1, Q_2 : I \rightarrow \Sigma$, and $p_1, p_2 \in [0, 1]$ such that $p_1 + p_2 = 1$. The only time Q_i is not uniquely determined is when $p_i = 0$, but in this case, it will turn out that the Q_i does not matter since it corresponds to a

branch of computation taken with probability zero. In this case, we can just make some fixed but arbitrary choice for Q_i . So the read operation makes the information of the probabilities p_1, p_2 and the states Q_1, Q_2 available.

In the following, we define the circuit execution time counterpart of Definition 5.1. It specifies the meaning of updating a quantum state by applying a quantum circuit, where the identity-on-object interpretation functor $J : \mathbf{M} \rightarrow \mathbf{Q}$ is needed for the definition.

Definition 5.9. Suppose $Q : I \rightarrow \Sigma_1, \Sigma_2$ is a morphism in $\mathbf{Q}$, and $C : \Sigma'_1 \rightarrow \Sigma_3$ is a morphism in $\mathbf{M}$, and there are typing judgements $\Sigma_1 \vdash a : S$ and $\Sigma'_1 \vdash a' : S$. We define $\text{applyTo}(C, Q, a', a)$ to be the following map in $\mathbf{Q}$.

$$(J(C \circ \llbracket a' \rrbracket^{-1} \circ \llbracket a \rrbracket) \otimes \llbracket \Sigma_2 \rrbracket) \circ Q : I \rightarrow \Sigma_3, \Sigma_2$$

We now define the operational semantics for circuit execution time. The underlying states of the evaluation are the states in $\mathbf{Q}$. The evaluation is of the form $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$. Its intuitive meaning is that the configuration (Q, M) can be reduced to (Q_i, V_i) with probability p_i . The notation $\sum_{i \in [n]} p_i(Q_i, V_i)$ is a short hand for the formal sum $p_1(Q_1, V_1) + \dots + p_n(Q_n, V_n)$, and we assume $\sum_{i \in [n]} p_i = 1$. We write $[n] = \{1, \dots, n\}$.

Definition 5.10 (Operational semantics for circuit execution time). The operational semantics for circuit execution time is defined in Figure 4.

In the *apply* rule, we use *applyTo* from Definition 5.9 instead of *append*. This allows a quantum circuit to be applied as a quantum operation. In the *dynlift* rule, for each (Q_i, ℓ_i) , we apply the operation $\text{read}(Q_i, \ell_i)$, which gives rise to two possible outcomes $(Q'_{i,1}, a_{i,1}), (Q'_{i,2}, a_{i,2})$ with probabilities $q_{i,1}, q_{i,2}$, where $a_{i,1}, a_{i,2} : \mathbf{Bool}$ and $a_{i,1} \neq a_{i,2}$. This is the only rule that gives rise to probabilistic results in the evaluation. In the *box* rule, the evaluation of $(\text{id}_S, M'_i a)$ uses the rules defined in Definition 5.2, so it is performed at circuit generation time.

We now define a well-typed configuration for evaluating a term under a quantum state.

Definition 5.11 (Well-typed configuration). We write $\vdash_\alpha (Q, M) : A; \Sigma'$ to mean there exists Σ'' such that $\Sigma'' \vdash_\alpha M : A$, and $\text{addr}(Q) = \Sigma'', \Sigma'$.

Since the evaluation rules in Definition 5.10 account for dynamic lifting, the above configuration allows the term M to have modality 0. The operational semantics defined in Definition 5.10 is type-safe in the following sense.

Theorem 5.12. If $\vdash_\alpha (Q, M) : A; \Sigma'$ and $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$, then $\vdash_1 (Q_i, V_i) : A; \Sigma'$ for all $i \in [n]$.

PROOF. By induction on $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$. □

The following theorem shows that if the configuration has modality 1, the evaluation is deterministic.

Theorem 5.13. If $\vdash_1 (Q, M) : A; \Sigma'$ and $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$, then $n = 1$. In other words, we actually have $(Q, M) \Downarrow (Q', V)$.

PROOF. The proof is by induction on the evaluation rules. □

In the following we interpret a well-typed configuration $\vdash_\alpha (Q, M) : A; \Sigma'$ as a map in the Kleisli category $Kl_T(\mathbf{A})$.

$$\begin{array}{c}
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \lambda x. M'_i) \\
(Q_i, N) \Downarrow \sum_{j \in [m]} q_{i,j}(Q'_{i,j}, V_{i,j}) \\
(Q'_{i,j}, [V_{i,j}/x] M'_i) \Downarrow \sum_{k \in [l]} s_{i,j,k}(Q''_{i,j,k}, V'_{i,j,k}) \\
\hline
(Q, MN) \Downarrow \sum_{(i,j,k) \in [n] \times [m] \times [l]} p_i q_{i,j} s_{i,j,k}(Q''_{i,j,k}, V'_{i,j,k}) \\
\\
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \text{lift } M'_i) \\
(Q_i, M'_i) \Downarrow \sum_{j \in [m]} q_{i,j}(Q'_{i,j}, V_{i,j}) \\
\hline
(Q, \text{force } M) \Downarrow \sum_{(i,j) \in [n] \times [m]} p_i q_{i,j}(Q'_{i,j}, V_{i,j}) \\
\\
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, (a_i, \mathcal{D}_i, b_i)) \\
(Q_i, N) \Downarrow \sum_{j \in [m]} q_{i,j}(Q'_{i,j}, c_{i,j}) \\
\text{applyTo}(\mathcal{D}_i, Q'_{i,j}, a_i, c_{i,j}) = Q''_{i,j} \\
\hline
(Q, \text{apply}(M, N)) \Downarrow \sum_{(i,j) \in [n] \times [m]} p_i q_{i,j}(Q''_{i,j}, b_i) \quad \text{apply} \\
\\
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \text{lift } M'_i) \\
\text{gen}(S) = a \\
(\text{id}_S, M'_i a) \Downarrow (\mathcal{D}_i, b_i) \\
\hline
(Q, \text{box } S M) \Downarrow \sum_{i \in [n]} p_i(Q_i, (a, \mathcal{D}_i, b_i)) \quad \text{box} \\
\\
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \ell_i) \\
\text{read}(Q_i, \ell_i) = q_{i,1}(Q'_{i,1}, a_{i,1}) + q_{i,2}(Q'_{i,2}, a_{i,2}) \\
\hline
(Q, \text{dynlift } M) \Downarrow \sum_{(i,j) \in [n] \times [2]} p_i q_{i,j}(Q'_{i,j}, a_{i,j}) \quad \text{dynlift} \\
\\
(Q, N) \Downarrow \sum_{i \in [n]} p_i(Q'_i, (V_i, V'_i)) \quad (Q'_i, [V_i/x, V'_i/y] M) \Downarrow \sum_{j \in m}(Q''_{i,j}, V''_{i,j}) \\
\hline
(Q, \text{let } (x, y) = N \text{ in } M) \Downarrow \sum_{(i,j) \in [n] \times [m]} (Q''_{i,j}, V''_{i,j}) \\
\\
(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i) \\
(Q_i, N) \Downarrow \sum_{j \in [m]} q_{i,j}(Q'_{i,j}, V'_{i,j}) \\
\hline
(Q, (M, N)) \Downarrow \sum_{i,j \in [n] \times [m]} (Q'_{i,j}, (V_i, V'_{i,j}))
\end{array}$$

Fig. 4. Operational semantics for circuit execution time

Definition 5.14. Suppose $\vdash_\alpha (Q, M) : A; \Sigma'$. We have $\phi Q : I \rightarrow T(\llbracket \Sigma'' \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\llbracket M \rrbracket : \llbracket \Sigma'' \rrbracket \rightarrow \alpha \llbracket A \rrbracket$ in **A**. We define $\llbracket Q, M \rrbracket$ by:

- If $\alpha = 1$, then

$$I \xrightarrow{\phi Q} T(\llbracket \Sigma'' \rrbracket \otimes \llbracket \Sigma' \rrbracket) \xrightarrow{T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket)} T(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket).$$

- If $\alpha = 0$, then

$$\begin{aligned}
I &\xrightarrow{\phi Q} T(\llbracket \Sigma'' \rrbracket \otimes \llbracket \Sigma' \rrbracket) \xrightarrow{T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket)} T(T\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket) \\
&\xrightarrow{T_S} TT(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket) \xrightarrow{\mu} T(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket).
\end{aligned}$$

The following theorem shows that the operational semantics in Definition 5.10 is sound with respect to the semantic model **A**.

Theorem 5.15 (Soundness). *If $\vdash_\alpha (Q, M) : A; \Sigma'$, and $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$, then*

$$\llbracket Q, M \rrbracket = \sum_{i \in [n]} p_i \llbracket Q_i, V_i \rrbracket : I \rightarrow T(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket).$$

PROOF. The proof is by induction on the evaluation rules. Here we only consider the rule for dynlift; see Appendix D for the remaining rules. We also assume $\alpha = 0$, because the other case is similar.

- Case

$$\frac{\begin{array}{c} (Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \ell_i) \\ \text{read}(Q_i, \ell_i) = \sum_{j \in [2]} q_{i,j}(Q'_{i,j}, a_{i,j}) \end{array}}{(Q, \text{dynlift } M) \Downarrow \sum_{(i,j) \in [n] \times [2]} p_i q_{i,j}(Q'_{i,j}, a_{i,j})}.$$

Suppose $\text{addr}(Q) = \Sigma'', \Sigma', \Sigma'' \vdash_0 M : A$, and

$$\frac{\Sigma'' \vdash_1 M : \mathbf{Bit}}{\Sigma'' \vdash_0 \text{dynlift } M : \mathbf{Bool}}.$$

In this case we have the following derivation.

$$\frac{\begin{array}{c} (Q, M) \Downarrow (Q', \ell) \\ \text{read}(Q', \ell) = q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True}) \end{array}}{(Q, \text{dynlift } M) \Downarrow q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True})}$$

Since $\text{read}(Q', \ell) = q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True})$, we have that $Q' = q_1(\text{inj}_1 \otimes Q'_1) + q_2(\text{inj}_2 \otimes Q'_2)$ in **Q**. So in **A**, we have

$$\phi Q' = q_1(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) + q_2(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2)),$$

where $\phi Q' : I \rightarrow T(\mathbf{Bit} \otimes \llbracket \Sigma' \rrbracket)$, and $\phi Q'_1, \phi Q'_2 : I \rightarrow T\llbracket \Sigma' \rrbracket$, and $\phi(\text{inj}_1), \phi(\text{inj}_2) : I \rightarrow T\mathbf{Bit}$. Note that by condition (e) of Definition 3.4, we have $\phi(\text{inj}_1) = \eta \circ \text{init} \circ \llbracket \text{False} \rrbracket$ and $\phi(\text{inj}_2) = \eta \circ \text{init} \circ \llbracket \text{True} \rrbracket$. We need to show that

$$\llbracket Q, \text{dynlift } M \rrbracket = q_1(T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1) + q_2(T(\llbracket \text{True} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_2).$$

By the induction hypothesis, we have $\llbracket Q, M \rrbracket = \llbracket Q', \ell \rrbracket$, i.e., $T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q = \phi Q'$. Thus

$$\begin{aligned} \llbracket Q, \text{dynlift } M \rrbracket &= \mu \circ Ts \circ T((\text{dynlift} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q' \\ &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ (q_1(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) \\ &\quad + q_2(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2))) \\ &= q_1(\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) \\ &\quad + q_2(\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2)). \end{aligned}$$

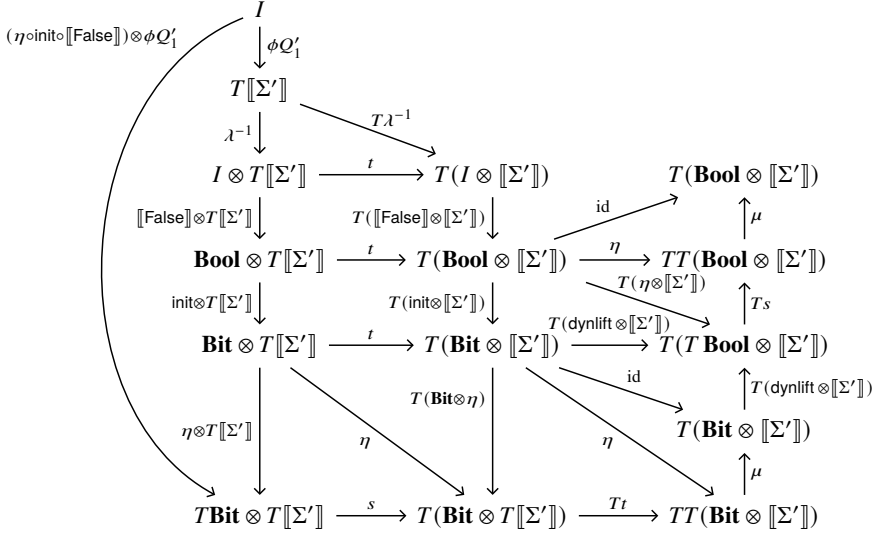


Fig. 5. A diagram used in the proof of Theorem 5.15

We just need to show

$$\begin{aligned} T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1) \\ &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \\ &\quad \circ ((\eta \circ \text{init} \circ \llbracket \text{False} \rrbracket) \otimes \phi Q'_1). \end{aligned}$$

This is true because of the commutative diagram in Figure 5. Note that the left-hand side in the above equation corresponds to the top leg of the diagram and the right-hand side corresponds to the bottom leg. The only non-trivial part of the diagram is the triangle involving `dynlift` and `init` in the middle, which commutes by the definition of `dynlift`.



Remark. In practice, a closed term M is always evaluated with the initial configuration (id_I, M) , where $\text{id}_I : I \rightarrow I$ is a state in $\mathbf{Q}$. When M has modality 0, we would need access to a quantum computer/simulator in order to evaluate (id_I, M) and each run of (id_I, M) could give a different value. When M has modality 1, the evaluation of (id_I, M) is deterministic, i.e., the top-level quantum state is updated in a deterministic fashion. In this case, instead of performing the quantum operations, we could also just generate a list of gates, which can be done entirely in a classical computer.

6 A concrete model for dynamic lifting

In this section, we will show how to construct a model for dynamic lifting from the categories $\mathbf{M}$ and $\mathbf{Q}$. We first give some background of enriched category theory. In Section 6.1, we will introduce the category of bisets and biset-enrichment. In Section 6.2, we will construct a biset-enriched category $\mathbf{C}$ based on $\mathbf{M}$ and $\mathbf{Q}$ and introduce its enriched presheaf category $\overline{\mathbf{C}}$. In Section 6.3, we will construct a reflective subcategory $\tilde{\mathbf{C}}$ of $\overline{\mathbf{C}}$ such that it is also monoidal

closed. In Section 6.4, we will show that there is a commutative strong monad $\tilde{T}$ on $\tilde{\mathbf{C}}$. In Section 6.5, we will demonstrate that the monad $\tilde{T}$ supports dynamic lifting. In Section 6.6, we will prove that the underlying category of $\tilde{\mathbf{C}}$ is a model for Proto-Quipper with dynamic lifting.

Recall that enriched categories are a generalization of categories where, instead of hom-sets, one works with hom-objects, which are objects in a monoidal category.

Definition 6.1. Let $\mathcal{V}$ be a monoidal category. A $\mathcal{V}$ -enriched category $\mathbf{B}$ is given by the following:

- A class of objects, also denoted $\mathbf{B}$.
- For any $A, B \in \mathbf{B}$, an object $\mathbf{B}(A, B)$ in $\mathcal{V}$.
- For any $A \in \mathbf{B}$, a morphism $u_A : I \rightarrow \mathbf{B}(A, A)$ in $\mathcal{V}$, called the *identity* on A .
- For any $A, B, C \in \mathbf{B}$, a morphism $c_{A,B,C} : \mathbf{B}(A, B) \otimes \mathbf{B}(B, C) \rightarrow \mathbf{B}(A, C)$ in $\mathcal{V}$, called *composition*.
- The composition and identity morphisms must satisfy suitable diagrams in $\mathcal{V}$ (see, e.g., Kelly [19] and Borceux [4]).

Remarks.

- Many concepts from the theory of non-enriched categories can be generalized to the enriched setting. For example, $\mathcal{V}$ -functors, $\mathcal{V}$ -natural transformations, $\mathcal{V}$ -adjunctions, and the $\mathcal{V}$ -Yoneda embedding are all straightforward generalizations of their non-enriched counterparts. We refer the reader to Kelly [19] and Borceux [4] for comprehensive introductions. Symmetric monoidal categories can also be generalized to the enriched setting (see Appendix E).
- In the rest of this section, when we speak of a map $f : A \rightarrow B$ in a $\mathcal{V}$ -enriched category $\mathbf{B}$, we mean a morphism of the form $f : I \rightarrow \mathbf{B}(A, B)$ in $\mathcal{V}$. Furthermore, when $g : B \rightarrow C$ is also a map in $\mathbf{B}$, we write $g \circ f : A \rightarrow C$ as a shorthand for

$$I \xrightarrow{f \otimes g} \mathbf{B}(A, B) \otimes \mathbf{B}(B, C) \xrightarrow{c} \mathbf{B}(A, C).$$

- A $\mathcal{V}$ -enriched category $\mathbf{B}$ gives rise to an ordinary (non-enriched) category $V(\mathbf{B})$, called the *underlying category* of $\mathbf{B}$. We use $V(\mathbf{A})$ to denote the underlying category, rather than the usual $U(\mathbf{A})$, because the letter U is used for other purposes in this paper. The objects of $V(\mathbf{B})$ are the objects of $\mathbf{B}$ and the hom-sets of $V(\mathbf{B})$ are defined as $V(\mathbf{B})(A, B) = \mathcal{V}(I, \mathbf{B}(A, B))$, for any $A, B \in V(\mathbf{B})$. Similarly, a $\mathcal{V}$ -functor $F : \mathbf{B} \rightarrow \mathbf{B}$ gives rise to a functor $VF : V(\mathbf{B}) \rightarrow V(\mathbf{B})$ and a $\mathcal{V}$ -natural transformation $\alpha : F \rightarrow G$ gives rise to a natural transformation $V\alpha : VF \rightarrow VG$.

6.1 Bisets

We now begin our construction of a concrete model satisfying Definition 3.5. Let $\mathbf{2}$ be the category with two objects $0, 1$ and one nontrivial arrow $0 \rightarrow 1$. Let $\mathbf{Set}^{2^{\text{op}}}$ be the category of functors from $\mathbf{2}^{\text{op}}$ to $\mathbf{Set}$. Concretely, the objects are triples (A_0, A_1, f) , where A_0, A_1 are sets and f is a function $A_1 \rightarrow A_0$. We call such a triple a *bisetet*. A morphism in $\mathbf{Set}^{2^{\text{op}}}$ from (A_0, A_1, f) to (B_0, B_1, g) is a pair (h_0, h_1) , where $h_0 : A_0 \rightarrow B_0$ and $h_1 : A_1 \rightarrow B_1$ are

functions such that the following diagram commutes.

$$\begin{array}{ccc} A_1 & \xrightarrow{h_1} & B_1 \\ \downarrow f & & \downarrow g \\ A_0 & \xrightarrow{h_0} & B_0 \end{array}$$

Because it is a presheaf category, the category of bisets $\mathbf{Set}^{2^{\text{op}}}$ is complete, cocomplete, and cartesian closed. We write $A \Rightarrow B$ to denote an exponential object. From now on, we will write $\mathcal{V}$ for $\mathbf{Set}^{2^{\text{op}}}$.

The category $\mathcal{V}$ is itself a $\mathcal{V}$ -category where the hom-object $\mathcal{V}(A, B)$ is given by the exponential object $A \Rightarrow B$. Since $\mathcal{V}$ can be regarded both as an ordinary category and as a $\mathcal{V}$ -category, we sometimes write $|\mathcal{V}|$ for $\mathcal{V}$, when we want to emphasize that we are regarding it as an ordinary category. In particular, $|\mathcal{V}|(A, B)$ is a set, whereas $\mathcal{V}(A, B)$ is a biset.

Any set X can be viewed as a trivial biset (X, X, id) . Therefore, any ordinary category can be viewed as a trivial biset-enriched category. For example, $\mathbf{Set}$ can be viewed as a $\mathcal{V}$ -category, where the hom-objects are given by $\mathbf{Set}(A, B) = (\mathbf{Set}(A, B), \mathbf{Set}(A, B), \text{id})$ for any $A, B \in \mathbf{Set}$.

Definition 6.2. We define $\mathcal{V}$ -functors $U_0(A_0, A_1, a) = A_0 : \mathcal{V} \rightarrow \mathbf{Set}$, and $\Delta(X) = (X, X, \text{id}) : \mathbf{Set} \rightarrow \mathcal{V}$.

The $\mathcal{V}$ -functor Δ is fully faithful and U_0 is strong monoidal. Note that there is also another functor $U_1(A_0, A_1, a) = A_1 : \mathcal{V} \rightarrow \mathbf{Set}$, but it is only an ordinary functor, not a $\mathcal{V}$ -functor. The functor U_1 will play no role in this paper, but the two $\mathcal{V}$ -functors U_0 and Δ will be important.

Theorem 6.3. *There is a $\mathcal{V}$ -adjunction $U_0 : \mathcal{V} \rightarrow \mathbf{Set} \dashv \Delta : \mathbf{Set} \rightarrow \mathcal{V}$. We write T for the $\mathcal{V}$ -monad $\Delta \circ U_0$. It is a commutative strong $\mathcal{V}$ -monad.*

PROOF. We have a $\mathcal{V}$ -natural isomorphism $\mathbf{Set}(U_0 A, Y) \cong \mathcal{V}(A, \Delta Y)$. Using the definition $T = \Delta \circ U_0$, it can be shown that $s : TA \otimes B \rightarrow T(A \otimes B)$ and $t : A \otimes TB \rightarrow T(A \otimes B)$ exist in $\mathcal{V}$ and satisfy the diagram for a commutative $\mathcal{V}$ -monad. $\square$

6.2 The $\mathcal{V}$ -categories $\mathbf{C}$ and $\overline{\mathbf{C}}$

Recall the categories $\mathbf{M}$ and $\mathbf{Q}$ from Assumption 3.3. In the following we define a $\mathcal{V}$ -category $\mathbf{C}$ from $\mathbf{M}$ and $\mathbf{Q}$.

Definition 6.4. We define the $\mathcal{V}$ -category $\mathbf{C}$ as follows.

- The objects of $\mathbf{C}$ are the same as those of $\mathbf{M}$ and $\mathbf{Q}$.
- For objects $A, B \in \mathbf{C}$, we define $\mathbf{C}(A, B)$ as the following object of $\mathcal{V}$,

$$\mathbf{C}(A, B) = (\mathbf{Q}(A, B), \mathbf{M}(A, B), J_{AB} : \mathbf{M}(A, B) \rightarrow \mathbf{Q}(A, B)),$$

where $J : \mathbf{M} \rightarrow \mathbf{Q}$ is the interpretation functor from Assumption 3.3 (3).

- For every object $A \in \mathbf{C}$, we have a morphism $u_A = (\tilde{\text{id}}_0, \tilde{\text{id}}_1) : 1 \rightarrow \mathbf{C}(A, A)$ in $\mathcal{V}$, where $\tilde{\text{id}}_1(*) = \text{id}_A : A \rightarrow A$ in $\mathbf{M}$ and $\tilde{\text{id}}_0(*) = \text{id}_A : A \rightarrow A$ in $\mathbf{Q}$.
- For any $A, B, C \in \mathbf{C}$, we have a morphism $c_{A,B,C} = (c_0, c_1) : \mathbf{C}(A, B) \times \mathbf{C}(B, C) \rightarrow \mathbf{C}(A, C)$ in $\mathcal{V}$, where $c_0 : \mathbf{Q}(A, B) \times \mathbf{Q}(B, C) \rightarrow \mathbf{Q}(A, C)$ and $c_1 : \mathbf{M}(A, B) \times \mathbf{M}(B, C) \rightarrow \mathbf{M}(A, C)$ are the compositions in $\mathbf{Q}$ and $\mathbf{M}$, respectively.

Note that we have a bijection $|\mathcal{V}|(1, \mathbf{C}(A, B)) \cong \mathbf{M}(A, B)$ in **Set**.

The biset-enriched category $\mathbf{C}$ is symmetric monoidal and it encompasses morphisms from both $\mathbf{M}$ and $\mathbf{Q}$. However, it is not monoidal closed. For that, we will need to work in the $\mathcal{V}$ -enriched presheaf category $\overline{\mathbf{C}}$.

Definition 6.5. We define the $\mathcal{V}$ -category $\overline{\mathbf{C}}$ to be $\mathcal{V}^{\mathbf{C}^{\text{op}}}$. Concretely, an object $F \in \overline{\mathbf{C}}$ is a $\mathcal{V}$ -functor $F : \mathbf{C}^{\text{op}} \rightarrow \mathcal{V}$. Because $\mathcal{V}$ is complete, for any $F, G \in \overline{\mathbf{C}}$, we have a hom-object $\overline{\mathbf{C}}(F, G) \in \mathcal{V}$ that represents $\mathcal{V}$ -natural transformations $F \rightarrow G$ [4].

An object in $\overline{\mathbf{C}}$ is a $\mathcal{V}$ -functor $F : \mathbf{C}^{\text{op}} \rightarrow \mathcal{V}$. This means that for each $A \in \mathbf{C}^{\text{op}}$, there is an object $FA \in \mathcal{V}$. And for any $A, B \in \mathbf{C}^{\text{op}}$, there is a morphism $F_{AB} : \mathbf{C}^{\text{op}}(A, B) \rightarrow (FA \Rightarrow FB)$ in $\mathcal{V}$, which is given by the following commutative diagram.

$$\begin{array}{ccc} \mathbf{M}(B, A) & \xrightarrow{F_{AB}^1} & (FA \Rightarrow FB)_1 = |\mathcal{V}|(FA, FB) \\ \downarrow J_{B,A} & & \downarrow p_0 \\ \mathbf{Q}(B, A) & \xrightarrow{F_{AB}^0} & (FA \Rightarrow FB)_0 = \mathbf{Set}((FA)_0, (FB)_0). \end{array}$$

Note that an element $h \in |\mathcal{V}|(FA, FB)$ is a pair of function (h_0, h_1) such that the following commutes.

$$\begin{array}{ccc} (FA)_1 & \xrightarrow{h_1} & (FB)_1 \\ \downarrow f & & \downarrow g \\ (FA)_0 & \xrightarrow{h_0} & (FB)_0 \end{array}$$

Thus we define $p_0(h_0, h_1) = h_0$. So a $\mathcal{V}$ -functor $F : \mathbf{C}^{\text{op}} \rightarrow \mathcal{V}$ induces an ordinary functor $F^0 : \mathbf{Q}^{\text{op}} \rightarrow \mathbf{Set}$, where $F^0(A) = (FA)_0$ and the function $\mathbf{Q}(B, A) \rightarrow \mathbf{Set}(F^0 A, F^0 B)$ is given by F_{AB}^0 for any $A, B \in \mathbf{Q}$.

Theorem 6.6. The $\mathcal{V}$ -category $\overline{\mathbf{C}}$ is a $\mathcal{V}$ -monoidal closed category, where the tensor product $\otimes_{\text{Day}}$ and linear exponential $\multimap_{\text{Day}}$ are given by Day's convolution. The tensor unit is defined by $I := yI = \mathbf{C}(-, I)$, where y is the $\mathcal{V}$ -enriched Yoneda embedding $y : \mathbf{C} \rightarrow \overline{\mathbf{C}}$.

PROOF. This is a result by Day [7]. □

The $\mathcal{V}$ -category $\overline{\mathbf{C}}$ has coproducts. Day's construction implies that the Day tensor product distributes over the coproducts, and that the $\mathcal{V}$ -enriched Yoneda embedding $y : \mathbf{C} \hookrightarrow \overline{\mathbf{C}}$ is strong monoidal.

The $\mathcal{V}$ -adjunction $U_0 \dashv \Delta$ and the $\mathcal{V}$ -monad T can be lifted to $\overline{\mathbf{C}}$.

Definition 6.7. We define $\mathcal{V}$ -functors $\overline{U}_0(F) := U_0 \circ F : \mathcal{V}^{\mathbf{C}^{\text{op}}} \rightarrow \mathbf{Set}^{\mathbf{C}^{\text{op}}}$, $\overline{\Delta}(F) := \Delta \circ F : \mathbf{Set}^{\mathbf{C}^{\text{op}}} \rightarrow \mathcal{V}^{\mathbf{C}^{\text{op}}}$, and $\overline{T} := \overline{\Delta} \circ \overline{U}_0 : \mathcal{V}^{\mathbf{C}^{\text{op}}} \rightarrow \mathcal{V}^{\mathbf{C}^{\text{op}}}$.

Note that $\overline{\Delta}$ is fully faithful and that $\overline{U}_0$ is strong monoidal.

Theorem 6.8. There is a $\mathcal{V}$ -adjunction $\overline{U}_0 : \mathcal{V}^{\mathbf{C}^{\text{op}}} \rightarrow \mathbf{Set}^{\mathbf{C}^{\text{op}}} \dashv \overline{\Delta} : \mathbf{Set}^{\mathbf{C}^{\text{op}}} \rightarrow \mathcal{V}^{\mathbf{C}^{\text{op}}}$.

PROOF. For any $F \in \mathbf{Set}^{\mathbf{C}^{\text{op}}}$, $G \in \overline{\mathbf{C}}$, we need to show that $\mathbf{Set}^{\mathbf{C}^{\text{op}}}(\overline{U}_0 F, G) \cong \mathcal{V}^{\mathbf{C}^{\text{op}}}(F, \overline{\Delta} G)$ that is $\mathcal{V}$ -natural in F and G . This is true since the following isomorphisms follow from properties of *end*.

$$\mathcal{V}^{\mathbf{C}^{\text{op}}}(F, \overline{\Delta} G) \cong \int_{A \in \mathbf{C}} \mathcal{V}(FA, \Delta GA) \cong \int_{A \in \mathbf{C}} \mathbf{Set}(U_0 FA, GA) \cong \mathbf{Set}^{\mathbf{C}^{\text{op}}}(\overline{U}_0 F, G). \quad \square$$

Theorem 6.9. *The monad $\bar{T}$ is a commutative strong monad.*

Theorem 6.9 is a consequence of the following more general theorem.

Theorem 6.10. *Let $\mathcal{V}$ be a complete, cocomplete, symmetric monoidal closed category. Let $\mathbf{A}$ be a $\mathcal{V}$ -category. If T is a commutative strong $\mathcal{V}$ -monad on $\mathcal{V}$, then $\bar{T}(F) = T \circ F$ is a commutative strong $\mathcal{V}$ -monad on $\mathcal{V}^{\mathbf{A}^{\text{op}}}$.*

PROOF. See Appendix F. □

Consider a $\mathcal{V}$ -functor $F : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$. For any $A, B \in \mathbf{C}$, $FA \in \mathbf{Set}$, and the map $F_{AB} : \mathbf{C}(B, A) \rightarrow \mathbf{Set}(FA, FB)$ is uniquely determined by the function $F_{AB}^0 : \mathbf{Q}(B, A) \rightarrow \mathbf{Set}(FA, FB)$. So F is uniquely determined by $F^0 : \mathbf{Q}^{\text{op}} \rightarrow \mathbf{Set}$. In fact, we have the following theorem.

Theorem 6.11. *We have $\mathbf{Set}^{\mathbf{C}^{\text{op}}} \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}$.*

PROOF. See Appendix G. □

The following theorem shows the maps in the Kleisli category of $\bar{T}$ are essentially maps in $\mathbf{Set}^{\mathbf{Q}^{\text{op}}}$.

Theorem 6.12. *For any $F, G \in \bar{\mathbf{C}}$, we have $\bar{\mathbf{C}}(F, \bar{T}G) \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0)$.*

PROOF. $\bar{\mathbf{C}}(F, \bar{T}G) = \bar{\mathbf{C}}(F, \Delta \bar{U}_0 G) \cong \mathbf{Set}^{\mathbf{C}^{\text{op}}}(\bar{U}_0 F, \bar{U}_0 G) \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0)$. □

6.3 A reflective subcategory $\tilde{\bar{\mathbf{C}}}$ of $\bar{\mathbf{C}}$

The $\mathcal{V}$ -category $\bar{\mathbf{C}}$ itself is not a model for Proto-Quipper with dynamic lifting. For example, it does not have a map $\mathbf{Bit} \rightarrow \bar{T} \mathbf{Bool}$ for dynamic lifting. Namely, we define $\mathbf{Bool} := yI + yI = \mathbf{C}(-, I) + \mathbf{C}(-, I)$ and $\mathbf{Bit} := y\mathbf{Bit} = \mathbf{C}(-, \mathbf{Bit}) \in \bar{\mathbf{C}}$, where $\mathbf{Bit} \in \mathbf{C}$. Note that $\mathbf{Bit} = I + I$ in $\mathbf{Q}$. Consider the following

$$\begin{aligned} \bar{\mathbf{C}}(\mathbf{Bit}, \bar{T} \mathbf{Bool}) &\cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(\mathbf{Bit}^0, \mathbf{Bool}^0) \\ &\cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(\mathbf{Q}(-, \mathbf{Bit}), \mathbf{Q}(-, I) + \mathbf{Q}(-, I)) \\ &= \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(\mathbf{Q}(-, I + I), \mathbf{Q}(-, I) + \mathbf{Q}(-, I)). \end{aligned}$$

So a map in $\bar{\mathbf{C}}(\mathbf{Bit}, \bar{T} \mathbf{Bool})$ is the same as a natural transformation from $\mathbf{Q}(-, I + I)$ to $\mathbf{Q}(-, I) + \mathbf{Q}(-, I)$ in $\mathbf{Set}^{\mathbf{Q}^{\text{op}}}$. Moreover, for condition (f) of Definition 3.5 to be satisfied, this natural transformation should be a left inverse of the canonical natural transformation $\mathbf{Q}(-, I) + \mathbf{Q}(-, I) \rightarrow \mathbf{Q}(-, I + I)$. On the other hand, by the Yoneda lemma, every natural transformation from $\mathbf{Q}(-, I + I)$ to $\mathbf{Q}(-, I) + \mathbf{Q}(-, I)$ either takes all of its values in the left component or in the right component of the disjoint union. Therefore, it cannot be a left inverse to $\mathbf{Q}(-, I) + \mathbf{Q}(-, I) \rightarrow \mathbf{Q}(-, I + I)$. It follows that dynamic lifting cannot be interpreted in $\bar{\mathbf{C}}$. To fix this, we now consider a reflective subcategory of $\bar{\mathbf{C}}$, in the style of Lambek [22].

Definition 6.13. A $\mathcal{V}$ -functor $F : \mathbf{C}^{\text{op}} \rightarrow \mathcal{V}$ is called *smooth* if $F^0 : \mathbf{Q}^{\text{op}} \rightarrow \mathbf{Set}$ is a product-preserving functor, i.e., $F^0(A + B) \cong F^0 A \times F^0 B$ for any $A, B \in \mathbf{Q}$.

Observe that for any $A \in \mathbf{C}$, the $\mathcal{V}$ -enriched Yoneda embedding y of A , which is $\mathbf{C}(-, A)$, is smooth. Because $\mathbf{C}(-, A)^0 = \mathbf{Q}(-, A)$, and for any $B_1, B_2 \in \mathbf{Q}$, we have $\mathbf{Q}(B_1 + B_2, A) \cong \mathbf{Q}(B_1, A) \times \mathbf{Q}(B_2, A)$. Thus, the codomain of y consists of smooth $\mathcal{V}$ -functors.

Definition 6.14. We define $\tilde{\mathbf{C}}$ to be the full $\mathcal{V}$ -subcategory of smooth functors.

Definition 6.15. We define the *Lambek embedding* $\bar{y} : \mathbf{C} \rightarrow \tilde{\mathbf{C}}$ to be the corestriction of the Yoneda embedding y , i.e., it is the unique $\mathcal{V}$ -functor such that the following diagram commutes.

$$\begin{array}{ccc} \mathbf{C} & & \\ \downarrow \bar{y} & \searrow y & \\ \tilde{\mathbf{C}} & \xrightarrow{i} & \bar{\mathbf{C}} \end{array}$$

Theorem 6.16. The $\mathcal{V}$ -category $\tilde{\mathbf{C}}$ is a reflective $\mathcal{V}$ -subcategory of $\bar{\mathbf{C}}$, i.e., the inclusion $\mathcal{V}$ -functor $i : \tilde{\mathbf{C}} \hookrightarrow \bar{\mathbf{C}}$ has a left adjoint K .

PROOF. See Appendix H. □

Using results of Day [8, 9] (see also Malherbe [25] for a more recent exposition), we can furthermore show that $\tilde{\mathbf{C}}$ is symmetric monoidal and K is strong monoidal. See Appendix I for further details. We now give an explicit definition of the monoidal closed structure in $\tilde{\mathbf{C}}$.

Definition 6.17. For any $F, G \in \tilde{\mathbf{C}}$, we define the tensor product, internal hom and tensor unit in $\tilde{\mathbf{C}}$ as $F \otimes_{\text{Lam}} G := K(iF \otimes_{\text{Day}} iG)$, $F \multimap_{\text{Lam}} G := iF \multimap_{\text{Day}} iG$, and $I := \bar{y}I = \mathbf{C}(-, I)$, respectively.

In the above definition, the linear exponential $F \multimap_{\text{Lam}} G$ is well-defined because $iF \multimap_{\text{Day}} iG$ is an object in $\tilde{\mathbf{C}}$ (Theorem I.1 in Appendix I).

Theorem 6.18. The $\mathcal{V}$ -category $\tilde{\mathbf{C}}$ is symmetric monoidal closed. For any $F, G, H \in \tilde{\mathbf{C}}$, there is a $\mathcal{V}$ -natural isomorphism $\tilde{\mathbf{C}}(F \otimes_{\text{Lam}} G, H) \cong \tilde{\mathbf{C}}(F, G \multimap_{\text{Lam}} H)$.

PROOF.

$$\begin{aligned} \tilde{\mathbf{C}}(F \otimes_{\text{Lam}} G, H) &\cong \tilde{\mathbf{C}}(K(iF \otimes_{\text{Day}} iG), H) \cong \bar{\mathbf{C}}(iF \otimes_{\text{Day}} iG, iH) \\ &\cong \bar{\mathbf{C}}(iF, iG \multimap_{\text{Day}} iH) \cong \tilde{\mathbf{C}}(F, G \multimap_{\text{Lam}} H) \end{aligned}$$

□

6.4 A commutative strong monad on $\tilde{\mathbf{C}}$

The $\mathcal{V}$ -category $\tilde{\mathbf{C}}$ has a commutative strong monad. In the following we write $[\mathcal{V}^{\text{C}^{\text{op}}}]_{\text{prod}}$ for $\tilde{\mathbf{C}}$ and $\mathcal{V}^{\text{C}^{\text{op}}}$ for $\bar{\mathbf{C}}$. We write $[\mathbf{Set}^{\text{Q}^{\text{op}}}]_{\text{prod}}$ for the full subcategory of product-preserving functors of $\mathbf{Set}^{\text{Q}^{\text{op}}}$. Consider the following diagram.

$$\begin{array}{ccc} \mathbf{Set}^{\text{C}^{\text{op}}} & \xrightleftharpoons[\bar{\Delta}]{\bar{U}_0} & \mathcal{V}^{\text{C}^{\text{op}}} \\ L \downarrow j & & K \downarrow i \\ [\mathbf{Set}^{\text{C}^{\text{op}}}]_{\text{prod}} & \xrightleftharpoons[\tilde{\Delta}]{\tilde{U}_0} & [\mathcal{V}^{\text{C}^{\text{op}}}]_{\text{prod}} \end{array}$$

We define the $\mathcal{V}$ -functor $\tilde{U}_0 : [\mathcal{V}^{\text{C}^{\text{op}}}]_{\text{prod}} \rightarrow [\mathbf{Set}^{\text{C}^{\text{op}}}]_{\text{prod}}$ by restricting the domain of $\bar{U}_0$ to $[\mathcal{V}^{\text{C}^{\text{op}}}]_{\text{prod}}$. Here $[\mathbf{Set}^{\text{C}^{\text{op}}}]_{\text{prod}}$ is the full $\mathcal{V}$ -subcategory of smooth $\mathcal{V}$ -functors. Similarly, $\tilde{\Delta} : [\mathbf{Set}^{\text{C}^{\text{op}}}]_{\text{prod}} \rightarrow [\mathcal{V}^{\text{C}^{\text{op}}}]_{\text{prod}}$ is a restriction of $\bar{\Delta}$. We have a monoidal adjunction $L \dashv j$, since $[\mathbf{Set}^{\text{Q}^{\text{op}}}]_{\text{prod}} \cong [\mathbf{Set}^{\text{C}^{\text{op}}}]_{\text{prod}}$, the full subcategory of product-preserving functors, is reflective in $\mathbf{Set}^{\text{Q}^{\text{op}}} \cong \mathbf{Set}^{\text{C}^{\text{op}}}$. The other adjunction $K \dashv i$ is from Theorem 6.16. We write $\tilde{T} = \tilde{\Delta} \circ \tilde{U}_0$. Observe that $\tilde{T}$ is $\bar{T}$ with a restricted domain.

Theorem 6.19. *We have $i \circ \tilde{\Delta} \cong \bar{\Delta} \circ j$ and $j \circ \tilde{U}_0 \cong \bar{U}_0 \circ i$, therefore $i \circ \tilde{T} \cong \bar{T} \circ i$. Moreover, $\tilde{U}_0 \circ K \cong L \circ \bar{U}_0$.*

PROOF. By definition of $\tilde{\Delta}$, $\tilde{U}_0$, L , and K . $\square$

Theorem 6.20. *We have a $\mathcal{V}$ -adjunction $\tilde{U}_0 \dashv \tilde{\Delta} : [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}} \rightarrow [\mathcal{V}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}$. And $\tilde{U}_0$ is strong monoidal.*

PROOF. For any $X \in [\mathcal{V}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}$, $Y \in [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}$, we have

$$\begin{aligned} [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 X, Y) &\cong \mathbf{Set}^{\mathbf{C}^{\text{op}}}(j\tilde{U}_0 X, jY) \cong \mathbf{Set}^{\mathbf{C}^{\text{op}}}(\bar{U}_0 iX, jY) \\ &\cong \mathcal{V}^{\mathbf{C}^{\text{op}}}(iX, \bar{\Delta} jY) \cong \mathcal{V}^{\mathbf{C}^{\text{op}}}(iX, i\tilde{\Delta} Y) \cong [\mathcal{V}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(X, \tilde{\Delta} Y). \end{aligned}$$

The $\mathcal{V}$ -functor $\tilde{U}_0$ is strong monoidal. For any $F, G \in [\mathcal{V}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}$, we have $\tilde{U}_0 I \cong \bar{U}_0 I \cong I$ and

$$\begin{aligned} \tilde{U}_0(F \otimes_{\text{Lam}} G) &= \tilde{U}_0 K(iF \otimes_{\text{Day}} iG) \cong L\bar{U}_0(iF \otimes_{\text{Day}} iG) \cong L(\bar{U}_0 iF \otimes_{\text{Day}} \bar{U}_0 iG) \\ &\cong L(j\tilde{U}_0 F \otimes_{\text{Day}} j\tilde{U}_0 G) = \tilde{U}_0 F \otimes_{\text{Lam}} \tilde{U}_0 G. \end{aligned} \quad \square$$

Theorem 6.21. *There is a $\mathcal{V}$ -natural transformation $\rho : K \circ \bar{T} \rightarrow \tilde{T} \circ K$.*

PROOF. For any $F \in \bar{\mathbf{C}}$, let $\eta_F : F \rightarrow iKF$ be the unit and $\epsilon_F : KiF \rightarrow F$ be the counit (which is an isomorphism). We define ρ_F to be the composition $K\bar{T}F \xrightarrow{K\bar{T}\eta_F} K\bar{T}iKF \xrightarrow{\cong} Ki\tilde{T}KF \xrightarrow{\epsilon_{\tilde{T}KF}} \tilde{T}KF$. $\square$

The natural transformation ρ is one of the components for defining the strength for $\tilde{T}$.

Theorem 6.22. *The $\mathcal{V}$ -functor $\tilde{T}$ is a commutative strong monad.*

PROOF. For any $F, G \in \tilde{\mathbf{C}}$, the strength of $\tilde{T}$ is given by

$$\begin{aligned} F \otimes_{\text{Lam}} \tilde{T}G &= K(iF \otimes_{\text{Day}} i\tilde{T}G) \xrightarrow{\cong} K(iF \otimes_{\text{Day}} \bar{T}iG) \\ &\xrightarrow{K\bar{T}} K\bar{T}(iF \otimes_{\text{Day}} iG) \xrightarrow{\rho} \tilde{T}K(iF \otimes_{\text{Day}} iG) = \tilde{T}(F \otimes_{\text{Lam}} G). \end{aligned}$$

Note that $\bar{t}$ is the strength for $\bar{T}$. The verification of the strength diagrams is in Appendix J. $\square$

Similarly to Theorem 6.12, we have the following theorem for $\tilde{T}$.

Theorem 6.23. *For any $F, G \in \tilde{\mathbf{C}}$, we have the following $\mathcal{V}$ -natural isomorphisms.*

$$\tilde{\mathbf{C}}(F, \tilde{T}G) \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0).$$

PROOF. We have

$$\tilde{\mathbf{C}}(F, \tilde{T}G) = \tilde{\mathbf{C}}(F, \tilde{\Delta}\tilde{U}_0 G) \cong [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 F, \tilde{U}_0 G) \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0).$$

Note that by Theorem 6.11, $[\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}} \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$. $\square$

Corollary 6.24. *Consider the underlying category $V(\tilde{\mathbf{C}})$ of $\tilde{\mathbf{C}}$. The functor $V\tilde{T}$ is a monad on $V(\tilde{\mathbf{C}})$. Moreover, for any $F, G \in V(\tilde{\mathbf{C}})$, we have the following isomorphisms.*

$$\begin{aligned} V(\tilde{\mathbf{C}})(F, V\tilde{T}G) &\cong |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 F, \tilde{U}_0 G)) \\ &\cong |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0)) \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0). \end{aligned}$$

6.5 Dynamic lifting in $\tilde{\mathbf{C}}$

Since $\overline{\mathbf{C}}$ has coproducts and $\tilde{\mathbf{C}}$ is a reflective subcategory, the coproduct of $A, B \in \tilde{\mathbf{C}}$ is defined as $A +' B = K(iA + iB)$. In $\tilde{\mathbf{C}}$, we define $\mathbf{Bool} := \bar{y}I +' \bar{y}I = K(\bar{y}I + \bar{y}I)$ and $\mathbf{Bit} := \bar{y}(\mathbf{Bit})$, where $I, \mathbf{Bit} \in \mathbf{C}$. There exists maps zero, one : $\bar{y}I \rightarrow \mathbf{Bit}$ in $\tilde{\mathbf{C}}$. We are now ready to define a map for dynamic lifting.

Theorem 6.25. *There are $\mathcal{V}$ -natural transformations $\text{init} : \mathbf{Bool} \rightarrow \mathbf{Bit}$ and $\text{dynlift} : \mathbf{Bit} \rightarrow \tilde{\mathbf{Bool}}$ in $\tilde{\mathbf{C}}$ such that the following diagram commutes.*

$$\begin{array}{ccc} & & \mathbf{Bit} \\ & \nearrow \text{init} & \downarrow \text{dynlift} \\ \mathbf{Bool} & \xrightarrow{\eta} & \tilde{\mathbf{Bool}} \end{array}$$

PROOF. We define $\text{init} = [\text{zero}, \text{one}] : \mathbf{Bool} \rightarrow \mathbf{Bit}$. Firstly, we want to show that $\tilde{\text{init}} : \tilde{\mathbf{Bool}} \rightarrow \tilde{\mathbf{Bit}}$ is an isomorphism. Using Yoneda's principle, we just need to show $\tilde{\mathbf{C}}(F, \tilde{\text{init}}) : \tilde{\mathbf{C}}(F, \tilde{\mathbf{Bool}}) \rightarrow \tilde{\mathbf{C}}(F, \tilde{\mathbf{Bit}})$ is an isomorphism for any $F \in \tilde{\mathbf{C}}$. By Theorem 6.23, this is equivalent to showing that

$$[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, \text{init}^0) : [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, \mathbf{Bool}^0) \rightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, \mathbf{Bit}^0)$$

is an isomorphism. This is the case because the Lambek embedding $\kappa : \mathbf{Q} \hookrightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ preserves coproducts, $\mathbf{Bit} = I + I \in \mathbf{Q}$, and the map $\text{init}^0 : \kappa I + \kappa I \rightarrow \kappa(I + I)$ is an isomorphism in $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$. We therefore define dynlift as the composition $(\tilde{\text{init}})^{-1} \circ \eta : \mathbf{Bit} \rightarrow \tilde{\mathbf{Bool}}$. As a result, we have the following commutative diagram.

$$\begin{array}{ccccc} \mathbf{Bool} & \xrightarrow{\eta} & \tilde{\mathbf{Bool}} & & \\ \downarrow \text{init} & & \downarrow \tilde{\text{init}} & \searrow \text{id} & \\ \mathbf{Bit} & \xrightarrow{\eta} & \tilde{\mathbf{Bit}} & \xrightarrow{(\tilde{\text{init}})^{-1}} & \tilde{\mathbf{Bool}} \end{array} \quad \square$$

Recall that the category $\mathbf{Q}$ is enriched in convex spaces, i.e., the hom-sets of $\mathbf{Q}$ are convex spaces and the composition is bilinear with respect to the convex sum. The following theorem shows that its Lambek embedding is also enriched in convex spaces.

Theorem 6.26. *The category $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ is enriched in convex spaces. Moreover, the Lambek embedding $\kappa : \mathbf{Q} \hookrightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ preserves the convex sum in $\mathbf{Q}$.*

PROOF. The proof is in Appendix B. □

The above theorem implies that for any $A, B \in \tilde{\mathbf{C}}$, the Kleisli-hom $\tilde{\mathbf{C}}(A, \tilde{\text{TB}})$ is convex because of the isomorphism $\tilde{\mathbf{C}}(A, \tilde{\text{TB}}) \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(A^0, B^0)$ from Theorem 6.23.

6.6 $\mathcal{V}(\tilde{\mathbf{C}})$ is a model for Proto-Quipper with dynamic lifting

We are now ready to prove our main theorem.

Theorem 6.27. *The category $\mathcal{V}(\tilde{\mathbf{C}})$ is a model for Proto-Quipper with dynamic lifting.*

PROOF. We show each of the conditions (a)–(f) from Definitions 3.1–3.5.

- (a) Since $\tilde{\mathbf{C}}$ is a biset-enriched symmetric monoidal category, the underlying category $\mathcal{V}(\tilde{\mathbf{C}})$ is an ordinary symmetric monoidal category. Since we have a natural isomorphism $\tilde{\mathbf{C}}(B \otimes A, C) \cong \tilde{\mathbf{C}}(B, A \multimap C)$ in $\mathcal{V}$, we also have a natural isomorphism $|\mathcal{V}|(1, \tilde{\mathbf{C}}(B \otimes A, C)) \cong |\mathcal{V}|(1, \tilde{\mathbf{C}}(B, A \multimap C))$ in $\mathbf{Set}$.

Since $\tilde{\mathbf{C}}$ has coproduct, $V(\tilde{\mathbf{C}})$ has coproduct as well.

$$\begin{aligned}
 V(\tilde{\mathbf{C}})(F + G, H) &= |\mathcal{V}|(1, \tilde{\mathbf{C}}(F + G, H)) \\
 &\cong |\mathcal{V}|(1, \tilde{\mathbf{C}}(F, H) \times \tilde{\mathbf{C}}(G, H)) \\
 &\cong |\mathcal{V}|(1, \tilde{\mathbf{C}}(F, H)) \times |\mathcal{V}|(1, \tilde{\mathbf{C}}(G, H)) \\
 &= V(\tilde{\mathbf{C}})(F, H) \times V(\tilde{\mathbf{C}})(G, H).
 \end{aligned}$$

(b) We define $p(X) = \sum_{x \in X} I$ and $\flat(A) = V(\mathcal{V})(I, A) = |\mathcal{V}|(1, \tilde{\mathbf{C}}(I, A))$. We can show that we have an adjunction $p \dashv \flat : \mathbf{Set} \rightarrow V(\tilde{\mathbf{C}})$.

$$\begin{aligned}
 V(\tilde{\mathbf{C}})(pX, A) &= |\mathcal{V}|(1, \tilde{\mathbf{C}}(pX, A)) \\
 &= |\mathcal{V}|(1, \tilde{\mathbf{C}}(\sum_{x \in X} I, A)) \\
 &\cong |\mathcal{V}|(1, \prod_{x \in X} \tilde{\mathbf{C}}(I, A)) \\
 &\cong \prod_{x \in X} |\mathcal{V}|(1, \tilde{\mathbf{C}}(I, A)) \\
 &\cong \mathbf{Set}(X, \flat(A)).
 \end{aligned}$$

(c) By Theorem 6.22, $V(\tilde{\mathbf{C}})$ have a commutative strong monad $V\tilde{T} : V(\tilde{\mathbf{C}}) \rightarrow V(\tilde{\mathbf{C}})$.

(d) By Corollary 6.24 and Theorem 6.26.

(e) • First we need to define a functor $\psi : \mathbf{M} \hookrightarrow V(\tilde{\mathbf{C}})$. We define it as the following composition.

$$\mathbf{M} \cong V(\mathbf{C}) \xrightarrow{V\bar{y}} V(\tilde{\mathbf{C}})$$

The functor ψ is strong monoidal, because $V\bar{y}$ is strong monoidal.

• Next we need to define a functor $\phi : \mathbf{Q} \hookrightarrow Kl_{V\tilde{T}}(V(\tilde{\mathbf{C}}))$. We write

$$\theta_{F,G} : \tilde{\mathbf{C}}(F, \tilde{T}G) \cong [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 F, \tilde{U}_0 G)$$

for any $F, G \in \tilde{\mathbf{C}}$. We also write $\Omega : [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}} \cong [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$. We have

$$\begin{aligned}
 Kl_{V\tilde{T}}(V(\tilde{\mathbf{C}}))(F, G) &= |\mathcal{V}|(1, \tilde{\mathbf{C}}(F, \tilde{T}G)) \\
 &\stackrel{|\mathcal{V}|(1, \theta_{F,G})}{\cong} |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 F, \tilde{U}_0 G)) \\
 &\stackrel{|\mathcal{V}|(1, \Omega)}{\cong} |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0)) \\
 &= [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(F^0, G^0)
 \end{aligned}$$

for any $F, G \in \tilde{\mathbf{C}}$. The category $Kl_{V\tilde{T}}(V(\tilde{\mathbf{C}}))$ is enriched in convex spaces because so is $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$.

Now we define ϕ . On objects, we define

$$\phi(S) = \bar{y}S = \mathbf{C}(-, S).$$

On morphisms, for any $S, U \in \mathbf{Q}$, we define $\phi_{S,U}$ by the following composition of isomorphisms.

$$\begin{aligned}
\mathbf{Q}(S, U) &\stackrel{\kappa_{S,U}}{\cong} [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(\kappa S, \kappa U) = [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}((\bar{y}S)^0, (\bar{y}U)^0) \\
&\cong |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}((\bar{y}S)^0, (\bar{y}U)^0)) \\
&\stackrel{|\mathcal{V}|(1, \Omega^{-1})}{\cong} |\mathcal{V}|(1, [\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\tilde{U}_0 \bar{y}S, \tilde{U}_0 \bar{y}U)) \\
&\stackrel{|\mathcal{V}|(1, \theta_{\bar{y}S, \bar{y}U}^{-1})}{\cong} |\mathcal{V}|(1, [\mathcal{V}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}(\bar{y}S, \tilde{\Delta} \tilde{U}_0 \bar{y}U)) = Kl_{\mathcal{V}\tilde{T}}(V(\tilde{\mathbf{C}}))(\bar{y}S, \bar{y}U)
\end{aligned}$$

Since the Lambek embedding $\kappa_{S,U}$ preserves convex sum (Theorem 6.26) and the composition $|\mathcal{V}|(1, \theta_{\bar{y}S, \bar{y}U}^{-1} \circ \Omega^{-1})$ is an isomorphism, we conclude that ϕ preserves convex sum.

Next we need to show ϕ is strong monoidal. Since κ is strong monoidal, we have the natural isomorphisms $I \stackrel{e'}{\cong} \kappa I$ and $\kappa S \otimes \kappa U \stackrel{m'_{S,U}}{\cong} \kappa(S \otimes U)$ for any $S, U \in \mathbf{Q}$. Recall that for any $S \in \mathbf{Q}$, $\kappa S = \mathbf{Q}(-, S) = \tilde{U}_0 \mathbf{C}(-, S)$ and $\tilde{U}_0$ is strong monoidal. Via the isomorphism Ω (which preserves the monoidal structure) we have the following natural isomorphisms in $[\mathbf{Set}^{\mathbf{C}^{\text{op}}}]_{\text{prod}}$.

$$\Omega^{-1}(e') : \tilde{U}_0 \mathbf{C}(-, I) \rightarrow \tilde{U}_0 \mathbf{C}(-, I)$$

$$\Omega^{-1}(m'_{S,U}) : \tilde{U}_0(\mathbf{C}(-, S) \otimes \mathbf{C}(-, U)) \rightarrow \tilde{U}_0 \mathbf{C}(-, S \otimes U).$$

Now let $m_{S,U} = \theta_{\bar{y}S, \bar{y}U}^{-1}(\Omega^{-1}(m'_{S,U})) : \mathbf{C}(-, S) \otimes \mathbf{C}(-, U) \rightarrow \tilde{T}\mathbf{C}(-, S \otimes U)$ and $e = \theta_{\bar{y}I, \bar{y}I}^{-1}(\Omega^{-1}(e')) : \mathbf{C}(-, I) \rightarrow \tilde{T}\mathbf{C}(-, I)$. Then it is obvious that e is an isomorphism in $Kl_{\mathcal{V}\tilde{T}}(V(\tilde{\mathbf{C}}))$. The inverse of $m_{S,U}$ is defined as $\theta_{\bar{y}S, \bar{y}U}^{-1}(\Omega^{-1}(m'_{S,U}{}^{-1}))$, which can be verified. We can furthermore show that $m_{S,U}$ is natural and that $e, m_{S,U}$ satisfies the strength diagrams for any $S, U \in \mathbf{Q}$. For example, showing $m_{S,U}$ is natural in $Kl_{\mathcal{V}\tilde{T}}(V(\tilde{\mathbf{C}}))$, via the adjunction $\tilde{U}_0 \dashv \tilde{\Delta}$, is equivalent to the naturality of $\Omega^{-1}(m'_{S,U})$.

- Lastly, we want to show that the following diagram commutes.

$$\begin{array}{ccc}
\mathbf{M}(S, U) & \xrightarrow{\psi_{S,U}} & |\mathcal{V}|(1, \tilde{\mathbf{C}}(\bar{y}S, \bar{y}U)) = V(\tilde{\mathbf{C}})(\bar{y}S, \bar{y}U) \\
\downarrow J_{S,U} & & \downarrow E_{S,U} \\
\mathbf{Q}(S, U) & \xleftarrow[\phi_{S,U}^{-1}]{} & |\mathcal{V}|(1, \tilde{\mathbf{C}}(\bar{y}S, \tilde{T}\bar{y}U)) = Kl_{\mathcal{V}\tilde{T}}(V(\tilde{\mathbf{C}}))(\bar{y}S, \bar{y}U)
\end{array}$$

Let $f \in \mathbf{M}(S, U)$. We write $(f, J_{S,U}f)$ for the corresponding map in $|\mathcal{V}|(1, \mathbf{C}(S, U))$. It corresponds to the following map in $\tilde{\mathbf{C}}$ via the enriched Yoneda embedding.

$$\mathbf{C}(-, S) \xrightarrow{\bar{y}(f, J_{S,U}f)} \mathbf{C}(-, U)$$

Composing η_U with the above map, we have the following map in $\mathcal{V}$.

$$\mathbf{C}(-, S) \xrightarrow{\bar{y}(f, J_{S,U}f)} \mathbf{C}(-, U) \xrightarrow{\eta_U} \tilde{T}\mathbf{C}(-, U)$$

So for any $A \in \mathbf{C}$, we have the following.

$$\begin{array}{ccccc}
\mathbf{M}(A, S) & \xrightarrow{\mathbf{M}(A, f)} & \mathbf{M}(A, U) & \xrightarrow{J_{A,U}} & \mathbf{Q}(A, U) \\
\downarrow J_{A,S} & & \downarrow J_{A,U} & & \downarrow \text{id} \\
\mathbf{Q}(A, S) & \xrightarrow{\mathbf{Q}(A, J_{S,U}f)} & \mathbf{Q}(A, U) & \xrightarrow{\text{id}} & \mathbf{Q}(A, U)
\end{array}$$

$$\begin{aligned}
& \text{Since } \phi_{S,U}^{-1} = \kappa_{S,U}^{-1} \circ \Omega_{\tilde{U}_0 \bar{y} S, \tilde{U}_0 \bar{y} U} \circ |\mathcal{V}|(1, \theta_{\bar{y} S, \bar{y} U}), \text{ we have} \\
& \phi_{S,U}^{-1}(\eta_U \circ \bar{y}(f, J_{S,U} f)) = \kappa_{S,U}^{-1} \Omega_{\tilde{U}_0 \bar{y} S, \tilde{U}_0 \bar{y} U} |\mathcal{V}|(1, \theta_{\bar{y} S, \bar{y} U})(\eta_U \circ \bar{y}(f, J_{S,U} f)) \\
& \quad = \kappa_{S,U}^{-1} \Omega_{\tilde{U}_0 \bar{y} S, \tilde{U}_0 \bar{y} U}(\tilde{U}_0 \bar{y}(f, J_{S,U} f)) \\
& \quad = \kappa_{S,U}^{-1}(\mathbf{Q}(-, J_{S,U} f)) = J_{S,U} f \in \mathbf{Q}(S, U).
\end{aligned}$$

(f) This is by Theorem 6.25.

□

7 Conclusion

We have given an axiomatization of a categorical semantics for Proto-Quipper with dynamic lifting. We defined a type system with a modality to keep track of functions that use dynamic lifting. The main benefit of our type system is that it statically ensures that the boxing operation can only be applied to a function that does not use dynamic lifting. We also gave an operational semantics for dynamic lifting. The operational semantics models both circuit generation and circuit execution. We proved that the operational semantics is sound with respect to the abstract categorical semantics. We also gave a construction of a concrete model based on biset-enrichment. Lastly, we gave some examples of quantum algorithms that rely on dynamic lifting.

There are many things left to be done. One of them is how to combine dynamic lifting with dependent types and/or recursion. At this point, we have a model for dynamic lifting, but it does not support dependent types or recursion. We also have a model for Proto-Quipper with dependent types [15], but it does not support dynamic lifting or recursion. Finally, Lindenhovius et al. [24] have a model for Proto-Quipper with recursion, but it does not support dynamic lifting or dependent types. We do not think that adding recursion to Proto-Quipper-Dyn would create fundamental difficulties at the level of the syntax, type system, or operational semantics (although languages with recursion are usually better handled by small-step operational semantics rather than the big-step semantics we considered here). On the other hand, finding a concrete denotational semantics gets more complicated the more features are included in the programming language, and how to add dependent types or recursion to a semantics of Proto-Quipper-Dyn is an open problem.

Acknowledgements

This work was supported by the U.S. Air Force Office of Scientific Research under Award No. FA9550-21-1-0041 and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- [1] Guillaume Allais. 2024. Scoped and typed staging by evaluation. In *Proceedings of the 2024 ACM SIGPLAN International Workshop on Partial Evaluation and Program Manipulation*. 83–93.
- [2] Nick Benton. 1995. A mixed linear and non-linear logic: proofs, terms and models (extended abstract). In *Proceedings of the 8th Workshop on Computer Science Logic, CSL'94, Selected Papers (Springer Lecture Notes in Computer Science 933)*. 121–135.
- [3] Benjamin Bichsel, Maximilian Baader, Timon Gehr, and Martin Vechev. 2020. Silq: A High-Level Quantum Language with Safe Uncomputation and Intuitive Semantics. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (London, UK) (PLDI 2020)*. Association for Computing Machinery, New York, NY, USA, 286–300. doi:10.1145/3385412.3386007

- [4] Francis Borceux. 1994. *Handbook of Categorical Algebra, Volume 2: Categories and Structures*. Cambridge University Press.
- [5] Sergey Bravyi and Alexei Kitaev. 2005. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A* 71, 2 (2005), 022316.
- [6] Andrea Colledan and Ugo Dal Lago. 2022. On Dynamic Lifting and Effect Typing in Circuit Description Languages (Extended Version). (2022). Available from [arXiv:2202.07636](https://arxiv.org/abs/2202.07636).
- [7] Brian Day. 1970. On closed categories of functors. In *Reports of the Midwest Category Seminar IV (Springer Lecture Notes in Mathematics, Vol. 137)*. 1–38.
- [8] Brian Day. 1972. A reflection theorem for closed categories. *Journal of Pure and Applied Algebra* 2, 1 (1972), 1–11.
- [9] Brian Day. 1973. Note on monoidal localisation. *Bulletin of the Australian Mathematical Society* 8, 1 (1973), 1–16.
- [10] Jana Dunfield and Neel Krishnaswami. 2021. Bidirectional Typing. *ACM Comput. Surv.* 54, 5, Article 98 (May 2021), 38 pages. doi:10.1145/3450952
- [11] Frank Fu. 2025. An implementation of Proto-Quipper. Available from <https://gitlab.com/frank-peng-fu/dpq-remake>.
- [12] Peng Fu, Kohei Kishida, Neil J. Ross, and Peter Selinger. 2020b. A tutorial introduction to quantum circuit programming in dependently typed Proto-Quipper. In *Proceedings of the 12th International Conference on Reversible Computation, RC 2020, Oslo, Norway (Lecture Notes in Computer Science, Vol. 12227)*. Springer, 153–168. doi:10.1007/978-3-030-52482-1_9 Also available from [arXiv:2005.08396](https://arxiv.org/abs/2005.08396).
- [13] Peng Fu, Kohei Kishida, Neil J. Ross, and Peter Selinger. 2022. A biset-enriched categorical model for Proto-Quipper with dynamic lifting. In *Proceedings 19th International Conference on Quantum Physics and Logic, QPL 2022, Wolfson College, Oxford, UK, 27 June - 1 July 2022 (EPTCS, Vol. 394)*, Stefano Gogioso and Matty Hoban (Eds.). 302–342. doi:10.4204/EPTCS.394.16
- [14] Peng Fu, Kohei Kishida, Neil J. Ross, and Peter Selinger. 2023. Proto-Quipper with Dynamic Lifting. *Proc. ACM Program. Lang.* 7, POPL 2023, Article 11 (jan 2023), 26 pages. doi:10.1145/3571204 Also available from [arXiv:2204.13041](https://arxiv.org/abs/2204.13041).
- [15] Peng Fu, Kohei Kishida, and Peter Selinger. 2020a. Linear dependent type theory for quantum programming languages. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2020, Saarbrücken, Germany*. 440–453. doi:10.1145/3373718.3394765 Also available from [arXiv:2004.13472](https://arxiv.org/abs/2004.13472).
- [16] Alexander Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. 2013a. Quipper: a scalable quantum programming language. In *Proceedings of the 34th Annual ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2013, Seattle (ACM SIGPLAN Notices, Vol. 48(6))*. 333–342. doi:10.1145/2499370.2462177 Also available from [arXiv:1304.3390](https://arxiv.org/abs/1304.3390).
- [17] Alexander Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. 2013b. An introduction to quantum programming in Quipper. In *Proceedings of the 5th International Conference on Reversible Computation, RC 2013, Victoria, British Columbia (Lecture Notes in Computer Science, Vol. 7948)*. Springer, 110–124. doi:10.1007/978-3-642-38986-3_10 Also available from [arXiv:1304.5485](https://arxiv.org/abs/1304.5485).
- [18] IBM. 2025. Classical feedforward and control flow. <https://docs.quantum.ibm.com/guides/classical-feedforward-and-control-flow>. Accessed: 2025-03-05.
- [19] G. M. Kelly. 1982. *Basic concepts of enriched category theory*. Lecture Notes of the London Mathematical Society, Vol. 64. Cambridge University Press.
- [20] Anders Kock. 1972. Strong functors and monoidal monads. *Archiv der Mathematik* 23, 1 (1972), 113–120.
- [21] András Kovács. 2022. Staged compilation with two-level type theory. *Proceedings of the ACM on Programming Languages* 6, ICFP (2022), 540–569.
- [22] Joachim Lambek. 1966. *Completions of categories: Seminar lectures given 1966 in Zürich*. Lecture Notes in Mathematics, Vol. 24. Springer. doi:10.1007/BFb0077265
- [23] Dongho Lee, Valentin Perrelle, Benoît Valiron, and Zhaowei Xu. 2021. Concrete categorical model of a quantum circuit description language with measurement. In *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021 (LIPIcs, Vol. 213)*, Mikolaj Bojanczyk and Chandra Chekuri (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 51:1–51:20. doi:10.4230/LIPIcs.FSTTCS.2021.51
- [24] Bert Lindenhovius, Michael Mislove, and Vladimir Zamdzhiev. 2018. Enriching a linear/non-linear lambda calculus: A programming language for string diagrams. In *Proceedings of the 33rd Annual ACM/IEEE*

- Symposium on Logic in Computer Science* (Oxford, United Kingdom) (*LICS '18*). Association for Computing Machinery, New York, NY, USA, 659–668. doi:10.1145/3209108.3209196
- [25] Octavio Malherbe. 2013. *Categorical Models of Computation: Partially Traced Categories and Presheaf Models of Quantum Computation*. Ph. D. Dissertation. University of Ottawa, Department of Mathematics and Statistics. Available from [arXiv:1301.5087](https://arxiv.org/abs/1301.5087).
 - [26] MetaOCaml 2020. MetaOCaml – an OCaml dialect for multi-stage programming. <https://okmij.org/ftp/ML/MetaOCaml.html>. Accessed: 2022-10-05.
 - [27] Eugenio Moggi. 1991. Notions of computation and monads. *Information and Computation* 93, 1 (1991), 55–92.
 - [28] Michael A. Nielsen and Isaac L. Chuang. 2002. *Quantum Computation and Quantum Information*. Cambridge University Press.
 - [29] Bernhard Ömer. 1998. *A Procedural Formalism for Quantum Computing*. Master's thesis. Department of Theoretical Physics, Technical University of Vienna. Available at <http://tph.tuwien.ac.at/~oemer/qcl.html>.
 - [30] Adam Paetznick and Krysta M. Svore. 2014. Repeat-Until-Success: Non-deterministic decomposition of single-qubit unitaries. *Quantum Information and Computation* 14, 15–16 (2014), 1277–1301. doi:10.26421/QIC14.15-16-2 Also available from [arXiv:1311.1074](https://arxiv.org/abs/1311.1074).
 - [31] Jennifer Paykin, Robert Rand, and Steve Zdancewic. 2017. QWIRE: a core language for quantum circuits. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (ACM SIGPLAN Notices, Vol. 52)*. ACM, 846–858.
 - [32] Benjamin C. Pierce and David N. Turner. 2000. Local type inference. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 22, 1 (2000), 1–44.
 - [33] Robert Rand, Jennifer Paykin, Dong-Ho Lee, and Steve Zdancewic. 2019. ReQWIRE: Reasoning about reversible quantum circuits. In *Proceedings of the 15th International Conference on Quantum Physics and Logic, QPL 2018, Halifax (Electronic Proceedings in Theoretical Computer Science, Vol. 287)*. 299–312. doi:10.4204/EPTCS.287.17 Available from [arXiv:1901.10118](https://arxiv.org/abs/1901.10118).
 - [34] Robert Rand, Jennifer Paykin, and Steve Zdancewic. 2018. QWIRE practice: Formal verification of quantum circuits in Coq. In *Proceedings of the 14th International Conference on Quantum Physics and Logic, QPL 2017, Nijmegen (Electronic Proceedings in Theoretical Computer Science, Vol. 266)*. 119–132. doi:10.4204/EPTCS.266.11 Available from [arXiv:1803.00699](https://arxiv.org/abs/1803.00699).
 - [35] Mathys Rennela and Sam Staton. 2020. Classical control, quantum circuits and linear logic in enriched category theory. *Logical Methods in Computer Science* 16, 1 (2020), 30:1–24. doi:10.23638/LMCS-16(1:30)2020
 - [36] Francisco Rios. 2021. *On a Categorically Sound Quantum Programming Language for Circuit Description*. Ph. D. Dissertation. Dalhousie University, Department of Mathematics and Statistics. Available from <http://hdl.handle.net/10222/80771>.
 - [37] Francisco Rios and Peter Selinger. 2018. A categorical model for a quantum circuit description language. Extended Abstract. In *Proceedings of the 14th International Conference on Quantum Physics and Logic, QPL 2017, Nijmegen (Electronic Proceedings in Theoretical Computer Science, Vol. 266)*. 164–178. doi:10.4204/EPTCS.266.11 Also available from [arXiv:1706.02630](https://arxiv.org/abs/1706.02630).
 - [38] Neil J. Ross. 2015. *Algebraic and Logical Methods in Quantum Computation*. Ph. D. Dissertation. Dalhousie University, Department of Mathematics and Statistics. Available from [arXiv:1510.02198](https://arxiv.org/abs/1510.02198).
 - [39] Peter Selinger. 2004. Towards a quantum programming language. *Mathematical Structures in Computer Science* 14, 4 (2004), 527–586.
 - [40] Peter Selinger and Benoît Valiron. 2009. Quantum Lambda Calculus. In *Semantic Techniques in Quantum Computation*, Simon Gay and Ian Mackie (Eds.). Cambridge University Press, Chapter 4, 135–172.
 - [41] Walid Taha and Tim Sheard. 2000. MetaML and multi-stage programming with explicit annotations. *Theoretical Computer Science* 248, 1 (2000), 211–242. doi:10.1016/S0304-3975(00)00053-0 PEPM'97.

A Diagrams for strong monoidal functors

Definition A.1. Suppose $\mathbf{A}, \mathbf{B}$ are categories. A functor $F : \mathbf{A} \rightarrow \mathbf{B}$ is *strong monoidal* if the following hold.

- In $\mathbf{B}$, there is an isomorphism $e : I \rightarrow FI$ and a natural isomorphism $m : FA \otimes FB \rightarrow F(A \otimes B)$.
- For all $A, B, C \in \mathbf{A}$, the following diagrams commute.

$$\begin{array}{ccc}
 (FA \otimes FB) \otimes FC & \xrightarrow{\alpha} & FA \otimes (FB \otimes FC) \\
 \downarrow m \otimes FC & & \downarrow FA \otimes m \\
 F(A \otimes B) \otimes FC & & FA \otimes F(B \otimes C) \\
 \downarrow m & & \downarrow m \\
 F((A \otimes B) \otimes C) & \xrightarrow{F\alpha} & F(A \otimes (B \otimes C))
 \end{array}$$

$$\begin{array}{ccc}
 I \otimes FA & \xrightarrow{e \otimes FA} & FI \otimes FA \\
 \downarrow l & & \downarrow m \\
 FA & \xleftarrow{Fl} & F(I \otimes A)
 \end{array}$$

$$\begin{array}{ccc}
 FA \otimes I & \xrightarrow{FA \otimes e} & FA \otimes FI \\
 \downarrow r & & \downarrow m \\
 FA & \xleftarrow{Fr} & F(A \otimes I)
 \end{array}$$

B Convexity

Let $[0, 1]$ denote the real unit interval.

Definition B.1. A *convexity structure* on a set X is an operation that assigns to all $p, q \in [0, 1]$ with $p + q = 1$ and all $x, y \in X$ an element $px + qy \in X$, subject to the following properties. Throughout, we assume $p + q = 1$.

- $px + qx = x$ for all $x \in X$.
- $px + qy = qy + px$ for all $x, y \in X$.
- $0x + 1y = y$ for all $x, y \in X$.
- $(a + b)(\frac{a}{a+b}x + \frac{b}{a+b}y) + (c + d)(\frac{c}{c+d}z + \frac{d}{c+d}w) = (a + c)(\frac{a}{a+c}x + \frac{c}{a+c}z) + (b + d)(\frac{b}{b+d}y + \frac{d}{b+d}w)$, where $a, b, c, d \in [0, 1]$ with $a + b + c + d = 1$ and all denominators are non-zero.

Remark. Property (d) can best be understood by realizing that both sides of the equation are equal to $ax + by + cz + dw$, decomposed in two different ways into convex sums of two elements at a time. In the literature, we sometimes find a different, but equivalent condition of the form $s(px + qy) + rz = spx + (qs + r)(\frac{qs}{qs+r}y + \frac{r}{qs+r}z)$. The latter axiom is arguably shorter, but harder to read.

We often expand the binary $+$ operation to a multi-arity operation, i.e., $\sum_i p_i x_i$, where $\sum_i p_i = 1$ and $x_i \in X$ for all i .

Theorem B.2. Let $\mathbf{A}$ be a symmetric monoidal category with a coproduct $I + I$, such that tensor distributes over this coproduct. The following are equivalent.

- The category $\mathbf{A}$ is enriched in convex spaces.

(b) There exists a family of maps $\langle\langle p, q \rangle\rangle : I \rightarrow I + I$, where $p, q \in [0, 1]$ with $p + q = 1$, such that the following diagrams commute:

$$\begin{array}{ccc}
 I \xrightarrow{\langle\langle p, q \rangle\rangle} I + I \xrightarrow{[\text{id}, \text{id}]} I & I \xrightarrow{\langle\langle p, q \rangle\rangle} I + I & I \xrightarrow{\text{inj}_2} I + I \\
 \text{id} \searrow & \swarrow \langle\langle q, p \rangle\rangle \downarrow [\text{inj}_2, \text{inj}_1] & \langle\langle 0, 1 \rangle\rangle \searrow \\
 & I + I &
 \end{array}$$

$$\begin{array}{ccc}
 & I & \\
 \langle\langle a+b, c+d \rangle\rangle \swarrow & & \searrow \langle\langle a+c, b+d \rangle\rangle \\
 I + I & & I + I \\
 \downarrow \langle\langle \frac{a}{a+b}, \frac{b}{a+b} \rangle\rangle + \langle\langle \frac{c}{c+d}, \frac{d}{c+d} \rangle\rangle & \xrightarrow{\text{iso}} & \downarrow \langle\langle \frac{a}{a+c}, \frac{c}{a+c} \rangle\rangle + \langle\langle \frac{b}{b+d}, \frac{d}{b+d} \rangle\rangle \\
 (I + I) + (I + I) & & (I + I) + (I + I)
 \end{array}$$

Here, in the last diagram, we have $a, b, c, d \in [0, 1]$ with $a + b + c + d = 1$, and we assume the denominators are non-zero. The map “iso” is the canonical isomorphism $(A + B) + (C + D) \cong (A + C) + (B + D)$.

PROOF. For the left-to-right implication, suppose $\mathbf{A}$ is enriched in convex spaces. We can define

$$\langle\langle p, q \rangle\rangle := p \text{inj}_1 + q \text{inj}_2 : I \rightarrow I + I.$$

It is easy to verify that this definition of $\langle\langle p, q \rangle\rangle$ satisfies the four diagrams above.

We now focus on the right-to-left implication.

- First we need to show that $\mathbf{A}(A, B)$ is convex for all $A, B \in \mathbf{A}$. Given $f, g \in \mathbf{A}(A, B)$, we define $pf + qg$ as follows.

$$A \xrightarrow{\lambda^{-1}} A \otimes I \xrightarrow{A \otimes \langle\langle p, q \rangle\rangle} A \otimes (I + I) \xrightarrow{d} A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[f, g]} B.$$

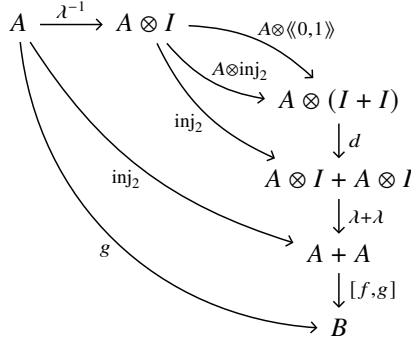
- $pf + qf = f$. This holds because the following diagram commutes.

$$\begin{array}{ccccccc}
 A & \xrightarrow{\lambda^{-1}} & A \otimes I & \xrightarrow{A \otimes \langle\langle p, q \rangle\rangle} & A \otimes (I + I) & \xrightarrow{d} & A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[f, f]} B \\
 f \downarrow & & \downarrow f \otimes I & & \downarrow f \otimes (I + I) & & \downarrow f \otimes I + f \otimes I \xrightarrow{f + f} \downarrow [\text{id}, \text{id}] \\
 B & \xrightarrow{\lambda^{-1}} & B \otimes I & \xrightarrow{B \otimes \langle\langle p, q \rangle\rangle} & B \otimes (I + I) & \xrightarrow{d} & B \otimes I + B \otimes I \xrightarrow{\lambda + \lambda} B + B \\
 & & \searrow \text{id} & & \downarrow B \otimes [\text{id}, \text{id}] & \nearrow [\text{id}, \text{id}] & \nearrow \lambda
 \end{array}$$

- $pf + qg = qg + pf$. This holds because the following diagram commutes.

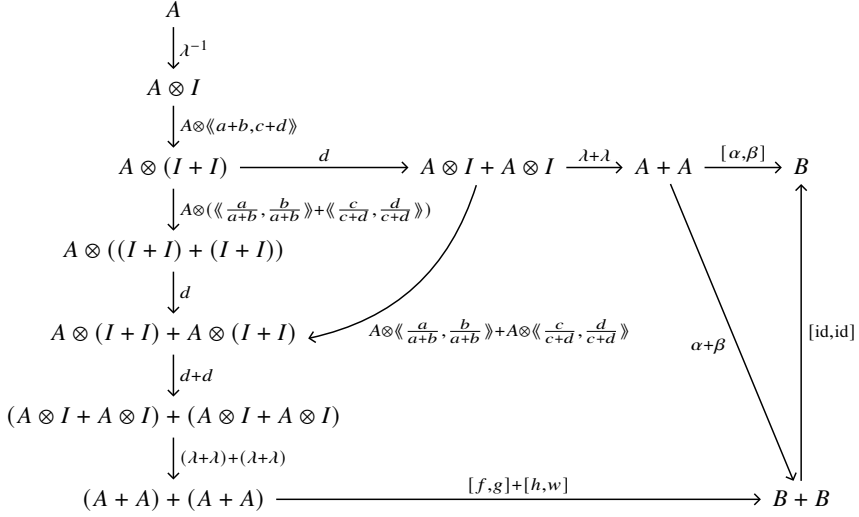
$$\begin{array}{ccccccc}
 A & \xrightarrow{\lambda^{-1}} & A \otimes I & \xrightarrow{A \otimes \langle\langle p, q \rangle\rangle} & A \otimes (I + I) & \xrightarrow{d} & A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[f, g]} B \\
 & & \searrow A \otimes \langle\langle q, p \rangle\rangle & & \downarrow A \otimes [\text{inj}_2, \text{inj}_1] & & \downarrow [\text{inj}_2, \text{inj}_1] \downarrow [\text{id}, \text{id}] \\
 & & & & A \otimes (I + I) & \xrightarrow{d} & A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[g, f]} B
 \end{array}$$

- $0f + 1g = g$. We have the following commutative diagram.



- $(a + b)(\frac{a}{a+b}f + \frac{b}{a+b}g) + (c + d)(\frac{c}{c+d}h + \frac{d}{c+d}w) = (a + c)(\frac{a}{a+c}f + \frac{c}{a+c}h) + (b + d)(\frac{b}{b+d}g + \frac{d}{b+d}w)$.

Let us write $\alpha = \frac{a}{a+b}f + \frac{b}{a+b}g$ and $\beta = \frac{c}{c+d}h + \frac{d}{c+d}w$. We have the following commutative diagram.



Thus

$$\begin{aligned}
& (a + b)(\frac{a}{a+b}f + \frac{b}{a+b}g) + (c + d)(\frac{c}{c+d}h + \frac{d}{c+d}w) \\
&= [\text{id}, \text{id}] \circ ([f, g] + [h, w]) \circ ((\lambda + \lambda) + (\lambda + \lambda)) \circ (d + d) \circ d \\
&\quad \circ (A \otimes (\langle \frac{a}{a+b}, \frac{b}{a+b} \rangle + \langle \frac{c}{c+d}, \frac{d}{c+d} \rangle)) \circ (A \otimes \langle a + b, c + d \rangle) \circ \lambda^{-1}.
\end{aligned}$$

Similarly, we can show that

$$\begin{aligned}
& (a + c)(\frac{a}{a+c}f + \frac{c}{a+c}h) + (b + d)(\frac{b}{b+d}g + \frac{d}{b+d}w) \\
&= [\text{id}, \text{id}] \circ ([f, h] + [g, w]) \circ ((\lambda + \lambda) + (\lambda + \lambda)) \circ (d + d) \circ d \\
&\quad \circ (A \otimes (\langle \frac{a}{a+c}, \frac{c}{a+c} \rangle + \langle \frac{b}{b+d}, \frac{d}{b+d} \rangle)) \circ (A \otimes \langle a + c, b + d \rangle) \circ \lambda^{-1}.
\end{aligned}$$

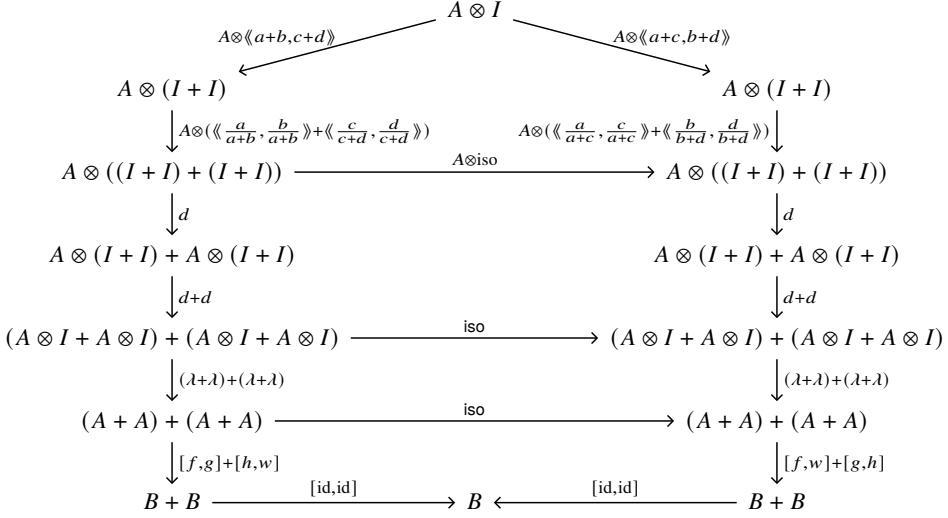


Fig. 6. A diagram used in the proof of Theorem B.2

Thus we can show

$$\begin{aligned} & (a+b)\left(\frac{a}{a+b}f + \frac{b}{a+b}g\right) + (c+d)\left(\frac{c}{c+d}h + \frac{d}{c+d}w\right) \\ &= (a+c)\left(\frac{a}{a+c}f + \frac{c}{a+c}h\right) + (b+d)\left(\frac{b}{b+d}g + \frac{d}{b+d}w\right) \end{aligned}$$

by the commutative diagram in Figure 6.

- $(pf + qg) \circ e = p(f \circ e) + q(g \circ e)$. This is by the following commutative diagram.

$$\begin{array}{ccccccc} C & \xrightarrow{e} & A & \xrightarrow{\lambda^{-1}} & A \otimes I & \xrightarrow{A \otimes \langle \langle p, q \rangle \rangle} & A \otimes (I + I) & \xrightarrow{d} & A \otimes I + A \otimes I & \xrightarrow{\lambda + \lambda} & A + A & \xrightarrow{[f, g]} & B \\ & \searrow \lambda^{-1} & & \uparrow e \otimes I & & \uparrow e \otimes (I + I) & & \uparrow e \otimes I + e \otimes I & & \uparrow e + e & & \nearrow [f \circ e, g \circ e] \\ & & C \otimes I & \xrightarrow{C \otimes \langle \langle p, q \rangle \rangle} & C \otimes (I + I) & \xrightarrow{d} & C \otimes I + C \otimes I & \xrightarrow{\lambda + \lambda} & C + C & & & & \end{array}$$

- $h \circ (pf + qg) = p(h \circ f) + q(h \circ g)$. This is by the following.

$$A \xrightarrow{\lambda^{-1}} A \otimes I \xrightarrow{A \otimes \langle \langle p, q \rangle \rangle} A \otimes (I + I) \xrightarrow{d} A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[f, g]} B \xrightarrow{h} C \quad \square$$

$[h \circ f, h \circ g]$

THEOREM 6.26. *The category $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ is enriched in convex spaces. Moreover, the Lambek embedding $\kappa : \mathbf{Q} \hookrightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ preserves the convex sum in $\mathbf{Q}$.*

PROOF. By Theorem B.2 (b), there exists a map $\langle \langle p, q \rangle \rangle : I \rightarrow I + I$ in $\mathbf{Q}$ for any $p, q \in [0, 1]$, $p + q = 1$, and it satisfies the four diagrams. Since κ preserves coproducts in $\mathbf{Q}$, the map $\kappa \langle \langle p, q \rangle \rangle : \kappa I \rightarrow \kappa I +' \kappa I$ in $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ also satisfies the four diagrams in Theorem B.2 (b). Therefore $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ is enriched in convex spaces.

For all $f, g \in \mathbf{Q}(A, B)$, the convex sum $pf + qg \in \mathbf{Q}(A, B)$ is defined to be the following.

$$A \xrightarrow{\lambda^{-1}} A \otimes I \xrightarrow{A \otimes \langle \langle p, q \rangle \rangle} A \otimes (I + I) \xrightarrow{d} A \otimes I + A \otimes I \xrightarrow{\lambda + \lambda} A + A \xrightarrow{[f, g]} B$$

Since κ preserves coproducts in $\mathbf{Q}$ and it is strong monoidal, we have $\kappa(pf + qg) = p\kappa(f) + q\kappa(g) \in [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(\kappa A, \kappa B)$.

□

C Proof of Theorem 4.7

THEOREM 4.7 (SEMANTIC SUBSTITUTION). *Suppose $\Phi, \Gamma_1, x : A, \Gamma_2 \vdash_{\alpha} M : B$ and $\Phi, \Gamma_3 \vdash_1 V : A$. We have the following equality.*

$$\begin{aligned}
 \llbracket [V/x]M \rrbracket &= \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\text{dup} \otimes \text{id}} \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\text{id} \otimes \gamma \otimes \text{id}} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\text{id} \otimes \llbracket V \rrbracket \otimes \text{id}} \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket A \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \\
 &\xrightarrow{\llbracket M \rrbracket} \alpha \llbracket B \rrbracket.
 \end{aligned}$$

PROOF. The proof is by induction on $\Phi, \Gamma_1, x : A, \Gamma_2 \vdash_{\alpha} M : B$. The only non-trivial cases are that of the *dynlift* rule, which is new in this paper compared to previous work, and the *lift* rule, which has some subtleties because the context must be a parameter context. All of the remaining cases are routine, and we show a few of them to illustrate the general pattern.

- Case

$$\frac{\Phi, \Gamma_1, x : A, \Gamma'_1 \vdash_{\alpha} M : \mathbf{Bit}}{\Phi, \Gamma_1, x : A, \Gamma'_1 \vdash_0 \text{dynlift } M : \mathbf{Bool}}.$$

Suppose $\Phi, \Gamma_2 \vdash_1 V : A$ and $\alpha = 0$. By the induction hypothesis, we have

$$\begin{aligned}
 \llbracket [V/x]M \rrbracket &= \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\
 &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\
 &\quad : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \rightarrow T \llbracket \mathbf{Bit} \rrbracket.
 \end{aligned}$$

Since $\Phi, \Gamma_1, \Gamma_2, \Gamma'_1 \vdash_0 \text{dynlift}[V/x]M : \mathbf{Bool}$, we have

$$\begin{aligned}
 \llbracket \text{dynlift}[V/x]M \rrbracket &= \mu \circ T \text{dynlift} \circ \llbracket [V/x]M \rrbracket \\
 &= (\mu \circ T \text{dynlift} \circ \llbracket M \rrbracket) \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\
 &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\
 &\quad : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \rightarrow T \llbracket \mathbf{Bool} \rrbracket.
 \end{aligned}$$

- Case

$$\frac{\Phi, x : P, \Phi' \vdash_{\alpha} M : A}{\Phi, x : P, \Phi' \vdash_1 \text{lift } M : !_A A}.$$

By Theorem 4.3, we have $\Phi \vdash_1 V : P$. Since $\Phi, \Phi' \vdash_{\alpha} [V/x]M : A$, by induction hypothesis, we have

$$\llbracket [V/x]M \rrbracket = \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Phi' \rrbracket) \circ (\text{dup} \otimes \llbracket \Phi' \rrbracket) : \llbracket \Phi \rrbracket \otimes \llbracket \Phi' \rrbracket \rightarrow \alpha \llbracket A \rrbracket.$$

Since $\Phi, \Phi' \vdash_1 \text{lift}[V/x]M : !_A A$, we have

$$\llbracket \text{lift}[V/x]M \rrbracket = p\delta \llbracket [V/x]M \rrbracket : \llbracket \Phi \rrbracket \otimes \llbracket \Phi' \rrbracket \rightarrow pb\alpha \llbracket A \rrbracket.$$

Note that $\delta \llbracket [V/x]M \rrbracket$ is the unique morphism such that

$$\text{force} \circ p\delta[[V/x]M] = [[V/x]M].$$

On the other hand, we have

$$\begin{aligned} & \text{force} \circ p\delta[M] \circ ([\Phi] \otimes [V] \otimes [\Phi']) \circ (\text{dup} \otimes [\Phi']) \\ &= [M] \circ ([\Phi] \otimes [V] \otimes [\Phi']) \circ (\text{dup} \otimes [\Phi']) \\ &= [[V/x]M]. \end{aligned}$$

Thus

$$\begin{aligned} [[\text{lift}[V/x]M]] &= p\delta[[V/x]M] \\ &= p\delta[M] \circ ([\Phi] \otimes [V] \otimes [\Phi']) \circ (\text{dup} \otimes [\Phi']) \\ &= [[\text{lift} M]] \circ ([\Phi] \otimes [V] \otimes [\Phi']) \circ (\text{dup} \otimes [\Phi']). \end{aligned}$$

• Case

$$\frac{\Phi, \Gamma_1, x : C, \Gamma'_1, y : A \vdash_\alpha M : B}{\Phi, \Gamma_1, x : C, \Gamma'_1 \vdash_1 \lambda y. M : A \multimap_\alpha B.}$$

Suppose $\Phi, \Gamma_2 \vdash_1 V : C$. By the induction hypothesis, we have

$$\begin{aligned} [[V/x]M] &= [M] \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1, y : A]) \\ &\quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1, y : A]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1, y : A]) \\ &\quad : [\Phi] \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1, y : A] \rightarrow \alpha[B]. \end{aligned}$$

Since $\Phi, \Gamma_1, \Gamma_2, \Gamma'_1 \vdash_1 \lambda y. [V/x]M : A \multimap_\alpha B$, we have

$$[\lambda y. [V/x]M] = \text{curry}[[V/x]M] : [\Phi] \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1] \rightarrow [A] \multimap \alpha[B],$$

which is the unique morphism such that

$$\epsilon \circ (\text{curry}[[V/x]M] \otimes [A]) = [[V/x]M].$$

On the other hand, we have

$$\begin{aligned} & \epsilon \circ ((\text{curry}[M] \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1]) \\ & \quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1])) \otimes [A]) \\ &= \epsilon \circ (\text{curry}[M] \otimes [A]) \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1, y : A]) \\ & \quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \\ &= [M] \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1, y : A]) \\ & \quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \\ &= [[V/x]M]. \end{aligned}$$

Thus

$$\begin{aligned} & [[\lambda y. [V/x]M]] \\ &= \text{curry}[M] \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1, y : A]) \\ & \quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \\ &= [\lambda y. M] \circ ([\Phi] \otimes [\Gamma_1] \otimes [V] \otimes [\Gamma'_1, y : A]) \\ & \quad \circ ([\Phi] \otimes \gamma \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]) \circ (\text{dup} \otimes [\Gamma_1] \otimes [\Gamma_2] \otimes [\Gamma'_1] \otimes [A]). \end{aligned}$$

- Case

$$\frac{\Phi, \Gamma_1, x : C, \Gamma'_1 \vdash_{\alpha_1} M : A \multimap_{\beta} B \quad \Phi, \Gamma_2 \vdash_{\alpha_2} N : A}{\Phi, \Gamma_1, x : C, \Gamma'_1, \Gamma_2 \vdash_{\alpha_1 \& \alpha_2 \& \beta} MN : B}$$

Suppose $\Phi, \Gamma_3 \vdash_1 V : C$ and $\alpha_1 = \alpha_2 = \beta = 0$. By the induction hypothesis, we have

$$\begin{aligned} \llbracket [V/x]M \rrbracket &= \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\ &\quad : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \rightarrow T(\llbracket A \rrbracket \multimap T\llbracket B \rrbracket). \end{aligned}$$

Moreover,

$$\begin{aligned} \llbracket [V/x]MN \rrbracket &= \mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (\llbracket [V/x]M \rrbracket \otimes \llbracket N \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ &= \mu \circ T\epsilon \circ \mu \circ Ts \circ t \\ &\quad \circ ((\llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket)) \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\ &\quad \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket)) \otimes \llbracket N \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket). \end{aligned}$$

On the other hand,

$$\begin{aligned} \llbracket MN \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ = \mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket) \\ \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket C \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket). \end{aligned}$$

Thus by reasoning about the box diagram, we have

$$\begin{aligned} \llbracket [V/x]MN \rrbracket &= \llbracket MN \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket) \\ &\quad \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_3 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket). \end{aligned}$$

- Case

$$\frac{\Phi, \Gamma_1, x : C, \Gamma'_1 \vdash_{\beta} M : !_A A}{\Phi, \Gamma_1, x : C, \Gamma'_1 \vdash_{\alpha \& \beta} \text{force } M : A.}$$

Suppose $\Phi, \Gamma_2 \vdash_1 V : C$ and $\alpha = \beta = 0$. By the induction hypothesis, we have

$$\begin{aligned} \llbracket [V/x]M \rrbracket &= \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket) \\ &\quad : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma'_1 \rrbracket \rightarrow TpbT\llbracket A \rrbracket. \end{aligned}$$

Since $\Gamma_1, \Gamma_2, \Gamma_1' \vdash_0 \text{force}[V/x]M : A$, we have

$$\begin{aligned} \llbracket \text{force}[V/x]M \rrbracket &= \mu \circ T \text{force} \circ \llbracket [V/x]M \rrbracket \\ &= \mu \circ T \text{force} \circ \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \\ &\quad : \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket \rightarrow T[A]. \end{aligned}$$

On the other hand, we have

$$\begin{aligned} &\llbracket \text{force} M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \\ &= \mu \circ T \text{force} \circ \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \\ &\quad \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket). \end{aligned}$$

So $\llbracket \text{force}[V/x]M \rrbracket = \llbracket \text{force} M \rrbracket \circ (\llbracket \Gamma_1 \rrbracket \otimes \llbracket V \rrbracket \otimes \llbracket \Phi \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \circ (\llbracket \Phi \rrbracket \otimes \gamma \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket) \circ (\text{dup} \otimes \llbracket \Gamma_1 \rrbracket \otimes \llbracket \Gamma_2 \rrbracket \otimes \llbracket \Gamma_1' \rrbracket)$.

• Case

$$\frac{\Phi, x : P \vdash_0 M : A \quad \Phi, x : P \vdash_0 N : B}{\Phi, x : P \vdash_0 (M, N) : A \otimes B}$$

Suppose $\Phi \vdash_1 V : P$. By the induction hypothesis, we have

$$\llbracket [V/x]M \rrbracket = \llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup} : \llbracket \Phi \rrbracket \rightarrow T[A]$$

and

$$\llbracket [V/x]N \rrbracket = \llbracket N \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup} : \llbracket \Phi \rrbracket \rightarrow T[B].$$

Thus

$$\begin{aligned} &\llbracket ([V/x]M, [V/x]N) \rrbracket \\ &= \mu \circ T\sigma \circ t \circ (\llbracket [V/x]M \rrbracket \otimes \llbracket [V/x]N \rrbracket) \circ \text{dup} \\ &= \mu \circ T\sigma \circ t \circ ((\llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup}) \otimes (\llbracket N \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup})) \circ \text{dup}. \end{aligned}$$

On the other hand,

$$\begin{aligned} &\llbracket (M, N) \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup} \\ &= \mu \circ T\sigma \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket) \circ \text{dup}_{\llbracket \Phi \rrbracket \otimes \llbracket P \rrbracket} \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup}_{\llbracket \Phi \rrbracket} \\ &= \mu \circ T\sigma \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket) \circ ((\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \otimes (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket)) \\ &\quad \circ \text{dup}_{\llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket} \circ \text{dup}_{\llbracket \Phi \rrbracket} \\ &= \mu \circ T\sigma \circ t \circ ((\llbracket M \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup}_{\llbracket \Phi \rrbracket}) \otimes (\llbracket N \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup}_{\llbracket \Phi \rrbracket})) \\ &\quad \circ \text{dup}_{\llbracket \Phi \rrbracket}. \end{aligned}$$

Thus $\llbracket ([V/x]M, [V/x]N) \rrbracket = \llbracket (M, N) \rrbracket \circ (\llbracket \Phi \rrbracket \otimes \llbracket V \rrbracket) \circ \text{dup}$.

□

D Proof of Theorem 5.15

THEOREM 5.15. *If $\vdash_\alpha (Q, M) : A; \Sigma'$, and $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$, then*

$$\llbracket Q, M \rrbracket = \sum_{i \in [n]} p_i \llbracket Q_i, V_i \rrbracket : I \rightarrow T(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket).$$

PROOF. In the case where $\alpha = 1$, there is no dynamic lifting when evaluating M , therefore the proof is mostly similar to the proof of Theorem 5.6. We consider just one typical case:

$$\frac{\begin{array}{c} (Q, M) \Downarrow (Q_1, \lambda x.M'_1) \\ (Q_1, N) \Downarrow (Q'_1, V_1) \\ (Q'_1, [V_1/x]M'_1) \Downarrow (Q''_1, V'_1) \end{array}}{(Q, MN) \Downarrow (Q''_1, V'_1)}$$

Suppose $\text{addr}(Q) = \Sigma'', \Sigma'$ and $\Sigma'' \vdash_1 MN : A$. By the typing rule for MN , we have

$$\frac{\Sigma''_1 \vdash_1 M : B \multimap_1 A \quad \Sigma''_2 \vdash_1 N : B}{\Sigma'' \vdash_1 MN : A},$$

where $\Sigma'' = \Sigma''_1, \Sigma''_2$. By Theorem 5.12, we have $\text{addr}(Q_1) = \Sigma''_1, \Sigma''_2, \Sigma'$, and $\Sigma''_1 \vdash_1 \lambda x.M'_1 : B \multimap_1 A$, and $\text{addr}(Q'_1) = \Sigma''_1, \Sigma''_2, \Sigma'$, and $\Sigma''_2 \vdash_1 V_1 : B$, and $\text{addr}(Q''_1) = \Sigma''''_1, \Sigma'$, and $\Sigma''''_1 \vdash_1 V'_1 : B$.

By the induction hypothesis, we have

$$\begin{aligned} \llbracket Q, M \rrbracket &= T(\llbracket M \rrbracket \otimes \llbracket \Sigma''_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= T(\llbracket \lambda x.M'_1 \rrbracket \otimes \llbracket \Sigma''_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_1 : I \rightarrow T(\llbracket B \multimap A \rrbracket \otimes \llbracket \Sigma''_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \end{aligned}$$

and (where δ is an isomorphism that permutes $\text{addr}(Q_1)$ and $\text{addr}(Q'_1)$)

$$\begin{aligned} \llbracket Q_1, N \rrbracket &= T(\llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket \otimes \llbracket \Sigma''_1 \rrbracket) \circ \phi(\delta \circ Q_1) \\ &= T(\llbracket \Sigma''_1 \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_1 \\ &= T(\llbracket V_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket \otimes \llbracket \Sigma''_1 \rrbracket) \circ \phi(\delta \circ Q'_1) \\ &= T(\llbracket \Sigma''_1 \rrbracket \otimes \llbracket V_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 : I \rightarrow T(\llbracket B \rrbracket \otimes \llbracket \Sigma' \rrbracket \otimes \llbracket \Sigma''_1 \rrbracket) \end{aligned}$$

and

$$\begin{aligned} \llbracket Q'_1, [V_1/x]M'_1 \rrbracket &= T(\llbracket [V_1/x]M'_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 \\ &= T((\llbracket M'_1 \rrbracket \circ (\llbracket \Sigma''_1 \rrbracket \otimes \llbracket V_1 \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 \\ &= T(\llbracket V'_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_1 : I \rightarrow T(\llbracket A \rrbracket \otimes \llbracket \Sigma' \rrbracket). \end{aligned}$$

We need to show that

$$\begin{aligned} \llbracket Q, MN \rrbracket &= T(\llbracket MN \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= T((\epsilon \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= T(\llbracket V'_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_1 \\ &= T((\llbracket M'_1 \rrbracket \circ (\llbracket V_1 \rrbracket \otimes \llbracket \Sigma''_1 \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1. \end{aligned}$$

This is the case because

$$\begin{aligned}
\llbracket Q, MN \rrbracket &= T(\llbracket MN \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= T((\epsilon \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket B \multimap_1 A \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma'_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket B \multimap_1 A \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \lambda x. M'_1 \rrbracket \otimes \llbracket \Sigma'_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_1 \\
&= T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \lambda x. M'_1 \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_1 \\
&= T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \lambda x. M'_1 \rrbracket \otimes \llbracket B \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \Sigma'_1 \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_1 \\
&= T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \lambda x. M'_1 \rrbracket \otimes \llbracket B \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \Sigma'_1 \rrbracket \otimes \llbracket V_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 \\
&= T(\llbracket M'_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \Sigma'_1 \rrbracket \otimes \llbracket V_1 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 \\
&= T((\llbracket M'_1 \rrbracket \circ (\llbracket \Sigma'_1 \rrbracket \otimes \llbracket V_1 \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1.
\end{aligned}$$

Now we consider the case where $\alpha = 0$. Suppose $\text{addr}(Q) = \Sigma'', \Sigma'$ and $\Sigma'' \vdash_0 M : A$. We proceed by induction on $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, V_i)$.

- Case

$$\frac{(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \ell_i) \quad \text{read}(Q_i, \ell_i) = \sum_{j \in [2]} q_{i,j}(Q'_{i,j}, a_{i,j})}{(Q, \text{dynlift } M) \Downarrow \sum_{(i,j) \in [n] \times [2]} p_i q_{i,j}(Q'_{i,j}, a_{i,j})}.$$

- Suppose we have

$$\frac{\Sigma'' \vdash_1 M : \mathbf{Bit}}{\Sigma'' \vdash_0 \text{dynlift } M : \mathbf{Bool}}.$$

In this case we have the following derivation.

$$\frac{(Q, M) \Downarrow (Q', \ell) \quad \text{read}(Q', \ell) = q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True})}{(Q, \text{dynlift } M) \Downarrow q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True})}$$

Since $\text{read}(Q', \ell) = q_1(Q'_1, \text{False}) + q_2(Q'_2, \text{True})$, we have that $Q' = q_1(\text{inj}_1 \otimes Q'_1) + q_2(\text{inj}_2 \otimes Q'_2)$ in $\mathbf{Q}$. So in $\mathbf{A}$, we have

$$\phi Q' = q_1(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) + q_2(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2)),$$

where $\phi Q' : I \rightarrow T(\mathbf{Bit} \otimes \llbracket \Sigma' \rrbracket)$, and $\phi Q'_1, \phi Q'_2 : I \rightarrow T\llbracket \Sigma' \rrbracket$, and $\phi(\text{inj}_1), \phi(\text{inj}_2) : I \rightarrow T\mathbf{Bit}$. Note that by condition (f) (Definition 3.5), we have $\phi(\text{inj}_1) = \eta \circ \text{init} \circ \llbracket \text{False} \rrbracket$ and $\phi(\text{inj}_2) = \eta \circ \text{init} \circ \llbracket \text{True} \rrbracket$. We need to show that

$$\llbracket Q, \text{dynlift } M \rrbracket = q_1(T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1) + q_2(T(\llbracket \text{True} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_2).$$

By the induction hypothesis, we have $\llbracket Q, M \rrbracket = \llbracket Q', \ell \rrbracket$, i.e., $T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q = \phi Q'$. Thus

$$\begin{aligned}
\llbracket Q, \text{dynlift } M \rrbracket &= \mu \circ Ts \circ T((\text{dynlift} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q' \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ (q_1(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) \\
&\quad \quad + q_2(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2))) \\
&= q_1(\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1)) \\
&\quad + q_2(\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_2)).
\end{aligned}$$

We just need to show

$$\begin{aligned}
T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_1 &= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_1) \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ ((\eta \circ \text{init} \circ \llbracket \text{False} \rrbracket) \otimes \phi Q'_1).
\end{aligned}$$

This is true because of the commutative diagram in Figure 5.

– Suppose we have

$$\frac{\Sigma'' \vdash_0 M : \mathbf{Bit}}{\Sigma'' \vdash_0 \text{dynlift } M : \mathbf{Bool}}.$$

In this case $(Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \ell_i)$ and $\text{read}(Q_i, \ell_i) = q_{i,1}(Q'_{i,1}, \text{False}) + q_{i,2}(Q'_{i,2}, \text{True})$, where $\text{addr}(Q_i) = \Sigma'$, $\ell_i : \mathbf{Bit}$ and $\text{addr}(Q'_{i,1}) = \text{addr}(Q'_{i,2}) = \Sigma'$. Thus we have the following in $\mathbf{A}$: $\phi Q_i = q_{i,1}(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_{i,1})) + q_{i,2}(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_{i,2}))$. Note that $\phi(\text{inj}_1) = \eta \circ \text{init} \circ \llbracket \text{False} \rrbracket$ and $\phi(\text{inj}_2) = \eta \circ \text{init} \circ \llbracket \text{True} \rrbracket$.

We need to show that

$$\begin{aligned}
\llbracket Q, \text{dynlift } M \rrbracket &= \sum_{i \in [n]} p_i q_{i,1}(T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \otimes \phi Q'_{i,1}) \\
&\quad + p_i q_{i,2}(T(\llbracket \text{True} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \otimes \phi Q'_{i,2}).
\end{aligned}$$

By the induction hypothesis, we have

$$\begin{aligned}
\llbracket Q, M \rrbracket &= \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \sum_{i \in [n]} p_i(T(\llbracket \ell_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \otimes \phi Q_i) \\
&= \sum_{i \in [n]} p_i \phi Q_i.
\end{aligned}$$

Thus

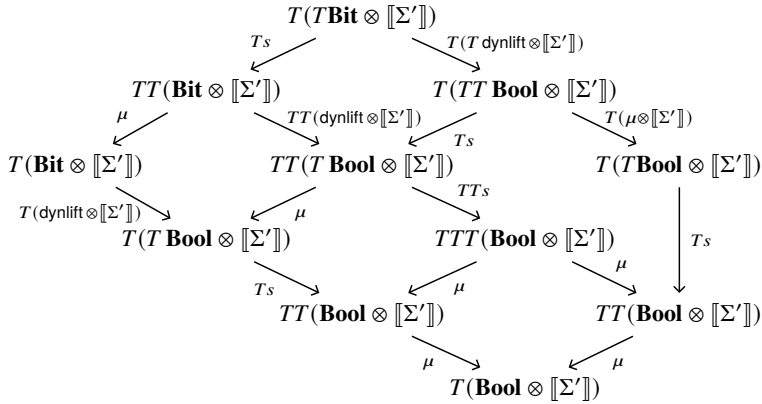
$$\begin{aligned}
& \llbracket Q, \text{dynlift } M \rrbracket \\
&= \mu \circ Ts \circ T((\mu \circ T \text{dynlift} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T((\mu \circ T \text{dynlift}) \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&\stackrel{(*)}{=} \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \left(\sum_{i \in [n]} p_i \phi Q_i \right) \\
&= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ (q_{i,1}(\mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_{i,1})) \\
&\quad + q_{i,2}(\mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_{i,2})))) \\
&= \sum_{i \in [n]} p_i q_{i,1} (\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_{i,1})) \\
&\quad + p_i q_{i,2} (\mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_2) \otimes \phi Q'_{i,2})).
\end{aligned}$$

So we just need to show

$$\begin{aligned}
& T(\llbracket \text{False} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \otimes \phi Q'_{i,1} \\
&= \mu \circ Ts \circ T(\text{dynlift} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Tt \circ s \circ (\phi(\text{inj}_1) \otimes \phi Q'_{i,1}).
\end{aligned}$$

We already showed this in the previous case.

The equality of (*) is by the following commutative diagram.



• Case

$$\frac{
\begin{aligned}
& (Q, M) \Downarrow \sum_{i \in [n]} p_i (Q_i, \lambda x. M'_i) \\
& (Q_i, N) \Downarrow \sum_{j \in [m]} q_{i,j} (Q'_{i,j}, V_{i,j}) \\
& (Q'_{i,j}, [V_{i,j}/x] M'_i) \Downarrow \sum_{k \in [l]} s_{i,j,k} (Q''_{i,j,k}, V'_{i,j,k})
\end{aligned}
}{
(Q, MN) \Downarrow \sum_{(i,j,k) \in [n] \times [m] \times [l]} p_i q_{i,j} s_{i,j,k} (Q''_{i,j,k}, V'_{i,j,k}).
}$$

Here we only consider the following typing judgment (the other cases are proved similarly).

$$\frac{\Sigma_1 \vdash_0 M : A \multimap_0 B \quad \Sigma_2 \vdash_0 N : A}{\Sigma_1, \Sigma_2 \vdash_0 MN : B}$$

We assume that $\text{addr}(Q) = \Sigma_1, \Sigma_2, \Sigma'$.

By the induction hypothesis, we have

$$\begin{aligned} \llbracket Q, M \rrbracket &= \mu \circ Ts \circ T(s \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \sum_{i \in [n]} p_i \llbracket Q_i, \lambda x. M'_i \rrbracket \\ &= \sum_{i \in [n]} p_i (T(\llbracket \lambda x. M'_i \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i), \end{aligned}$$

where $\phi Q_i : I \rightarrow T(\llbracket \Sigma'_i \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\Sigma'_i \vdash_1 \lambda x. M'_i : A \multimap_0 B$. The induction hypothesis also gives

$$\begin{aligned} \llbracket Q_i, N \rrbracket &= \mu \circ Ts \circ T(t \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \Sigma'_i \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i \\ &= \sum_{j \in [m]} q_{i,j} \llbracket Q'_{i,j}, V_{i,j} \rrbracket \\ &= \sum_{j \in [m]} q_{i,j} (T(\llbracket \Sigma'_i \rrbracket \otimes \llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}), \end{aligned}$$

where $Q'_{i,j} : I \rightarrow T(\llbracket \Sigma'_i \rrbracket \otimes \llbracket \Sigma''_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\Sigma''_{i,j} \vdash_1 V_{i,j} : A$. Moreover,

$$\begin{aligned} \llbracket Q'_{i,j}, [V_{i,j}/x] M'_i \rrbracket &= \mu \circ Ts \circ T(\llbracket [V_{i,j}/x] M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j} \\ &= \mu \circ Ts \circ T((\llbracket M'_i \rrbracket \circ (\llbracket \Sigma'_i \rrbracket \otimes \llbracket V_{i,j} \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j} \\ &= \sum_{k \in [l]} s_{i,j,k} \llbracket Q''_{i,j,k}, V'_{i,j,k} \rrbracket \\ &= \sum_{k \in [l]} s_{i,j,k} (T(\llbracket V'_{i,j,k} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_{i,j,k}), \end{aligned}$$

where $\phi Q''_{i,j,k} : I \rightarrow T(\llbracket \Sigma'''_{i,j,k} \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\Sigma'''_{i,j,k} \vdash_1 V'_{i,j,k} : B$.

We need to show

$$\begin{aligned} \llbracket Q, MN \rrbracket &= \mu \circ Ts \circ T(\llbracket MN \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \mu \circ Ts \circ T((\mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \sum_{i,j,k \in [n] \times [m] \times [l]} p_i q_{i,j} s_{i,j,k} (T(\llbracket V'_{i,j,k} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_{i,j,k}). \end{aligned}$$

On the right hand side, we have

$$\begin{aligned} RHS &= \sum_{i,j,k \in [n] \times [m] \times [l]} p_i q_{i,j} s_{i,j,k} (T(\llbracket V'_{i,j,k} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_{i,j,k}) \\ &= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} \sum_{k \in [l]} s_{i,j,k} (T(\llbracket V'_{i,j,k} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_{i,j,k}) \\ &\stackrel{IH}{=} \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (\mu \circ Ts \circ T((\llbracket M'_i \rrbracket \circ (\llbracket \Sigma'_i \rrbracket \otimes \llbracket V_{i,j} \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}). \end{aligned}$$

On the left hand side, we have

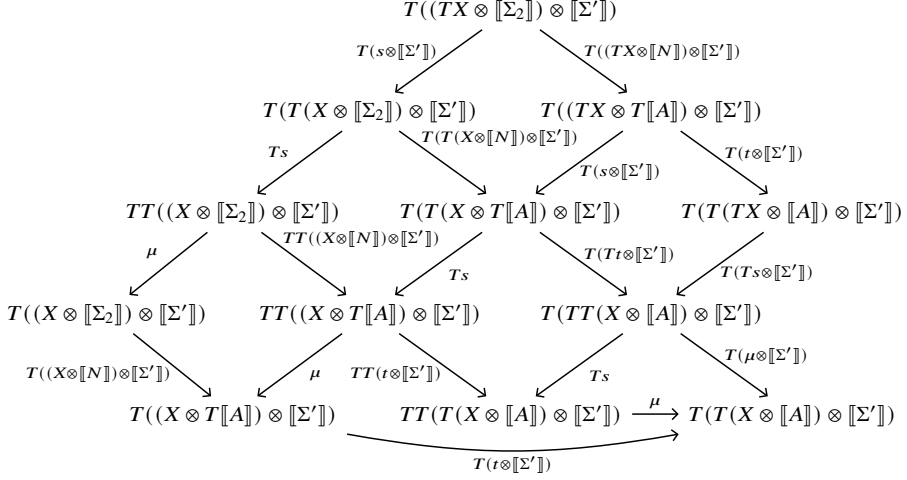
LHS

$$\begin{aligned}
&= \mu \circ Ts \circ T((\mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ ([M] \otimes [N])) \otimes [\Sigma']) \circ \phi Q \\
&= \mu \circ Ts \circ T((\mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (T([A] \multimap T[B]) \otimes [N]) \circ ([M] \otimes [\Sigma_2])) \otimes [\Sigma']) \circ \phi Q \\
&= \mu \circ Ts \circ T(\mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (T([A] \multimap T[B]) \otimes [N]) \otimes [\Sigma']) \circ T([M] \otimes [\Sigma_2] \otimes [\Sigma']) \circ \phi Q \\
&\stackrel{(1)}{=} \mu \circ Ts \circ T(\mu \otimes [\Sigma']) \circ T(T\epsilon \otimes [\Sigma']) \circ T(t \otimes [\Sigma']) \circ T([A] \multimap T[B] \otimes [N] \otimes [\Sigma']) \\
&\quad \circ \mu \circ Ts \circ T(s \otimes [\Sigma']) \circ T([M] \otimes [\Sigma_2] \otimes [\Sigma']) \circ \phi Q \\
&\stackrel{IH}{=} \mu \circ Ts \circ T(\mu \otimes [\Sigma']) \circ T(T\epsilon \otimes [\Sigma']) \circ T(t \otimes [\Sigma']) \circ T([A] \multimap T[B] \otimes [N] \otimes [\Sigma']) \circ \\
&\quad \sum_{i \in [n]} p_i(T([\lambda x.M'_i] \otimes [\Sigma_2] \otimes [\Sigma']) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T(\mu \otimes [\Sigma']) \circ T(T\epsilon \otimes [\Sigma']) \circ T(t \otimes [\Sigma']) \circ T([A] \multimap T[B] \otimes [N] \otimes [\Sigma']) \\
&\quad \circ T([\lambda x.M'_i] \otimes [\Sigma_2] \otimes [\Sigma']) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T((\mu \circ T\epsilon \circ t \circ ([A] \multimap T[B]) \otimes [N]) \circ ([\lambda x.M'_i] \otimes [\Sigma_2])) \otimes [\Sigma']) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T((\mu \circ T\epsilon \circ t \circ ([\lambda x.M'_i] \otimes [N])) \otimes [\Sigma']) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T((\mu \circ T\epsilon \circ t \circ ([\lambda x.M'_i] \otimes T[A]) \circ ([\Sigma'_i] \otimes [N])) \otimes [\Sigma']) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T((\mu \circ T\epsilon \circ t \circ ([\lambda x.M'_i] \otimes T[A])) \otimes [\Sigma']) \circ T([\Sigma'_i] \otimes [N] \otimes [\Sigma']) \circ \phi Q_i) \\
&\stackrel{(2)}{=} \sum_{i \in [n]} p_i(\mu \circ Ts \circ T(\epsilon \otimes [\Sigma']) \circ T([\lambda x.M'_i] \otimes [A] \otimes [\Sigma']) \circ \mu \circ Ts \circ T(t \otimes [\Sigma']) \\
&\quad \circ T([\Sigma'_i] \otimes [N] \otimes [\Sigma']) \circ \phi Q_i) \\
&\stackrel{IH}{=} \sum_{i \in [n]} p_i(\mu \circ Ts \circ T(\epsilon \otimes [\Sigma']) \circ T([\lambda x.M'_i] \otimes [A] \otimes [\Sigma']) \\
&\quad \circ \sum_{j \in [m]} q_{i,j}(T([\Sigma'_i] \otimes [V_{i,j}] \otimes [\Sigma']) \circ \phi Q'_{i,j})) \\
&= \sum_{i \in [n]} p_i(\mu \circ Ts \circ T([M'_i] \otimes [\Sigma']) \circ \sum_{j \in [m]} q_{i,j}(T([\Sigma'_i] \otimes [V_{i,j}] \otimes [\Sigma']) \circ \phi Q'_{i,j})) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j}(\mu \circ Ts \circ T([M'_i] \circ ([\Sigma'_i] \otimes [V_{i,j}])) \otimes [\Sigma']) \circ \phi Q'_{i,j}) \\
&= RHS.
\end{aligned}$$

To show (1) holds, we just need to show

$$\begin{aligned}
&T((\mu \circ T\epsilon \circ \mu \circ Ts \circ t \circ (T([A] \multimap T[B]) \otimes [N]) \otimes [\Sigma'])) \\
&= T(\mu \otimes [\Sigma']) \circ T(T\epsilon \otimes [\Sigma']) \circ T(t \otimes [\Sigma']) \circ T([A] \multimap T[B] \otimes [N] \otimes [\Sigma']) \\
&\quad \circ \mu \circ Ts \circ T(s \otimes [\Sigma']).
\end{aligned}$$

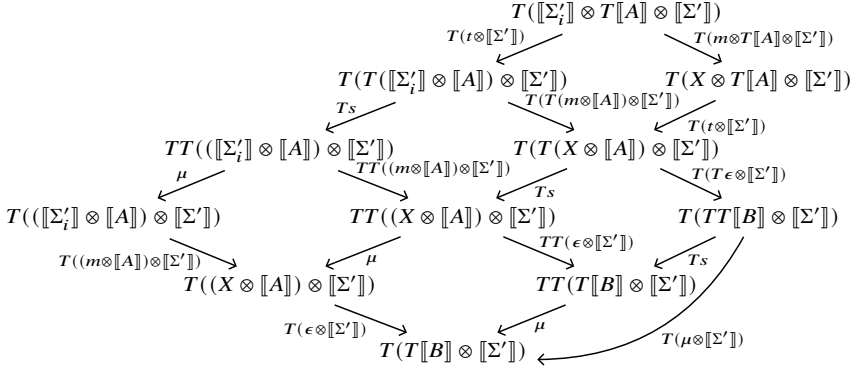
This is true because the following diagram commutes. Here, we have written $X = \llbracket A \rrbracket \multimap T\llbracket B \rrbracket$.



To show (2) holds, we need to show

$$\begin{aligned} & T((\mu \circ T\epsilon \circ t \circ (\llbracket \lambda x.M'_i \rrbracket \otimes T\llbracket A \rrbracket)) \otimes [\Sigma']) \\ &= T(\epsilon \otimes [\Sigma']) \circ T(\llbracket \lambda x.M'_i \rrbracket \otimes \llbracket A \rrbracket \otimes [\Sigma']) \circ \mu \circ Ts \circ T(t \otimes [\Sigma']). \end{aligned}$$

This is true because the following diagram commutes. Here, we have written $m = \llbracket \lambda x.M_i \rrbracket$ and $X = \llbracket A \rrbracket \multimap T\llbracket B \rrbracket$. Note that the right leg of the diagram is the left-hand side by functoriality of $T((-) \otimes [\Sigma'])$.



- Case

$$\frac{\begin{array}{c} (Q, M) \Downarrow \sum_{i \in [n]} p_i(Q_i, \text{lift } M'_i) \\ \text{gen}(S) = a \\ (\text{id}_{\Sigma_a}, M'_i a) \Downarrow (\mathcal{D}_i, b_i) \end{array}}{(Q, \text{box } S M) \Downarrow \sum_{i \in [n]} p_i(Q_i, (a, \mathcal{D}_i, b_i)).}$$

Suppose $\text{addr}(Q) = \Sigma_1, \Sigma'$. Consider the following.

$$\frac{\Sigma_1 \vdash_0 M : !_1(S \multimap_1 U)}{\Sigma_1 \vdash_0 \text{box } S M : \mathbf{Circ}(S, U)}$$

By the induction hypothesis, we have

$$\begin{aligned}
 \llbracket Q, M \rrbracket &= \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \sum_{i \in [n]} p_i \llbracket Q_i, \text{lift } M'_i \rrbracket \\
 &= \sum_{i \in [n]} p_i T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i
 \end{aligned}$$

where $\phi Q_i : I \rightarrow T[\Sigma']$ and $\vdash_1 \text{lift } M'_i : !_1(S \multimap_1 U)$. Note that $\Sigma_a \vdash a : S$. By Theorem 5.6, we have

$$\llbracket \text{id}_{\Sigma_a}, M'_i a \rrbracket = \epsilon \circ (\llbracket M'_i \rrbracket \otimes \llbracket a \rrbracket) = \llbracket b_i \rrbracket \circ \mathcal{D}_i : \llbracket \Sigma_a \rrbracket \rightarrow \llbracket U \rrbracket.$$

Thus $\text{uncurry} \llbracket M'_i \rrbracket = \llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1}$. We need to show

$$\begin{aligned}
 \llbracket Q, \text{box}_S M \rrbracket &= \mu \circ Ts \circ T(\llbracket \text{box}_S M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \mu \circ Ts \circ T((Tp\text{box} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \sum_{i \in [n]} p_n \llbracket Q_i, (a, \mathcal{D}_i, b) \rrbracket \\
 &= \sum_{i \in [n]} p_n (T(p1_\theta) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i,
 \end{aligned}$$

where $\theta = \llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a \rrbracket^{-1} : \llbracket S \rrbracket \rightarrow \llbracket U \rrbracket$.

This is true because we have the following.

$$\begin{aligned}
 &\mu \circ Ts \circ T((Tp\text{box} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= T(p\text{box} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &\stackrel{IH}{=} T(p\text{box} \otimes \llbracket \Sigma' \rrbracket) \circ \sum_{i \in [n]} p_i (T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
 &= \sum_{i \in [n]} p_i (T(p\text{box} \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
 &= \sum_{i \in [n]} p_n (T((p\text{box} \circ p\delta \llbracket M'_i \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i).
 \end{aligned}$$

Thus we just need to show $\text{box} \circ \delta \llbracket M'_i \rrbracket = 1_{\text{uncurry} \llbracket M'_i \rrbracket}$. This is true by the definition of box .

- Case

$$\frac{
 \begin{array}{l}
 (Q, M) \Downarrow \sum_{i \in [n]} p_i (Q_i, \text{lift } M'_i) \\
 (Q_i, M'_i) \Downarrow \sum_{j \in [m]} q_{i,j} (Q'_{i,j}, V_{i,j})
 \end{array}
 }{
 (Q, \text{force } M) \Downarrow \sum_{(i,j) \in [n] \times [m]} p_i q_{i,j} (Q'_{i,j}, V_{i,j}).
 }$$

Suppose $\text{addr}(Q) = \Sigma_1, \Sigma'$ and

$$\frac{\Sigma_1 \vdash_0 M : !_0 A}{\Sigma_1 \vdash_0 \text{force } M : A.}$$

By the induction hypothesis, we have

$$\begin{aligned}
 \llbracket Q, M \rrbracket &= \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \sum_{i \in [n]} p_i \llbracket Q_i, \text{lift } M'_i \rrbracket \\
 &= \sum_{i \in [n]} p_i (T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i),
 \end{aligned}$$

where $\phi Q_i : I \rightarrow T[\llbracket \Sigma' \rrbracket]$ and $\emptyset \vdash_1 \text{lift } M_i : !_0 A$.

Also by the induction hypothesis, we have

$$\begin{aligned}
 \llbracket Q_i, M'_i \rrbracket &= \mu \circ Ts \circ T(\llbracket M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i \\
 &= \sum_{j \in [m]} q_{i,j} \llbracket Q'_{i,j}, V_{i,j} \rrbracket \\
 &= \sum_{j \in [m]} q_{i,j} (T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}),
 \end{aligned}$$

where $\phi Q'_{i,j} : I \rightarrow T(\llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\Sigma_2 \vdash_1 V_{i,j} : A$.

We need to show

$$\begin{aligned}
 \llbracket Q, \text{force } M \rrbracket &= \mu \circ Ts \circ T((\mu \circ T \text{force} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} \llbracket Q'_{i,j}, V_{i,j} \rrbracket \\
 &= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}.
 \end{aligned}$$

This is true because

$$\begin{aligned}
 &\mu \circ Ts \circ T((\mu \circ T \text{force} \circ \llbracket M \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \mu \circ Ts \circ T((\mu \circ T \text{force}) \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &= \mu \circ Ts \circ T(\text{force} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Ts \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
 &\stackrel{IH}{=} \mu \circ Ts \circ T(\text{force} \otimes \llbracket \Sigma' \rrbracket) \circ \sum_{i \in [n]} p_i (T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
 &= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T(\text{force} \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \text{lift } M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
 &= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T(\llbracket M'_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
 &\stackrel{IH}{=} \sum_{i \in [n]} p_i \sum_{j \in [m]} q_{i,j} (T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}) \\
 &= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}).
 \end{aligned}$$

• Case

$$\begin{array}{c}
 (Q, M) \Downarrow \sum_{i \in [n]} p_i (Q_i, (a_i, \mathcal{D}_i, b_i)) \\
 (Q_i, N) \Downarrow \sum_{j \in [m]} q_{i,j} (Q'_{i,j}, V_{i,j}) \\
 \text{applyTo}(\mathcal{D}_i, Q'_{i,j}, a_i, V_{i,j}) = Q''_{i,j} \\
 \hline
 (Q, \text{apply}(M, N)) \Downarrow \sum_{(i,j) \in [n] \times [m]} p_i q_{i,j} (Q''_{i,j}, b_i).
 \end{array}$$

Suppose $\text{addr}(Q) = \Sigma_1, \Sigma_2, \Sigma'$ and

$$\frac{\Sigma_1 \vdash_0 M : \mathbf{Circ}(S, U) \quad \Sigma_2 \vdash_0 N : S}{\Sigma_1, \Sigma_2 \vdash_0 \text{apply}(M, N) : U.}$$

By the induction hypothesis, we have

$$\begin{aligned} \llbracket Q, M \rrbracket &= \mu \circ Ts \circ T(s \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\ &= \sum_{i \in [n]} p_i \llbracket Q_i, (a_i, \mathcal{D}_i, b_i) \rrbracket \\ &= \sum_{i \in [n]} p_i T(\llbracket a_i, \mathcal{D}_i, b_i \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i \\ &= \sum_{i \in [n]} p_i T(p1_\theta \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i, \\ &= \sum_{i \in [n]} p_i T((p \text{ box} \circ p\delta\text{curry}\theta) \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i, \end{aligned}$$

where $\phi Q_i : I \rightarrow T(\llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket)$, and $\theta = \llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} : \llbracket S \rrbracket \rightarrow \llbracket U \rrbracket$, and $\mathcal{D}_i : \llbracket \Sigma_{a_i} \rrbracket \rightarrow \llbracket \Sigma_{b_i} \rrbracket$, and $\Sigma_{a_i} \vdash_1 a_i : S$ and $\Sigma_{b_i} \vdash_1 b_i : U$. Note that the last equality is by definition of box .

Moreover, by the induction hypothesis, we also have

$$\begin{aligned} \llbracket Q_i, N \rrbracket &= \mu \circ Ts \circ T(\llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i \\ &= \sum_{j \in [m]} q_{i,j} \llbracket Q'_{i,j}, V_{i,j} \rrbracket \\ &= \sum_{j \in [m]} q_{i,j} T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}, \end{aligned}$$

where $\phi Q'_{i,j} : I \rightarrow T(\llbracket \Sigma'_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket)$ and $\Sigma'_2 \vdash_1 V_{i,j} : S$.

Since $\text{applyTo}(\mathcal{D}_i, Q'_{i,j}, a_i, V_{i,j}) = Q''_{i,j}$, we have the following in $\mathbf{Q}$.

$$Q''_{i,j} = ((J\mathcal{D}_i \circ J\llbracket a_i \rrbracket^{-1} \circ J\llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ Q'_{i,j} : I \rightarrow \Sigma_{b_i}, \Sigma'.$$

Thus in $\mathbf{A}$, we have

$$\begin{aligned} \phi Q''_{i,j} &= \phi((J\mathcal{D}_i \circ J\llbracket a_i \rrbracket^{-1} \circ J\llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ Q'_{i,j} \\ &= \mu \circ T\phi((J\mathcal{D}_i \circ J\llbracket a_i \rrbracket^{-1} \circ J\llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j} \\ &= \mu \circ T(\mu \circ Ts \circ t \circ \phi(J\mathcal{D}_i \circ J\llbracket a_i \rrbracket^{-1} \circ J\llbracket V_{i,j} \rrbracket) \otimes \eta_{\llbracket \Sigma' \rrbracket}) \circ \phi Q'_{i,j} \\ &\stackrel{(*)}{=} \mu \circ T(\mu \circ Ts \circ t \circ (\eta_{\llbracket \Sigma_{b_i} \rrbracket} \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \eta_{\llbracket \Sigma' \rrbracket}) \circ \phi Q'_{i,j} \\ &= \mu \circ T(s \circ (\eta_{\llbracket \Sigma_{b_i} \rrbracket} \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j} \\ &= \mu \circ T(\eta_{\llbracket \Sigma'_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket} \circ ((\mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket)) \circ \phi Q'_{i,j} \\ &= T((\mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j} : I \rightarrow T(\llbracket \Sigma_{b_i} \rrbracket \otimes \llbracket \Sigma' \rrbracket). \end{aligned}$$

Note that $(*)$ is by $\phi \circ J = E \circ \psi$ in condition (f) (Definition 3.5).

We need to show

$$\begin{aligned}
& \llbracket Q, \text{apply}(M, N) \rrbracket \\
&= \mu \circ Ts \circ T(\llbracket \text{apply}(M, N) \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket)) \\
&\quad \circ \mu \circ Ts \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} \llbracket Q''_{i,j}, b_i \rrbracket \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T(\llbracket b_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q''_{i,j}) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T(\llbracket b_i \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T((\mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T((\llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}).
\end{aligned}$$

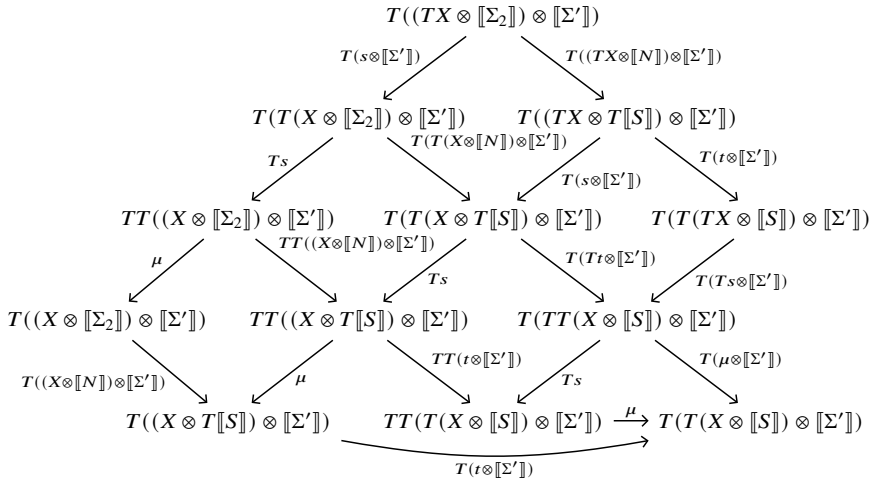
This is true because

$$\begin{aligned}
& \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket)) \circ \mu \circ Ts \circ t \circ (\llbracket M \rrbracket \otimes \llbracket N \rrbracket)) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ \mu \circ Ts \circ t) \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ T((\llbracket M \rrbracket \otimes \llbracket N \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&= \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ \mu \circ Ts \circ t) \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ T(T\llbracket \text{Circ}(S, U) \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&\stackrel{(1)}{=} \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ t) \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \text{Circ}(S, U) \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ \mu \circ Ts \circ T(s \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket M \rrbracket \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q \\
&\stackrel{IH}{=} \mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ t) \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket \text{Circ}(S, U) \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ \sum_{i \in [n]} p_i (T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ t) \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ T(\llbracket \text{Circ}(S, U) \rrbracket \otimes \llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes \llbracket \Sigma_2 \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i (\mu \circ Ts \circ T((T\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ t) \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes T\llbracket S \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
&\stackrel{(2)}{=} \sum_{i \in [n]} p_i (T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket \Sigma' \rrbracket) \circ T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes \llbracket S \rrbracket \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ T(\lambda^{-1} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Ts \circ T(\llbracket N \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q_i) \\
&= \sum_{i \in [n]} p_i (T(\epsilon \otimes \llbracket \Sigma' \rrbracket) \circ T(\text{curry} \theta \otimes \llbracket S \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ T(\lambda^{-1} \otimes \llbracket \Sigma' \rrbracket) \\
&\quad \circ \sum_{j \in [m]} q_{i,j} (T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j})) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T(\theta \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T((\llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1}) \otimes \llbracket \Sigma' \rrbracket) \circ T(\llbracket V_{i,j} \rrbracket \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}) \\
&= \sum_{i,j \in [n] \times [m]} p_i q_{i,j} (T((\llbracket b_i \rrbracket \circ \mathcal{D}_i \circ \llbracket a_i \rrbracket^{-1} \circ \llbracket V_{i,j} \rrbracket) \otimes \llbracket \Sigma' \rrbracket) \circ \phi Q'_{i,j}).
\end{aligned}$$

To prove (1), we just need to prove

$$\begin{aligned} & T((\mu \circ Ts \circ t) \otimes [\Sigma']) \circ T(T[\mathbf{Circ}(S, U)] \otimes [N] \otimes [\Sigma']) \\ &= T((t \otimes [\Sigma']) \circ T([\mathbf{Circ}(S, U)] \otimes [N] \otimes [\Sigma']) \circ \mu \circ Ts \circ T(s \otimes [\Sigma'])). \end{aligned}$$

This is true because the following diagram commutes. Here, we have written $X = \llbracket \mathbf{Circ}(S, U) \rrbracket$.

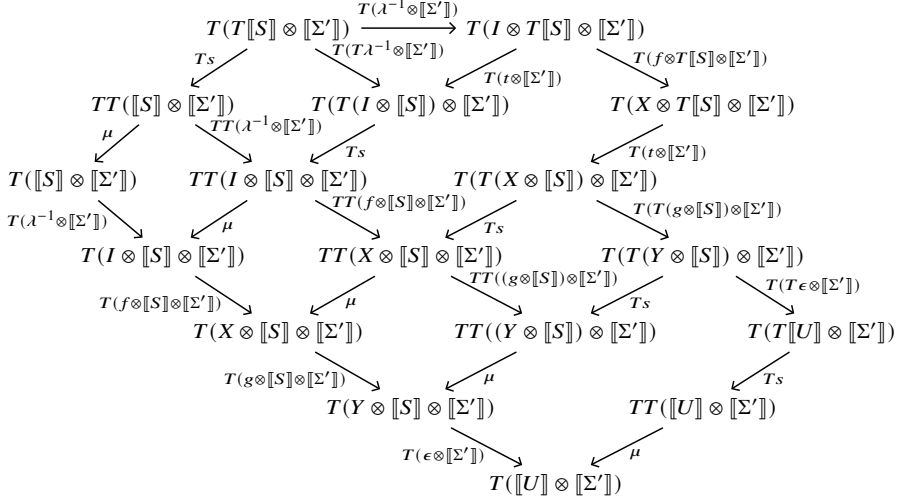


To prove (2), we just need to show

$$\begin{aligned}
& \mu \circ Ts \circ T((\epsilon \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket S \rrbracket) \circ t) \otimes \llbracket \Sigma' \rrbracket)) \\
& \quad \circ T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes T\llbracket S \rrbracket \otimes \llbracket \Sigma' \rrbracket)) \\
& = T(\epsilon \circ \llbracket \Sigma' \rrbracket) \circ T((\text{force} \circ p \text{ unbox}) \otimes \llbracket \Sigma' \rrbracket)) \\
& \quad \circ T((p \text{ box} \circ p \delta \text{ curry} \theta) \otimes \llbracket S \rrbracket \otimes \llbracket \Sigma' \rrbracket)) \circ T(\lambda^{-1} \otimes \llbracket \Sigma' \rrbracket) \circ \mu \circ Ts.
\end{aligned}$$

This is true because the following diagram commutes. Here, we have written $X = \llbracket \mathbf{Circ}(S, U) \rrbracket$ and $Y = \llbracket S \rrbracket \multimap \llbracket U \rrbracket$, as well as $f = p \mathbf{box} \circ p \delta \mathbf{curry} \theta$ and $g =$

force $\circ p$ unbox.



□

E Enriched symmetric monoidal categories

Definition E.1. Let $\mathcal{V}$ be a symmetric monoidal category. A $\mathcal{V}$ -category $\mathbf{A}$ is symmetric monoidal if it is equipped with the following:

- There is an object I , called the *tensor unit*. For all $A, B \in \mathbf{A}$, there is an object $A \otimes B \in \mathbf{A}$. Moreover, for all $A_1, A_2, B_1, B_2 \in \mathbf{A}$, there is a map

$$\text{Tensor} : \mathbf{A}(A_1, B_1) \otimes \mathbf{A}(A_2, B_2) \rightarrow \mathbf{A}(A_1 \otimes A_2, B_1 \otimes B_2)$$

in $\mathcal{V}$. The tensor product is a bifunctor in the sense that $\text{Tensor} \circ (u_A \otimes u_B) = u_{A \otimes B}$ for the identity maps $u_A, u_B, u_{A \otimes B}$, and the following diagram commutes for any $A_1, A_2, B_1, B_2, C_1, C_2 \in \mathbf{A}$.

$$\begin{array}{ccc} \mathbf{A}(A_1, B_1) \otimes \mathbf{A}(A_2, B_2) \otimes \mathbf{A}(B_1, C_1) \otimes \mathbf{A}(B_2, C_2) & \xrightarrow{\text{id} \otimes \gamma \otimes \text{id}; c \otimes c} & \mathbf{A}(A_1, C_1) \otimes \mathbf{A}(A_2, C_2) \\ \downarrow \text{Tensor} \otimes \text{Tensor} & & \downarrow \text{Tensor} \\ \mathbf{A}(A_1 \otimes A_2, B_1 \otimes B_2) \otimes \mathbf{A}(B_1 \otimes B_2, C_1 \otimes C_2) & \xrightarrow{c} & \mathbf{A}(A_1 \otimes A_2, C_1 \otimes C_2) \end{array}$$

- There are the following $\mathcal{V}$ -natural isomorphisms in $\mathbf{A}$ and they satisfy the same coherence diagrams as for symmetric monoidal categories, and analogous naturality conditions.

$$l_A : I \otimes A \rightarrow A$$

$$r_A : A \otimes I \rightarrow A$$

$$\gamma_{A,B} : A \otimes B \rightarrow B \otimes A$$

$$\alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$$

If the $\mathcal{V}$ -category $\mathbf{A}$ is symmetric monoidal, for all maps $f : A_1 \rightarrow B_1, g : A_2 \rightarrow B_2$ in $\mathbf{A}$, we write $f \otimes g : A_1 \otimes A_2 \rightarrow B_1 \otimes B_2$ as a shorthand for the following composition.

$$I \xrightarrow{f \otimes g} \mathbf{A}(A_1, B_1) \otimes \mathbf{A}(A_2, B_2) \xrightarrow{\text{Tensor}} \mathbf{A}(A_1 \otimes A_2, B_1 \otimes B_2)$$

F Proof of Theorem 6.10

In this section, we assume $\mathcal{V}$ to be a complete, cocomplete, symmetric monoidal closed category. The following theorem is due to Kock [20].

Theorem F.1. *Let $T : \mathcal{V} \rightarrow \mathcal{V}$ be a $\mathcal{V}$ -monad. Then T is a strong monad with strength $t : A \otimes TB \rightarrow T(A \otimes B)$ given by the following commutative diagram. Note that η is the unit of the adjunction $- \otimes A \dashv A \multimap -$.*

$$\begin{array}{ccc} A & \xrightarrow{\text{curry}(t)} & TB \multimap T(A \otimes B) \\ \eta \downarrow & \nearrow T_{B, A \otimes B} & \\ B \multimap A \otimes B & & \end{array}$$

Theorem F.2. *Let T be a strong monad on $\mathcal{V}$ and $F : \mathbf{A}^{\text{op}} \rightarrow \mathcal{V}$ be a $\mathcal{V}$ -functor. For all $A, B \in \mathbf{A}$, the functors F and T induce the following maps in $\mathcal{V}$.*

$$F_{AB} : \mathbf{A}(B, A) \rightarrow FA \multimap FB$$

and

$$(TF)_{AB} : \mathbf{A}(B, A) \rightarrow TFA \multimap TFB.$$

The following diagram commutes.

$$\begin{array}{ccc} \mathbf{A}(B, A) \otimes TFA & & \\ \downarrow t & \searrow \text{uncurry}((TF)_{AB}) & \\ T(\mathbf{A}(B, A) \otimes FA) & \xrightarrow{T\text{uncurry}(F_{AB})} & TFB \end{array}$$

PROOF. By currying the diagram above, we just need to show the right triangle commutes in the following diagram.

$$\begin{array}{ccccc} & & \mathbf{A}(B, A) & & \\ & \swarrow F_{AB} & \downarrow \eta & \searrow (TF)_{AB} & \searrow \text{curry}(t) \\ & FA \multimap \mathbf{A}(B, A) \otimes FA & & & TFA \multimap T(\mathbf{A}(B, A) \otimes FA) \\ & \swarrow FA \multimap \text{uncurry}(F_{AB}) & & \searrow T_{FA, \mathbf{A}(B, A) \otimes FA} & \\ FA \multimap FB & \xrightarrow{T_{FA, FB}} & TFA \multimap TFB & \xleftarrow{TFA \multimap T\text{uncurry}(F_{AB})} & \end{array}$$

Note that the bottom square commutes because of the $\mathcal{V}$ -naturality of T . The left triangle commutes by the property of monoidal closedness. The front triangle commutes by definition of $(TF)_{AB}$. The back triangle commutes by Theorem F.1. $\square$

Theorem F.3. *Let $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A} \rightarrow \mathcal{V}$ be a $\mathcal{V}$ -functor and let T be a strong monad on $\mathcal{V}$. Then there exists a natural map*

$$\xi_F : \int^{A \in \mathbf{A}} TF(A, A) \rightarrow T \int^{A \in \mathbf{A}} F(A, A).$$

PROOF. Recall that by definition of coend, we have the following coequalizers.

$$\sum_{A, B \in \mathbf{A}} \mathbf{A}(B, A) \otimes F(A, B) \xrightleftharpoons[\rho_2]{\rho_1} \sum_{A \in \mathbf{A}} F(A, A) \xrightarrow{e} \int^{A \in \mathbf{A}} F(A, A)$$

$$\sum_{A,B \in \mathbf{A}} \mathbf{A}(B, A) \otimes TF(A, B) \xrightarrow[\rho'_2]{\rho'_1} \sum_{A \in \mathbf{A}} TF(A, A) \xrightarrow{e'} \int^{A \in \mathbf{A}} TF(A, A)$$

For any $A \in \mathbf{A}$, the functor $F(A, -) : \mathbf{A} \rightarrow \mathcal{V}$ gives rise to a map

$$F(A, -)_{BA} : \mathbf{A}(B, A) \rightarrow F(A, B) \multimap F(A, A)$$

for each $B \in \mathbf{A}$. The map ρ_1 is defined as the coproduct pairing $[\text{inj}_A \circ \text{uncurry}(F(A, -)_{BA})]_{A, B \in \mathbf{A}}$.

For any $B \in \mathbf{A}$, the functor $F(-, B) : \mathbf{A}^{\text{op}} \rightarrow \mathcal{V}$ gives rise to a map

$$F(-, B)_{AB} : \mathbf{A}(B, A) \rightarrow F(A, B) \multimap F(B, B)$$

for each $A \in \mathbf{A}$. The map ρ_2 is defined as the coproduct pairing $[\text{inj}_B \circ \text{uncurry}(F(-, B)_{AB})]_{A, B \in \mathbf{A}}$.

The maps ρ'_1, ρ'_2 are induced similarly.

Consider the following diagram.

$$\begin{array}{ccc} \int^A TF(A, A) & \xrightarrow{\quad \xi \quad} & T \int^A F(A, A) \\ e' \uparrow & & \uparrow Te \\ \sum_A TF(A, A) & \xrightarrow{[\text{Tinj}_A]_A} & T \sum_A F(A, A) \\ \rho'_1 \uparrow \uparrow \rho'_2 & & T\rho_1 \uparrow \uparrow T\rho_2 \\ \sum_{A,B} \mathbf{A}(B, A) \otimes TF(A, B) & & T \sum_{A,B} \mathbf{A}(B, A) \otimes F(A, B) \\ \searrow \Sigma_{A,B} t & & \nearrow [\text{Tinj}_{A,B}]_{A,B} \\ & \Sigma_{A,B} T(\mathbf{A}(B, A) \otimes F(A, B)) & \end{array}$$

Note that $[\text{Tinj}_A]_A$ and $[\text{Tinj}_{A,B}]_{A,B}$ are coproduct pairings. The morphism $t : \mathbf{A}(B, A) \otimes TF(A, B) \rightarrow T(\mathbf{A}(B, A) \otimes F(A, B))$ is the strength map for T .

To show the existence of ξ , we just need to show $Te \circ [\text{Tinj}_A]_A \circ \rho'_1 = Te \circ [\text{Tinj}_A]_A \circ \rho'_2$, which is to show the bottom square commutes for ρ'_1 and $T\rho_1$ (ρ'_2 and $T\rho_2$). This is the case because of the following commutative diagram (for the A, B component). Note that the left triangle commutes by Theorem F.2.

$$\begin{array}{ccccc} \mathbf{A}(B, A) \otimes TF(A, B) & \xrightarrow{\rho'_1(A,B)} & TF(A, A) & \xrightarrow{\text{Tinj}_A} & T \sum_A F(A, A) \\ t \downarrow & \nearrow T\rho_1(A,B) & & & \uparrow T\rho_1 \\ T(\mathbf{A}(B, A) \otimes F(A, B)) & \xrightarrow{\text{Tinj}_{A,B}} & T \sum_{A,B} \mathbf{A}(B, A) \otimes F(A, B) & & \end{array}$$

Note that $\rho'_1(A, B)$ is a component of ρ'_1 and $\rho_1(A, B)$ is a component of ρ_1 . By the universal property of the coequalizer e' , there exists a unique arrow

$$\xi : \int^{A \in \mathbf{A}} TF(A, A) \rightarrow T \int^{A \in \mathbf{A}} F(A, A). \quad \square$$

Theorem F.4. Suppose $F : \mathbf{A}^{\text{op}} \rightarrow \mathcal{V}$. For all $B, C \in \mathbf{A}$, the following diagram commutes.

$$\begin{array}{ccc} \mathbf{A}(C, B) \otimes F(B) & \xrightarrow{e_B} & \int^B \mathbf{A}(C, B) \otimes F(B) \\ \text{uncurry}(F_{BC}) \downarrow & & \swarrow y \\ F(C) & & \end{array}$$

Note that y is an isomorphism expressing the Yoneda lemma in the language of coends, and $F_{BC} : \mathbf{A}(C, B) \rightarrow F(B) \multimap F(C)$, and e_B is the unit of the coend.

PROOF. Note that the map $\text{uncurry}(F_{BC}) : \mathbf{A}(C, B) \otimes F(B) \rightarrow F(C)$ is $\mathcal{V}$ -natural in B . By the universal property of coends, there exists a map $y : \int^B \mathbf{A}(C, B) \otimes F(B) \rightarrow F(C)$ such that the diagram above commutes. Moreover, y is an isomorphism [19, Chapter 2.4]. $\square$

Theorem F.5. Let T be a strong monad on $\mathcal{V}$. For all $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A} \rightarrow \mathcal{V}$, the map $\xi_F : \int^{A \in \mathbf{A}} TF(A, A) \rightarrow T \int^{A \in \mathbf{A}} F(A, A)$ makes the following diagrams commute.

(a)

$$\begin{array}{ccc} \int^A F(A, A) & & \\ \eta \downarrow & \searrow \int \eta & \\ T \int^A F(A, A) & \xleftarrow{\xi} & \int^A TF(A, A) \end{array}$$

(b)

$$\begin{array}{ccc} \int^A TTF(A, A) & \xrightarrow{\xi} & T \int^A TF(A, A) \xrightarrow{T\xi} TT \int^A F(A, A) \\ \int \mu \downarrow & & \downarrow \mu \\ \int^A TF(A, A) & \xrightarrow{\xi} & T \int^A F(A, A) \end{array}$$

(c) Suppose $G : \mathbf{A}^{\text{op}} \rightarrow \mathcal{V}$ and $A \in \mathbf{A}$.

$$\begin{array}{ccc} \int^B \mathbf{A}(A, B) \otimes TGB & \xrightarrow{y'} & TGA \\ \int t \downarrow & & \uparrow Ty \\ \int^B T(\mathbf{A}(A, B) \otimes GB) & \xrightarrow{\xi} & T \int^B \mathbf{A}(A, B) \otimes GB \end{array}$$

Note that y', y are isomorphisms induced by the Yoneda lemma.

(d) Suppose $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A} \rightarrow \mathcal{V}$ and $X \in \mathcal{V}$.

$$\begin{array}{ccc} (\int^A F(A, A)) \otimes TX & \xrightarrow{t} & T((\int^A F(A, A)) \otimes X) \\ \cong \downarrow & & \downarrow \cong \\ \int^A (F(A, A) \otimes TX) & & \\ \int t \downarrow & & \downarrow \\ \int^A T(F(A, A) \otimes X) & \xrightarrow{\xi} & T \int^A (F(A, A) \otimes X) \end{array}$$

(e) Suppose $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A} \rightarrow \mathcal{V}$ and $X \in \mathcal{V}$.

$$\begin{array}{ccc} \int^A (X \otimes TF(A, A)) & \xrightarrow{\int t} & \int^A T(X \otimes F(A, A)) \\ \cong \downarrow & & \downarrow \xi \\ X \otimes \int^A TF(A, A) & & T \int^A (X \otimes F(A, A)) \\ X \otimes \xi \downarrow & & \downarrow \cong \\ X \otimes T \int^A F(A, A) & \xrightarrow{t} & T(X \otimes \int^A F(A, A)) \end{array}$$

(f) Suppose $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A}^{\text{op}} \otimes \mathbf{A} \otimes \mathbf{A} \rightarrow \mathcal{V}$.

$$\begin{array}{ccc}
 \int^A \int^B TF(A, B, A, B) & \xrightarrow{\int^A \xi} & \int^A T \int^B F(A, B, A, B) \xrightarrow{\xi} T \int^A \int^B F(A, B, A, B) \\
 \cong \downarrow & & \downarrow \cong \\
 \int^{A,B} TF(A, B, A, B) & \xrightarrow{\xi} & T \int^{A,B} F(A, B, A, B)
 \end{array}$$

PROOF. (a) We need to show that the following commutes.

$$\begin{array}{ccc}
 \int^A F(A, A) & & \\
 \eta \downarrow & \searrow \int \eta & \\
 T \int^A F(A, A) & \xleftarrow{\xi} & \int^A TF(A, A)
 \end{array}$$

Consider the following diagram. We write η_1 for the map $F(A, A) \rightarrow TF(A, A)$ and η_2 for the map $\int^A F(A, A) \rightarrow T \int^A F(A, A)$.

$$\begin{array}{ccccc}
 \int^A F(A, A) & \xrightarrow{\int \eta_1} & \int^A TF(A, A) & & \\
 \uparrow e & \searrow \eta_2 & \uparrow e' & \searrow \xi & \\
 \Sigma_A F(A, A) & \xrightarrow{\Sigma \eta_1} & \Sigma_A TF(A, A) & \xrightarrow{[T\text{inj}]_A} & T \int^A F(A, A) \\
 & \searrow \eta & & \uparrow Te & \\
 & & & T \Sigma_A F(A, A) &
 \end{array}$$

We need to show that the top triangle commutes. Since e is an epimorphism, we just need to show $\xi \circ \int \eta_1 \circ e = \eta_2 \circ e$. This is the case because the bottom triangle commutes and all three square faces commute. The bottom triangle commutes by the universal property of coproducts. The square with ξ commutes by definition of ξ . Also note that $e' \circ \Sigma \eta_1 = \int \eta_1 \circ e$ is a property of coends (see 19, Eqn. 2.4).

(b) The proof is similar to (1).

(c) Next we need to show that the following commutes (where y, y' are isomorphisms induced by the Yoneda lemma).

$$\begin{array}{ccc}
 \int^B \mathbf{A}(A, B) \otimes TGB & \xrightarrow{\int^t} & \int^B T(\mathbf{A}(A, B) \otimes GB) \\
 y' \downarrow & & \downarrow \xi \\
 TGA & \xleftarrow{Ty} & T \int^B \mathbf{A}(A, B) \otimes GB
 \end{array}$$

The above diagram commutes because the following diagram commutes for all $A, B \in \mathbf{A}$.

$$\begin{array}{ccccc}
 \mathbf{A}(A, B) \otimes TGB & \xrightarrow{e_1} & \int^B \mathbf{A}(A, B) \otimes TGB & & \\
 \text{uncurry}((TG)_{BA}) \swarrow & & \searrow y' & & \downarrow \int t \\
 TGA & \xrightarrow{\text{id}} & TGA & & \\
 \uparrow T\text{uncurry}(G_{BA}) & & \uparrow Ty & & \\
 T(\mathbf{A}(A, B) \otimes GB) & \xrightarrow{e_2} & \int^B T(\mathbf{A}(A, B) \otimes GB) & & \\
 \text{id} \swarrow & & \searrow \xi & & \\
 T(\mathbf{A}(A, B) \otimes G(B)) & \xrightarrow{Te_3} & T \int^B \mathbf{A}(A, B) \otimes GB & &
 \end{array}$$

t (vertical arrow from $\mathbf{A}(A, B) \otimes TGB$ to $T(\mathbf{A}(A, B) \otimes GB)$)
 Ty (vertical arrow from TGA to $T \int^B \mathbf{A}(A, B) \otimes GB$)

Since G and TG are contravariant $\mathcal{V}$ -functors, there are the following maps in $\mathcal{V}$.

$$G_{BA} : \mathbf{A}(A, B) \rightarrow GB \Rightarrow GA$$

$$(TG)_{BA} : \mathbf{A}(A, B) \rightarrow TGB \Rightarrow TGA$$

The bottom square commutes by the definition of ξ , and the back square (with e_1, e_2) commutes by naturality of coends. The top and the front squares commutes because of Theorem F.4. Thus we just need to show that the left square commutes, i.e.,

$$\begin{array}{ccc}
 \mathbf{C}(C, B) \otimes TFB & \xrightarrow{t} & T(\mathbf{C}(C, B) \otimes FB) \\
 \text{uncurry}(TG)_{BA} \downarrow & \swarrow T(\text{uncurry}G_{BA}) & \\
 TFC & &
 \end{array}$$

This commutes by Theorem F.2.

(d) Next we need to prove that the following commutes.

$$\begin{array}{ccc}
 (\int^A F(A, A)) \otimes TX & \xrightarrow{t} & T((\int^A F(A, A)) \otimes X) \\
 \cong \downarrow & & \downarrow \cong \\
 \int^A (F(A, A) \otimes TX) & & \\
 \int t \downarrow & & \downarrow \\
 \int^A T(F(A, A) \otimes X) & \xrightarrow{\xi} & T \int^A (F(A, A) \otimes X)
 \end{array}$$

First observe that the following commutes (each arrow is canonical).

$$\begin{array}{ccc}
 (\sum_A F(A, A)) \otimes TX & \longrightarrow & T((\sum_A F(A, A)) \otimes X) \\
 \cong \downarrow & & \downarrow \cong \\
 \sum_A (F(A, A) \otimes TX) & & \\
 \downarrow & & \downarrow \\
 \sum_A T(F(A, A) \otimes X) & \longrightarrow & T \sum_A (F(A, A) \otimes X)
 \end{array}$$

Now let us consider the following cube.

$$\begin{array}{ccccc}
 (\int^A F(A, A)) \otimes TX & \xrightarrow{t} & T((\int^A F(A, A)) \otimes X) & & \\
 \nearrow & \downarrow \cong & \nearrow T(e \otimes X) & & \downarrow \cong \\
 (\sum_A F(A, A)) \otimes TX & \xrightarrow{t} & T((\sum_A F(A, A)) \otimes X) & & \\
 \downarrow \cong & \downarrow \int^A (F(A, A) \otimes TX) & \downarrow \cong & & \\
 \sum_A (F(A, A) \otimes TX) & \xrightarrow{\int t} & \sum_A T(F(A, A) \otimes X) & \xrightarrow{\xi} & T \int^A (F(A, A) \otimes X) \\
 \downarrow \sum_A t & \downarrow \int^A T(F(A, A) \otimes X) & \downarrow \cong & \nearrow T e' & \\
 \sum_A T(F(A, A) \otimes X) & \xrightarrow{[T \text{inj}_A]_A} & T \sum_A (F(A, A) \otimes X) & &
 \end{array}$$

Note that the top square commutes, by naturality of t . The bottom square commutes, by definition of ξ . The left square involving $\sum_A t$, $\int t$ commutes by naturality of coend. The right square and the left top square commute for the same reason. For simplicity, consider the following diagram.

$$\begin{array}{ccccc}
 (\sum_{A,B} \mathbf{A}(B, A) \otimes F(A, B)) \otimes X & \rightrightarrows & (\sum_A F(A, A)) \otimes X & \twoheadrightarrow & (\int^A F(A, A)) \otimes X \\
 \downarrow \cong & & \downarrow \cong & & \downarrow \cong \\
 \sum_{A,B} (\mathbf{A}(B, A) \otimes F(A, B) \otimes X) & \rightrightarrows & \sum_A (F(A, A) \otimes X) & \twoheadrightarrow & \int^A (F(A, A) \otimes X)
 \end{array}$$

Note that the right square is the same square as the right square in the cube. And $- \otimes X$ preserves coequalizers. The left square commutes by naturality. This implies that the right square commutes, by the universal property of coequalizers. Therefore the back side of the cube above commutes.

(e) Next, we need to show that the following diagram commutes.

$$\begin{array}{ccc}
 \int^A (X \otimes TF(A, A)) & \xrightarrow{\int t} & \int^A T(X \otimes F(A, A)) \\
 \downarrow \cong & & \downarrow \xi \\
 X \otimes \int^A TF(A, A) & & T \int^A (X \otimes F(A, A)) \\
 \text{id} \otimes \xi \downarrow & & \downarrow \cong \\
 X \otimes T \int^A F(A, A) & \xrightarrow{t} & T(X \otimes \int^A F(A, A))
 \end{array}$$

First, observe that the following diagram commutes.

$$\begin{array}{ccc}
 \sum_A (X \otimes TF(A, A)) & \longrightarrow & \sum_A T(X \otimes F(A, A)) \\
 \downarrow \cong & & \downarrow \\
 X \otimes \sum_A TF(A, A) & & T \sum_A (X \otimes F(A, A)) \\
 \downarrow & & \cong \downarrow \\
 X \otimes T \sum_A F(A, A) & \longrightarrow & T(X \otimes \sum_A F(A, A))
 \end{array}$$

Now let us consider the following cube.

$$\begin{array}{ccccc}
 & \int^A (X \otimes TF(A, A)) & \xrightarrow{f^t} & \int^A T(X \otimes F(A, A)) & \\
 \nearrow & \downarrow \Sigma t & \cong & \downarrow \Sigma_A & \nearrow \\
 \Sigma_A (X \otimes TF(A, A)) & & \Sigma_A T(X \otimes F(A, A)) & & \\
 \downarrow \cong & \nearrow X \otimes \int^A TF(A, A) & \downarrow X \otimes \xi & \nearrow T \int^A (X \otimes F(A, A)) & \downarrow \xi \\
 X \otimes \Sigma_A TF(A, A) & & X \otimes T \Sigma_A (X \otimes F(A, A)) & & T \int^A (X \otimes F(A, A)) \\
 \downarrow & \nearrow X \otimes T \int^A F(A, A) & \downarrow X \otimes \xi & \nearrow T \int^A (X \otimes F(A, A)) & \downarrow \cong \\
 X \otimes T \Sigma_A F(A, A) & & X \otimes T \int^A F(A, A) & \xrightarrow{t} & T(X \otimes \int^A F(A, A)) \\
 \downarrow & \nearrow & \downarrow & \nearrow Te' & \\
 X \otimes T \Sigma_A F(A, A) & \xrightarrow{t} & T(X \otimes \Sigma_A F(A, A)) & &
 \end{array}$$

Note that the top square commutes by naturality of coends. The bottom square commutes by naturality of t . The left bottom and the right top square involving ξ commute by definition. The left top and the right bottom square commute for the same reason. Consider the following diagram.

$$\begin{array}{ccccc}
 \Sigma_{A,B}(\mathbf{A}(B, A) \otimes X \otimes F(A, B)) & \rightrightarrows & \Sigma_A(X \otimes F(A, A)) & \twoheadrightarrow & \int^A (X \otimes F(A, A)) \\
 \downarrow \cong & & \downarrow \cong & & \downarrow \cong \\
 X \otimes \Sigma_{A,B}(\mathbf{A}(B, A) \otimes F(A, B)) & \rightrightarrows & X \otimes \Sigma_A F(A, A) & \twoheadrightarrow & X \otimes \int^A F(A, A)
 \end{array}$$

Note that the right square is the same square as the right bottom square in the cube under the functor T . And $X \otimes -$ preserves coequalizers. The left square commutes by naturality. This implies that the right square commutes, by the universal property of coequalizers. Therefore the backside of the cube above commutes.

- (f) Let $F : \mathbf{A}^{\text{op}} \otimes \mathbf{A}^{\text{op}} \otimes \mathbf{A} \otimes \mathbf{A} \rightarrow \mathcal{V}$. We need to show that the following diagram commutes.

$$\begin{array}{ccc}
 \int^A \int^B TF(A, B, A, B) & \xrightarrow{\int^A \xi} & \int^A T \int^B F(A, B, A, B) \xrightarrow{\xi} T \int^A \int^B F(A, B, A, B) \\
 \downarrow f & & \downarrow Tf \\
 \int^{A,B} TF(A, B, A, B) & \xrightarrow{\xi} & T \int^{A,B} F(A, B, A, B)
 \end{array}$$

First, the isomorphism f is an instance of the so-called Fubini theorem for coends, which also gives rise to the following commutative diagram.

$$\begin{array}{ccc}
 F(A, B, A, B) & \xrightarrow{e_{A,B}} & \int^{A,B} F(A, B, A, B) \\
 e_B \downarrow & & \uparrow f \\
 \int^B F(A, B, A, B) & \xrightarrow{e_A} & \int^A \int^B F(A, B, A, B)
 \end{array}$$

The following diagram commutes by properties of the Fubini theorem, the definition of ξ , and the $\mathcal{V}$ -naturality of coends.

$$\begin{array}{ccccc}
 \int^A \int^B TF(A, B, A, B) & \xrightarrow{\int^A \xi} & \int^A T \int^B F(A, B, A, B) & \xrightarrow{\xi} & T \int^A \int^B F(A, B, A, B) \\
 \uparrow e'_A & & \uparrow e''_A & \nearrow Te_A & \downarrow Tf \\
 \int^B TF(A, B, A, B) & \xrightarrow{\xi} & T \int^B F(A, B, A, B) & & T \int^{A,B} F(A, B, A, B) \\
 \uparrow e'_B & \nearrow Te_B & & \nearrow Te_{A,B} & \uparrow \xi \\
 TF(A, B, A, B) & \xrightarrow{\quad} & \int^{A,B} TF(A, B, A, B) & & \int^{A,B} TF(A, B, A, B) \\
 \downarrow e'_B & & \xleftarrow{e'_{A,B}} & & \uparrow f \\
 \int^B TF(A, B, A, B) & \xrightarrow{\quad} & \int^A \int^B TF(A, B, A, B) & & \int^A \int^B TF(A, B, A, B) \\
 & & \xleftarrow{e'_A} & &
 \end{array}$$

This implies the claim since e'_A and e'_B are epimorphisms. $\square$

THEOREM 6.10. *Let $\mathbf{A}$ be a $\mathcal{V}$ -category. If T is a commutative strong $\mathcal{V}$ -monad on $\mathcal{V}$ (the strength is given by the map $t_{A,B} : A \otimes TB \rightarrow T(A \otimes B)$ for any $A, B \in \mathbf{A}$), then $\bar{T}(F) = T \circ F$ is a commutative strong $\mathcal{V}$ -monad on $\mathcal{V}^{\mathbf{A}^{\text{op}}}$.*

PROOF. It is straightforward to verify that $\bar{T}$ is a monad. We define the strength $\bar{t}$ to be the following composition.

$$\begin{aligned}
 (F \otimes_{\text{Day}} \bar{T}G)(C) &= \int^{(A,B) \in \mathbf{A} \otimes \mathbf{A}} \mathbf{A}(C, A \otimes B) \otimes FA \otimes TGB \\
 &\xrightarrow{\int^{(A,B)} t} \int^{(A,B) \in \mathbf{A} \otimes \mathbf{A}} T(\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) \\
 &\xrightarrow{\xi} T \int^{(A,B) \in \mathbf{A} \otimes \mathbf{A}} (\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) = \bar{T}(F \otimes_{\text{Day}} G)(C)
 \end{aligned}$$

Now to show that $\bar{T}$ is a commutative strong monad, we need to show the following properties.

- We must show that this diagram commutes.

$$\begin{array}{ccc}
 F \otimes_{\text{Day}} G & \xrightarrow{F \otimes_{\text{Day}} \eta} & F \otimes_{\text{Day}} \bar{T}G \\
 \eta \downarrow & \nwarrow \bar{t} & \\
 \bar{T}(F \otimes_{\text{Day}} G) & &
 \end{array}$$

To show this, we just need to show that the following diagram commutes for any $C \in \mathbf{A}$.

$$\begin{array}{ccc}
 \int^{A,B} \mathbf{A}(C, A \otimes B) \otimes FA \otimes GB & \xrightarrow{\int^{A,B} \eta'} & \int^{A,B} \mathbf{A}(C, A \otimes B) \otimes FA \otimes TGB \\
 \eta \downarrow & \searrow \int^{A,B} \eta & \downarrow \int^{A,B} t \\
 T(\int^{A,B} \mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) & \xleftarrow{\xi} & \int^{A,B} T(\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB)
 \end{array}$$

Note that $\int^{A,B} \eta'$ is a shorthand for $\int^{A,B} \mathbf{A}(C, A \otimes B) \otimes FA \otimes \eta$ in the above diagram. Similarly, $\int^{A,B} t$ is a shorthand for $\int^{A,B} t_{\mathbf{A}(C, A \otimes B) \otimes FA, GB}$. The bottom triangle commutes because of Theorem F.5(a). The top triangle commutes by properties of t .

- We must show that the following diagram commutes.

$$\begin{array}{ccc}
 I \otimes_{\text{Day}} \bar{T}F & \xrightarrow{\bar{t}_{I,F}} & \bar{T}(I \otimes_{\text{Day}} F) \\
 & \searrow \lambda_{\bar{T}F} & \downarrow \bar{T}\lambda_F \\
 & & \bar{T}F
 \end{array}$$

If we unfold the definition of the Day tensor and apply coend version of the Yoneda lemma, we have the following diagram for any $C \in \mathbf{A}$.

$$\begin{array}{ccccc}
 \int^B \mathbf{A}(C, I \otimes B) \otimes TFB & \xrightarrow{\int t} & \int^B T(\mathbf{A}(C, I \otimes B) \otimes FB) & & \\
 \searrow \int \lambda & & \searrow \int T\lambda & & \searrow \xi \\
 \int^B \mathbf{A}(C, B) \otimes TFB & \xrightarrow{\int t'} & \int^B T(\mathbf{A}(C, B) \otimes FB) & & T \int^B \mathbf{A}(C, I \otimes B) \otimes FB \\
 \searrow \cong & & \searrow \xi & & \swarrow T \int \lambda \\
 & TFC & \xrightarrow{\cong} & T \int^B \mathbf{A}(C, B) \otimes FB &
 \end{array}$$

Note that for any $C \in \mathbf{A}$, the following commutes by naturality of t .

$$\begin{array}{ccc}
 \int^B \mathbf{A}(C, I \otimes B) \otimes TFB & \xrightarrow{\int t} & \int^B T(\mathbf{A}(C, I \otimes B) \otimes FB) \\
 \int \lambda_B \downarrow & & \downarrow \int T\lambda_B \\
 \int^B \mathbf{A}(C, B) \otimes TFB & \xrightarrow{\int t'} & \int^B T(\mathbf{A}(C, B) \otimes FB)
 \end{array}$$

Note that $I = \mathbf{A}(-, I) \in \mathcal{V}^{\mathbf{A}^{\text{op}}}$, and $\int t$ is a shorthand for $\int^B t_{\mathbf{A}(C, I \otimes B), FB}$, and $\int \lambda_B$ is a shorthand for $\int^B \mathbf{A}(C, \lambda_B) \otimes \text{id}_{TFB}$, and $\int T\lambda_B$ is a shorthand for $\int^B T\mathbf{A}(C, \lambda_B) \otimes \text{id}_{FB}$, and $\int t'$ is a shorthand for $\int^B t_{\mathbf{A}(C, B), FB}$.

The bottom square commutes because of Theorem F.5(c). The right square commutes by the naturality of ξ .

- Next we need to show that the following diagram commutes.

$$\begin{array}{ccc}
 F \otimes_{\text{Day}} \bar{T}TG & \xrightarrow{t} & \bar{T}(F \otimes_{\text{Day}} \bar{T}G) \xrightarrow{\bar{T}t} \bar{T}\bar{T}(F \otimes_{\text{Day}} G) \\
 \text{id}_F \otimes_{\text{Day}} \mu \downarrow & & \downarrow \mu \\
 F \otimes_{\text{Day}} \bar{T}G & \xrightarrow{t} & \bar{T}(F \otimes_{\text{Day}} G)
 \end{array}$$

The above diagram commutes because for any $C \in \mathbf{A}$, we have the following commutative diagram, where $X = \mathbf{A}(C, A \otimes B)$.

$$\begin{array}{ccccc}
 \int^{A,B} X \otimes FA \otimes TTGB & \xrightarrow{\int t} & \int^{A,B} T(X \otimes FA \otimes TGB) & \xrightarrow{\xi} & T \int^{A,B} (X \otimes FA \otimes TGB) \\
 \int \text{id} \otimes \mu \downarrow & & \downarrow \int Tt & & \downarrow T \int t \\
 \int^{A,B} X \otimes FA \otimes TGB & & \int^{A,B} TT(X \otimes FA \otimes GB) & \xrightarrow{\xi} & T \int^{A,B} T(X \otimes FA \otimes GB) \\
 \int t \downarrow & \swarrow \int \mu & & & \downarrow T\xi \\
 \int^{A,B} T(X \otimes FA \otimes GB) & \xrightarrow{\xi} & T \int^{A,B} (X \otimes FA \otimes GB) & \xleftarrow{\mu} & TT \int^{A,B} (X \otimes FA \otimes GB)
 \end{array}$$

Note that the top right square commutes by naturality of ξ , the bottom diagram commutes by Theorem F.5 (b), and the left diagram commutes by properties of t .

- Next we need to show that the following diagram commutes.

$$\begin{array}{ccc}
 (F \otimes_{\text{Day}} G) \otimes_{\text{Day}} \bar{T}H & \xrightarrow{t} & \bar{T}((F \otimes_{\text{Day}} G) \otimes_{\text{Day}} H) \\
 \alpha \downarrow & & \downarrow \bar{T}\alpha \\
 F \otimes_{\text{Day}} (G \otimes_{\text{Day}} \bar{T}H) & \xrightarrow{\text{id}_F \otimes_{\text{Day}} t} F \otimes_{\text{Day}} \bar{T}(G \otimes_{\text{Day}} H) \xrightarrow{t} \bar{T}(F \otimes_{\text{Day}} (G \otimes_{\text{Day}} H))
 \end{array}$$

For any $C \in \mathbf{A}$, we have

$$((F \otimes_{\text{Day}} G) \otimes_{\text{Day}} \bar{T}H)(C) \cong \int^{B \in \mathbf{A}} \int^{(X,Y) \in \mathbf{A} \otimes \mathbf{A}} \mathbf{A}(C, (X \otimes Y) \otimes B) \otimes FX \otimes GY \otimes THB$$

and

$$\begin{aligned}
 (F \otimes_{\text{Day}} (G \otimes_{\text{Day}} \bar{T}H))(C) &\cong \int^{X \in \mathbf{A}} \int^{(Y,B) \in \mathbf{A} \otimes \mathbf{A}} \mathbf{A}(C, X \otimes (Y \otimes B)) \otimes FX \otimes GY \otimes THB \\
 &\cong \int^{X,Y,B} FX \otimes GY \otimes THB \otimes \int^A \mathbf{A}(A, Y \otimes B) \otimes \mathbf{A}(C, X \otimes A).
 \end{aligned}$$

Consider the following diagram. We need to show that the outermost diagram commutes. Note that $\int y, T \int y, a, a'$ are all isomorphisms. For space reasons, we have written AB for $A \otimes B$ etc.

$$\begin{array}{c}
 \int^{B,X,Y} \mathbf{A}(C, (XY)B) \otimes FX \otimes GY \otimes THB \\
 \swarrow \int \alpha \quad \searrow \int t \\
 \int^{B,X,Y} \mathbf{A}(C, X(YB)) \otimes FX \otimes GY \otimes THB \quad \int^{B,X,Y} T(\mathbf{A}(C, (XY)B)) \otimes FX \otimes GY \otimes HB \\
 \downarrow \cong \quad \searrow \int t \quad \swarrow \int T\alpha \quad \searrow \xi \\
 \int^{A,X} \mathbf{A}(C, XA) \otimes FX \otimes \int^{Y,B} \mathbf{A}(A, YB) \otimes GY \otimes THB \quad T \int^{B,X,Y} \mathbf{A}(C, (XY)B) \otimes FX \otimes GY \otimes HB \\
 \downarrow \int \text{id} \otimes \int t \quad \searrow \xi \\
 \int^{A,X} \mathbf{A}(C, XA) \otimes FX \otimes \int^{Y,B} T(\mathbf{A}(A, YB)) \otimes GY \otimes HB \quad T \int^{B,X,Y} \mathbf{A}(C, X(YB)) \otimes FX \otimes GY \otimes HB \\
 \downarrow \int \text{id} \otimes \xi \quad \searrow \cong \\
 \int^{A,X} \mathbf{A}(C, XA) \otimes FX \otimes T \int^{Y,B} \mathbf{A}(A, YB) \otimes GY \otimes HB \quad T \int^{X,Y,B} \int^A \mathbf{A}(C, XA) \otimes \mathbf{A}(A, YB) \otimes FX \otimes GY \otimes HB \\
 \downarrow \int t \quad \searrow \cong \\
 \int^{A,X} T(\mathbf{A}(C, XA)) \otimes FX \otimes \int^{Y,B} \mathbf{A}(A, YB) \otimes GY \otimes HB \quad T \int^{A,X} \mathbf{A}(C, XA) \otimes FX \otimes \int^{Y,B} \mathbf{A}(A, YB) \otimes GY \otimes HB
 \end{array}$$

Note that the top square and the top right square commute by naturality of t and ξ . We just need to show that the bottom diagram commutes. The expanded bottom diagram is shown in Figure 7. Note that all the inner diagrams commute, by Theorem P.3(d)–(f) and naturality. Therefore the outer diagram commutes.

- Lastly, since $\mathcal{V}^{\mathbf{A}^{\text{op}}}$ is a symmetric monoidal $\mathcal{V}$ -category with $\gamma_{F,G} : F \otimes_{\text{Day}} G \rightarrow G \otimes_{\text{Day}} F$, we can define the costrength as the following for any $F, B \in \mathcal{V}^{\mathbf{A}^{\text{op}}}$.

$$\sigma_{F,G} := \bar{T}\gamma_{G,F} \circ t_{G,F} \circ \gamma_{\bar{T}F,G} : \bar{T}F \otimes_{\text{Day}} G \rightarrow \bar{T}(F \otimes_{\text{Day}} G)$$

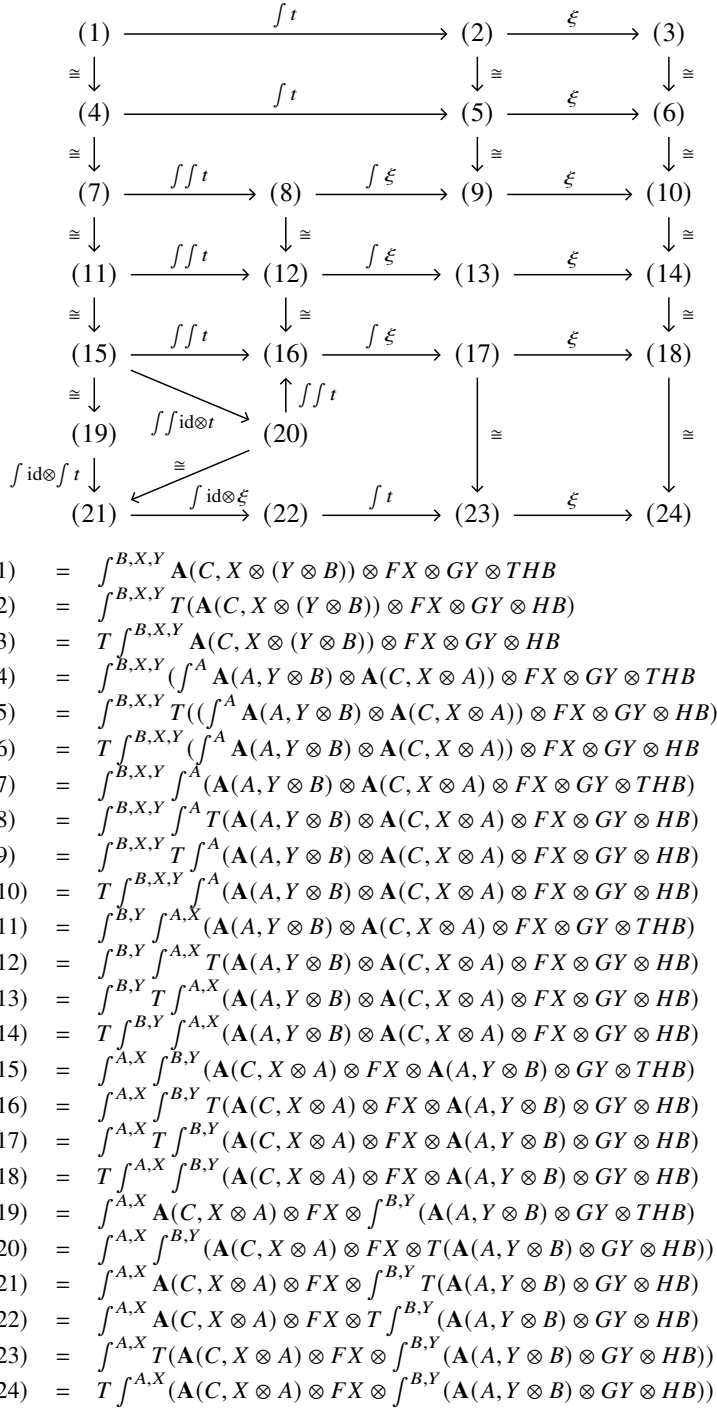


Fig. 7. A diagram used in the proof of Theorem 6.10. The objects (1) – (24) are as shown.

We need to show that the following diagram commutes.

$$\begin{array}{ccccc}
 & \bar{T}F \otimes_{\text{Day}} \bar{T}G & & & \\
 \swarrow \sigma & & \searrow t & & \\
 \bar{T}(F \otimes_{\text{Day}} \bar{T}G) & & & & \bar{T}(\bar{T}F \otimes_{\text{Day}} G) \\
 \bar{T}t \downarrow & & & & \downarrow \bar{T}\sigma \\
 \bar{TT}(F \otimes_{\text{Day}} G) & & & & \bar{TT}(F \otimes_{\text{Day}} G) \\
 \searrow \mu & & \swarrow \mu & & \\
 & \bar{T}(F \otimes_{\text{Day}} G) & & &
 \end{array}$$

For any $C \in \mathbf{A}$, the above diagram can be expanded to the following.

$$\begin{array}{c}
 \int^{A,B} \mathbf{A}(C, A \otimes B) \otimes TFA \otimes TGB \\
 \swarrow \int \gamma \quad \downarrow \int t \\
 \int^{A,B} \mathbf{A}(C, B \otimes A) \otimes TGB \otimes TFA \quad \downarrow \int t \\
 \downarrow \int t \quad \int^{A,B} T(\mathbf{A}(C, A \otimes B) \otimes TFA \otimes GB) \\
 \downarrow \int t \quad \downarrow \int T\gamma \quad \searrow \xi \\
 \int^{A,B} T(\mathbf{A}(C, B \otimes A) \otimes TGB \otimes FA) \quad \downarrow \int T\gamma \quad T \int^{A,B} (\mathbf{A}(C, A \otimes B) \otimes TFA \otimes GB) \\
 \swarrow \xi \quad \downarrow \int T\gamma \quad \downarrow \int Tt \quad \searrow \xi \quad \downarrow T \int \gamma \\
 T \int^{A,B} (\mathbf{A}(C, B \otimes A) \otimes TGB \otimes FA) \quad \downarrow \int T\gamma \quad \int^{A,B} T(\mathbf{A}(C, B \otimes A) \otimes GB \otimes TFA) \quad \downarrow T \int \gamma \\
 \downarrow T \int \gamma \quad \downarrow \int Tt \quad \downarrow \int Tt \quad \searrow \xi \quad \downarrow T \int t \\
 T \int^{A,B} (\mathbf{A}(C, A \otimes B) \otimes FA \otimes TGB) \quad \downarrow \int Tt \quad \int^{A,B} TT(\mathbf{A}(C, B \otimes A) \otimes GB \otimes FA) \quad \downarrow T \int t \\
 \downarrow T \int t \quad \downarrow \int \mu \quad \downarrow \int \mu \quad \searrow \xi \quad \downarrow T \int \gamma \\
 T \int^{A,B} TT(\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) \quad \downarrow \int \mu \quad T \int^{A,B} T(\mathbf{A}(C, B \otimes A) \otimes GB \otimes FA) \\
 \swarrow \xi \quad \downarrow \int T\gamma \quad \downarrow T \xi \\
 T \int^{A,B} T(\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) \quad \downarrow \int T\gamma \quad \int^{A,B} T(\mathbf{A}(C, B \otimes A) \otimes GB \otimes FA) \quad \downarrow T \xi \\
 \downarrow T \xi \quad \downarrow \int \mu \quad \downarrow \int \mu \quad \searrow \xi \quad \downarrow TT \int \gamma \\
 TT \int^{A,B} (\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) \quad \downarrow \int \mu \quad T \int^{A,B} (\mathbf{A}(C, B \otimes A) \otimes GB \otimes FA) \quad \downarrow TT \int \gamma \\
 \searrow \mu \quad \downarrow T \int \gamma \quad \downarrow \mu \\
 T \int^{A,B} (\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB) \quad \xleftarrow{\mu} \quad TT \int^{A,B} (\mathbf{A}(C, A \otimes B) \otimes FA \otimes GB)
 \end{array}$$

This commutes because every inner diagram commutes (by naturality and Theorem F.3(b)). $\square$

G Proof of Theorem 6.11

Notations. Let $\mathbf{A}, \mathbf{B}$ be $\mathcal{V}$ -categories. For all $A, B \in \mathbf{A}$, we have

$$\mathbf{A}(A, B) = (\mathbf{A}(A, B)_0, \mathbf{A}(A, B)_1, \varphi^A : \mathbf{A}(A, B)_1 \rightarrow \mathbf{A}(A, B)_0).$$

So we write $A \rightarrow_1 B := \mathbf{A}(A, B)_1$ and $A \rightarrow_0 B := \mathbf{A}(A, B)_0$. Moreover, for all $f : A \rightarrow_1 B$, we have $\varphi^A(f) : A \rightarrow_0 B$. A $\mathcal{V}$ -functor $F : \mathbf{A} \rightarrow \mathbf{B}$ gives rise to the following commutative diagram for all $A, B \in \mathbf{A}$.

$$\begin{array}{ccc}
\mathbf{A}(A, B)_1 & \xrightarrow{F_{A,B}^1} & \mathbf{B}(FA, FB)_1 \\
\downarrow \varphi^A & & \downarrow \varphi^B \\
\mathbf{A}(A, B)_0 & \xrightarrow{F_{A,B}^0} & \mathbf{B}(FA, FB)_0
\end{array}$$

For all $f : A \rightarrow_1 B$, we have $F_{A,B}^1 f : FA \rightarrow_1 FB$. Similarly, for all $g : A \rightarrow_0 B$, we have $F_{A,B}^0 g : FA \rightarrow_0 FB$.

For any $\mathcal{V}$ -functors $F, G : \mathbf{A} \rightarrow \mathbf{B}$, we define a biset $(F \Rightarrow_0 G, F \Rightarrow_1 G, p : F \Rightarrow_1 G \rightarrow F \Rightarrow_0 G)$ as follows.

$$F \Rightarrow_0 G := \{(\beta_A : FA \rightarrow_0 GA)_{A \in \mathbf{A}} \mid \forall A, B \in \mathbf{A}, \forall g : A \rightarrow_0 B, \beta_B \circ F_{AB}^0 g = G_{AB}^0 g \circ \beta_A\}$$

$$\begin{aligned}
F \Rightarrow_1 G := \{(\alpha_A : FA \rightarrow_1 GA)_{A \in \mathbf{A}} \mid \forall A, B \in \mathbf{A}, \forall f : A \rightarrow_1 B, \alpha_B \circ F_{AB}^1 f = G_{AB}^1 f \circ \alpha_A, \\
\forall A, B \in \mathbf{A}, \forall g : A \rightarrow_0 B, \varphi^B(\alpha_B) \circ F_{AB}^0 g = G_{AB}^0 g \circ \varphi^B(\alpha_A)\}
\end{aligned}$$

$$p((\alpha_A : FA \rightarrow_1 GA)_{A \in \mathbf{A}}) := (\varphi^B(\alpha_A) : FA \rightarrow_0 GA)_{A \in \mathbf{A}} : F \Rightarrow_1 G \rightarrow F \Rightarrow_0 G$$

Theorem G.1. *Suppose $\mathbf{A}, \mathbf{B}$ are $\mathcal{V}$ -categories. Since the category of bisets $\mathcal{V}$ is complete, the functor category $\mathbf{B}^{\mathbf{A}}$ is $\mathcal{V}$ -enriched. For all $\mathcal{V}$ -functors $F, G : \mathbf{A} \rightarrow \mathbf{B}$, we have*

$$\mathbf{B}^{\mathbf{A}}(F, G) := \int_{A \in \mathbf{A}} \mathbf{B}(FA, GA) \cong (F \Rightarrow_0 G, F \Rightarrow_1 G, p : F \Rightarrow_1 G \rightarrow F \Rightarrow_0 G)$$

PROOF. By definition of end, we have the following equalizer diagram in $\mathcal{V}$.

$$\int_{A \in \mathbf{A}} \mathbf{B}(FA, GA) := eq(u, v) \xrightarrow{k} \prod_{A \in \mathbf{A}} \mathbf{B}(FA, GA) \xrightarrow[u]{u} \prod_{A, B \in \mathbf{A}} \mathbf{A}(A, B) \Rightarrow \mathbf{B}(FA, GB)$$

Note that $u = \langle \text{curry}(c \circ (\pi_A \times G_{AB})) \rangle_{A, B \in \mathbf{A}}$, where $c \circ (\pi_A \times G_{AB})$ is the following.

$$(\prod_A \mathbf{B}(FA, GA)) \times \mathbf{A}(A, B) \xrightarrow{\pi_A \times G_{AB}} \mathbf{B}(FA, GA) \times \mathbf{B}(GA, GB) \xrightarrow{c} \mathbf{B}(FA, GB)$$

We have $v = \langle \text{curry}(c \circ (\pi_B \times F_{AB})) \rangle_{A, B \in \mathbf{A}}$, where $c \circ (\pi_B \times F_{AB})$ is the following.

$$(\prod_A \mathbf{B}(FA, GA)) \times \mathbf{A}(A, B) \xrightarrow{\pi_B \times F_{AB}} \mathbf{B}(FB, GB) \times \mathbf{B}(FA, FB) \xrightarrow{c} \mathbf{B}(FA, GB)$$

We can show $(\int_{A \in \mathbf{A}} \mathbf{B}(FA, GA))_1 = eq(u_1, v_1) \cong F \Rightarrow_1 G$ and $(\int_{A \in \mathbf{A}} \mathbf{B}(FA, GA))_0 = eq(u_0, v_0) \cong F \Rightarrow_0 G$. $\square$

THEOREM 6.11. *The biset-enriched categories $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ and $\mathbf{Set}^{\mathbf{Q}^{\text{op}}}$ are isomorphic.*

PROOF. Let us define a $\mathcal{V}$ -enriched functor $\Omega : \mathbf{Set}^{\mathbf{C}^{\text{op}}} \rightarrow \mathbf{Set}^{\mathbf{Q}^{\text{op}}}$. On objects, $\Omega(F) = F^0$ for any $F \in \mathbf{Set}^{\mathbf{C}^{\text{op}}}$. Since $F : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$ is uniquely determined by F^0 , the function Ω is bijective on objects.

Suppose $F, G : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$. We claim that $\mathbf{Set}^{\mathbf{C}^{\text{op}}}(F, G) \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0)$. This will allow us to define $\Omega_{F, G}$ to be this isomorphism. To show $\mathbf{Set}^{\mathbf{C}^{\text{op}}}(F, G) \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0)$, first of all, we have

$$\mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0) = (X, X, \text{id}),$$

where

$$X = \{(\alpha_A : F^0 A \rightarrow G^0 A)_{A \in \mathbf{Q}} \mid \forall A, B \in \mathbf{Q}, \forall f : A \rightarrow B \in \mathbf{Q}, \alpha_B \circ F_{AB}^0 f = G_{AB}^0 f \circ \alpha_A\}.$$

Next,

$$\mathbf{Set}^{\mathbf{C}^{\text{op}}}(F, G) = (F \Rightarrow_1 G, F \Rightarrow_0 G, p),$$

where

$$F \Rightarrow_0 G = \{(\alpha_A : FA \rightarrow_0 GA)_{A \in \mathbf{C}} \mid \forall A, B \in \mathbf{C}, \forall f : A \rightarrow_0 B \in \mathbf{C}, \alpha_B \circ F_{AB}^0 f = G_{AB}^0 f \circ \alpha_A\} \\ \cong X$$

and

$$F \Rightarrow_1 G := \{(\alpha_A : FA \rightarrow_1 GA)_{A \in \mathbf{A}} \mid \forall A, B \in \mathbf{C}, \forall f : A \rightarrow_1 B, \alpha_B \circ F_{AB}^1 f = G_{AB}^1 f \circ \alpha_A, \\ \forall A, B \in \mathbf{C}, \forall g : A \rightarrow_0 B, \varphi^{\mathbf{Set}}(\alpha_B) \circ F_{AB}^0 g = G_{AB}^0 g \circ \varphi^{\mathbf{Set}}(\alpha_A)\}.$$

Since $F_{A,B}^1 = F_{A,B}^0 \circ \varphi^{\mathbf{C}^{\text{op}}}$, and $\varphi^{\mathbf{Set}} = \text{id}$, and $\varphi^{\mathbf{C}^{\text{op}}}(f) : A \rightarrow_0 B$ for any $f : A \rightarrow_1 B$ with $A, B \in \mathbf{C}$, therefore $\forall A, B \in \mathbf{C}, \forall g : A \rightarrow_0 B, \varphi^{\mathbf{Set}}(\alpha_B) \circ F_{AB}^0 g = G_{AB}^0 g \circ \varphi^{\mathbf{Set}}(\alpha_A)$ implies $\forall A, B \in \mathbf{C}, \forall f : A \rightarrow_1 B, \alpha_B \circ F_{AB}^1 f = G_{AB}^1 f \circ \alpha_A$. So $F \Rightarrow_1 G \cong F \Rightarrow_0 G \cong X$ and $p = \text{id}$. $\square$

H Proof of Theorem 6.16

We write $[\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ for the full subcategory of $\mathbf{Set}^{\mathbf{Q}^{\text{op}}}$ consisting of product-preserving functors. We write $L : \mathbf{Set}^{\mathbf{Q}^{\text{op}}} \rightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$ for the left adjoint of the inclusion functor $i : [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}} \hookrightarrow \mathbf{Set}^{\mathbf{Q}^{\text{op}}}$. We write $\eta : \text{id} \rightarrow iL$ for the unit of the adjunction.

Definition H.1. We define a function $K : \overline{\mathbf{C}} \rightarrow \tilde{\mathbf{C}}$ as follows.

- For any $F \in \overline{\mathbf{C}}$, we define $K(F) = G$ such that for all $A \in \mathbf{C}$,

$$G(A) = ((iLF^0)A, (FA)_1, (\eta_{F^0})_A \circ h_A : (FA)_1 \rightarrow (iLF^0)A) \in \mathcal{V},$$

where $h_A : (FA)_1 \rightarrow (FA)_0$ and $(\eta_{F^0})_A : F^0 A \rightarrow (iLF^0)A$.

For any $A, B \in \mathbf{C}$, we define G_{AB}^0 and G_{AB}^1 by the following.

$$\begin{array}{ccc} f \in \mathbf{M}^{\text{op}}(A, B) & \xrightarrow{G_{AB}^1} & (F_{AB}^1 f, (iLF^0)(J_{AB} f)) \in \mathcal{V}(GA, GB) \\ \downarrow J_{AB} & & \downarrow p_0 \\ J_{AB} f \in \mathbf{Q}^{\text{op}}(A, B) & \xrightarrow{G_{AB}^0} & (iLF^0)(J_{AB} f) \in \mathbf{Set}((iLF^0)A, (iLF^0)B) \end{array}$$

Note that G is smooth, hence $G \in \tilde{\mathbf{C}}$.

Proposition H.2. $K : \overline{\mathbf{C}} \rightarrow \tilde{\mathbf{C}}$ is a $\mathcal{V}$ -functor.

PROOF. We just need to show that for all $F, G \in \overline{\mathbf{C}}$, there is a morphism

$$K_{FG} : \overline{\mathbf{C}}(F, G) \rightarrow \tilde{\mathbf{C}}(KF, KG)$$

in $\mathcal{V}$. This is provided by the following commuting square in $\mathbf{Set}$.

$$\begin{array}{ccc} \mathcal{V}\text{-Nat}(F, G) & \xrightarrow{K_{FG}^1} & \mathcal{V}\text{-Nat}(iKF, iKG) \\ \downarrow & & \downarrow \\ \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, G^0) & \xrightarrow{K_{FG}^0} & [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(LF^0, LG^0) \end{array}$$

We write $\mathcal{V}\text{-Nat}(F, G)$ for the set of $\mathcal{V}$ -natural transformations from F to G . The arrow K_{FG}^0 is given by the functor $L : \mathbf{Set}^{\mathbf{Q}^{\text{op}}} \rightarrow [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}$. And the arrow K_{FG}^1 is given by extending

the commuting square $\alpha_A : FA \rightarrow GA$ with $\eta : \text{id} \rightarrow iL$, as in the following diagram. Note that for each $\mathcal{V}$ -natural transformation $\alpha \in \mathcal{V}\text{Nat}(F, G)$, we have $\alpha^0 : F^0 \rightarrow G^0$.

$$\begin{array}{ccc}
 (FA)_1 & \xrightarrow{\alpha_A^1} & (GA)_1 \\
 \downarrow & & \downarrow \\
 (FA)_0 & \xrightarrow{\alpha_A^0} & (GA)_0 \\
 (\eta_{F^0})_A \downarrow & & \downarrow (\eta_{G^0})_A \\
 (iLF^0)_A & \xrightarrow{(iL)\alpha_A^0} & (iLG^0)_A
 \end{array}$$

□

THEOREM 6.16. *The $\mathcal{V}$ -category $\tilde{\mathbf{C}}$ is a reflective $\mathcal{V}$ -subcategory of $\overline{\mathbf{C}}$, i.e., the inclusion $\mathcal{V}$ -functor $i : \tilde{\mathbf{C}} \hookrightarrow \overline{\mathbf{C}}$ has a left adjoint K .*

PROOF. K was defined in Definition H.1. We need to show $\overline{\mathbf{C}}(F, iG) \cong \tilde{\mathbf{C}}(KF, G)$ for any $F \in \overline{\mathbf{C}}, G \in \tilde{\mathbf{C}}$ and it is $\mathcal{V}$ -natural in F and G . We just need to show the following diagram commutes.

$$\begin{array}{ccc}
 \mathcal{V}\text{-Nat}(F, iG) & \xrightarrow{\cong} & \mathcal{V}\text{-Nat}(iKF, iG) \\
 \downarrow & & \downarrow \\
 \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(F^0, iG^0) & \xrightarrow{\cong} & [\mathbf{Set}^{\mathbf{Q}^{\text{op}}}]_{\text{prod}}(LF^0, G^0)
 \end{array}$$

The bottom arrow is an isomorphism because $L \dashv i$. The top arrow is an isomorphism because for any $A \in \mathbf{C}$ and $\mathcal{V}$ -natural transformation $\gamma : F \rightarrow iG$, we have the following commutative diagram.

$$\begin{array}{ccc}
 (FA)_1 & \xrightarrow{\gamma_A^1} & ((iG)A)_1 \\
 \downarrow & & \downarrow \\
 (FA)_0 & \xrightarrow{\gamma_A^0} & ((iG)A)_0 \\
 (\eta_{F^0})_A \downarrow & \nearrow i(\gamma^0)_A & \\
 (iLF^0)_A & &
 \end{array}$$

□

I Day's reflection theorem for $\tilde{\mathbf{C}}$

Theorem I.1. *If $H \in \tilde{\mathbf{C}}$, then $G \multimap_{\text{Day}} iH$ is also a smooth functor for any $G \in \overline{\mathbf{C}}$.*

PROOF. Suppose $H \in \tilde{\mathbf{C}}$ and $G \in \overline{\mathbf{C}}$. For any $C \in \mathbf{C}$, we have

$$(G \multimap_{\text{Day}} iH)(C) = \int_{A \in \mathbf{C}} GA \Rightarrow iH(C \otimes A) \cong \overline{\mathbf{C}}(G, iH(C \otimes -)).$$

Thus

$$\begin{aligned}
 (G \multimap_{\text{Day}} iH)(C)_0 &\cong \overline{\mathbf{C}}(G, iH(C \otimes -))_0 \cong \mathbf{Set}^{\mathbf{Q}^{\text{op}}}(G^0, (iH)^0(C \otimes -)) \\
 &\cong \int_{A \in \mathbf{Q}} \mathbf{Set}(G^0 A, (iH)^0(C \otimes A)) \cong (G^0 \multimap_{\text{Day}} (iH)^0)(C),
 \end{aligned}$$

where $G^0 \multimap_{\text{Day}} (iH)^0$ is an exponential in $\mathbf{Set}^{\mathbf{Q}^{\text{op}}}$. Since H^0 preserves products, so does $G^0 \multimap_{\text{Day}} (iH)^0$, thus $G \multimap_{\text{Day}} iH$ is smooth and $G \multimap_{\text{Day}} iH \in \tilde{\mathbf{C}}$. The functor $G^0 \multimap_{\text{Day}}$

$(iH)^0$ preserves products in $\mathbf{Q}^{\text{op}}$ because for any $C_1, C_2 \in \mathbf{C}$, we have

$$\begin{aligned}
 (G^0 \multimap_{\text{Day}} iH^0)(C_1 + C_2) &= \int_A \mathbf{Set}(G^0 A, iH^0((C_1 + C_2) \otimes A)) \\
 &\cong \int_A \mathbf{Set}(G^0 A, iH^0(C_1 \otimes A + C_2 \otimes A)) \\
 &\cong \int_A \mathbf{Set}(G^0 A, iH^0(C_1 \otimes A) \times iH^0(C_2 \otimes A)) \\
 &\cong \int_A \mathbf{Set}(G^0 A, iH^0(C_1 \otimes A)) \times \mathbf{Set}(G^0 A, iH^0(C_2 \otimes A)) \\
 &\cong \int_A \mathbf{Set}(G^0 A, iH^0(C_1 \otimes A)) \times \int_A \mathbf{Set}(G^0 A, iH^0(C_2 \otimes A)) \\
 &= (G^0 \multimap_{\text{Day}} iH^0)(C_1) \times (G^0 \multimap_{\text{Day}} iH^0)(C_2). \quad \square
 \end{aligned}$$

The above theorem implies that for any $G \in \tilde{\mathbf{C}}$, $F \in \overline{\mathbf{C}}$, the unit $\eta_{F \multimap_{\text{Day}} iG} : F \multimap_{\text{Day}} iG \rightarrow iK(F \multimap_{\text{Day}} iG)$ is an isomorphism, which gives rise to the following theorem.

Theorem I.2. *For any $F, H \in \overline{\mathbf{C}}$, we have*

$$K(F \otimes_{\text{Day}} H) \xrightarrow{K(\eta_{F \otimes_{\text{Day}} H})} K(iKF \otimes_{\text{Day}} H).$$

PROOF. For any $G \in \tilde{\mathbf{C}}$, we have the following commutative diagram.

$$\begin{array}{ccc}
 \tilde{\mathbf{C}}(K(iKF \otimes_{\text{Day}} H), G) & \xrightarrow{\tilde{\mathbf{C}}(K(\eta_{F \otimes_{\text{Day}} H}), G)} & \tilde{\mathbf{C}}(K(F \otimes_{\text{Day}} H), G) \\
 \downarrow \cong & & \downarrow \cong \\
 \overline{\mathbf{C}}(iKF \otimes_{\text{Day}} H, iG) & \xrightarrow{\overline{\mathbf{C}}(\eta_{F \otimes_{\text{Day}} H}, iG)} & \overline{\mathbf{C}}(F \otimes_{\text{Day}} H, iG) \\
 \downarrow \cong & & \downarrow \cong \\
 \overline{\mathbf{C}}(iKF, H \multimap_{\text{Day}} iG) & \xrightarrow{\overline{\mathbf{C}}(\eta_F, H \multimap_{\text{Day}} iG)} & \overline{\mathbf{C}}(F, H \multimap_{\text{Day}} iG) \\
 \downarrow \tilde{\mathbf{C}}(iKF, \eta_{H \multimap_{\text{Day}} iG}) & & \downarrow \tilde{\mathbf{C}}(F, \eta_{H \multimap_{\text{Day}} iG}) \\
 \overline{\mathbf{C}}(iKF, iK(H \multimap_{\text{Day}} iG)) & \xrightarrow{\overline{\mathbf{C}}(\eta_F, iK(H \multimap_{\text{Day}} iG))} & \tilde{\mathbf{C}}(F, iK(H \multimap_{\text{Day}} iG)) \\
 \downarrow \cong & \nearrow \text{id} & \\
 \tilde{\mathbf{C}}(KF, K(H \multimap_{\text{Day}} iG)) & & \\
 \downarrow \cong & & \\
 \overline{\mathbf{C}}(F, iK(H \multimap_{\text{Day}} iG)) & &
 \end{array}$$

The top two squares commute by naturality of the adjunctions, the third square commutes by the bi-functoriality of $\overline{\mathbf{C}}(-, -)$ and the bottom triangle commutes by properties of the adjunction $K \dashv i$. $\square$

With the help of Theorem I.2, one can verify that K is strong monoidal, e.g., for any $F, G \in \overline{\mathbf{C}}$, $KF \otimes_{\text{Lam}} KG = K(iKF \otimes_{\text{Day}} iKG) \cong K(F \otimes_{\text{Day}} G)$.

J Proof of Theorem 6.22

We write $\beta : i \circ \tilde{T} \rightarrow \overline{T} \circ i$ to denote the isomorphism $i \circ \tilde{T} \cong \overline{T} \circ i$.

Theorem J.1. *The following diagrams commute.*

(a) Suppose $F \in \tilde{\mathbf{C}}$.

$$\begin{array}{ccc}
 iF & \xrightarrow{i\eta^{\tilde{T}}} & i\tilde{T}F \\
 \searrow \eta^{\overline{T}} & & \downarrow \beta \\
 & & \overline{T}iF
 \end{array}$$

(b) Suppose $F \in \overline{\mathbf{C}}$.

$$\begin{array}{ccc} KF & \xrightarrow{K\eta^{\overline{T}}} & K\overline{T}F \\ & \searrow \eta^{\tilde{T}} & \downarrow \rho \\ & & \tilde{T}KF \end{array}$$

(c) Suppose $F \in \tilde{\mathbf{C}}$.

$$\begin{array}{ccccc} i\tilde{T}\tilde{T}F & \xrightarrow{\beta} & \overline{T}i\tilde{T}F & \xrightarrow{\overline{T}\beta} & \overline{T}\overline{T}iF \\ \downarrow i\mu^{\tilde{T}} & & & & \downarrow \mu^{\overline{T}} \\ i\tilde{T}F & \xrightarrow{\beta} & \overline{T}iF & & \end{array}$$

(d) Suppose $F \in \overline{\mathbf{C}}$.

$$\begin{array}{ccccc} K\overline{T}\overline{T}F & \xrightarrow{\rho} & \tilde{T}K\overline{T}F & \xrightarrow{\tilde{T}\rho} & \tilde{T}\tilde{T}KF \\ \downarrow K\mu^{\overline{T}} & & & & \downarrow \mu^{\tilde{T}} \\ K\overline{T}F & \xrightarrow{\rho} & \tilde{T}KF & & \end{array}$$

(e) Suppose $F \in \tilde{\mathbf{C}}$.

$$\begin{array}{ccc} Ki\tilde{T}F & \xrightarrow{K\beta} & K\overline{T}iF \\ \downarrow \epsilon & & \downarrow \rho \\ \tilde{T}F & \xleftarrow{\tilde{T}\epsilon} & \tilde{T}KiF \end{array}$$

(f) Suppose $F \in \overline{\mathbf{C}}, G \in \tilde{\mathbf{C}}$.

$$\begin{array}{ccccc} K\overline{T}(i\tilde{T}G \otimes_D F) & \xrightarrow{K\overline{T}(\beta \otimes_D F)} & K\overline{T}(\overline{T}iG \otimes_D F) & \xrightarrow{K\overline{T}\gamma} & K\overline{T}(F \otimes_D \overline{T}iG) \\ \downarrow \rho & & & & \downarrow \rho \\ \tilde{T}K(i\tilde{T}G \otimes_D F) & \xrightarrow{\tilde{T}K\gamma} & \tilde{T}K(F \otimes_D i\tilde{T}G) & \xrightarrow{\tilde{T}K(F \otimes_D \beta)} & \tilde{T}K(F \otimes_D \overline{T}iG) \end{array}$$

(g) Suppose $F, G \in \overline{\mathbf{C}}$.

$$\begin{array}{ccccccc} K(F \otimes_D \overline{T}G) & \xrightarrow{K\tilde{i}} & K\overline{T}(F \otimes_D G) & \xrightarrow{\rho} & \tilde{T}K(F \otimes_D G) & \xrightarrow{\tilde{T}K(F \otimes_D \eta^{iK})} & \tilde{T}K(F \otimes_D iKG) \\ \downarrow K(F \otimes_D \eta^{iK}) & & & & & & \uparrow \rho \\ K(F \otimes_D iK\overline{T}G) & \xrightarrow{K(F \otimes_D i\rho)} & K(F \otimes_D i\tilde{T}KG) & \xrightarrow{K(F \otimes_D \beta)} & K(F \otimes_D \overline{T}iKG) & \xrightarrow{K\tilde{i}} & K\overline{T}(F \otimes_D iKG) \end{array}$$

PROOF. (a) This is from definition of $\overline{T}$ and $\tilde{T}$. Moreover, we have

$$\begin{aligned} \overline{\mathbf{C}}(iF, \overline{T}iF) &= \overline{\mathbf{C}}(iF, \overline{\Delta U}_0 iF) \cong \mathbf{Set}^{\text{Cop}}(\overline{U}_0 iF, \overline{U}_0 iF) \cong \mathbf{Set}^{\text{Cop}}(j\tilde{U}_0 F, j\tilde{U}_0 F) \\ &\cong [\mathbf{Set}^{\text{Cop}}]_{\text{prod}}(\tilde{U}_0 F, \tilde{U}_0 F) \cong \tilde{\mathbf{C}}(F, \tilde{T}F) \cong \overline{\mathbf{C}}(iF, i\tilde{T}F). \end{aligned}$$

(b) If we unfold the definition of ρ , we have the following diagram.

$$\begin{array}{ccc} KF & \xrightarrow{K\eta^{\overline{T}}} & K\overline{T}F \\ \eta^{\tilde{T}} \downarrow & & \downarrow K\overline{T}\eta^{iK} \\ & & K\overline{T}iKF \\ & & \downarrow K\beta^{-1} \\ \tilde{T}KF & \xleftarrow{\epsilon} & Ki\tilde{T}KF \end{array}$$

Note that we have the following commutative diagram, by naturality and (a).

$$\begin{array}{ccccccc} F & \xrightarrow{\eta^{\overline{T}}} & \overline{T}F & \xrightarrow{\overline{T}\eta^{iK}} & \overline{T}iKF & \xrightarrow{\beta^{-1}} & i\tilde{T}KF \\ \eta^K \downarrow & \nearrow \eta^{\tilde{T}} & & & & \nearrow i\eta^{\tilde{T}} & \\ iKF & & & & & & \end{array}$$

Therefore we just need to show the following diagram commutes (and indeed it does).

$$\begin{array}{ccccc}
 KF & \xrightarrow{K\eta^{iK}} & KiKF & \xrightarrow{Ki\tilde{\eta}^T} & Ki\tilde{TK}F \\
 & \searrow \text{id} & \downarrow \epsilon & & \downarrow \epsilon \\
 & & KF & \xrightarrow{\eta^T} & \tilde{TK}F
 \end{array}$$

- (c) Since each arrow is an identity, the diagram commutes.
 (d) If we unfold the definition of ρ , we have the following diagram.

$$\begin{array}{ccccc}
 K\bar{T}\bar{T}F & \xrightarrow{K\bar{T}\eta^{iK}} & K\bar{T}iK\bar{T}F & \xrightarrow{K\beta^{-1}} & Ki\tilde{TK}\bar{T}F & \xrightarrow{\epsilon} & \tilde{TK}\bar{T}F \\
 \downarrow K\mu^T & & & & \downarrow Ki\tilde{TK}\bar{T}\eta^{iK} & & \downarrow \tilde{TK}\bar{T}\eta^{iK} \\
 K\bar{T}F & & & & Ki\tilde{TK}\bar{T}iKF & \xrightarrow{\epsilon} & \tilde{TK}\bar{T}iKF \\
 \downarrow K\bar{T}\eta^K & & & & \downarrow Ki\tilde{TK}\beta^{-1} & & \downarrow \tilde{TK}\beta^{-1} \\
 K\bar{T}iKF & & & & Ki\tilde{TK}i\tilde{TK}F & \xrightarrow{\epsilon} & \tilde{TK}i\tilde{TK}F \\
 \downarrow K\beta^{-1} & & & & \downarrow Ki\tilde{T}\epsilon & & \downarrow \tilde{T}\epsilon \\
 Ki\tilde{TK}F & \xleftarrow{Ki\mu^T} & Ki\tilde{T}\tilde{TK}F & \xrightarrow{\epsilon} & \tilde{T}\tilde{TK}F & & \\
 \downarrow \epsilon & & & & \swarrow \mu^T & & \\
 \tilde{TK}F & & & & & &
 \end{array}$$

All of the squares commute by naturality. We just need to show the left diagram commutes. It commutes because the following commutes.

$$\begin{array}{ccccc}
 \bar{T}\bar{T}F & \xrightarrow{\bar{T}\eta^{iK}} & \bar{T}iK\bar{T}F & \xrightarrow{\beta^{-1}} & i\tilde{TK}\bar{T}F \\
 \mu \downarrow & \searrow \bar{T}\eta^{iK} & \downarrow \bar{T}iK\bar{T}\eta^{iK} & & \downarrow iK\bar{T}\eta^{iK} \\
 \bar{T}F & \xrightarrow{\bar{T}\eta^{iK}} & \bar{T}\bar{T}iKF & \xrightarrow{\bar{T}\eta^{iK}} & \bar{T}iK\bar{T}iKF & \xrightarrow{\beta^{-1}} & i\tilde{TK}\bar{T}iKF \\
 \bar{T}\eta^{iK} \downarrow & \swarrow \mu & \downarrow \bar{T}\beta^{-1} & \downarrow \bar{T}iK\beta^{-1} & \downarrow \beta^{-1} & & \downarrow i\tilde{TK}\beta^{-1} \\
 \bar{T}iKF & \xrightarrow{\bar{T}\eta^{iK}} & \bar{T}i\tilde{TK}F & \xrightarrow{\bar{T}\eta^{iK}} & \bar{T}iKi\tilde{TK}F & \xrightarrow{\beta^{-1}} & i\tilde{TK}Ki\tilde{TK}F \\
 \beta^{-1} \downarrow & & \downarrow \beta^{-1} & & \downarrow \beta^{-1} & & \downarrow i\tilde{TK}\beta^{-1} \\
 i\tilde{TK}F & \xleftarrow{i\mu} & i\tilde{T}\tilde{TK}F & \xleftarrow{i\tilde{T}\epsilon} & i\tilde{TK}i\tilde{TK}F & &
 \end{array}$$

The above diagram commutes, because all the squares commutes by naturality. The bottom left corner diagram commutes by (c).

- (e) If we unfold the definition of ρ , we have the following diagram.

$$\begin{array}{ccc}
 Ki\tilde{TK}F & \xrightarrow{K\beta} & K\bar{T}iF \\
 K\beta \downarrow & & \downarrow K\bar{T}\eta^{iK} \\
 K\bar{T}iF & \xleftarrow{K\bar{T}i\epsilon} & K\bar{T}iKiF \\
 K\beta^{-1} \downarrow & & \downarrow K\beta^{-1} \\
 Ki\tilde{TK}F & \xleftarrow{Ki\tilde{T}\epsilon} & Ki\tilde{TK}iF \\
 \epsilon \downarrow & & \downarrow \epsilon \\
 \tilde{TK}F & \xleftarrow{\tilde{T}\epsilon} & \tilde{TK}iF
 \end{array}$$

The bottom two squares commute by naturality. The top square commutes because $i\epsilon \circ \eta^{iK} = \text{id}$.

(f) See the following commutative diagram.

$$\begin{array}{ccccc}
 K\bar{T}(i\tilde{T}G \otimes_D F) & \xrightarrow{K\bar{T}(\beta \otimes_D F)} & K\bar{T}(\bar{T}iG \otimes_D F) & \xrightarrow{K\bar{T}\gamma} & K\bar{T}(F \otimes_D \bar{T}iG) \\
 \downarrow \rho & & \downarrow \rho & & \downarrow \rho \\
 & \nearrow \tilde{T}K(\beta \otimes_D F) & \tilde{T}K(\bar{T}iG \otimes_D F) & \searrow \tilde{T}K\gamma & \\
 \tilde{T}K(i\tilde{T}G \otimes_D F) & \xrightarrow{\tilde{T}K\gamma} & \tilde{T}K(F \otimes_D i\tilde{T}G) & \xrightarrow{\tilde{T}K(F \otimes_D \beta)} & \tilde{T}K(F \otimes_D \bar{T}iG)
 \end{array}$$

(g) Consider the following diagram.

$$\begin{array}{ccccccc}
 K(F \otimes_D \bar{T}G) & \xrightarrow{K\bar{i}} & K\bar{T}(F \otimes_D G) & \xrightarrow{\rho} & \tilde{T}K(F \otimes_D G) & \xrightarrow{\tilde{T}K(F \otimes_D \eta^{iK})} & \tilde{T}K(F \otimes_D iKG) \\
 \downarrow K(F \otimes_D \eta^{iK}) & & \searrow K(F \otimes_D \bar{T}\eta^{iK}) & & \searrow K\bar{T}(F \otimes_D \eta^{iK}) & & \uparrow \rho \\
 K(F \otimes_D iK\bar{T}G) & \xrightarrow{K(F \otimes_D i\rho)} & K(F \otimes_D i\tilde{T}KG) & \xrightarrow{K(F \otimes_D \beta)} & K(F \otimes_D \bar{T}iKG) & \xrightarrow{K\bar{i}} & K\bar{T}(F \otimes_D iKG)
 \end{array}$$

Note that the very right diagram and the middle square commute by naturality. The left diagram commutes because the following diagram commutes (with ρ unfolded).

$$\begin{array}{ccccc}
 K(F \otimes_D \bar{T}G) & \xrightarrow{K(F \otimes_D \bar{T}\eta^{iK})} & K(F \otimes_D \bar{T}iKG) & & \\
 \downarrow K(F \otimes_D \eta^{iK}) & \searrow K(F \otimes_D \bar{T}\eta^{iK}) & \nearrow \text{id} & & \\
 K(F \otimes_D iK\bar{T}G) & & K(F \otimes_D \bar{T}iKG) & \xrightarrow{K(F \otimes_D \beta^{-1})} & K(F \otimes_D i\tilde{T}KG) \\
 \downarrow K(F \otimes_D iK\bar{T}\eta^{iK}) & \searrow K(F \otimes_D \eta^{iK}) & \nearrow K(F \otimes_D \eta^{iK}) & & \downarrow \text{id} \\
 K(F \otimes_D iK\bar{T}iKG) & \xrightarrow{K(F \otimes_D iK\beta^{-1})} & K(F \otimes_D iK\tilde{T}KG) & \xrightarrow{K(F \otimes_D i\epsilon)} & K(F \otimes_D i\tilde{T}KG)
 \end{array}$$

□

THEOREM 6.22. *The $\mathcal{V}$ -functor $\tilde{T}$ is a commutative strong monad. The strength is given by $\tilde{i}_{F,G} : F \otimes_{\text{Lam}} \tilde{T}G \rightarrow \tilde{T}(F \otimes_{\text{Lam}} G)$ for any $F, G \in \tilde{\mathbf{C}}$.*

PROOF. For any $F, G \in \tilde{\mathbf{C}}$, we define $\tilde{i}_{F,G}$ by the following composition.

$$\begin{aligned}
 F \otimes_{\text{Lam}} \tilde{T}G &= K(iF \otimes_{\text{Day}} i\tilde{T}G) \\
 &\stackrel{\beta}{\cong} K(iF \otimes_{\text{Day}} \bar{T}iG) \xrightarrow{\bar{i}} K(\bar{T}(iF \otimes_{\text{Day}} iG)) \xrightarrow{\rho} \tilde{T}(K(iF \otimes_{\text{Day}} iG)) = \tilde{T}(F \otimes_{\text{Lam}} G).
 \end{aligned}$$

Now we need to show $\tilde{T}$ is a commutative strong monad.

- We must show:

$$\begin{array}{ccc}
 F \otimes_{\text{Lam}} G & \xrightarrow{\text{id}_F \otimes_{\text{Lam}} \eta} & F \otimes_{\text{Lam}} \tilde{T}G \\
 \eta \downarrow & \swarrow \tilde{i} & \\
 \tilde{T}(F \otimes_{\text{Lam}} G) & &
 \end{array}$$

The above diagram commutes because the following diagram commutes (by properties of $\tilde{i}$ and Theorem J.1(a)+(b)).

$$\begin{array}{ccccc}
 K(iF \otimes_{\text{Day}} iG) & \xrightarrow{K(\text{id} \otimes_D i\eta^{\tilde{T}})} & K(iF \otimes_{\text{Day}} i\tilde{T}G) & \xrightarrow{K(\text{id} \otimes_D \beta)} & K(iF \otimes_{\text{Day}} \bar{T}iG) \\
 \downarrow \eta^{\tilde{T}} & \searrow K(\text{id} \otimes_D \eta^{\tilde{T}}) & & \searrow K\eta^{\tilde{T}} & \downarrow K\bar{i} \\
 \tilde{T}K(iF \otimes_{\text{Day}} iG) & & & & K\bar{T}(iF \otimes_{\text{Day}} iG) \\
 & \swarrow \rho & & & \\
 & & & &
 \end{array}$$

- We must show:

$$\begin{array}{ccc}
 F \otimes_{\text{Lam}} \tilde{T}\tilde{T}G & \xrightarrow{\tilde{i}} & \tilde{T}(F \otimes_{\text{Lam}} \tilde{T}G) \xrightarrow{\tilde{T}\tilde{i}} \tilde{T}\tilde{T}(F \otimes_{\text{Lam}} G) \\
 \downarrow F \otimes_{\text{Lam}} \mu^{\tilde{T}} & & \downarrow \mu^{\tilde{T}} \\
 F \otimes_{\text{Lam}} \tilde{T}G & \xrightarrow{\tilde{i}} & \tilde{T}(F \otimes_{\text{Lam}} G)
 \end{array}$$

The above diagram commutes because the following diagram commutes (all the inner diagrams commute, by naturality, properties of $\tilde{i}$, and Theorem J.1(c)+(d)).

$$\begin{array}{ccccc}
 K(iF \otimes_D i\tilde{T}\tilde{T}G) & \xrightarrow{K(\text{id} \otimes_D \beta)} & K(iF \otimes_D \bar{T}i\tilde{T}G) & \xrightarrow{K\bar{i}} & K\bar{T}(iF \otimes_D i\tilde{T}G) \\
 \downarrow K(\text{id} \otimes_D i\mu^{\tilde{T}}) & & \downarrow K(\text{id} \otimes_D \bar{T}\beta) & & \downarrow K\bar{T}(\text{id} \otimes_D \beta) \\
 K(iF \otimes_D i\tilde{T}G) & & K(iF \otimes_D \bar{T}\bar{T}iG) & & \tilde{T}K(iF \otimes_D i\tilde{T}G) \\
 \downarrow K(\text{id} \otimes_D \beta) & \swarrow K(\text{id} \otimes_D \mu^{\tilde{T}}) & \downarrow K\bar{i} & \swarrow \rho & \downarrow \tilde{T}K(\text{id} \otimes_D \beta) \\
 K(iF \otimes_D \bar{T}iG) & & K\bar{T}(iF \otimes_D \bar{T}iG) & \xrightarrow{\rho} & \tilde{T}K(iF \otimes_D \bar{T}iG) \\
 \downarrow K\bar{i} & & \downarrow K\bar{T}\bar{i} & & \downarrow \tilde{T}K\bar{i} \\
 K\bar{T}(iF \otimes_D iG) & \xleftarrow{K\mu^{\tilde{T}}} & K\bar{T}\bar{T}(iF \otimes_D iG) & \xrightarrow{\rho} & \tilde{T}K\bar{T}(iF \otimes_D iG) \\
 \downarrow \rho & & & & \downarrow \tilde{T}\rho \\
 \tilde{T}K(iF \otimes_D iG) & \xleftarrow{\mu^{\tilde{T}}} & \tilde{T}\tilde{T}K(iF \otimes_D iG) & &
 \end{array}$$

- We must show:

$$\begin{array}{ccc}
 (F \otimes_{\text{Lam}} G) \otimes_{\text{Lam}} \tilde{T}H & \xrightarrow{\tilde{i}} & \tilde{T}((F \otimes_{\text{Lam}} G) \otimes_{\text{Lam}} H) \\
 \downarrow \alpha & & \downarrow \tilde{T}\alpha \\
 F \otimes_{\text{Lam}} (G \otimes_{\text{Lam}} \tilde{T}H) & \xrightarrow{\text{id}_F \otimes_{\text{Lam}} \tilde{i}} F \otimes_{\text{Lam}} \tilde{T}(G \otimes_{\text{Lam}} H) \xrightarrow{\tilde{i}} \tilde{T}(F \otimes_{\text{Lam}} (G \otimes_{\text{Lam}} H))
 \end{array}$$

Again, the above diagram commutes because the following diagram commutes (all of the inner diagrams commute by naturality, properties of $\tilde{i}$, and Theorem J.1(g)).

$$\begin{array}{c}
K(iK(iF \otimes_D iG) \otimes_D i\tilde{T}H) \xrightarrow{K(\text{id} \otimes_D \beta)} K(iK(iF \otimes_D iG) \otimes_D \bar{T}iH) \xrightarrow{K\bar{i}} K\bar{T}(iK(iF \otimes_D iG) \otimes_D iH) \\
\downarrow (K(\eta^{iK} \otimes_D \text{id}))^{-1} \quad \downarrow (K(\eta^{iK} \otimes_D \text{id}))^{-1} \quad \downarrow \rho \\
K((iF \otimes_D iG) \otimes_D i\tilde{T}H) \xrightarrow{K(\text{id} \otimes_D \beta)} K((iF \otimes_D iG) \otimes_D \bar{T}iH) \xrightarrow{K\bar{i}} \tilde{T}K(iK(iF \otimes_D iG) \otimes_D iH) \\
\downarrow K\alpha \quad \downarrow K\alpha \quad \downarrow K\bar{T}((iF \otimes_D iG) \otimes_D iH) \quad \downarrow \tilde{T}(K(\eta^{iK} \otimes_D \text{id}))^{-1} \\
K(iF \otimes_D iG \otimes_D i\tilde{T}H) \xrightarrow{K(\text{id} \otimes_D (\text{id} \otimes_D \beta))} K(iF \otimes_D (iG \otimes_D \bar{T}iH)) \quad \downarrow \tilde{T}K((iF \otimes_D iG) \otimes_D iH) \\
\downarrow K(\text{id} \otimes_D \eta^{iK}) \quad \downarrow K(\text{id} \otimes_D \eta^{iK}) \quad \downarrow K(\text{id} \otimes_D \bar{i}) \quad \downarrow \tilde{T}K\alpha \\
K(iF \otimes_D iK(iG \otimes_D i\tilde{T}H)) \xrightarrow{K(\text{id} \otimes_D \eta^{iK})} K(iF \otimes_D \bar{T}(iG \otimes_D iH)) \xrightarrow{K\bar{T}\alpha} \tilde{T}K(iF \otimes_D (iG \otimes_D iH)) \\
\downarrow K(\text{id} \otimes_D iK(\text{id} \otimes_D \beta)) \quad \downarrow K\bar{i} \quad \downarrow \rho \\
K(iF \otimes_D iK(iG \otimes_D \bar{T}iH)) \xrightarrow{K(\text{id} \otimes_D \eta^{iK})} K\bar{T}(iF \otimes_D (iG \otimes_D iH)) \xrightarrow{\rho} \tilde{T}K(iF \otimes_D (iG \otimes_D iH)) \\
\downarrow K(\text{id} \otimes_D iK\bar{i}) \quad \downarrow K(\text{id} \otimes_D \eta^{iK}) \quad \downarrow \tilde{T}(K(\text{id} \otimes_D \eta^{iK})) \\
K(iF \otimes_D iK\bar{T}(iG \otimes_D iH)) \xrightarrow{K(\text{id} \otimes_D i\rho)} \tilde{T}K(iF \otimes_D iK(iG \otimes_D iH)) \\
\downarrow K(\text{id} \otimes_D i\rho) \quad \downarrow K\bar{i} \\
K(iF \otimes_D i\tilde{T}K(iG \otimes_D iH)) \xrightarrow{K(\text{id} \otimes_D \beta)} K(iF \otimes_D \bar{T}iK(iG \otimes_D iH)) \xrightarrow{K\bar{i}} K\bar{T}(iF \otimes_D iK(iG \otimes_D iH))
\end{array}$$

- We must show:

$$\begin{array}{ccc}
I \otimes_{\text{Lam}} \tilde{T}F & \xrightarrow{\tilde{i}_{I,F}} & \tilde{T}(I \otimes_{\text{Lam}} F) \\
& \searrow \lambda_{\tilde{T}F} & \downarrow \tilde{T}\lambda_F \\
& & \tilde{T}F
\end{array}$$

The above diagram commutes because the following commutes (all the inner diagrams commute, by naturality, property of $\bar{i}$, and Theorem J.1 (e)).

$$\begin{array}{ccccc}
K(I \otimes_D i\tilde{T}F) & \xrightarrow{K(\text{id} \otimes_D \beta)} & K(I \otimes_D \bar{T}iF) & \xrightarrow{K\bar{i}} & K\bar{T}(I \otimes_D iF) \\
\downarrow K\lambda & & \downarrow K\lambda & \swarrow K\bar{T}\lambda & \downarrow \rho \\
Ki\tilde{T}F & \xrightarrow{K\beta} & K\bar{T}iF & & \tilde{T}K(I \otimes_D iF) \\
\downarrow \epsilon & & & \searrow \rho & \downarrow \tilde{T}K\lambda \\
\tilde{T}F & \xleftarrow{\tilde{T}\epsilon} & \tilde{T}KiF & & \tilde{T}KiF
\end{array}$$

- Lastly, since $\tilde{\mathcal{C}}$ is symmetric monoidal with $\gamma_{F,G} : F \otimes_{\text{Lam}} G \rightarrow G \otimes_{\text{Lam}} F$. We define the costrength $\sigma_{F,G} := \tilde{T}\gamma_{G,F} \circ \tilde{i}_{G,F} \circ \gamma_{\tilde{T}F,G} : \tilde{T}F \otimes_{\text{Lam}} G \rightarrow \tilde{T}(F \otimes_{\text{Lam}} G)$. We need to show the following diagram commutes.

$$\begin{array}{ccc}
& \tilde{T}F \otimes_{\text{Lam}} \tilde{T}G & \\
\sigma \swarrow & & \searrow \tilde{i} \\
\tilde{T}(F \otimes_{\text{Lam}} \tilde{T}G) & & \tilde{T}(\tilde{T}F \otimes_{\text{Lam}} G) \\
\tilde{T}\bar{i} \downarrow & & \downarrow \tilde{T}\sigma \\
\tilde{T}\tilde{T}(F \otimes_{\text{Lam}} G) & & \tilde{T}\tilde{T}(F \otimes_{\text{Lam}} G) \\
\mu^{\tilde{T}} \swarrow & & \swarrow \mu^{\tilde{T}} \\
& \tilde{T}(F \otimes_{\text{Lam}} G) &
\end{array}$$

Again, the above diagram commutes because the following diagram commutes (all the inner diagrams commute, by naturality, properties of $\bar{i}$, and Theorem J.1(d)+(f)).

$$\begin{array}{c}
K(i\tilde{T}G \otimes_D i\tilde{T}F) \xleftarrow{K\gamma} K(i\tilde{T}F \otimes_D i\tilde{T}G) \\
\downarrow K(\text{id} \otimes_D \beta) \quad \downarrow K(\beta \otimes_D \text{id}) \quad \downarrow K(\text{id} \otimes_D \beta) \\
K(i\tilde{T}G \otimes_D i\tilde{T}F) \quad K(\bar{T}iF \otimes_D i\tilde{T}G) \quad K(i\tilde{T}F \otimes_D \bar{T}iG) \\
\downarrow K\gamma \quad \downarrow K(\text{id} \otimes_D \beta) \quad \downarrow K(\beta \otimes_D \text{id}) \quad \downarrow K\bar{\iota} \\
K(i\tilde{T}G \otimes_D \bar{T}iF) \quad K(\bar{T}iF \otimes_D \bar{T}iG) \quad K\bar{T}(i\tilde{T}F \otimes_D iG) \\
\downarrow K\bar{\iota} \quad \downarrow K(\beta \otimes_D \text{id}) \quad \downarrow K\gamma \quad \downarrow K\bar{\iota} \quad \downarrow K\bar{T}(\beta \otimes_D \text{id}) \quad \downarrow \rho \\
K\bar{T}(i\tilde{T}G \otimes_D iF) \quad K(\bar{T}iG \otimes_D \bar{T}iF) \quad K\bar{T}(\bar{T}iF \otimes_D iG) \quad \tilde{T}K(i\tilde{T}F \otimes_D iG) \\
\downarrow \rho \quad \downarrow K\bar{T}(\beta \otimes_D \text{id}) \quad \downarrow K\bar{\iota} \quad \downarrow K\bar{T}\gamma \quad \downarrow \rho \quad \downarrow \tilde{T}K(\beta \otimes_D \text{id}) \quad \downarrow \tilde{T}K\lambda \\
\tilde{T}K(i\tilde{T}G \otimes_D iF) \quad K\bar{T}(\bar{T}iG \otimes_D iF) \quad K\bar{T}(iG \otimes_D \bar{T}iF) \quad \tilde{T}K(\bar{T}iF \otimes_D iG) \quad \tilde{T}K(iG \otimes_D i\tilde{T}F) \\
\downarrow \tilde{T}K\gamma \quad \downarrow K\bar{T}\gamma \quad \downarrow K\bar{T}\bar{\iota} \quad \downarrow \rho \quad \downarrow \tilde{T}K\lambda \quad \downarrow \tilde{T}K(\text{id} \otimes_D \beta) \\
\tilde{T}K(iF \otimes_D i\tilde{T}G) \quad K\bar{T}(iF \otimes_D \bar{T}iG) \quad K\bar{T}\bar{T}(iG \otimes_D iF) \quad \tilde{T}K(\bar{T}iG \otimes_D iF) \\
\downarrow \tilde{T}K(\text{id} \otimes_D \beta) \quad \downarrow \rho \quad \downarrow K\bar{T}\bar{\iota} \quad \downarrow K\bar{T}\gamma \quad \downarrow K\mu \quad \downarrow \rho \quad \downarrow \tilde{T}K\bar{\iota} \\
\tilde{T}K(iF \otimes_D \bar{T}iG) \quad K\bar{T}\bar{T}(iF \otimes_D iG) \quad K\bar{T}(iG \otimes_D iF) \quad \tilde{T}K\bar{T}(iG \otimes_D iF) \\
\downarrow \tilde{T}K\bar{\iota} \quad \downarrow \rho \quad \downarrow K\mu \quad \downarrow K\bar{T}\gamma \quad \downarrow \rho \quad \downarrow \tilde{T}K\gamma \\
\tilde{T}K\bar{T}(iF \otimes_D iG) \quad K\bar{T}(iF \otimes_D iG) \quad \tilde{T}K(iG \otimes_D iF) \quad \tilde{T}K\gamma \\
\downarrow \tilde{T}\rho \quad \downarrow \rho \quad \downarrow \tilde{T}K\gamma \quad \downarrow \tilde{T}\rho \\
\tilde{T}\tilde{T}K(iF \otimes_D iG) \xrightarrow{\mu} \tilde{T}K(iF \otimes_D iG) \xleftarrow{\mu} \tilde{T}\tilde{T}K(iF \otimes_D iG)
\end{array}$$

□